A USER'S MANUAL FOR
ELECTROMAGNETIC SURFACE PATCH (ESP) CODE:
VERSION II - POLYGONAL PLATES AND WIRES

E.H. Newman
P. Alexandropoulos

The Ohio State University
ElectroScience Laboratory
Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 712692-3
Contract No. N00014-78-C-0049
September 1983
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.
This report serves as a user's manual for the Electromagnetic Surface Patch (ESP) Code. This code is a method of moments solution for interconnections of thin wires and polygonal plates. The code can compute currents, input impedance, efficiency, mutual coupling, and far-zone radiation and scattering patterns. In addition to describing the code input and output, the use of the code is illustrated by simple examples. Subroutine descriptions are also given.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER I: INTRODUCTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAPTER II: FORMULATION OF THE MOMENT METHOD SOLUTION</td>
<td>1</td>
</tr>
<tr>
<td>A. THEORY</td>
<td>4</td>
</tr>
<tr>
<td>B. EXPANSION MODES</td>
<td>8</td>
</tr>
<tr>
<td>1. Wire Mode</td>
<td>8</td>
</tr>
<tr>
<td>2. Surface Patch Mode</td>
<td>10</td>
</tr>
<tr>
<td>3. Attachment Mode</td>
<td>14</td>
</tr>
<tr>
<td>4. Overlap Mode</td>
<td>16</td>
</tr>
<tr>
<td>C. TEST MODES</td>
<td>16</td>
</tr>
<tr>
<td>D. COMPUTATION OF THE IMPEDANCE ELEMENTS</td>
<td>17</td>
</tr>
<tr>
<td>1. Surface Patch Monopoles</td>
<td>20</td>
</tr>
<tr>
<td>2. Two Parallel Surface Patch Monopoles</td>
<td>22</td>
</tr>
<tr>
<td>3. Touching Surface Patch Monopoles</td>
<td>22</td>
</tr>
<tr>
<td>4. Toeplitz Properties</td>
<td>24</td>
</tr>
<tr>
<td>5. Disk Parallel To Plate</td>
<td>25</td>
</tr>
<tr>
<td>CHAPTER III: MAIN PROGRAM INPUTS</td>
<td>28</td>
</tr>
<tr>
<td>A. READ INPUT STATEMENTS</td>
<td>29</td>
</tr>
<tr>
<td>1. READ 1</td>
<td>29</td>
</tr>
<tr>
<td>2. READ's 2-5</td>
<td>33</td>
</tr>
<tr>
<td>3. READ 6</td>
<td>36</td>
</tr>
</tbody>
</table>
4. READ's 7-9 36
5. READ 10 38
6. READ's 11-15 42

B. SUBROUTINE WGEOM 47
   1. Example of WGEOM Subroutines 49

C. ARRAY DIMENSIONS 55

D. PROGRAM FILE DESCRIPTIONS 58

E. DESIGN EXAMPLES 62
   1. DESIGN EXAMPLE 1 63
   2. DESIGN EXAMPLE 2 69
   3. DESIGN EXAMPLE 3 78
   4. DESIGN EXAMPLE 4 81

CHAPTER IV: GENERAL PARAMETERS 93

A. GENERAL PARAMETERS 93
B. WIRE PARAMETERS 96
C. PLATE MODE PARAMETERS 99
D. ATTACHMENT MODE PARAMETERS 101
E. NOTES ON THE IMPEDANCE MATRIX 103
F. MISCELLANEOUS NOTES 104
   1. SUBROUTINE PLPLCK 106
   2. SUBROUTINE PLATE3 107
   3. SUBROUTINE POPLOV 111
   4. SUBROUTINE FGPOV 115
   5. SUBROUTINE FMDC 119
6. SUBROUTINE MOPLOT
7. SUBROUTINE MPLLOT
8. SUBROUTINE GPLLOT2
9. SUBROUTINE ZTOT
10. SUBROUTINE TOPO
11. SUBROUTINE PLTST2
12. SUBROUTINE ZWTPE2
13. SUBROUTINE PLTTS1
14. SUBROUTINE PPLTS
15. SUBROUTINE ZWTPE
16. SUBROUTINE ZWTDE
17. SUBROUTINE ZWTWE
18. SUBROUTINE ZGSMM
19. SUBROUTINE GGS1
20. SUBROUTINE DSKTS2
21. SUBROUTINE DSKTST
22. SUBROUTINE ZATAT2
23. SUBROUTINE PDPZ1
24. SUBROUTINE PDPZ
25. SUBROUTINE ERDSK
26. SUBROUTINE COUPLE
27. SUBROUTINE ANTV
28. SUBROUTINE AGDISS
29. SUBROUTINE ARITE
<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>30. SUBROUTINE CROUT</td>
</tr>
<tr>
<td>31. SUBROUTINE SORTB</td>
</tr>
<tr>
<td>32. SUBROUTINE SURMFF</td>
</tr>
<tr>
<td>33. SUBROUTINE SURFFP</td>
</tr>
<tr>
<td>34. SUBROUTINE DSKFF</td>
</tr>
</tbody>
</table>

CHAPTER V: SUMMARY

REFERENCES

APPENDICES

<p>| 1: OUTPUT FOR DESIGN EXAMPLE 1 | 184 |
| 2: OUTPUT FOR DESIGN EXAMPLE 2 | 190 |
| 3: OUTPUT FOR DESIGN EXAMPLE 3 | 194 |
| 4: OUTPUT FOR DESIGN EXAMPLE 4 | 198 |
| 5: OUTPUT FOR DESIGN EXAMPLE 5 | 203 |
| 6: SUBROUTINE PLPLCK | 205 |
| 7: SUBROUTINE PLATE 3 | 206 |
| 8: SUBROUTINE POPLOV | 212 |
| 9: SUBROUTINE FGPOV | 219 |
| 10: SUBROUTINE FMDC | 225 |
| 11: SUBROUTINE MPLcot | 227 |
| 12: SUBROUTINE MOPLOT | 232 |
| 13: SUBROUTINE GLOT2 | 236 |
| 14: SUBROUTINE ZTOT | 239 |
| 15: SUBROUTINE TOPO | 253 |</p>
<table>
<thead>
<tr>
<th>Number</th>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:</td>
<td>SUBROUTINE PLTTST</td>
<td>254</td>
</tr>
<tr>
<td>17:</td>
<td>SUBROUTINE PPLTS</td>
<td>256</td>
</tr>
<tr>
<td>18:</td>
<td>SUBROUTINE PLTST2</td>
<td>259</td>
</tr>
<tr>
<td>19:</td>
<td>SUBROUTINE ZWTPE</td>
<td>262</td>
</tr>
<tr>
<td>20:</td>
<td>SUBROUTINE ZWTDE</td>
<td>263</td>
</tr>
<tr>
<td>21:</td>
<td>SUBROUTINE ZWTWE</td>
<td>265</td>
</tr>
<tr>
<td>22:</td>
<td>SUBROUTINE ZWTPE2</td>
<td>266</td>
</tr>
<tr>
<td>23:</td>
<td>SUBROUTINE DSKTST</td>
<td>270</td>
</tr>
<tr>
<td>24:</td>
<td>SUBROUTINE DSKTS2</td>
<td>272</td>
</tr>
<tr>
<td>25:</td>
<td>SUBROUTINE ZATAT2</td>
<td>274</td>
</tr>
<tr>
<td>26:</td>
<td>SUBROUTINE PDPZ</td>
<td>276</td>
</tr>
<tr>
<td>27:</td>
<td>SUBROUTINE PDPZ1</td>
<td>277</td>
</tr>
<tr>
<td>28:</td>
<td>SUBROUTINE ERDSK</td>
<td>279</td>
</tr>
<tr>
<td>29:</td>
<td>SUBROUTINE ZGSM1</td>
<td>280</td>
</tr>
<tr>
<td>30:</td>
<td>SUBROUTINE GGS1</td>
<td>282</td>
</tr>
<tr>
<td>31:</td>
<td>SUBROUTINE COUPLE</td>
<td>284</td>
</tr>
<tr>
<td>32:</td>
<td>SUBROUTINE ANTV</td>
<td>285</td>
</tr>
<tr>
<td>33:</td>
<td>SUBROUTINE CROUT</td>
<td>287</td>
</tr>
<tr>
<td>34:</td>
<td>SUBROUTINE AGDISS</td>
<td>289</td>
</tr>
<tr>
<td>35:</td>
<td>SUBROUTINE ARITE</td>
<td>290</td>
</tr>
<tr>
<td>36:</td>
<td>SUBROUTINE SORTB</td>
<td>292</td>
</tr>
<tr>
<td>37:</td>
<td>SUBROUTINE SURFFP</td>
<td>297</td>
</tr>
<tr>
<td>38:</td>
<td>SUBROUTINE SURNF</td>
<td>299</td>
</tr>
</tbody>
</table>
39: SUBROUTINE DSKFF

40: SYMBOLS
CHAPTER I
INTRODUCTION

This report deals with the development of a computer program based on the Moment Method (MM) solution for antenna and scattering problems. The program can solve for geometries consisting of polygonal plates, wires, wire/plate junctions and multiplate junctions. The advantages of the program are accuracy, versatility and the flexibility in the input of the problem geometry. It can calculate far-zone radiation and scattering patterns as well as antenna input impedances and efficiencies. The disadvantages of the program are the limitation to geometries not large in terms of wavelength and the lack of analytical results which can provide physical insight into the problem.

The first implementation of the Moment Method solution into a useful computer program involved the thin-wire formulation [1,2]. These programs gave good results for most wire antennas and by forming wire grid models of solid surfaces reasonable results were obtained [3]. The wire grid approach was limited to solid surfaces whose dimensions in terms of wavelength were very small. Also it did not give accurate results for near-zone parameters such as current distributions and input impedances.
The next development was the use of surface patches for the modeling of three dimensional surfaces. Surface patches give a much more accurate approximation to the currents on a three-dimensional surface and require less unknowns than the wire grid model. Oshiro [4] used pulse basis functions and point matching to solve the Magnetic Field Integral Equation (MFIE) for various three dimensional surfaces. Albertsen, et al. [5], used pulse test modes and the MFIE formulation to model wire, plates and wire/plate attachments. However, their solution was limited to closed surfaces since the MFIE applies to that type of surface only. Based on the MFIE formulation, the Numerical Electromagnetic Code (NEC) was developed by Burke and Poggio [6] to solve for geometries consisting of wires and closed surfaces. Wang and Richmond [7] used piecewise sinusoidal (PWS) rectangular surface patches to model rectangular plates and wires. Surface patch models using triangular patches and pulse test modes have been developed by Glisson [8] and can handle wires, plates and wire/plate junctions.

This report incorporates the work of Newman and Pozar [9,10] and Tulyathan [11].

Chapter Two gives a brief review of the MM solution for electromagnetic scattering and radiation problems, based on the Reaction Integral Equation (RIE) [12]. The wire, plate, attachment and overlap modes are defined. Various special methods for calculating the impedance matrix efficiently and accurately are also discussed.
Chapter Three describes the READ input statements of the main program. For every READ statement a detailed explanation of all the parameters introduced is given. Subroutine WGEOM is described and two examples illustrating the methodology of creating and the advantages of using such a subroutine are given. Finally, several design examples are given for a better understanding of the input parameters.

Chapter Four contains descriptions of every subroutine called by the main program. Excluded are the subroutines documented in [1] and also several subroutines that are system dependent. For every subroutine the following format is used:

1. A brief statement of its purpose.
2. The general calling form.
3. Detailed definition of every input and output parameter.
4. A brief outline of the subroutine's inner workings, unless self evident.

Chapter Five is the summary.
CHAPTER II
FORMULATION OF THE MOMENT METHOD SOLUTION

A. THEORY

This chapter gives a brief outline of the solution of the electromagnetic scattering or antenna problem by the Method of Moments (MM). A description of the expansion (basis) functions is given, along with a discussion of various impedance calculation methods.

Consider an arbitrarily shaped scatterer in a homogeneous medium. Let \( S \) represent the surface of the scatterer and \( \hat{n} \) the unit outward normal to the surface. \((\mathbf{j}_i,\mathbf{m}_i)\) is an impressed source which radiates fields \((\mathbf{E}_i,\mathbf{H}_i)\) in free space and fields \((\mathbf{E},\mathbf{H})\) in the presence of the scatterer (see Figure 2-1).

From Schelkunoff's surface equivalence theorem \([13]\) the field interior to the surface \( S \) will vanish without changing the exterior field \((\mathbf{E},\mathbf{H})\) if one introduces the following electric and magnetic surface current densities on surface \( S \):

\[
\mathbf{J}_s = \hat{n} \times \mathbf{H} \quad (2.1)
\]

\[
\mathbf{M}_s = \mathbf{E} \times \hat{n} \quad (2.2)
\]
Figure 2-1. Source \((J_i, M_i)\) radiates fields \((E, H)\) in the presence of the scatterer.

Figure 2-2. Surface currents \((J_s, M_s)\) placed on the surface of the scatterer do not change the exterior fields \((E, H)\).
where \( \mathbf{E} \) and \( \mathbf{H} \) are the electric and magnetic fields, respectively, on the surface \( S \). These surface current densities radiate, in the ambient medium, the scattered fields \( (\mathbf{E}_s,\mathbf{H}_s) \) (see Figure 2-2) which are defined by:

\[
\mathbf{E}_s = \mathbf{E} - \mathbf{E}_i \tag{2.3}
\]

\[
\mathbf{H}_s = \mathbf{H} - \mathbf{H}_i. \tag{2.4}
\]

If one place a test source \( (\mathbf{J}_m,\mathbf{M}_m) \) in the volume interior to surface \( S \) its reaction with the sources \( (\mathbf{J}_i,\mathbf{M}_i) \) and \( (\mathbf{J}_s,\mathbf{M}_s) \) will be zero since the field interior to surface \( S \) is zero, i.e.,

\[
\iint_S (\mathbf{J}_m \mathbf{E}_s - \mathbf{M}_m \mathbf{H}_s) \, ds = -\iint_S (\mathbf{J}_m \mathbf{E}_i - \mathbf{M}_m \mathbf{H}_i) \, ds. \tag{2.5}
\]

Using the reciprocity theorem, Equation (2.5) can be written as

\[
\iint_S (\mathbf{J}_s \mathbf{E}_m - \mathbf{M}_s \mathbf{H}_m) \, ds + \iiint_V (\mathbf{J}_i \mathbf{E}_m - \mathbf{M}_i \mathbf{H}_m) \, dv = 0 \tag{2.6}
\]

where \( V \) is the volume occupied by source \( (\mathbf{J}_i,\mathbf{M}_i) \).

This is the Reaction Integral Equation (RIE). If one uses a set of electric test sources, the RIE reduces to the Electric Field Integral Equation (EFIE). If a set of magnetic test sources is used, the RIE reduces to the Magnetic Field Integral Equation (MFIE). The EFIE is used in this work since it applies to both closed and open surfaces while the MFIE applies to closed surfaces only. Thus one can take \( \mathbf{M}_m = 0 \). Also, perfect conductivity is assumed for the scatterer surface and thus \( \mathbf{M}_s = 0 \).
All of the above analysis was based on the assumption that the surfaces in consideration are closed. However it can be shown that the analysis is valid for open surfaces such as surface plates. This is very important since surface plate modeling is the core of the Electromagnetic Surface Patch Code (ESP). The plates used in the ESP code are fictitious in the sense that they have zero thickness. In general different currents exist on the top and bottom surfaces of a real plate. As the thickness of the plate goes to zero the fields radiated by the top and bottom currents become equivalent to the field radiated by a single current located at the center of the plate. This current, which is \( J_s \) of Equation (2.6), is the vector sum of the top and bottom surface currents of the plate [14].

\( J_s \) represents the unknown current on the surface of the scatterer. The Moment Method solution begins by expanding \( J_s \) in terms of \( N \) expansion (basis) functions \( F_n \), i.e.,

\[
J_s = \sum_{n=1}^{N} I_n F_n . \tag{2.7}
\]

Substituting \( J_s \) from Equation (2.7) into Equation (2.6) one obtains:

\[
\sum_{n=1}^{N} I_n Z_{mn} = V_m ; \quad m = 1, N \tag{2.8}
\]

where

\[
Z_{mn} = - \iint_{n} E_m F_{nds} , \tag{2.9}
\]

\[
V_m = \iiint_{V} J_m E dV . \tag{2.10}
\]
The integral in Equation (2.9) is over the surface of the n-th expansion mode while the one in Equation (2.10) is over the volume occupied by source $(J_i,M_i)$. $V_m$ is called the excitation voltage. The detailed evaluations of Equations (2.9) and (2.10) are described in references [9,10].

B. EXPANSION MODES

Three types of basis functions (modes) are used in the moment method computer code, i.e., wire, surface patch and attachment dipole modes. With this choice of modes one can model geometries consisting of wires and/or polygonal plates or any geometry that can be approximated by wires and polygonal plates.

1. **Wire Mode**

The wire mode is the piecewise sinusoidal V-dipole consisting of two sinusoidal monopoles. Figure 2-3 shows a V-dipole with 180 degree interior angle [10]. The current on this dipole is given by

$$J_s = \frac{a}{2\pi a} \left[ p_1 \frac{\sin(k(z-z_1))}{\sin(k(z_1-z_1))} + p_2 \frac{\sin(k(z_3-z))}{\sin(k(z_3-z_2))} \right], \quad (2.11)$$

where

$$p_1 = \begin{cases} 1 & z_1 < z < z_2 \\ 0 & \text{elsewhere} \end{cases}$$

$$p_2 = \begin{cases} 1 & z_2 < z < z_3 \\ 0 & \text{elsewhere} \end{cases}$$

$a =$ the wire radius and $k = 2\pi/\lambda$.  

8
Figure 2-3. A wire PWS dipole mode.

Figure 2-4. Array of overlapping PWS wire dipoles representing the current on a wire.
When wire modes are used to represent the current on a real wire they are placed in an overlapping array as shown in Figure 2-4. This ensures the continuity of current on the wire.

2. **Surface Patch Mode**

Two kinds of surface patch modes are used, i.e., a rectangular and a quadrilateral. The rectangular mode is a surface V-dipole consisting of two rectangular sinusoidal surface patch monopoles. A surface V-dipole with an interior angle of 180 degrees is shown in Figure 2-5 [10].

The current density on this dipole is given by

\[
J_s = \hat{z} \frac{kP_1 \sin(k(z-z_1)) \cos(ky)}{2 \sin(z_2-z_1) \sin(kw)} + \hat{z} \frac{kP_2 \sin(k(z_3-z)) \cos(ky)}{\sin(z_3-z_2) \sin(kw)} \tag{2.12}
\]

where \(P_1\) and \(P_2\) are the unit pulse functions as described for the wire dipole.

Two orthogonal and overlapping arrays of rectangular surface patch modes are used to represent the current density on a rectangular plate as shown in Figure 2-6. Each arrow represents a V-dipole. This modal outlay ensures continuity of current on the surface of the plate and it makes the current density a two dimensional vector.

If the plate is not rectangular then quadrilateral V-dipole surface patches are used. A typical quadrilateral surface patch mode is shown in Figure 2-7.
Figure 2-5. A PWS rectangular surface patch dipole mode.

Figure 2-6. A two dimensional array of overlapping rectangular surface patch dipole modes representing the current density on a rectangular plate. The modes are represented by arrows.
Figure 2-7. A quadrilateral surface patch dipole mode.
To describe the current density on the patch consider a point \( C \) interior to monopole \( A \). Draw a line \( X \) intersecting sides "a" and "b" in such a way that \( u/U = v/V \). The intersection of sides a and b is point \( 0 \). A line from point \( 0 \) to the end side of monopole \( A \) will be divided by line \( X \) such that \( \ell/L = u/U = v/V \). \( L \) is the segment of the line drawn from \( 0 \) between the terminal and end side. The coordinate along segment \( L \) is \( \ell \) (\( \ell = 0 \) at the terminal and \( \ell = L \) at the end) and \( W(\ell/L) \) is the length of the segment of line \( X \) between sides "a" and "b".

Now the current density on monopole \( A \) of the surface patch mode is

\[
J_\text{SA} = c' \frac{\ell \sin \ell}{\sin L \cdot W(\ell/L)}.
\] (2.13)

The constant \( C \) is chosen so that the current at the terminal side of monopole \( A \) is one ampere.

The density on monopole \( B \) is

\[
J_\text{SB} = c' \frac{\ell \sin \ell}{\sin t \cdot W(\ell/T)}.
\] (2.14)

where \( t \) and \( T \) are defined the same way as \( \ell \) and \( L \) were defined for monopole \( A \). The constant \( C' \) is chosen so that the current at the terminal side of monopole \( B \) is equal to one ampere.

This surface patch mode is similar to the rectangular patch mode used by Newman and Pozar [9].
3. Attachment Mode

The attachment mode, shown in Figure 2-8(a), is used to model the wire/plate junction. Note that the wire is not necessarily always perpendicular to the disk. The attachment mode serves two purposes:

1. Ensures the continuity of current at the junction.
2. Ensures the proper \( \rho \) polarization and \( 1/\rho \) dependence of the current density at the junction.

It is composed of two monopoles, the wire monopole which is similar to the wire monopole described by Equation (2.11) and a disk monopole. The current density of the wire part is

\[
\frac{J_W}{S} = \frac{1}{2\pi a} \frac{\sin(kz_2 - z)}{\sin k z_2} \hat{z} ; \quad 0 < z < z_2 , \tag{2.15}
\]

while the current density on the disk is

\[
\frac{J_D}{S} = -\frac{\sin(k(b - \rho))}{2\pi \rho \sin(k(b - a))} \hat{\rho} ; \quad a < \rho < b \tag{2.16}
\]

\( a \) = radius of the wire and \( b \) = outer radius of disk. See Figure 2-8(b).

Note that the disk density at \( \rho = a \) equals the wire density at \( z = 0 \), insuring continuity of current at the junction. Also, the density at the edge of the disk (\( \rho = b \)) is zero to maintain continuity of current on the plate where the disk is placed.
Figure 2-8. (a) Wire attachment dipole mode, and (b) Current density on the wire monopole (top) and the disk monopole (bottom).
The attachment mode is placed directly over the surface of the plate at the wire/plate junction. The only restriction is that the wire/plate junction be at least $0.1\lambda$ away from all edges of the plate. Through convergence tests it was found that a good value for the outer radius $b$ is $0.2\lambda$.

4. Overlap Mode

The overlap modes are identical in mathematical description to the surface patch modes. They allow for the continuity of current at a plate-to-plate intersection. The edges of the overlap modes need not coincide with the edges of the surface patch modes on either plate. However, the closer they match the better the results seem to become. The program automatically searches for plates with touching sides and places the corresponding overlap modes.

C. TEST MODES

Normally the test modes used in the MM solution are identical to the expansion modes (Galerkin's method). This results in a symmetric impedance matrix and only its lower triangular part is evaluated.

Most of the computer CPU time is spent in calculating the elements of the impedance matrix. Substantial CPU time can be saved, without compromising the accuracy of the solution, by representing the test modes as single filaments. The endpoints of a filament are defined by the midpoints of the terminal and end sides of the surface patch it represents.
D. COMPUTATION OF THE IMPEDANCE ELEMENTS

Since there are three types of expansion and test modes the impedance matrix consists of nine types of impedance elements. This is shown graphically in Figure 2-9. It should be noted that the mutual impedance between two dipole modes is simply the sum of four monopole-to-monopole impedances.

Of the nine types of dipole-to-dipole impedances shown in Figure 2-9, five involve wire dipoles and require little computation time. Attachment-to-attachment impedances occur infrequently and thus do not require very much CPU time. Since most of the computation time is spent for surface patch-to-surface patch and disk-to-surface patch mutual impedances, the discussion that follows refers to those type of calculations.

A general impedance matrix entry $Z_{mn}$ is defined by Equation (2.9), or more explicitly

$$Z_{mn} = - \int \int E_m(e_1,e_2) \cdot J_n(e_1,e_2) \, de_1 de_2 \quad (2.17)$$

where $e_1,e_2$ are independent coordinates on the surface of the n-th expansion mode, $J_n$ is the current density of the n-th expansion mode and $E_m$ is the electric field in free space of the m-th test mode. In particular, one finds that

$$E_m(e_1,e_2) = \int \int \bar{E}_0 (e_1,e_2;t_1,t_2) \cdot J_m(t_1,t_2) \, dt_1 dt_2 \quad (2.18)$$
Figure 2-9. Symbolic representation of the nine different blocks of the impedance matrix.

<table>
<thead>
<tr>
<th>W/W</th>
<th>W/P</th>
<th>W/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>P/W</td>
<td>P/P</td>
<td>P/A</td>
</tr>
<tr>
<td>A/W</td>
<td>A/P</td>
<td>A/A</td>
</tr>
</tbody>
</table>

W = WIRE
P = PLATE
A = ATTACHMENT
where $J_m$ is the current density of the $m$-th test mode, $t_1$ and $t_2$ are independent coordinates on the surface of the $m$-th test mode, and $\overline{u_0}$ is the free space Dyadic Green's function. For a piecewise sinusoidal mode the field $E_m$ is known in closed form for a wire [15] and surface patch [16] monopole and can be evaluated with a numerical integration for a disk monopole.

There is a way of avoiding the double integration of Equation (2.18) if we consider both the expansion and test mode as being made up of piecewise sinusoidal filaments. Then the impedance between a filament on the expansion mode (in the $e_1$ direction) and a filament on the test mode (in the $t_1$ direction) is given by (see Figure 2-10 for the filamentary representation of two surface patch monopoles)

$$Z_{mn} = - \int_{e_1} J_n(e_1, e_2) \cdot \int_{t_1} \overline{u_0}(e_1, e_2; t_1, t_2) \cdot J_m(t_1, t_2) dt_1 de_1 .$$

This expression is known in closed form for piecewise sinusoids [17]. The total impedance between the expansion and the test monopole is

$$Z_{mn} = \int_{e_2} \int_{t_2} Z_{mn}(e_2, t_2) de_2 dt_2 .$$

The evaluation of the double integral is done numerically in the code, usually using a Simpson's rule integration or Spline integration. The main advantage of using the second method (Equations 2.19 and 2.20) is that it results in a simpler computer program. In particular only one subroutine is needed to evaluate $Z_{mn}$ while for method one (Equation 2.17) three different subroutines are needed for calculating $E_m$, the
electric field of each type of test monopole. However, both methods are used for efficient computation of the impedance elements. Several examples of how each method is used are given below along with other special computation techniques.

1. **Surface Patches Monopoles**

A surface patch monopole is represented by NPT filaments (see Figure 2-10) and the mutual impedance between two surface patch monopoles is then the weighted sum of their mutual filament to filament impedances. The choice of NPT is important for the accuracy of the impedance matrix elements and the computation time required to evaluate those elements. Through extensive convergence tests it was found that NPT should be set based on DIST, where DIST is the center to center distance between two surface patch monopoles, as follows:

1. If \(0 < \text{DIST} \leq 0.25\lambda\), \(\text{NPT} = 8\)
2. If \(0.25\lambda < \text{DIST} \leq 0.35\lambda\), \(\text{NPT} = 4\)
3. If \(0.35\lambda < \text{DIST} \leq 0.6\lambda\), \(\text{NPT} = 2\)
4. If \(\text{DIST} > 0.6\lambda\), \(\text{NPT} = 1\).

The one-filament representation for each surface patch monopole can be quickly and easily modified to increase accuracy. If \(Z_{ff}\) is the filament-to-filament impedance then the variation due to the actual size and orientation of the \(n\)-th surface patch expansion monopole is taken into account by \([11]\)
Figure 2-10. Filamentary representation of a test surface patch monopole and an expansion surface patch monopole.

Figure 2-11. Single filament representation of two surface patch monopoles. Points A, B, C, D are the midpoints of their respective segments.
\[ Z_{fn} = Z_{ff} \left( \frac{\text{Re} - jkR}{R} \frac{e^{-jkR_1}}{R_1} + 4e^{-jkR} + \frac{e^{-jkR_2}}{R_2} \right) \]  \hspace{1cm} (2.21)

where \( R_1, R_2, R_3 \) are the average distances from the test filament to the closer, center and furthest edge, respectively of the expansion surface patch monopole (see Figure 2-11).

If the variation due to the \( m \)-th test monopole is taken into consideration then

\[ Z_{mn} = Z_{fn} \left( \frac{\text{Re} - jkR}{R} \frac{e^{-jkR_1}}{R_1} + 4e^{-jkR} + \frac{e^{-jkR_2}}{R_2} \right). \]  \hspace{1cm} (2.22)

2. **Two Parallel Surface Patch Monopoles**

Another advantage of the use of Equation (2.20) comes when calculating the mutual impedances between two rectangular surface patch monopoles which have current vectors parallel. This includes monopoles which have the vectors transverse to the current direction vectors parallel (see Figure 2-12). If each monopole is represented by \( M \) filaments, then \( M^2 \) filament-to-filament impedances need to be calculated. However, no more than \( 2M \) impedances are different and the rest can be evaluated from those \( 2M \) entries. This makes the computation time proportional to \( 2M \) instead of \( M^2 \).

3. **Touching Surface Patch Monopoles**

The integral of Equation (2.20) converges slowly when computing the mutual impedance between two touching monopoles. This is due to the
CURRENT DIRECTIONS ARE PARALLEL

VECTORS TRANSVERSE TO CURRENT DIRECTIONS ARE PARALLEL

Figure 2-12. Two cases of parallel rectangular surface patch monopoles.
fact that the imaginary part of the mutual impedance of two piecewise
sinusoidals has a logarithmic singularity as the separation between the
two filaments gets smaller [18]. For a small separation $x$ the reactance
between the two filaments can be written as

$$X(x) = C_1 + C_2 \ln(x).$$ (2.23)

The constants $C_1$ and $C_2$ can be evaluated and the $\ln$ singularity is
integrated analytically. If $\Delta x$ is a numerical integration interval then
the equivalent value of the reactance at $x = 0$ is

$$X(0) = -X(\Delta x) + 2C_1 + 2C_2[\ln(\Delta x) - 1].$$ (2.26)

where

$$C_2 = \frac{X(\Delta x) - X(\Delta x/2)}{\ln^2}$$ (2.24)

$$C_1 = X(\Delta x) - C_2 \ln(\Delta x)$$ (2.25)

$X(0)$ is not the reactance at $x = 0$ (which would be infinite), but rather
the value of $X$ at zero that makes the numerical integration correct
[10].

4. **Toeplitz Properties**

The impedance matrix for a single rectangular plate displays a
great deal of toeplitz properties which can be used to reduce the
computation time.

24
This property comes into play when computing the mutual impedance between modes on the same rectangular plate. Consider the typical modal layout for a 0.5\(\lambda\) by 1\(\lambda\) rectangular plate as shown in Figure 2-13.

Modes with the same current polarization are of equal size. In the example of Figure 2-13 modes 1, 2, 3, 4 are the same and modes 5, 6, 7, 8, 9, 10 are the same. It is obvious that \(Z_{51} = -Z_{52}\), \(Z_{58} = Z_{69} = Z_{710}\), \(Z_{68} = -Z_{59}\), etc. This indicates that in general one does not need to calculate all of the mutual impedances between modes on the same rectangular plate. It is only necessary to compute the mutual impedances between the first mode in the 1-2 direction and all the modes on the plate and the mutual between the first mode in the 2-3 direction and all the 2-3 modes. This is shown graphically in Figure 2-14 where X's represent mutuals that are calculated and 0's represent mutuals that can be obtained using the Toeplitz properties. If \(N_{12}\) is the number of modes in the 1-2 direction and \(N_{23}\) the number of modes in the 2-3 direction, then instead of calculating \((N_{12} + N_{23})^2/2 + (N_{12} + N_{23})/2\) mutual impedances one only has to calculate \(N_{12} + 2*N_{23}\).

5. Disk Parallel To Plate

Whenever a wire attachment is used substantial time can be saved in computing impedances between the disk monopole and plate modes on any plate parallel to the disk monopole. Here advantage is taken of the fact that the electric field of the disk part of the attachment mode has a radial component which is a function of the radial distance only. Using this property a table containing the values of \(E_\rho\) versus \(\rho\) is
Figure 2-13. Model layout on a rectangular plate. The mode numbers are shown next to the arrays.

Figure 2-14. Symbolic representation of a P/P impedance matrix block. All entries in the block are the mutual impedances between modes on the same rectangular plate.
created and then used to extrapolate the value of \( E_m \) to be used in
Equation (2.17). The integral in Equation (2.17) is actually evaluated
numerically in the code as described in subroutines DSKTST and DSKTS2.
CHAPTER III
MAIN PROGRAM INPUTS

The inputs to the Electromagnetic Surface Patch Code (ESP) are explained below. They are used to describe to the program the detailed geometry of the problem and indicate the type of calculation desired. The input data can be broken into four major groups as follows:

1. Miscellaneous parameters.
2. Type of calculation desired.
3. Plate geometry.
4. Wire and attachment geometry.

The first three parts are always defined by an input file. The wire and attachment geometry can be defined either by an input file or by a FORTRAN subroutine called WGEOM. At first, the use of a separate subroutine for describing the wire geometry might seem as an unnecessary complication. However, experience has shown that subroutine WGEOM is very useful for cases where the wire structure has a regular or periodic geometry or a shape that can be defined by an analytic expression. Examples are monopole and dipole antennas, loop antennas, helical antennas, log periodic antennas and arrays. For further explanation about the use of WGEOM see Section 3.
A. READ INPUT STATEMENTS

A description of every READ input statement is given below along with a definition of every parameter introduced. The fifteen READ input statements are shown in Figure 3-1. Also shown is some of the main program logic, indicating the order and number of times each READ is called. All READ input statements use a free format input on logical unit 5.

1. READ 1

The first READ input statement defines the following control parameters:

NGO = run indicator.

= 0 implies input and print out problem geometry and then stop, i.e., do not calculate any patterns or data. An NGO = 0 run should precede any pattern or data calculations. It gives the user the opportunity to review the accuracy of the problem geometry as defined by the input file.

= 1 implies go through the whole program, i.e. input the geometry and calculate the required patterns or data.

NPRINT = print indicator.

= 1 implies print wire and plate geometry.

= 2 implies print both the input parameters and the wire/plate geometry. Normally NPRINT = 2.

= 3 implies print nothing.
Figure 3-1. The 15 READ input statements.
NRUNS = the number of runs to be made, i.e., the limit of the DO 700 loop in Figure 3-1.

NWGS = the number of wire geometries for each run, i.e., the limit of the 600 loop.

IWR = indicator for writing out the induced modal currents.

= 0 implies do not print the induced modal currents.

= 1 implies print the induced modal currents plus the detailed wire and plate modal geometry.

IWRZT = indicator for writing the impedance matrix on the output file.

= 0 implies do not write the impedance matrix on the output file.

= 1 implies write the impedance matrix on the output file.

INT = the number of Simpson's rule integration intervals used for the evaluation of the wire-to-wire impedances. INT is always an even integer, usually equal to 4.

= 0 implies the impedance calculations are to be done using the exact closed form expression. Self or overlapping wire impedances are always
calculated by the closed form expression because it is more accurate than the numerical integration. However, the closed form expression is more time consuming than the INT = 4 numerical integration.

\[ \text{INTP} = \text{the number of Simpson's rule integration intervals used in integrating over the surface patch monopoles. INTP is always an even integer, typically chosen as 8.} \]

\[ \text{INTD} = \text{the number of Simpson's rule integration intervals used in integrating over the disk monopoles. INTD is always an even integer, typically chosen as 18.} \]

\[ \text{INWR} = \text{wire indicator.} \]

\[ = 0 \text{ implies geometry does not contain any wires.} \]

\[ = 1 \text{ implies geometry contains wires.} \]

\[ \text{IHGM} = \text{indicator for choosing the method of defining the wire geometry.} \]

\[ = 0 \text{ implies the wire geometry is to be defined by subroutine WGEOM.} \]

\[ = 1 \text{ implies the wire geometry is to be read in via the input file.} \]

\[ \text{IFIL} = \text{indicator for choosing the type of test plate modes.} \]
= 0 implies full surface patch test plate modes.
= 1 implies filamentary test plate modes.

2. READ's 2-5

READ's 2-5 specify the far-zone pattern calculations desired. READ's 2 and 3 specify the elevation and azimuth radiation patterns, respectively. READ's 4 and 5 specify the elevation and azimuth scattering patterns, respectively.

READ 2 defines the following:

IFE = indicator for calculating the far zone elevation radiation pattern.
= 0 implies do not compute far zone radiation pattern in the elevation plane.
= 1 implies compute far zone radiation pattern in the elevation plane.

IPFE = plot indicator.
= 0 implies do not plot far zone radiation pattern in the elevation plane.
= 1 implies plot far zone radiation pattern in the elevation plane.

NDFE = angle increment in degrees for far zone radiation pattern in the elevation plane.

PHFE = phi angle in degrees for far zone radiation pattern in the elevation plane.
READ 3 defines the following:

IFA, PFA, N DFA = same as IFE, IPFE, NDFE but for azimuth plane.

THFA = theta angle in degrees for far zone radiation pattern in the azimuth plane.

READ 4 defines the following:

ISE = indicator for calculating the far zone elevation plane scattering pattern. Scattering implies either backscattering (ISE = 1) or bistatic scattering (ISE = 2).

= 0 implies do not compute far zone scattering pattern in the elevation plane.

= 1 implies compute backscattering pattern in the elevation plane.

= 2 implies compute bistatic scattering in the elevation plane.

IPSE = plot indicator.

= 0 implies do not plot far zone scattering pattern in the elevation plane.

= 1 implies plot far zone scattering pattern in the elevation plane.

NDSE = angle increment in degrees for far zone scattering pattern in the elevation plane.

PHSE = phi angle in degrees for far zone scattering pattern in the elevation plane.
THIN = theta angle of the incident wave for bistatic scattering calculations (i.e., ISE = 2 or ISA = 2).

PHIN = phi angle of the incident wave for bistatic scattering calculations.

READ 5 defines the following:

ISA, IPSA, NDSA = same as ISE, IPSE, NDSE but for scattering in the azimuth plane.

THSA = theta angle in degrees for far zone scattering pattern in the azimuth plane.

NOTES:

If ISA or ISE are set to -1 or -2, then the image of the incident wave is included for the azimuth or elevation scattering calculations, respectively. This option is useful for treating problems over an infinite ground plane using image theory. The image of the geometry of the scatterer structure has to be defined in the input file. However, the program defines the image incident wave automatically.

On the same run one can obtain one of each, i.e., either a radiation pattern or a scattering pattern or a bistatic scattering pattern. For each type of calculation one can obtain both polarizations. For more information see the output section of SUBROUTINE SORTB in Chapter IV.

To obtain different patterns of the same antenna or scatterer structure see READ 10 input statement.
3. **READ 6**

READ 6 defines the following:

- **FMC** = frequency in megahertz.
- **CMM** = wire conductivity in megamohms/meter. CMM = -1.0 implies a perfect conductor.
- **A** = the wire radius in meters.

4. **READ'S 7-9**

READ's 7-9 define the plate geometry. In particular READ 7 defines the following:

- **NPLTS** = the total number of plates.

READ 8 defines the following for every plate NPL:

- **NCNRS(NPL)** = the number of corners on plate NPL.
- **SEGM(NPL)** = the maximum segment size of the surface patch monopoles on plate NPL (in wavelengths). It should not exceed 0.25λ and is typically chosen 0.25λ. If more accuracy is needed, **SEGM** can be chosen less than 0.25λ with a substantial sacrifice of computation time since the number of modes increases.
- **IREC(NPL)** = rectangular/polygonal plate indicator for plate NPL.
  - = 0 implies plate NPL is polygonal.
  - = 1 implies plate NPL is rectangular.
IPN(NPL) = polarization indicator. It has meaning only for quadrilateral plates.

= 1 implies modes are to be placed on the quadrilateral plate NPL to cover the first polarization only.

= 2 implies modes are to be placed on the quadrilateral plate NPL to cover the second polarization only.

= 3 implies both polarizations are to be placed on plate NPL. Also for a polygonal plate NPL IPN(NPL) = 3.

The term first polarization implies the current flowing in the direction of side 1-2. The second polarization implies the current flowing in the direction of side 2-3.

IGS(NPL) = number of generating side in SUBROUTINE PLATE3. The generating side is the reference side subroutine PLATE3 uses to divide plate NPL into modes (see description of PLATE3 in chapter 4). Normally IGS(NPL) = 0 which implies that subroutine PLATE3 will use the longest side of the plate NPL as the generating side. However, the ensuing modal segmentation may not be the optimal one in the sense of minimum number of modes and accurate representation of the current
flow on the plate. In such cases the user might want to use a different generating side by setting IGS(NPL) = the side number of the desired reference side, i.e.,

0 < IGS(NPL) < NCNRS(NPL).

The program automatically checks for plates which intersect along a common edge and inserts surface patch overlap modes to ensure the continuity of current along the common edge. If more than two plates intersect along a common edge the program finds the minimum linearly independent set of overlapping plates. For a detailed description of how overlap plate modes are defined see SUBROUTINE POPLOV, Chapter IV.

READ 9 statement inputs the coordinates of the corners of plate NPL. It is executed NCNRS(NPL) times for plate NPL and it defines the following:

PCN(1,NCNR,NPL),(PCN(2,NCNR,NPL),PCN(3,NCNR,NPL) = x,y,z coordinates, respectively, of the corner NCNR (1 < NCNR < NCNRS(NPL)) of plate NPL (in meters).

5. READ 10

At times a user may wish to run several consecutive problems for which the impedance matrix either does not change or only certain blocks of it change. For example, the impedance matrix will not change for the following cases:

1. if different far-zone patterns are desired,
2. if different voltage excitations are used, or
3. if different angles of incidence are used in a bistatic scattering calculation.

Obviously in these cases it would be extremely wasteful to recompute the entire impedance matrix. At other times the geometry may change only slightly from one run to the other. For example, consider the problem of locating a monopole on a ship such that a desired impedance and/or pattern is achieved. In order to solve this problem one would construct a model of the ship from several intersecting plates, possibly requiring hundreds of surface patch modes. One attachment mode would be required where the monopole physically connects to a plate. The user would then analyze this configuration for many monopole locations in search of the optimum location. The impedance matrix of this (and in general of any) MM problem can be visualized as shown in Figure 3-2. As the monopole is moved around, the P/P block of the matrix does not change; only the blocks involving wire and attachments change. Thus a considerable savings in time will result if on the first run the entire matrix is stored on a disk file. On subsequent runs the stored matrix is read in and only the blocks involving wires and attachments are recomputed.

The operation of storing, reading and selecting the blocks of the impedance matrix to be recomputed is controlled by the parameters IWRZM and IROZM. Specifically:

\[
IWRZM = \text{indicator for writing the impedance matrix on a disk file.}
\]

\[
= 0 \text{ implies do not write out the impedance matrix.}
\]
<table>
<thead>
<tr>
<th>W/W</th>
<th>W/P</th>
<th>W/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>P/W</td>
<td>P/P</td>
<td>P/A</td>
</tr>
<tr>
<td>A/W</td>
<td>A/P</td>
<td>A/A</td>
</tr>
</tbody>
</table>

**W** = WIRE  
**P** = PLATE  
**A** = ATTACHMENT

Figure 3-2. Symbolic representation of the nine different blocks of the impedance matrix.
= 1 implies write out the impedance matrix.
IRDZM = indicator for reading in the impedance matrix
calculated during the previous run.
= 0 implies do not read in the previous matrix
and calculate the entire new matrix.
= 1 implies read in the previous matrix and compute
new matrix except for the W/W and A/A blocks.
= 2 implies read in the previous matrix and compute
new matrix except for the P/P block.
= 3 implies read in old matrix and use as new
matrix, i.e., do not calculate any impedance
elements.

NOTES:

Thus IRDZM = 2 if the plate geometry is unchanged from the last
run, IRDZM = 1 if the wire and attachment geometry is unchanged from the
last run and IRDZM = 3 if the entire geometry is unchanged.
Whenever IRDZM>0 the following should be true:
1. IWRZM must be 1 on the previous or first run, and
2. the number of wire, plate and wire attachment modes is
unchanged from the IWRZM = 1 run.

The impedance matrix is read from and is written on the disk file
ZMAT.DAT on logical unit 1.
6. **READ'S 11-15**

The following read statements input the wire geometry of the problem including the impedance loads, voltage generators and wire-to-plate attachments. These read statements are executed only if INWR = 1 and IRGM = 1 (see READ 1). The wire geometry input will be described with the aid of the example shown in Figure 3-3. The structure consists of a T-shaped wire with one load and one generator. The wire is defined by four points, shown as heavy black dots in Figure 3-3, and three wire segments. The wire point numbering scheme shown in Figure 3-3 is arbitrary. The wire point numbers are shown adjacent to the dots and the segment numbers are shown as circled numbers next to the segments.

The following rules apply for wires:

1. The wire geometry consists of interconnected straight wire segments.
2. Each segment should not exceed a quarter wavelength in length.
3. Two intersecting segments should form an acute angle no less than 30 degrees.
4. Single isolated wire segments are not permitted.
5. There is no limit to the number of wire segments intersecting at a given point.

**READ 11** inputs the following parameters:

- \( \text{NM} = \) total number of segments on the wire structure.
- \( \text{NP} = \) total number of points on the wire structure.
Figure 3-3. A wire geometry showing points, segments, a load, a generator and an attachment point.
\( \text{NAT} = \text{total number of wire-to-plate attachment points.} \)

\( \text{NFPT} = \text{total number of feed locations on the wire} \)

\( \text{structure, excluding feeds at wire-to-plate} \)

\( \text{attachment points.} \)

There are times when the mutual coupling between two feed locations (points) on the wire structure is needed. By specifying

\( \text{NFS1} = \text{wire "location" of the first feed point, and} \)

\( \text{NFS2} = \text{wire "location" of the second feed point,} \)

the program will calculate the maximum coupling between feed points \( \text{NFS1} \) and \( \text{NFS2} \). Also, the impedance matrix relating the two feed points is calculated. If no coupling calculations are needed then \( \text{NFS1} = \text{NFS2} = 0. \)

The term wire "location" implies either endpoint A or endpoint B of a particular segment. Specifically, if endpoint A of segment \( L \) is meant then the wire "location" is \( L \). If endpoint B is meant then the wire "location" is \( L + \text{NM} \), where \( \text{NM} \) is the total number of wire points.

For the example of Figure 3-3 READ 11 would be:

\( 3 \ 4 \ 1 \ 1 \ 0 \ 0 \) (no coupling is specified).

READ 12 requires \( NP \) lines of inputs to define the \( x,y,z \) coordinates of every point on the wire structure (in meters):

\( X(I) = \text{the x coordinate of point I.} \)

\( Y(I) = \text{the y coordinate of point I.} \)

\( Z(I) = \text{the z coordinate of point I.} \)
For the geometry of Figure 3-3 the NP = 4 lines of input for READ 12 are:

0.0 0.0 0.0
0.0 0.0 0.25
0.0 0.0 0.5
-0.3 0.0 0.25

READ 13 requires NM lines of input to define the endpoints of every segment. Each segment has two endpoints denoted by A and B. The user can arbitrarily select which end is A and which is B. READ 13 defines:

IA(J) = endpoint A of wire segment J.
IB(J) = endpoint B of wire segment J.

By arbitrarily choosing the endpoint with the smaller point number as A, the NM = 3 lines of input for READ 13 would be:

1 2
2 3
2 4

READ 14 defines for every feed point its wire "location" and the complex value of the generator and load at that location. In this code one always think of generators and loads as being inserted into segments, either by endpoint A or B of the segment. One should not think of feeds as being by a point in the wire. For example, for the geometry of Figure 3-3 it is not sufficient to specify a 50 ohm load by point 2. There are three locations (although physically close, electrically very different) which could be taken as point 2, i.e., endpoint B of segment 1, endpoint A of segment 2, or endpoint A of segment 3. The last location is the correct location specification for the 50 ohm load. READ 14 defines the following:
IFM = segment number of feed point.
IAB = indicator specifying by which endpoint of segment IFM the feed point is located.
   = 0 implies feed point is by endpoint A of segment IFM.
   = 1 implies feed point is by endpoint B of segment IFM.

VLG = complex voltage generator at the feed point (in volts). Positive polarity is from endpoint A to endpoint B of segment IFM.

For the geometry of Figure 3-3 the NFPT = 1 line of input for READ 14 would be:
3 0 (0.0,0.0) (50.0,0.0) .

Note that there is no voltage generator at this wire "location".

READ 15 specifies the wire-to-plate attachment geometry along with the complex values of the generators and loads at the attachment locations. Specifically, READ 15 defines the following for each of the NAT attachments:

NAS = the number of the segment which attaches to the plate.

IAB = indicator specifying which endpoint of segment NAS attaches to the plate.
   = 0 implies endpoint A of segment NAS.
   = 1 implies endpoint B of segment NAS.
NPLA(I) = plate where the attachment point I is located.

VGA(I) = complex voltage generator at attachment point I.

ZLDA(I) = complex impedance loading at the attachment point I.

BDSK(I) = outer disk radius in meters of the disk monopole of the Ith attachment mode.

Experience has shown that for accurate impedance and pattern results the disk radius should be between $0.1\lambda$ and $0.25\lambda$. A good average choice is $0.2\lambda$. Also, the center of the disk should be at least $0.1\lambda$ away from all plate edges.

Assuming a frequency of 300.0 Mhz, READ 15 for the geometry of Figure 3-3 would require NAT = 1 lines of input:

```
1 0 1 (1.0,0.0) (0.0,0.0) 0.2
```

Note that there is no impedance load at the attachment point.

B. SUBROUTINE WGEOM

If INWR = 1 and IRGM = 0 (See READ 1), then the wire and attachment geometry is defined by subroutine WGEOM, which has to be written by the user. The general form of subroutine WGEOM is:
SUBROUTINE WGEOM(IA,IB,X,Y,Z,NM,NP,NAT,NSA,NPLA,VGA,BDSK, ZLDA,NWG,VG,ZLD,WV,NFS1,NFS2)

DIMENSION IA(1),IB(1),X(1),Y(1),Z(1),NSA(1),NPLA(1),BDSK(1)
COMPLEX VGA(1),ZLDA(1),VG(1),ZLD(1)

MAIN PROGRAM
RETURN
END

The following parameters are inputs and are defined in the main program:

NWG = indicator for the number of wire geometries.
WV = wavelength.

The following parameters are outputs:

IA(I) = endpoint A of wire segment I (I = 1,NM).
IB(I) = endpoint B of wire segment I (I = 1,NM).
X(J),Y(J),Z(J) = x,y,z coordinates of point J (J =1,NP).
NM = the total number of wire segments.
NP = the total number of wire points.
NAT = the total number of attachment points.
NSA(K) = wire "location" of attachment point K.
NPLA(K) = the number of the plate where attachment point K is located.
VGA(K) = complex voltage generator at attachment point K.
BDSK(K) = outer disk radius of the disk monopole of attachment mode K.
ZLDA(K) = complex impedance load at attachment point K.
VG(L) = complex voltage generator at wire "location" L.

ZLD(L) = complex impedance load at wire "location" L.

NFS1 = wire "location" of the first feed point.
NFS2 = wire "location" of the second feed point.

The parameters NFS1 and NFS2 are used when the mutual coupling between two feed points on the wire structure is required. When no mutual port coupling calculation is needed then NFS1 = NFS2 = 0.

All of the above outputs must be defined by the user via FORTRAN statements in subroutine WGEOM. Usually WGEOM is written on a separate file and is linked with the main program. This procedure saves compiling time when debugging or changing WGEOM.

1. Example of WGEOM Subroutines:

Consider the problem of the center-fed dipole. If one wants to study different dipole lengths and/or segmentations, it is more efficient to write a subroutine to generate the dipole geometry for arbitrary length and segmentation. An arbitrary dipole consisting of NM segments and DH segment length is shown in Figure 3-4. A subroutine describing the geometry of this arbitrary dipole should define the following parameters:

1. The number of points NP (NP = NM + 1).
2. The segment size DH (DH = H/NM).
Figure 3-4. Segmentation of a straight wire.
3. The coordinates X(I), Y(I), Z(I) of the Ith point, i.e.,
\[ X(I) = 0.0 \]
\[ Y(I) = 0.0 \]
\[ Z(I) = DH*\text{(I-1)} \].

4. The endpoints A and B of segment J, i.e.,
\[ IA(J) = J \]
\[ IB(J) = J + 1 \].

5. If the dipole is to be center fed then NM must be an even number and the generator wire "location" is:
\[ IGN = (\text{NM}/2) + 1 \] or
\[ IGN = \text{NM} + \text{NM}/2 \].

6. There are no attachments, i.e., NAT=0.

7. No coupling calculations are desired, i.e., NFS1=0, NFS2=0.

A possible WGEOM subroutine to handle the center-fed dipole is shown in Figure 3-5. The length of the dipole is set at H = 0.5 meters and the number of segments NM = 4. The advantage of writing a subroutine WGEOM for this problem is that a user can obtain dipoles of different lengths and/or segmentations. This can implemented by simply changing the parameters H and NM. Otherwise, for every different dipole length or segmentation a whole new wire geometry would have to be defined.

As a second example consider the problem of describing a regular polygon loop of arbitrary radius, number of sides and segment size.

Let:
SUBROUTINE WGEOM(IA, IB, X, Y, Z, NM, NP, NAT, NSA, NPLA, VGA, BDSK,
ZLDA, NWG, VG, ZLD, W, NFS1, NFS2)
DIMENSION IA(1), IB(1), X(1), Y(1), Z(1), NSA(1), NPLA(1), BDSK(1)
COMPLEX VGA(1), ZLDA(1), VG(1), ZLD(1)

C GEOMETRY FOR A CENTER FED DIPOLE
C
C SPECIFY H = WIRE LENGTH AND NM = NUMBER OF SEGMENTS
H=0.5
NM=4
C INSURE THAT NM IS EVEN
NM=2*((NM+1)/2)
C THE NUMBER OF POINTS IS
NP=NM+1
C THE SEGMENT SIZE IS
DH=H/NM
C DEFINE COORDINATES OF NP POINTS AND NM SEGMENTS
DO 100 I=1,NP
X(I)=0.0
Y(I)=0.0
Z(I)=(I-1)*DH
IA(I)=I
IB(I)=I+1
100 CONTINUE
C DEFINE GENERATOR LOCATION AND VALUE
IGN=NM/2+1
VG(IGN)=(1.0,0.0)
C INDICATE NO ATTACHMENTS
NAT=0
C INDICATE NO COUPLING
NFS1=0
NFS2=0
RETURN
END

Figure 3-5. A subroutine WGEOM to describe the center fed dipole of Figure 3-4.
$R$ = the loop radius in meters.

$NS$ = the number of sides on the loop.

$SWX$ = the maximum segment size in $\lambda$.

Figure 3-6 shows a hexagon loop with two segments per side. For a general loop subroutine, WGEOM should define the following parameters:

1. The length $SL$ of each side.
2. The number of segments per side ($NMS$) and the length of each segment ($DSL$).
3. The total number of segments ($NM = NMS \times NS$) and the total number of points ($NP = NM$).
4. The $x,y,z$ coordinates of the endpoints of side $I$, i.e.,
   \[X_1 = R \cos(PH_1), \quad PH_1 = (I-1) \times 360/NS\]
   \[Y_1 = R \sin(PH_1)\]
   \[X_2 = R \cos(PH_2), \quad PH_2 = I \times 360/NS\]
   \[Y_2 = R \sin(PH_2)\]
5. The $x,y,z$ coordinates of point $K$ which is point $J$ on side $I$, i.e.,
   \[K = (I-1) \times NMS + J\]
   \[X(K) = X_1 + (J-1) \times DX12, \quad DX12 = (X_2 - X_1)/NMS\]
   \[Y(K) = Y_2 + (J-1) \times DY12, \quad DY12 = (Y_2 - Y_1)/NMS\]
   \[Z(K) = 0.0\]
6. The endpoints $A$ and $B$ of segment $K$ are:
   \[IA(K) = K\]
   \[IB(K) = K + 1 \text{ except } IB(NM) = 1\]
Figure 3-6. Segmentation of a hexagon loop.
7. The feed point is by endpoint A of segment 1, i.e., IGN = 1.
8. There are no attachments, i.e., NAT=0.
9. No coupling calculations are desired, i.e., NFS1=0 NFS2=0.

A possible WGEOM subroutine is shown in Figure 3-7. Here NS = 6, R = 0.3 m and SWX = 0.2λ. The advantage of writing a subroutine WGEOM for this problem is that a user can define regular polygon loops with different radii, number of sides and segment sizes. This can be accomplished by changing only the parameters R, NS and SWX. Also, since SWX is specified in wavelengths the subroutine is frequency independent. This feature is especially desirable for an analysis of the wire antenna over a broad frequency range.

C. ARRAY DIMENSIONS

The array dimensions are defined by DIMENSION and COMPLEX statements at the top of the main program. All arrays have either fixed dimensions, independent of the geometry being run, or are dimensioned according to one of the following dimension indicators:

INM = maximum number of wire segments.
ICJ = maximum number of wire modes.
IPLM = maximum number of plate modes.
IPL = maximum number of plates.
IAT = maximum number of wire to plate attachments.
INP = maximum number of wire points.
SUBROUTINE WGEOM (IA, IB, NM, NP, NAT, NSA, NPLA, VGA, BDSK,
2 ILDA, NGC, VG, ILD, W, NFS1, NFS2)
DIMENSION IA(I), IB(I), X(I), Y(I), X(I), Y(I), Z(I), NSA(I), NPLA(I), BDSK(I)
COMPLEX VGA(I), ILDA(I), VG(I), ILD(I)
C
C GEOMETRY FOR POLYGONAL LOOP
C
C SPECIFY R = LOOP RADIUS IN METERS, NS = NUMBER OF
C SIDES IN POLYGONAL LOOP, AND SWX = MAXIMUM SEGMENT SIZE
C IN WAVELENGTHS.
C
R=0.3
NS=6
SWX=0.2
C
C FIND SL* SIDE LENGTH
PI=4.0*ATAN(1.0)
DPH=2.0*PI/NS
SL=2.0*R*SIN(DPH/2.0)
C
C FIND NMS = NUMBER OF SEGMENTS PER SIDE AND DSL* THE
C SEGMENT LENGTH.
DSL=SL/NMS
DSL=SL/NMS
DSL=SL/NMS
DSL=SL/NMS
C
C FIND NM= THE TOTAL NUMBER OF SEGMENTS AND NP= THE TOTAL
C NUMBER OF POINTS.
NM=NS*NMS
NP=NM
C
C DEFINING NMS POINTS AND SEGMENTS ON EACH OF THE NS SIDES
DO 100 I=1,NS
C
C THE COORDINATES OF THE FIRST END OF SIDE I
PHI=(I-1)*DPH
X1=R*COS(PHI)
Y1=R*SIN(PHI)
C
C THE COORDINATES OF THE SECOND END OF SIDE I
PH2=I*DPH
X2=R*COS(PH2)
Y2=R*SIN(PH2)
C
C EACH POINT ON SIDE I WILL BE
DX12=(X2-X1)/NMS
DY12=(Y2-Y1)/NMS
C
C FROM THE LAST POINT ON SIDE I
DO 200 J=1,NMS
C
C DEFINE THE KTH POINT AND SEGMENT
K=(I-1)*NMS + J
X(K)=X+(J-1)*DX12
Y(K)=Y+(J-1)*DY12
Z(K)=0.0
IA(K)=K
IB(K)=K+1
IF(K.EQ.NMS) IB(K)=1
200 CONTINUE
100 CONTINUE
C
C PLACE A 1 VOLT GENERATOR AT THE X AXIS
IGN=1
VG(IGN)=(1.0,0.0)
C
C INDICATE NO ATTACHMENTS
NAT=0
C
C INDICATE NO COUPLING
NFS1=0
NFS2=0
RETURN
END

Figure 3-7. A subroutine WGEOM to describe the hexagon loop of Figure 3-6.
ITOT = maximum total number of modes (wire + plate + attachment) if full surface patch test plate modes are used (IFIL = 0).

IDZT = maximum number of entries in impedance array ZT(IDZT).

ICC = maximum number of modes if filamentary test plate modes are used (IFIL = 1). ICC is the dimension indicator for impedance matrix ZTF(ICC, ICC).

The dimensions indicators are defined below the DIMENSION and COMPLEX statements and typically have the values:

INM = 491
ICJ = 492
IPLM = 490
IPL = 30
IAT = 2
INP = 493
ITOT = 495
IDZT = (ITOT*ITOT + ITOT)/2
ICC = 360.

Because of limited memory allocation space, when IFIL = 0 then ICC should be set to 1. Similarly, when IFIL = 1, IDZT should be set to 1. Also note that while the number of wire modes can be up to 492, the number of plate modes up to 490 or the number of attachment modes up to...
4, the total number of modes can not exceed 495, if IFIL = 0, or 360 if IFIL = 1.

Two steps are required in order to change the dimensions:
1. Change the appropriate dimension indicator.
2. Re-dimension all the arrays associated with that dimension indicator.

Arrays dimensioned by the same indicator are grouped together and are clearly identified by COMMENT statements at the beginning of the main program.

D. PROGRAM FILE DESCRIPTIONS

The computer code is contained in several files stored on disk in the ElectroScience Laboratory's computer, which is a VAX 11/780 manufactured by DIGITAL EQUIPMENT CORPORATION. A listing of the FORTRAN version of the files follows (except for 'PLOTLIB which is an object file):

STOMM2.FOR - the main program plus all the subroutines except the thin wire subroutines.
THNWRS.FOR - thin wire subroutines.
WGEOM.FOR - subroutine describing the wire structure geometry, written by the user (see section 2.3).
'GRP11LIB - contains various special library subroutines.

At present only the function subroutine GETCP(I) is used where I = clock reading in hundredths of a second. Since
this subroutine tends to be hardware dependent; it is not
included when the program is sent outside the ElectroScience
Laboratory.

GPLOT2.FOR - subroutine for making three-view orthographic
plot of wire and plate geometry.

'PLOTLIB contains various plotting subroutines. When
the program is sent outside the ElectroScience Laboratory
some routines must be omitted due to contactual
restrictions. When this file is supplied to an outside
user it will be called PLOTLIB.FOR. Of the subroutines
omitted, the only four used in the program are:

VPLOTS(I,0,0) - reserves the plotter.

I = 1 implies the plot is for the Versatek paper plotter.

= 2 implies the plot is for the Megatek plotter.

= 0 implies the program gives the user a choice of plotter.

PLOT(X,Y,I) - moves the plotter "pen", with pen up or down.

X,Y = the x,y coordinates of the point where the pen is
going.

I = 2 implies the pen is lowered before moving.

= 3 implies the pen is raised before moving.

= -2 or -3 implies the same as 2 or 3 except that
the origin is reset after moving.

= -999 implies go to the lower left corner of next
page with pen up and reset the origin.
= 999 implies this is the last plotting call, i.e.,
all plotting is terminated by calling
PLOT(X,Y,999).
NUMBER(X,Y,HT,FPN,ANGLE,N) - plots out a floating point number.
X,Y = are the x,y coordinates of the lower corner
of the number in inches.
HT = height of the output number in inches. If
HT>0, then the output is plotted to the right
of X,Y; if HT<0, it will be plotted to the
left of X,Y.
FPN = floating point number to be plotted.
ANGLE = angle in degrees(counter clockwise) with respect
to the X axis at which FPN is to be plotted.
N = integer specifying the output format. If the
absolute value of N is less than 100, then
F.N will be plotted in the "F" format. If N>0
then N digits will be plotted after the decimal
point, in addition to all the digits before the
decimal point. If N<0, then no digits will be
plotted after the decimal point and the decimal
point plus the first -(N + 1) digits to the left
of the decimal point will be suppressed. If the
absolute value of N is larger than 100, then FPN
will be plotted in the "E" format or exponential
scientific format. If N>100, there will be one
digit to the left and \( N - 100 \) digits to the right of the decimal point in the mantissa.

If \( N < -100 \), then the mantissa will be an integer with \( -(N + 100) \) digits.

\[
\text{SYMBOL}(X, Y, HT, \text{LABEL}, \text{ANGLE}, NC) - \text{plots a character or string of characters.}
\]

\[
X, Y = \text{coordinates in inches of the lower left hand corner of the symbol to be drawn.}
\]

\[
HT = \text{the height in inches of the character to be drawn. HT should be a multiple of 7 times the plotter increment.}
\]

\[
\text{LABEL} = \text{if } NC > 0 \text{ LABEL is a literal variable or constant representing the character strung to be plotted. NC is the number of characters to be plotted. If } NC = -1, \text{ then LABEL is an integer expression, ranging from 0 to 127, which represents a single character. These symbol and their codes are shown in Appendix 40.}
\]

\[
\text{ANGLE} = \text{angle in degrees between the symbol to be plotted and the X axis.}
\]

\[
NC = \text{see description of LABEL.}
\]

IF the user cannot supply a subroutine GETCP, then all calls to this subroutine should be deleted and the program will not supply the run time information at the end of the program run. If subroutines VPLOTS, PLOT, NUMBER and SYMBOL are not available in the system, all calls to them should be commented out. If plotting is not desired, then
the calls to subroutine GPLOT2, MPLOT, and MPLOT2 should be removed along with all calls to VPLOTS, PLOT, NUMBER and SYMBOL.

In summary, when the code is supplied outside the ElectroScience Laboratory the following FORTRAN files are included in a single file called OSUESP.FOR:

STDMM2.FOR
THNWRS.FOR
WGEOM.FOR
GPLOT2.FOR
PLOTLIB.FOR .

Note that the WGEOM.FOR supplied is for a dipole antenna. To obtain a new geometry the user must write a new subroutine WGEOM and replace the one supplied.

E. DESIGN EXAMPLES

This section will present several design or example runs illustrating the use of the code. The purpose of the examples is:

1. to illustrate the input data,
2. to illustrate the output data, and
3. to provide trial or debugging runs for a new user.
1. DESIGN EXAMPLE 1

The currents, input impedance and far-zone elevation plane radiation patterns in the \( \phi = 0.0 \) plane for the geometry of Figure 3-3 are desired. The wire is located in the center of a one meter square plate. The frequency is 150 MHz, the wire is made of aluminum (conductivity = 38 megamhos/meter) and the wire radius is 0.001 meter.

The input file for this problem is shown below:

```
READ 1-----1 2 1 1 1 0 4 10 18 1 1 1
READ 2-----1 1 3 0.0
READ 3-----0 1 3 90.0
READ 4-----0 1 3 0.0 90.0 0.0
READ 5-----0 1 3 90.0
READ 6-----150.0 38.0 0.001
READ 7-----1
READ 8-----4 0.25 1 3 0
READ 9-----0.5 -0.5 0.0
  0.5 -0.5 0.0
  0.5 0.5 0.0
-0.5 0.5 0.0
READ 10-----1 0
READ 11-----3 4 1 1 0 0
READ 12-----0.0 0.0 0.0
  0.0 0.0 0.25
  0.0 0.0 0.5
-0.3 0.0 0.25
```
Note that filamentary test plate modes are used, i.e., IFIT = 1 in READ 1.

Before computing any patterns or data the accuracy of the geometry as defined by the input file should be checked. A three-view orthographic plot of the geometry, shown in Figure 3-8, is obtained by setting NGO = 0. The edges of the plate are shown as solid lines. Wire segments are shown as solid lines with small circles representing the endpoints. This plot can be used as a first check of the accuracy of the input file.

The output for this problem is shown in Appendix 1 and could be broken in the following blocks:

a. Input Data

A listing of some of the input quantities such as frequency, wire radius, wire conductivity, integration parameters and the indicator for the type of test plate modes.
Figure 3-8. Three-view plot of the geometry of Example 1.
b. Plate Geometry

For every plate its type is specified (rectangular or polygonal) along with the coordinates of every plate corner, the maximum segment size (SEGM), the polarization indicator (IPN) and the generating side indicator (IGS). Since IWR = 1 a detailed printout of the x,y,z coordinates of every surface patch monopole on the plate is included. Figure 3-9 shows a typical surface patch dipole mode consisting of monopole A and monopole B. Monopole A is defined by the x,y,z coordinates of its four corners A1,A2,A3,A4, while monopole B is defined by its four corners B1,B2,B3,B4. By convention positive current flows from monopole A to monopole B.

c. Wire Geometry

First the x,y,z coordinates of the NP wire points are printed along with the the maximum and minimum number of modes at any point. Since IWR = 1, the wire modal layout is printed by specifying I1, I2, I3, JA, JB for every wire mode. Figure 3-10 shows a typical wire dipole mode defined by points I1, I2,I3 and segments JA and JB. By convention positive current flows from JA to JB. Next the endpoints and length of the NM wire segments are printed.

d. Attachment Geometry

For every attachment point the following parameters are printed:
Figure 3-9. A surface patch dipole mode.

Figure 3-10. A surface patch dipole mode.
SEGMENT = the wire segment which attaches to the plate.
END = same as IAB, the indicator for which wire segment endpoint attaches to the plate.
PLATE = the plate where the wire segment attaches to.
B = outer disk radius of the disk monopole of the attachment mode.

e. Loads And Generators

The wire "location" and complex value of every impedance load and voltage generator is printed. Finally the total number of wire, plate and attachment modes is printed. The output file ends here if NGO = 0. The user should check the output file carefully to make sure everything is defined correctly.

If NGO = 1 the program proceeds with the calculations and outputs the following additional information:

f. Antenna Modal Currents

If IWR = 1, the induced modal currents are printed. For every modal current its relative magnitude (with respect to the largest modal current), its absolute magnitude (in amperes) and its phase (in degrees) are printed.
g. Antenna Impedance And Patterns

The input impedance shown is for a unit voltage generator, i.e.,
\((1 + j0)\) volts. The far-zone patterns are printed as:

\[ G_{\Theta} = \text{the gain in db of the } \Theta \text{ component of the } \]
\[ \text{electric field.} \]

\[ G_{\Phi} = \text{the gain in db of the } \Phi \text{ component of the } \]
\[ \text{electric field.} \]

Figures 3-11a and b are the polar plots of \( G_{\Theta} \) and \( G_{\Phi} \),
respectively. At the end of the file the total CPU time for the run is
printed.

2. DESIGN EXAMPLE 2

The backscattering from the corner reflector of Figure 3-12 is
desired. It consists of two 1.0\( \lambda \) by 0.5\( \lambda \) rectangular plates
intersecting along the z-axis. The pattern is to be taken in the azimuth
plane at \( \theta = 90.0 \) degrees. This is specified in the input file by
setting \( ISA = 1 \) and \( THSA = 90.0 \). The input file for this problem is as
follows:

```
READ 1----1 2 1 1 0 0 4 1 0 1 8 0 1 1
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----0 1 3 90.0 90.0 45.0
READ 5----2 1 3 90.0
READ 6----300.0 -1.0 0.001
```
Figure 3-11. (a) $\theta$ polarized radiation pattern, and

70
Figure 3-11. (b) $\phi$ polarized radiation pattern.
Figure 3-12. Geometry for the corner reflector of Example 2.
READ 7----2
READ 8----4 0.25 1 3 0
READ 9----0.0 0.0 0.0
  0.0 0.0 1.0
  0.5 0.0 1.0
  0.5 0.0 0.0
READ 8----4 0.25 1 3 0
READ 9----0.0 0.0 0.0
  0.0 0.0 1.0
  0.0 0.5 1.0
  0.0 0.5 0.0
READ 10---1 0.

Figure 3-13 shows the three-view orthographic plot, obtained by setting NGO = 0. Also, if IWR = 1, plots of the modal layouts on both plates are obtained (Figures 3-14a and b) along with a plot of the overlap modal layout (Figure 3-14c). The output for this problem is given in Appendix 2. The program automatically inserts the necessary overlap modes between the two plates. Note that after specifying the x,y,z coordinates of the corners of the two plates, the output indicates that four overlap modes were inserted between side 1 of plate 1 and side 1 of plate 2.

Finally, a printout of all the various cross sections is included, i.e.,
Figure 3-13. A three-view plot of the geometry of Example 2.
Figure 3-14. (a) Modal outlay on plate 1 of the corner reflector of Example 1,
4 MODES FOR SECOND POLARIZ.

6 MODES FOR FIRST POLARIZ.

10 TOTAL MODES ON PLATE 2

Figure 3-14. (b) Modal outlay on plate 2 of the corner reflector of Example 2,
4 OVERLAP MODES BETWEEN PLATE 1, SIDE 1 AND PLATE 2, SIDE 1

Figure 3-14. (c) modal outlay on the overlap region between plates 1 and 2 of Example 2.
STTM = scattering cross section with incident and scattered fields theta polarized.
SPPM = scattering cross section with incident and scattered fields phi polarized.
STPM = scattering cross section with incident field $\theta$ polarized and scattered field $\phi$ polarized.
SPTM = scattering cross section with incident field $\phi$ polarized and scattered field $\theta$ polarized.

Both the magnitude (in $\text{db}/\lambda^2$) and the phase (in degrees) are given.
Figures 3-15a and b are the polar plots of the magnitudes of STTM and SPPM, respectively.

3. DESIGN EXAMPLE 3

The bistatic scattering pattern in the azimuth plane from the corner reflector of example 2 is examined. This calculation is specified by setting $\text{ISA} = 2$ and $\text{THSA} = 90.0$ in the input file. The incident field is coming from $\text{THIN} = 90.0$ and $\text{PHIN} = 45.0$. The input file for this problem is:

```
READ 1----1 2 1 1 0 0 4 8 18 0 1 1
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----2 1 3 90.0 90.0 45.0
READ 5----0 1 3 90.0
READ 6----300.0 -1.0 0.001
```
Figure 3-15. (a) $\theta$ polarized azimuth backscattering pattern for Example 2,
Figure 3-15. (b) \( \phi \) polarized azimuth backscattering pattern for Example 2.
READ 7 ---- 2
READ 8 ---- 4 0.25 1 3 0
READ 9 ---- 0.0 0.0 0.0
        0.0 0.0 1.0
        0.5 0.0 1.0
        0.5 0.0 0.0
READ 8 ---- 4 0.25 1 3 0
READ 9 ---- 0.0 0.0 0.0
        0.0 0.0 1.0
        0.0 0.5 1.0
        0.0 0.5 0.0
READ 10 --- 1 0.

The output of this problem (with IWR = 0 and IWRZT = 0) is shown in Appendix 3. Figures 3-16a and b are the polar plots of the magnitude of STTM and SPPM, respectively.

4. DESIGN EXAMPLE 4

This example will illustrate the use of the READ 10 statement to save CPU time for the impedance matrix calculation. Consider the problem of calculating the input impedance of a quarter-wave monopole at two locations on plate 2 of the three-plate bend shown in Figure 3-17a. Location 1 is at \((x,y,z) = (0.0,0.0,0.0)\) and location 2 is at \((x,y,z) = (0.0,0.3,0.0)\). The input file for this problem is:
Figure 3-16. (a) ε polarized azimuth bistatic scattering pattern for Example 3.
Figure 3-16. (b) φ polarized azimuth bistatic scattering pattern for Example 2.
Figure 3-17. (a) Geometry for the three-plate bend of Example 4, (b) Geometry for the wire of Example 4.
READ 1----1 2 1 2 0 0 4 8 18 1 1 0
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----0 1 3 90.0 90.0 45.0
READ 5----0 1 3 90.0
READ 6----300.0 -1.0 0.001
READ 7----3
READ 8----4 0.25 1 3 0
READ 9----0.5 -0.5 -1.0
   0.5 -0.5 0.0
   -0.5 -0.5 1.0
   -0.5 -0.5 0.0
READ 8----4 0.25 1 3 0
READ 9----0.5 -0.5 0.0
   0.5 0.5 0.0
   -0.5 0.5 0.0
   -0.5 -0.5 0.0
READ 8----4 0.25 1 3 0
READ 9----0.25 0.5 0.0
   -0.25 0.5 0.0
   -0.25 0.5 1.0
   0.25 0.5 1.0
READ 10----1 0
READ 11----2 3 1 0 0 0
Note that there is no READ 14 statement since there are no loads or generators in the wire, except the ones at the attachment point.

To find the input impedance at these two locations one sets NWG = 2 indicating that there are two separate wire geometries. For the first wire geometry one sets IWRZM = 1 and IRDZM = 0. Thus the entire impedance matrix will be calculated and then written into file ZMAT.DAT on logical unit 1. For the second geometry IRDZM = 2, indicating that the impedance matrix is to be read in from file ZMAT.DAT and that the P/P block of the matrix is not to be recomputed, since the plate geometry has not changed. IWRZM is set to 0 or 1, depending on whether
READ 1----1 2 1 2 0 0 4 8 1 8 1 1 0
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----0 1 3 90.0 90.0 45.0
READ 5----0 1 3 90.0
READ 6----300.0 -1.0 0.001
READ 7----3
READ 8----4 0.25 1 3 0
READ 9----0.5 -0.5 -1.0
0.5 -0.5 0.0
-0.5 -0.5 1.0
-0.5 -0.5 0.0
READ 8----4 0.25 1 3 0
READ 9----0.5 -0.5 0.0
0.5 0.5 0.0
-0.5 0.5 0.0
-0.5 -0.5 0.0
READ 8----4 0.25 1 3 0
READ 9----0.25 0.5 0.0
-0.25 0.5 0.0
-0.25 0.5 1.0
0.25 0.5 1.0
READ 10----1 0
READ 11----2 3 1 0 0 0
<table>
<thead>
<tr>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.25</td>
</tr>
</tbody>
</table>

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A
Figure 3-18. A three-view plot of the geometry of Example 4.
Figure 3-19. Geometry for the polygon plate of Example 5.
Since \( IWR = 1 \), a detailed plot of the modal layout on the plate is obtained (Figure 3-20). Figure 3-21 is the three-view orthographic plot of the regular octagon of Example 5. The output for this problem is shown in Appendix 5.
6 MODES FOR SECOND POLARIZ.

6 MODES FOR FIRST POLARIZ.

12 TOTAL MODES ON PLATE 1

Figure 3-20. Modal layout on the plate of Example 5.
Figure 3-21. Three-view orthographic plot of the geometry of Example 5.
CHAPTER IV
SUBROUTINE DESCRIPTIONS

Several parameters dealing with the geometrical features of the modes are used in more than one subroutine. For reasons of clarity and brevity they are described in detail here, and when they appear later in a subroutine only a brief title description will be given.

A. GENERAL PARAMETERS

Three types of modes are used in this code; wire dipole, surface patch dipole and attachment dipole. Each dipole mode is composed of two monopoles (see Figure 4-1). A wire dipole consists of two wire monopoles, a surface patch dipole consists of two surface patch monopoles and an attachment dipole consists of a wire monopole and a circular disk monopole. The mutual impedance between two dipole modes is the sum of four monopole-to-monopole impedances. In order to evaluate a particular monopole-to-monopole impedance the code needs to know the type of monopoles involved.

IOP = test monopole type indicator.

= 1 implies test monopole is a surface patch.
Figure 4-1. (a) Wire dipole mode, (b) arbitrary quadrilateral surface patch dipole mode, and (c) wire attachment dipole mode.
= 2 implies test monopole is a disk.
= 3 implies test monopole is a wire.
= expansion monopole type indicator.
= 1 implies expansion monopole is a surface patch.
= 2 implies expansion monopole is a disk.
= 3 implies expansion monopole is a wire.

**IM12** = polarity indicator for the direction of current flow on
a test monopole. Consider the three types of modes
depicted in Figure 4-1 with their terminals and ends
clearly identified. The arrows indicate the direction of
positive current flow on the modes.
= 1 implies positive current flows from terminal to end.
= -1 implies positive current flows from end to terminal.

**IN12** = polarity indicator for the direction of current flow on
an expansion monopole.
= 1 implies positive current flows from terminal to end.
= -1 implies positive current flows from end to terminal.

For monopoles A of the dipoles in Figures 4-1(a), 4-1(b),
4-1(c), IM12 (or IN12) is -1, -1 and +1, respectively. For
monopoles B of the same dipoles, IM12 (or IN12) is +1, +1
and -1, respectively.

To accurately represent the current density on a plate we need two
orthogonal current polarizations. For a quadrilateral plate it is often
sufficient to place modes to cover only one current polarization. For
example, consider two overlapping quadrilateral plates such that their touching sides coincide with the overlap segment, i.e., the segment common to both plates. One can place modes on the plates to cover only the polarization parallel to the overlap segment and let the overlap modes cover the other polarization.

\[ \text{IPN}(J) = 0 \] implies place no modes on plate \( J \).

\[ = 1 \] implies place modes on plate \( J \) to cover the first current polarization only.

\[ = 2 \] implies place modes on plate \( J \) to cover the second current polarization only.

\[ = 3 \] implies place modes on plate \( J \) to cover both current polarizations.

For a non-rectangular plate \( \text{IPN}(J) = 3 \).

B. WIRE PARAMETERS

Every wire dipole mode is composed of two wire segments or monopoles. A wire segment is defined by the \( x,y,z \) coordinates of its two endpoints as shown Figure 4-2. The first endpoint to be defined is \( A \) and the second one is \( B \).

\[ \text{IA}(I) = \text{the number of point } A \text{ of segment } I. \]

\[ \text{IB}(I) = \text{the number of point } B \text{ of segment } I. \]

A wire dipole mode is composed of two wire segments or monopoles (see Figure 4-3).
Figure 4-2. Wire segment I.

Figure 4-3. Wire dipole mode J.
I1(J) = number of endpoint 1 of wire dipole mode J.
I2(J) = number of terminal point of wire dipole mode J.
I3(J) = number of endpoint 2 of wire dipole mode J.
JA(J) = first segment of wire dipole mode J.
JB(J) = second segment of wire dipole mode J.
MD(J,K) = array containing list of wire dipole modes sharing wire segment J.

All dipoles modes used in this code are sinusoidal, involving the free space propagation constant $\Gamma$ which in general is defined as follows:

$$\Gamma = \sqrt{(\sigma + j\omega\varepsilon)(j\omega\mu)}$$

but $\sigma = 0$, $\varepsilon = \varepsilon_0$, $\mu = \mu_0$ in this work; so

$$\Gamma = -j\omega\sqrt{\mu_0\varepsilon_0}. \quad (4.1)$$

$\eta$ = complex impedance of the medium

$$\eta = \sqrt{j\omega\mu/\sigma + j\omega\varepsilon}$$

but $\sigma = 0$, $\varepsilon = \varepsilon_0$, $\mu = \mu_0$ in this work; so

$$\eta = \sqrt{\mu_0/\varepsilon_0}. \quad (4.2)$$

D(I) = length of wire segment I.
SGD(I) = sinh($\Gamma$D(I)).
CGD(I) = cosh($\Gamma$D(I)).
Whenever we include the generators or loads of the wire structure of an antenna it is necessary to know at which wire segment and by what endpoint they are located. If a generator or load is by endpoint A of segment J then its "location" on the wire structure is J. If a generator or load is located by endpoint B of segment J then its "location" on the wire structure is J + NM, where NM is the total number of wire segments.

C. PLATE MODE PARAMETERS

A general plate mode is composed of two arbitrary quadrilateral surface patch monopoles as shown in Figure 4-1. Two types of surface patch monopoles are used in this code; a rectangular surface patch and a quadrilateral (but not rectangular) surface patch (see Figure 4-4).

The following parameters deal with the geometrical features of the surface patch monopole:

IACM = monopole shape indicator for identifying the type of monopoles of a particular test plate dipole mode.

= -3 implies both monopoles of the dipole mode are rectangular surface patches.

= 0 implies either or both monopoles of the dipole mode are quadrilateral surface patches.
Figure 4-4. (a) Rectangular surface patch monopole, and (b) quadrilateral surface patch monopole.

Figure 4-5. Disk monopole on a plate; BDSK = ρ and A = ρ'.

100
IACN = monopole shape indicator for identifying the type of monopoles of a particular expansion plate dipole mode.

-3 implies both monopoles of the dipole mode are rectangular surface patches.

= 0 implies either or both monopoles of the dipole mode are quadrilateral surface patches.

A rectangular monopole is defined by three consecutive corners as shown in Figure 4-4(a). The number in parantheses indicates an equivalent way of defining the monopole. In either case side 2-3 is the end side and point 1 is on the terminal side. Positive current flows from the terminal to the end side. A quadrilateral surface patch monopole is defined by the x, y, z coordinates of its four corners as shown in Figure 4-4(b). Note that side 1-2 is the terminal side and side 2-3 is the end side. Again positive current flows from the terminal side to the end side.

D. ATTACHMENT MODE PARAMETERS

An attachment mode is composed of a wire monopole and a disk monopole. The wire monopole is defined by the x, y, z coordinates of its two endpoints as described for a wire segment above.

The disk monopole is defined (see Figure 4-5) by the x, y, z coordinates of its center and two points on its plane. Normally those two points are the first two corners of the plate the disk lies on.
The following parameters deal with the geometry of the disk monopole:

\[ \text{BDSK}(I) = \text{outer radius of the disk monopole } I. \]

From now on the term disk monopole I will imply the disk monopole of attachment dipole mode I.

\[ A = \text{inner radius of the disk monopole. Normally } A \text{ is the wire monopole radius.} \]

The following parameters deal with the electric field of a disk monopole parallel to a surface patch monopole:

\[ \text{ERVSR}(K,JJ) = \text{array containing values of the radial component of the electric field of disk monopole } K \text{ versus the radial distance } \rho. \]

\[ \text{RMIN}(K) = \text{the minimum distance between the center of disk monopole } K \text{ and any point on the surface patch monopole.} \]

\[ \text{RMAX} = \text{the maximum distance from the disk monopole center to any point on the surface patch monopole.} \]

\[ \text{DR}(K) = \text{the increment in the value of } \rho, \text{ i.e., } \rho = \text{RMIN}(K) + \text{DR}(K) \times JJ \text{ where } JJ \text{ is the JJ-th point the electric field is evaluated.} \]

\[ \text{DIST} = \text{the distance between the planes of the disk and surface patch monopoles.} \]

The term attachment or wire attachment point implies the point where a wire segment attaches to a plate. The number of attachment points is the same as the number of attachment dipole modes.
E. NOTES ON THE IMPEDANCE MATRIX

The general expression for an impedance matrix element is given by Equation (2.9). Since every test and expansion mode is made up of two monopoles the mutual impedance between two dipole modes is the sum of four monopole-to-monopole impedances. In particular

\[ Z_{mn} = Z_{t1e1} + Z_{t1e2} + Z_{t2e1} + Z_{t2e2} \]

where \( t_1 \) and \( e_1 \) refer to monopole 1 of the test and expansion dipole modes, respectively. Similarly, \( t_2 \) and \( e_2 \) refer to monopole 2 of the test and expansion modes, respectively.

When full surface patch test dipole modes are used (IFIL = 0), the impedance matrix is symmetric because the test and expansion modes are identical (Galerkin's method). Because of the symmetry only the lower triangular part of the matrix is calculated. An impedance matrix element \( Z_{mn} \) is stored in the linear array \( Z_T(K) \) at location \( K = (n-1)(NT) - (n-m)/2 + m \). \( NT \) is the total number of dipole modes in the problem.

If the surface patch test modes are represented as single filaments (IFIL = 1), the impedance matrix is no longer symmetric and the whole matrix has to be calculated. An impedance matrix element \( Z_{mn} \) is stored in the two dimensional array \( Z_{TF}(M,N) \).
F. MISCELLANEOUS NOTES:

A common block defining several parameters used by most of the subroutines in the code is defined in the main program. It has the following form:

```
COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
```

where

- **WV** = wavelength in meters.
- **PI** = 3.141592
- **A** = wire radius in meters.
- **Q** = 0.001*WV.
- **GAM** = complex free space propagation constant (0.0 - j2*PI/WV).
- **ETA** = complex free space impedance (376.7 + j0.0).
- **XK** = free space wavenumber (2*PI/WV).

The following are general comments about the program:

1. All values of lengths and distances are in meters (m) unless otherwise noted.
2. All values of angles are in degrees unless otherwise noted.
3. Many numerical integrations are done using Simpson's rule integration. The number of Simpson's rule integration intervals, specified with such parameters as INTP, NPT, INTD and NINT, should always be an even number.
4. All the subroutines explained in this chapter are listed in the Appendix section in the order that they appear here. The subroutine listings are Appendices 6 through 40.

5. The far zone electric field of a mode is a function of the spherical coordinates $r$, $\theta$ and $\phi$. In particular, $E(r,\theta,\phi)$ can be written as

$$E(r,\theta,\phi) = e^{-jkr} \frac{E_f(\theta,\phi)}{r}.$$

6. Whenever the far field of a monopole is mentioned in this code it is assumed that we mean $E_f(\theta,\phi)$ and that the $\exp(-jkr)/r$ dependence is suppressed.

7. The expressions for the near zone fields of PWS monopole do not include the contributions from the point or line charges at the endpoints of the monopole, since these charges disappear when two monopoles are connected to form a dipole.

The following subroutines dealing with wire monopoles are included with permission of Professor J. H. Richmond:

SORT,SGANT,CBES,DSHELL,GGS,GGMM,EXPJ,GANT1,SQROT,GFF.

They are documented in Richmond, J. H., "Computer Program for Thin-Wire Structures in a Homogeneous Conducting Medium", Report 2902-12, August 1973, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering, Columbus, Ohio.
1. SUBROUTINE PLPLCK

PURPOSE:

Subroutine PLPLCK checks every plate to ensure that all of its corners lie on the same plane.

GENERAL FORM:

PLPLCK(PCN,ICN,IPL,NC,TOUCH,NP,IOK).

THE FOLLOWING ARE INPUTS:

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of plate K.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

NC = number of plate corners.

TOUCH = touch indicator. If the distance of a plate corner from the plane of the plate (defined below in the NOTES section) is greater than TOUCH/2 then the geometry of the plate is being defined incorrectly in the INPUT file.

NC = plate number.
THE FOLLOWING ARE OUTPUTS:

IOK = status indicator. If IOK = 0 the run is aborted and PLPLCK returns to the main program.

NOTES:

Subroutine PLPLCK defines the x-y plane by corners 1, 2 and NC of the plate. Then for every corner IC 1, 2, NC its distance ZP from the x-y plane is evaluated. If ZP is larger than TOUCH/2, the run is aborted and an error message is printed. If ZP is less than TOUCH/2, corner IC is redefined as the projection of the old IC on the x-y plane.

2. SUBROUTINE PLATE3

PURPOSE:

Subroutine PLATE3 generates the modal layout on a polygonal plate. The only restriction is that the plate does not have more than one interior angle greater than 180 degrees.

GENERAL FORM:

PLATE3(PC, NC, ICN, NP, NDNPLT, PA, PB, IPLM, SEGM, IQUAD, WV, IRE, IP, MPL1, MPL2, IOK, NM12, NM23, IGS).

THE FOLLOWING ARE INPUTS:

PC(I,J) = x,y,z coordinates (I = 1,2,3) of the J-th plate corner.

NC = number of plate corners.
ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NP = plate number.

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

SEGM = the maximum permissible segment size for a surface patch monopole side. Normally SEGM = 0.25*WV for accurate calculation of the impedance matrix elements.

WV = wavelength.

IRE = rectangular/polygonal plate indicator.

0 implies the plate is polygonal.

1 implies the plate is rectangular.

IP = polarization indicator, same as IPN(NP).

IGS = generating plate side indicator. If IGS is an integer greater than 0 but less than NC, it specifies the number of the plate side to be used as the generating side. If IGS = 0, then PLATE3 chooses the largest plate side as the generating side.

THE FOLLOWING ARE OUTPUTS:

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the J-th corner of monopole A of the I-th plate mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the J-th corner of monopole B of the I-th plate mode.

IQUAD(I) = monopole shape indicator for identifying the type of monopoles of plate mode I.
MPL1 = the total number of modes covering the first polarization.
MPL2 = the total number of modes on the plate.
I0K = status indicator. If I0K = 0, the run is aborted and an error message is printed.
NM12 = the number of segments on the 1-2 side of a rectangular plate (if IRE = 1).
NM23 = the number of segments on the 2-3 side of a rectangular plate (if IRE = 1).

NOTES:

Consider the arbitrary polygon plate shown in Figure 4-6. First, PLATE3 checks the number of interior angles larger than 180 degrees and also finds the length DMX of the longest side ISMX with endpoints IC and IC2. If there is more than one interior angle greater than 180 degrees, PLATE3 returns to the main program.

Side ISMX is the generating side. PLATE3 moves from IC until it finds the next corner different from IC2 and repeats the same procedure from corner IC2. The resulting quadrilateral is defined by corners IC, IC2, B, A. Let NAS and NBS be the required number of segments along DA and DB, respectively. If NAS - NBS is larger than MDM, a constant specified at the beginning of PLATE3, then either side DA or DB is made shorter until NAS - NBS is less than MDM. The purpose of this test is to minimize the number of modes on the plate. This procedure is repeated until the polygon plate is broken into quadrilaterals. Triangles are
Figure 4-6. A three-side section of an arbitrary polygonal plate.
treated as quadrilaterals by defining a fourth corner at the midpoint of their longest side.

Finally, the modes along the direction of ISMX are defined followed by the orthogonal set of modes.

3. SUBROUTINE POPLOV

PURPOSE:

Whenever two or more plates are touching, overlap modes are needed to allow for a continuous current at the plate-to-plate junctions. Subroutine POPLOV sets up those overlap modes and for a multiplate junction finds the minimum linearly independent set of overlap modes.

GENERAL FORM:

POPLOV(NPLTS, PCN, NCNRS, TOUCH, SEGM, PA, PB, NOVT, NPLTM, IPL, IPLM, ICN, IOVT, DOVL, ITK, NOPL, IQAD, WV, NDNPLT, OVEP).

THE FOLLOWING ARE INPUTS:

NPLTS = the total number of plates.
PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.
NCNRS(I) = the number of corners on plate I.
TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.
SEG M = the maximum permissible segment size for a surface patch monopole side. Normally SEGM = 0.25*WV for accurate calculation of the impedance matrix elements.

NPLTM = total number of plate modes.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NDNPLT(I) = total number of plate modes through plate I.

WV = wavelength.

THE FOLLOWING ARE OUTPUTS:

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch mode.

NOVT = total number of overlap modes.

IOVT(I,J) = indicator for identifying the plates or common sides of the overlap plate pair I.

IOVT(I,1) = plate A of pair I.

IOVT(I,2) = side A of pair I.

IOVT(I,3) = plate B of pair I.

IOVT(I,4) = side B of pair I.
DOVL(I) = length in meters of the segment common to both plates of the overlap pair I.

ITK(I) = number of overlap modes in overlap pair I.

NOPL = total number of overlap plate pairs.

IQUAD(J) = monopole shape indicator for identifying the type of surface patch monopoles of overlap plate mode J.

For more explanations refer to Figure 4-7.

NOTES:

Subroutine POPLOV is divided into three sections. Section 1 (lines 18 through 107) determines the existence of all overlap plate pairs. An overlap pair is defined by two touching plates as if those were the only overlapping plates. Section 1 calculates and stores the parameters NOPL, IOVT, DOVL, OVEP and ITK.

Section 2 (lines 110 through 319) eliminates unnecessary overlap modes by checking for linear dependencies. All plate overlap pairs along the same overlap line are compared against its others three at a time. Overlap pair I is defined in DO 170 loop, overlap pair J is defined in DO 175 loop and overlap pair K is defined in DO loop 180.

The first criterion for a linear dependency is that plate A, side A of pair I be the same as plate A, side A of pair J; plate B, side B of pair I be the same as plate A, side A of pair K; and plate B, side B of pair J be the same as plate B, side B of pair K.
Figure 4-7. The overlap region between touching polygonal plates.
The second criterion for linear dependency is that the overlap modes of at least two of the three overlap pairs must mesh together. If the overlap segment of all three pairs is the same and at least two of the three overlap pairs contain the same number of modes, a linear dependency exists and one overlap pair can be eliminated. If that test fails, then in order to have a linear dependency at least two of the three overlap mode lengths must be the same and the endpoints of each overlap segment must coincide with an endpoint of an overlap mode in the other two overlap pairs. An overlap mode length is defined as the overlap segment length divided by the number of overlap modes. If it is determined that one of the pairs can be eliminated, say I, then ITK(I) = 0.

Section 3 (lines 320 through 377) constructs the overlap modes. The coordinates of the modes are stored in arrays PA and PB and the monopole shape indicators for every mode are stored in array IQUAD.

4. SUBROUTINE FGPOV

PURPOSE:

Subroutine FGPOV finds the four points that define the overlap regions on both plates (See Figure 4-7) of an overlap plate pair. All overlap modes lie within those regions. FGPOV also finds NOV, the minimum number of overlap modes needed to cover that region.
GENERAL FORM:

FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,SEGM,NDNPLT,PA,PB,
IPLM,WV,TOUCH,TPSI,SPSI,OMSP,NOV).

THE FOLLOWING ARE INPUTS:

NPA = plate A of the overlap plate pair.
ISA = side A of the overlap plate pair.
NPB = plate B of the overlap plate pair.
ISB = side B of the overlap plate pair.

OE(I,J) = x,y,z coordinates (I = 1,2,3) of the two endpoints
(J = 1,2) of the common overlap segment.

DOV = length of the common overlap segment.

NCNRS(I) = number of corners on plate I.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner
of the Kth plate.

ICN = dimension indicator for the maximum number of plate
corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for the maximum number of plates used
in array PCN(3,ICN,IPL).

SEGM = the maximum permissible segment size for a surface
patch monopole side. Normally SEGM = 0.25*WV for accurate
calculation of the impedance matrix elements.

NDNPLT(I) = total number of plate modes through plate I.

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of
monopole A of the Ith surface patch monopole.
PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch monopole.

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

WV = wavelength.

TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.

TPSI = cos(ψ)

SPSI = sin(ψ) (Refer to Figure 4-8).

THE FOLLOWING ARE OUTPUTS:

OMSP(I,J,K) = x,y,z coordinates (I = 1,2,3) of the four corners (J = 1,2,3,4) defining the overlap region on plate NPA(K = 1) or NPB(K = 2).

NOV = number of overlap modes between plate NPA and NPB.

NOTES:

First, preliminary values of OMSP are found from the plate corners and calls to subroutine FMDC. In most cases the final value for two of the OMSP points on a plate are given by the overlap segment endpoints. For the remaining points consider the case in Figure 4-8 where OMSP(I,4,1) is determined. The preliminary value of OMSP(I,4,1) is given by point 1. If OE(I,1) is also a corner of plate A or if the
Figure 4-8. Definition of the overlap region corners.
angle $\theta'$ is greater than $\psi$ and the distance between point 1 and $OE(I,1)$ is less than $SEGM$, then $OMSP(I,4,1)$ is point 1.

Otherwise $OMSP(I,4,1)$ is moved along the vector $CA$ until it reaches point 2 at the line $CMD$, which is at an angle $\psi$ with the overlap line. If the distance between point 2 and $OE(I,1)$ is larger than $SEGM$, $OMSP(I,4,1)$ continues moving along vector $CA$ until the distance between $OMSP(I,4,1)$ and $OE(I,1)$ is equal to $SEGM$ (see point 3). All remaining $OMSP$ points are found using the same procedure.

5. SUBROUTINE FMDC

PURPOSE:

Subroutine FMDC finds the preliminary value of the overlap region corner $OMSP(I,MC,IAB)$ which is point 1 of Figure 4-8.

GENERAL FORM:

$FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,NC,NAC,MC,NP,IAB,CE,TOUCH,OMSP,WV,NPO)$.

THE FOLLOWING ARE INPUTS:

$NDNPLT(I)$ = Total number of plate modes through plate $I$.
$PCN(I,J,K)$ = $x,y,z$ coordinates ($I = 1,2,3$) of the Jth corner of the Kth plate.

ICN = dimension indicator for the maximum number of plate corners used in array $PCN(3,ICN,IPL)$. 
IPL = dimension indicator for the maximum number of plates used in array PCN(3, ICN, IPL).

PA(I, J, K) = x, y, z coordinates (K = 1, 2, 3) of the Jth corner of monopole A of the Ith surface patch monopole.

PB(I, J, K) = x, y, z coordinates (K = 1, 2, 3) of the Jth corner of monopole B of the Ith surface patch monopole.

IPLM = dimension indicator for the maximum number of plate mode used in arrays PA(IPLM, 4, 3) and PB(IPLM, 4, 3).

NC = plate corner that lies on the overlap line.

NAC = plate corner adjacent to NC but not on the overlap line.

MC = overlap region corner to be defined.

NP = plate number.

IAB = plate A (IAB = 1) or plate B (IAB = 2).

CE = directional cosine of the overlap line.

TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.

NPO = number of the other plate in the overlap pair.

For further explanation see Figure 4-9.

THE FOLLOWING ARE OUTPUTS:

OMSP(I, J, K) = x, y, z coordinates (I = 1, 2, 3) of the four corners (J = 1, 2, 3, 4) of the overlap region on plate NPA (K = 1) or NPB (K = 2).
Figure 4-9. Definition of the preliminary overlap region corners.
NOTES:

Subroutine FMDC searches through all of the modes on plate NP and determines the mode which minimizes the distance between corner 2 or 3 of the monopole A or B and the plate corner PCN(I,NC,NP). OMSP(I,MC,IAB) is then given by corner 1 or 4 of the same monopole.

6. SUBROUTINE MOPLT

PURPOSE:

Subroutine MOPLT gives an orthographic plot of two touching plates and of the overlap modes existing between the two plates. The plot indicates the plate numbers and the plate side numbers of the sides along the overlap line. The orthographic plot is what one would see if he unfolded the two plates so that they lie on the same plane.

GENERAL FORM:

MOPLT(PCN,NCNRS,IPL,ICN,PA,PB,IPLM,IVT,ITK,NOPL,NPLTM,NOVT)

THE FOLLOWING ARE INPUTS:

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.
NCNRS(I) = number of corners on plate I.
IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

122
ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith mode.

IPLM = dimension indicator for maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

IOVT(I,J) = indicator for identifying the plates or common sides of the overlap plate pair I.

IOVT(I,1) = plate A of pair I.

IOVT(I,2) = side A of pair I.

IOVT(I,3) = plate B of pair I.

IOVT(I,4) = side B of pair I.

ITK(I) = number of overlap modes in group I.

NOPL = total number of plate overlap plate pairs.

NPLTM = total number of plate modes.

NOVT = total number of overlap plate modes.

See Figure 3-14(c) for an example.

7. SUBROUTINE MPLOT

PURPOSE:

Subroutine MPLOT plots the modal layout of a particular plate as it is defined by subroutine PLATE3.
GENERAL FORM:

MPLT(NCNRS,PCN,NPL,ICN,IPL,IPLM,NPL11,NPL22,NDNPLT, PA,PB,IPN)

THE FOLLOWING ARE INPUTS:

NCNRS(I) = the number of corners of plate I.

PCN(K,I,J) = x,y,z coordinates (K=1,2,3) of the Ith corner of
the Jth plate.

NPL = the total number of plates.

ICN = dimension indicator for maximum number of plate corners
used in array PCN(3,ICN,IPL).

IPL = dimension indicator for maximum number of plates used in
array PCN(3,ICN,IPL).

IPLM = dimension indicator for maximum number of plate modes
used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

NPL11(I) = the total number of modes covering the first
polarization on plate I.

NPL22(I) = the total number of modes on plate I.

NDNPLT(I) = the total number of modes through plate I.

PA(I,J,K) = x,y,z coordinates (K=1,2,3) of Jth corner of
monopole A of the Ith plate dipole mode.

PB(I,J,K) = x,y,z coordinates (K=1,2,3) of Jth corner of
monopole B of the Ith plate dipole mode.

IPN(I) = polarization indicator for plate I.
NOTES:

Subroutine MPLOT draws two outlines of the plate in consideration on the same page. Then using arrays PA and PB it draws the modal grid on both outlines. The bottom grid represents the modes covering the other polarization. Each mode is identified by drawing an arrow from monopole A to monopole B of the mode.

Also on the same page MPLOT outputs the total number of modes for every polarization and the total number of modes on the whole plate. See Figure 3-14(a) and (b) for an example.

8. SUBROUTINE GPNLOT

PURPOSE:

Subroutine GPNLOT gives an othographic plot of the antenna or scatterer geometry. In particular, it gives a projected view of the geometry as seen along the x, y and z axis. Plate sides are shown in solid lines and wire segments are shown as solid lines with small circles at the endpoints. GPNLOT also gives a summary of the wire, plate and attachment modes of the geometry as well as a scale indicating what one inch is in wavelengths.

GENERAL FORM:

GPNLOT2(NM,NP,X,Y,Z,IA,IB,NPLTS,PCN,IPL,NWR,NPLTM,NAT, WV,ICN,NCNRS)
THE FOLLOWING ARE INPUTS:

\[ \text{NM} = \text{the total number of wire segments.} \]
\[ \text{NP} = \text{the total number of wire points.} \]
\[ X(I), Y(I), Z(I) = x, y, z \text{ coordinates of the } I^\text{th} \text{ wire point.} \]
\[ I_A(J) = \text{endpoint A of wire segment } J. \]
\[ I_B(J) = \text{endpoint B of wire segment } J. \]
\[ \text{NPLTS} = \text{the total number of plates.} \]
\[ PCN(I,J,K) = x, y, z \text{ coordinates }(I = 1, 2, 3) \text{ of the } J^\text{th} \text{ corner of plate } K. \]
\[ \text{IPL} = \text{dimension indicator for the maximum number of plates used in array } PCN(3, ICN, IPL). \]
\[ \text{NWR} = \text{total number of wire modes.} \]
\[ \text{NPLTM} = \text{total number of plate modes.} \]
\[ \text{NAT} = \text{total number of wire attachments.} \]
\[ WV = \text{wavelength.} \]
\[ ICN = \text{dimension indicator for the maximum number of plate corners used in array } PCN(3, ICN, IPL). \]
\[ NCNRS(I) = \text{number of corners of plate } I. \]

See Figure 3-13 for an example.
9. SUBROUTINE ZTOT

PURPOSE:

Subroutine ZTOT evaluates the Moment Method impedance matrix elements Zmn, described by Equation (2.9). Every Zmn is the sum of four monopole-to-monopole impedances which are calculated i., ZTOT by calling the appropriate monopole-to-monopole impedance subroutines.

If full test surface patch monopoles are used (IFIL = 0), then the impedance matrix can be visualized as shown below:

\[
\begin{array}{ccc}
| W/W | W/P | W/A | W = Wire \\
| P/W | P/P | P/A | P = Plate \\
| A/W | A/P | A/A | A = Attachment.
\end{array}
\]

Subroutine ZTOT evaluates only the lower triangular part of the matrix and stores the entries in the linear array ZT(MN) where MN = (N-1)*NTOT - (N*N-N)/2.0 + M.

If filamentary test surface patch monopoles are used (IFIL = 1), then the impedance matrix reduces to

\[
\begin{array}{ccc}
| W/W | W/A | \\
| A/W | A/A | \\
| - | - |
\end{array}
\]

In this case the whole matrix is evaluated and its entries are stored in the two dimensional array ZTF(M,N).
GENERAL FORM:

\[
\text{ZTOT}(IA, IB, INM, I1, I2, I3, JA, JB, MD, NWR, ND, NM, NP, CGD, SGD, D, X, Y, Z, ZLD, NPL, NAT, ZS, IDZM, ZLDA, PA, PB, NSA, NPLA, PCN, IPL, IPLM, BDSK, ZT, ZTF, NM12N, NM23N, ICM, NONPLT, NOVT, INT, INTP, INTD, CMM, ERVSR, RMIN, DR, IAT, IPN, IQUAD, NCNRS, IFIL, IREC, ICC)
\]

THE FOLLOWING ARE INPUTS:

\begin{align*}
IA(I) &= \text{endpoint A of wire segment I.} \\
IB(I) &= \text{endpoint B of wire segment I.} \\
INM &= \text{dimension indicator for array MD(INM,4).} \\
I1(J) &= \text{endpoint 1 of wire dipole J.} \\
I2(J) &= \text{terminal point of wire dipole J.} \\
I3(J) &= \text{endpoint 2 of wire dipole J.} \\
JA(J) &= \text{segment A of wire dipole mode J.} \\
JB(J) &= \text{segment B of wire dipole mode J.} \\
MD(J,L) &= \text{list of wire dipoles modes sharing wire segment J.} \\
NWR &= \text{total number of wire modes.} \\
ND(J) &= \text{total number of wire dipoles modes sharing wire segment J.} \\
NM &= \text{total number of wire segments.} \\
NP &= \text{total number of wire points.} \\
D(J) &= \text{length of wire segment J.} \\
CGD(J) &= \text{cosh(GAM*D(J))}. \\
SGD(J) &= \text{sinh(GAM*D(J))}. \\
X(I), Y(I), Z(I) &= \text{x, y, z coordinates of wire point I.} \\
ZLD(II) &= \text{complex impedance load at wire "location" II.} \\
NPL &= \text{total number of plates.}
\end{align*}
NAT = total number of wire attachments points.

ZS = complex wire surface impedance.

IRDZM = read indicator.

IRDZM = 0 implies do not read in existing impedance matrix and calculate the whole new impedance matrix.

IRDZM = 1 implies read in the matrix and calculate the whole new impedance matrix except the W/W block.

IRDZM = 2 implies read in the matrix and calculate the whole new impedance matrix except the P/P block.

IRDZM = 3 implies use existing impedance matrix.

ZLDA(K) = complex impedance load at attachment K.

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch mode.

NSA(K) = the wire segment "location" of wire attachment K.

NPLA(K) = number of plate where wire attachment K is located.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

BDSK(K) = outer radius of disk monopole of Kth attachment mode.
NM12N(J) = the total number of modes in the 1-2 direction on rectangular plate J.

NM23N(J) = the total number of modes in the 2-3 direction on rectangular plate J.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN, IPL).

NDNPLT(J) = total number of plate modes through plate J.

NOVT = total number of overlap modes.

INT = number of Simpson's rule intervals used in the filament to filament integrations.

INTP = number of Simpson's rule intervals used in integrating over the surface patch expansion monopoles.

INTD = number of Simpson's rule integration intervals used in integrating over the disk expansion monopoles.

CMM = wire conductivity in megamhoms/meter. If CMM = -1.0, this implies perfect conductivity.

ERVSR(K,JJ) = array containing the values of the $p$ component of the electric field of disk monopole K versus the radial distance $p$.

RMIN(K) = the minimum value of $p$ corresponding to ERVSR(K,1).

DR(K) = the increment in the value of $p$, i.e., $p = RMIN(K) + DR(K) \times JJ$.

IAT = dimension indicator for the maximum number of attachments used in array ERVSR(IAT,400).
IPN(NPL) = polarization indicator for plate NPL.

IQUAD(J) = monopole shape indicator for identifying the type of surface patch monopoles composing plate mode J.

NCNRS(NPL) = total number of corners on plate NPL.

IFIL = indicator for choosing either full surface patch test monopoles (IFIL = 0) or filamentary surface patch test monopoles (IFIL = 1).

IREC(I) = rectangular/polygonal plate indicator.

= 0 implies plate I is polygonal (not rectangular).

= 1 implies plate I is rectangular.

ICC = Dimension indicator for array ZTF.

THE FOLLOWING ARE OUTPUTS:

ZT(MN) = complex impedance linear array used when IFIL = 0.

ZTF(M,N) = complex impedance matrix used when IFIL = 1.

NOTES:

Because of the importance of subroutine ZTOT it is included in Appendix (14) with extensive comments.

10. SUBROUTINE TOPO

PURPOSE:

The impedance matrix for a single rectangular plate has a great deal of Toeplitz properties. Subroutine TOPO identifies impedance elements which are equal (within a ±, -sign) by virtue of the Toeplitz properties.
GENERAL FORM:

TOPO(NM12,NM23,K,L,MT,NT,SGN).

THE FOLLOWING ARE INPUTS:

NM12 = the number of segments in the 1-2 direction.
NM23 = the number of segments in the 2-3 direction.
K = local row number of desired impedance element, i.e., as if the first mode of the plate was mode 1.
L = local column number of the desired impedance element.

THE FOLLOWING PARAMETERS ARE OUTPUTS:

MT,NT = local row and column number corresponding to K and L; i.e., entry Z(K,L) = Z(MT,NT)*SGN.

SGN = sign factor (+1.0 or -1.0).

11. SUBROUTINE PLTST2

PURPOSE:

Subroutine PLTST2 calculates the mutual impedance between test monopole M and expansion monopole N. M is always a quadrilateral surface patch monopole and N is either a quadrilateral surface patch or a wire monopole. The current distributions on a surface-patch and a wire monopole are given by Equations (2.13) and (2.11), respectively.
GENERAL FORM:

\[ \text{PLTST2}(X_{M1}, Y_{M1}, Z_{M1}, X_{M2}, Y_{M2}, Z_{M2}, X_{M3}, Y_{M3}, Z_{M3}, X_{M4}, Y_{M4}, Z_{M4}, \text{IM12}, \text{JOP}, \ X_{N1}, Y_{N1}, Z_{N1}, X_{N2}, Y_{N2}, Z_{N2}, X_{N3}, Y_{N3}, Z_{N3}, X_{N4}, Y_{N4}, Z_{N4}, \text{IN12}, \text{NPT}, \text{NINT}, \ I_{ACM}, I_{ACN}, Z_{MN}) \]

THE FOLLOWING PARAMETERS ARE INPUTS:

COMMON /A/ WV, PI, A, Q, GAM, ETA, XK .

PM1=(X_{M1}, Y_{M1}, Z_{M1}), PM2=(X_{M2}, Y_{M2}, Z_{M2}), PM3=(X_{M3}, Y_{M3}, Z_{M3}), PM4
=(X_{M4}, Y_{M4}, Z_{M4}) = x, y, z coordinates of the four corners of the surface
patch test monopole M.

IM12 = polarity indicator for the direction of current flow
on the test monopole.

JOP = expansion monopole type indicator.

PN1=(X_{N1}, Y_{N1}, Z_{N1}), PN2=(X_{N2}, Y_{N2}, Z_{N2}), PN3=(X_{N3}, Y_{N3}, Z_{N3}), PN4=(X_{N4}, Y_{N4}, Z_{N4})
=x, y, z coordinates of four points of the expansion monopole. If the
expansion monopole is a surface patch PN1, PN2, PN3, PN4 are its four
corners. If the expansion monopole is a wire PN1, PN2 are its endpoints
and PN3, PN4 are not used.

IN12 = same as IM12 but for expansion monopole.

NPT = the number of Simpson's rule integration intervals used
in integrating over the test monopole. This numerical
integration is implemented in the DO 10 loop.

NINT = the number of Simpson's rule integration intervals used
in integrating over the surface patch expansion monopole (if
JOP = 1). This integration takes place in subroutine ZWTPE2.
IACM = monopole shape indicator for identifying the type of surface patch monopoles of the particular test plate mode.

IACN = same as IACM but for expansion mode.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

Expansion monopole is a surface patch (JOP = 1):

If NPT = 1, both the expansion and test monopoles are represented by one filament. Their mutual impedance is evaluated by a single call to ZWTPE2 and subsequent averaging of this filament to filament impedance, as described by Equations (21) and (22), gives ZMN.

If NPT is larger than 1 the test monopole is represented by NPT + 1 filaments. The mutual impedance of each filament to the expansion monopole is evaluated by ZWTPE2 and all those partial impedances are summed using a Simpson's rule weighting.

Expansion monopole is a wire (JOP = 3):

NPT is always larger than one. The test monopole is represented by NPT + 1 filaments and the mutual impedance between every filament and the wire monopole is evaluated by ZWTWE. The final monopole-to-monopole impedance is the Spline rule summation of the partial filament-to-wire monopole impedances.
12. SUBROUTINE ZWTPE2

PURPOSE:

Subroutine ZWTPE2 calculates the mutual impedance between a wire test monopole M and a quadrilateral surface patch monopole N. The currents on the wire and the polygonal surface patch monopole are given by Equations (2.11) and (2.13), respectively.

GENERAL FORM:

\[ \text{ZWTPE2}(X_{M1}, Y_{M1}, Z_{M1}, X_{M2}, Y_{M2}, Z_{M2}, DM, IM_{12}, X_{N1}, Y_{N1}, Z_{N1}, X_{N2}, Y_{N2}, Z_{N2}, X_{N3}, Y_{N3}, Z_{N3}, X_{N4}, Y_{N4}, Z_{N4}, IN_{12}, NPLS, IACN, ZMN, KINT) } \]

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV, PI, A, Q, GAM, ETA, XK
PM1 = (X_{M1}, Y_{M1}, Z_{M1}), PM2 = (X_{M2}, Y_{M2}, Z_{M2}) = x, y, z coordinates of the two wire monopole endpoints.

DM = length of test wire monopole.
IM_{12} = polarity indicator for the direction of current flow on the test wire monopole.
PN1 = (X_{N1}, Y_{N1}, Z_{N1}), PN2 = (X_{N2}, Y_{N2}, Z_{N2}), PN3 = (X_{N3}, Y_{N3}, Z_{N3}), PN4 = (X_{N4}, Y_{N4}, Z_{N4}) = x, y, z coordinates of the four surface patch monopole corners.
IN12 = same as IM12 but for expansion monopole.

NPLS = number of Spline rule integration intervals used in the integration over the plate expansion monopole.

IACN = monopole shape indicator for identifying the type of surface patch monopoles of the particular expansion plate mode.

KINT = indicator for setting the INT integration parameter used for the filament-to-filament impedance calculations in subroutine GGS1. If KINT = 0, then ZWTPE2 assigns the value of INT. If KINT = 1, then INT = 0.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

If NPLS = 1, ZWTPE2 evaluates ZMN by calling GGS1 once and modifying the intermediate result using Equations (2.21) and (2.22).

If NPLS is larger than 1 ZWTPE2, represents the expansion monopole N by NPLS filaments and evaluates the mutual impedance between every filament and the wire test monopole using GGS1. Then it sums all those partial impedances by a Spline rule integration weighting to obtain ZMN. A substantial part of the subroutine is spent in deciding the value of INT. If INT = 0, the impedance calculations in GGS1 are done in closed form; If INT = 2 or 4, the calculations are done using 2 or 4 interval Simpson's rule integration, respectively.
13. SUBROUTINE PLTTST

PURPOSE:

Subroutine PLTTST calculates the mutual impedance between test monopole M and expansion monopole N. M is always a rectangular surface patch and N can be a wire, rectangular surface patch or disk monopole with current distributions given by Equations (2.11), (2.12) and (2.16), respectively.

GENERAL FORM:

PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,JOP,XN1,YN1,ZN1,
XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,NINT,BN, ZMN)

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
PM1= (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2),PM3=(XM3,YM3,ZM3) = x,y,z
coordinates of three consecutive corners of the rectangular surface patch test monopole.
IM12 = polarity indicator for the direction of current flow on the test monopole.
JOP = expansion monopole type indicator.
PN1= (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z
coordinates of three points on the expansion monopole. If the expansion monopole is a surface patch, PN1,PN2,PN3 are three consecutive corners of the patch. If it is a wire, PN1,PN2 are
its endpoints and PN3 is not used. If it is a disk, PN1 is the center of the disk, PN2 and PN3 are two points on the plane of the disk and PN4 is not defined.

\[ \text{IN12} = \text{same as IM12 but for expansion monopole.} \]

\[ \text{INTP} = \text{the number of Simpson's rule integration intervals used in integrating over the test surface patch monopole. The integration is implemented in the DO 10 loop.} \]

\[ \text{NINT} = \text{the number of Simpson's rule integration intervals used in integrating over the expansion surface patch (JOP = 1) or expansion disk (JOP = 2) monopole. The integration is implemented in subroutines ZWTPE (JOP = 1) or ZWTDE(JOP = 2).} \]

\[ \text{BN} = \text{outer radius of expansion disk monopole (if JOP = 2).} \]

THE FOLLOWING ARE OUTPUTS:

\[ \text{ZMN} = \text{the mutual impedance between monopole M and monopole N.} \]

NOTES:

If the expansion monopole is a surface patch, PLTTST checks for parallel current density vectors or parallel vectors transverse to the current density vectors (see Chapter II, section D.2). If either is true, then ZMN is calculated by calling subroutine PPLTS. If neither is true, the test surface patch monopole is represented by INTP+1 filaments and the mutual impedance between every filament and the expansion monopole is evaluated by calling subroutine ZWTPE. The partial filament-
to-monopole impedances are summed using a Simpson's rule weighting to give ZMN.

If the expansion monopole is not a plate (JOP = 2 or JOP = 3), then again the test monopole is represented by INTP+1 filaments and the mutual impedance between every filament and the expansion monopole is evaluated by calling the appropriate subroutine; i.e., ZWTDE if JOP = 2 or ZWTWE if JOP = 3. All those partial filament-to-monopole impedances are summed via a Simpson's rule weighting to give ZMN.

14. SUBROUTINE PPLTS

PURPOSE:

Subroutine PPLTS calculates the mutual impedance between two rectangular surface patch monopoles that are parallel in the sense described in Figure 4-10.

GENERAL FORM:

PPLTS(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,XN1,YN1,ZN1,XN2,YN2,
ZN2,XN3,YN3,ZN3,IN12,NPT,ZMN).

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK.
PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2),PM3=(XM3,YM3,ZM3) = x,y,z
coordinates of three consecutive corners of the surface patch test monopole.
Figure 4-10. (a) Current direction vectors are parallel.
(b) Vectors transverse to the current direction vectors are parallel.
IM12 = polarity indicator for the direction of current flow on the test monopole.

PN1 = (XN1,YN1,ZN1), PN2 = (XN2,YN2,ZN2), PN3 = (XN3,YN3,ZN3) = x,y,z coordinates of three consecutive corners of the surface patch expansion monopole.

IN12 = same as IM12 but for the expansion monopole.

NPT = number of Simpson's rule integration intervals used in integrating over the test and expansion monopoles.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the two rectangular surface patch monopoles.

NOTES:

Subroutine PPLTS finds DMIN = the minimum separation and DMAX = the maximum separation between the two surface patch monopoles. DM12 = the length of side PM1-PM2 of the test monopole and DN12 = the length of side PN1-PN2 of the expansion monopole. Then it sets up a wire monopole with length DM12 parallel to a wire monopole with length DN12 and lets their separation D vary between DMIN and DMAX. It calculates the corresponding mutual impedances and stores them in array ZVSD. If a log singularity arises because the separation is too small, it is removed analytically.

Subsequently the two surface patches are represented by filaments and array ZVSD is used to evaluate the filament-to-filament impedances.
by linear extrapolation. ZMN is the Simpson's rule weighted sum of all
the filament-to-filament impedances.

15. SUBROUTINE ZWTPE

PURPOSE:

Subroutine ZWTPE calculates the mutual impedance between a wire
test monopole and a rectangular surface patch monopole.

GENERAL FORM:

\[
ZWTPE(XM1, YM1, ZM1, XM2, YM2, ZM2, IM12, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, INTP, ZMN) .
\]

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV, PI, A, Q, GAM, ETA, XK .

PM1 = (XM1, YM1, ZM1), PM2=(XM2, YM2, ZM2) = x, y, z coordinates of the
test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test
monopole.

PN1 = (XN1, YN1, ZN1), PN2=(XN2, YN2, ZN2), PN3=(XN3, YN3, ZN3) = x, y, z
coordinates of three consecutive corners of the expansion
surface patch monopole.

IN12 = same as IM12 but for the expansion monopole.
INTP = the number of Simpson's rule integration intervals used in integrating over the expansion surface patch monopole. The integration is implemented in the DO 10 loop.

THE FOLLOWING ARE OUTPUTS:

ZMN = The mutual impedance between the wire and surface patch monopole.

NOTES:

The surface patch expansion monopole is represented by INTP+1 filaments. The mutual impedance between each filament and the wire test monopole is evaluated by ZGSMM and all those partial impedances are summed with a Simpson's rule weighting to give ZMN.

The following parameters are defined in ZWTP to be used by subroutine ZGSMM. Refer to Figure 4-11.

- DM = the length of test wire monopole.
- GAM = the complex free space propagation constant defined by Equation (4.1).
- CGDM = \( \cosh(\text{GAM} \cdot \text{DM}) \)
- SGDM = \( \sinh(\text{GAM} \cdot \text{DM}) \)
- DN = the length of side 1-2 of the expansion surface patch monopole.
- SGDN = \( \sinh(\text{GAM} \cdot \text{DN}) \)
Figure 4-11. Lengths DM and DN as defined in subroutine ZWTPE.
16. SUBROUTINE ZWTDE

PURPOSE:

Subroutine ZWTDE calculates the mutual impedance between a wire test monopole and a disk expansion monopole.

GENERAL FORM:

ZWTDE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XNO,YNO,ZNO,XN1,YN1,ZN1,XN2,YN2,ZN2, INTD,B,ZMN).

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV, PI, A, Q, GAM, ETA, XK.

PM1 = (XM1,YM1,ZM1), PM2 = (XM2,YM2,ZM2) = x, y, z coordinates of the two test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PO = (XNO,YNO,ZNO) = x, y, z coordinates of disk monopole center P0.

PN1 = (XN1,YN1,ZN1), PN2 = (XN2,YN2,ZN2) = x, y, z coordinates of two points on the plane of the disk monopole.

INTD = number of Simpson's rule integration intervals used in integrating over the disk expansion monopole. This integration is implemented in the DO 10 loop.

B = outer radius of disk monopole.
THE FOLLOWING ARE OUTPUTS:

\[ Z_{MN} = \text{the mutual impedance between the wire test monopole } M \]
\[ \text{and the disk expansion monopole } N. \]

NOTES:

The disk monopole is represented by INTD radial filaments. The mutual impedance between every filament and the wire monopole is evaluated by ZGSMM and all those partial mutual impedances are summed by a Simpson's rule weighting to give \[ Z_{MN}. \]

The following parameters are defined in ZWTDE and used by subroutine ZGSMM. Refer to Figure 4-12.

\[ \Gamma = \text{complex free space propagation constant defined by Equation (4.1).} \]
\[ D_M = \text{the length of the test wire monopole.} \]
\[ D_N = \text{the difference between the disk outer and inner radius.} \]
\[ S_{GDM} = \sinh(\Gamma D_M). \]
\[ C_{GDM} = \cosh(\Gamma D_M). \]
\[ S_{GDN} = \sinh(\Gamma D_N). \]
Figure 4-12. Lengths DM and DN as defined in Subroutine ZWTDE.
17. SUBROUTINE ZWTWE

PURPOSE:

Subroutine ZWTWE calculates the mutual impedance between a wire test monopole M and a wire expansion monopole N. First it sets up certain geometric parameters dealing with both monopoles. Then it calls subroutine ZGSMM to evaluate the mutual impedance between the two wire monopoles.

GENERAL FORM:

ZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN,IWW)

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK

PM1 = (XM1,YM1,ZM1), PM2 = (XM2,YM2,ZM2) = x,y,z coordinates of the test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PN1 = (XN1,YN1,ZN1), PN2 = (XN2,YN2,ZN2) = x,y,z coordinates of the expansion wire monopole endpoints.

IN12 = same as IM12 but for the expansion monopole.

IWW = indicator for choosing the minimum distance between two filaments which share one or more points. IWW = 1 implies the minimum separation is A, the wire radius. IWW = 0 implies the minimum separation is Q, normally chosen as 0.001*WV.
THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the wire test monopole M and the wire expansion monopole N.

NOTES:

The following parameters defined in ZWTWE are used by ZGSMM which ZWTWE calls.

GAM = complex free space propagation constant defined by Equation (4.1).

DM = the length of the test wire monopole.

DN = the length of the expansion wire monopole.

SGDM = sinh(GAM*DM).

CGDM = cosh(GAM*DM).

SGDN = sinh(GAM*DN).

18. SUBROUTINE ZGSMM

PURPOSE:

Subroutine ZGSMM calculates the mutual impedance between two PWS filaments.

GENERAL FORM:

ZGSMM(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,A,D1,CGD1,SGD1,D2,SGD2,Z12)
THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK.

PA = (XA,YA,ZA), PB=(XB,YB,ZB) = x,y,z coordinates of the test PWS monopole endpoints.

P1 = (X1,Y1,Z1), P2=(X2,Y2,Z2) = x,y,z coordinates of the expansion PWS monopole endpoints.

A = wire radius.

D1,CGD1,SGD1 = same as DM, CGDM, SGDM defined in subroutines ZWTPE, ZWTDE and ZWTWE.

D2,SGD2 = same as DN, SGDN defined in subroutines ZWTPE, ZWTDE and ZWTWE.

THE FOLLOWING ARE OUTPUTS:

Z12 = the mutual impedance between the two piecewise sinusoidal filaments.

NOTES:

Depending on the orientation and separation of the two filaments, subroutine ZGSMM sets the integration parameter INT to either zero or two. Then it calls GGS1 which calculates the mutual impedance by a closed form expression (INT = 0) or by a two interval Simpson's rule integration (INT = 2).
19. SUBROUTINE GGS1

PURPOSE:

Subroutine GGS1 calculates the mutual impedance between two filamentary monopoles with sinusoidal current distribution. This subroutine is the same as subroutine GGS in reference [1] except that GAM = jk where k is real.

GENERAL FORM

GGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,DS,CGDS,SGDS,DT,SGDT, INT,ETA,GAM,P11,P12,P21,P22).

THE FOLLOWING PARAMETERS ARE INPUTS:

PA = (XA,YA,ZA), PB=(XB,YB,ZB) = x,y,z coordinates of the test monopole endpoints.
P1 = (X1,Y1,Z1), P2=(X2,Y2,Z2) = x,y,z coordinates of the expansion filament endpoints.
AM = wire radius.
DS = length of the test filament.
SGDS = sinh(GAM*DS).
CGDS = cosh(GAM*DS).
DT = length of the expansion filament.
SGDT = sinh(GAM*DT).
INT = the number of Simpson's rule intervals in the integration over the expansion filament. If INT = 0, the integration is done in closed form. Otherwise, INT is always a even integer number.
ETA = complex impedance of free space (376.7 + j0.0).
GAM = complex propagation constant of free space (0.0 - jk)
where k = 2*PI/WV.

THE FOLLOWING ARE OUTPUTS:

P11, P12, P21 and P22 = mutual impedance between the two filaments. The first subscript refers to the endpoint of filament one with non-zero current. The second subscript refers to the endpoint of filament two with non-zero current.

20. SUBROUTINE DSKTS2

PURPOSE:

Subroutine DSKTS2 calculates the mutual impedance between a disk test monopole M and a wire, polygonal surface patch or disk expansion monopole N. The current distributions on the wire, polygonal surface patch and disk monopole are given by Equations (2.11), (2.13) and (2.16), respectively.

GENERAL FORM:

DSKTS2(XMO,YMO,ZMO,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,XN1,YN1,ZN1,XN2,YN2,ZN2,
    XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,INTP,NDT,BM,BN,ZMN).

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK.
PO = (XMO,YMO,ZMO) = x,y,z coordinates of the disk monopole center.

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of two points on the disk plane.

JOP = expansion monopole type indicator.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3),
PN4 = (XN4,YN4,ZN4) = x,y,z coordinates of four points on the expansion monopole. If JOP = 1, PN1,PN2,PN3,PN4 are the four corners of the plate. If JOP = 2, PN1,PN2,PN3 are the same as PO,PM1,PM2 for the test disk monopole and PN4 is not used. If JOP = 3, PN1,PN2 are the wire endpoints and PN3,PN4 are not used.

IN12 = polarity indicator for the direction of current flow on the expansion monopole N.

INTP = the number of Simpson's rule intervals in the integration over the expansion monopole used in subroutines ZWTPE2 (if JOP = 1) or ZWTDE (if JOP = 2).

NDT = the number of Simpson's rule integration intervals used in the integration over the disk test monopole.

BM = outer test disk monopole radius.

BN = outer expansion disk monopole radius (If JOP = 2).

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between test monopole M and expansion monopole N.
NOTES:

The disk test monopole is represented by NDT filaments and the impedance of each filament to the expansion monopole is evaluated by calling the appropriate subroutine; i.e., ZWTPE2 for a surface patch expansion monopole, ZWTDE for a disk expansion monopole and ZWTWE for a wire expansion monopole. All those partial impedances are summed using a Simpson's rule weighting to give ZMN.

21. SUBROUTINE DSKTST

PURPOSE:

Subroutine DSKTST calculates the mutual impedance between a disk test monopole M and a wire, rectangular surface patch or disk expansion monopole N. The current distributions on the wire, surface patch and disk monopoles are given by Equations (2.11), (2.12) and (2.16) respectively.

GENERAL FORM:

DSKTST(XMO,YMO,ZMO,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,XN1,YN1,ZN1,XN2,YN2, ZN2,XN3,YN3,ZN3,IN12,INTD,NINT,BM,BN,ZMN).

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK.

PO = (XMO,YMO,ZMO) = x,y,z coordinates of the disk monopole center.
PM1 = (XM1,YM1,ZM1), PM2=(XM2,YM2,ZM2) = x,y,z coordinates of two points on the plane of the disk.

JOP = expansion monopole type indicator.

PN1 = (XN1,YN1,ZN1), PN2=(XN2,YN2,ZN2), PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three points on the expansion monopole. If N is a plate, PN1,PN2,PN3 are three consecutive corners; if N is a disk, PN1,PN2,PN3 are the same as PO,PM1, PM2 for a test disk monopole; if N is a wire, PN1,PN2 are its endpoints and PN3 is not used.

IN12 = polarity indicator for the direction of current flow on expansion monopole N.

INTD = number of Simpson's rule intervals used in integrating over the disk test monopole.

NINT = number of Simpson's rule intervals in the integration over the expansion surface patch monopole (if JOP =1) or the expansion disk monopole (if JOP = 2). The integrations are implemented in subroutines ZWTPE (if JOP = 1) or ZWTDE (if JOP = 2).

BM = outer radius of test disk monopole.

BN = outer radius of expansion disk monopole (if JOP = 2).

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.
NOTES:

A disk test monopole is represented by INTD filaments and the mutual impedance between every filament and the expansion monopole is calculated using ZWTPE, ZWTDE, ZWTWE depending on the type of expansion monopole. Then all those partial impedances are summed using a Simpson's rule weighting.

22. SUBROUTINE ZATAT2

PURPOSE:

Subroutine ZATAT2 calculates the self impedance of an attachment mode.

GENERAL FORM:

ZATAT2(B,H,Z,NL,ZS,ALFD).

THE FOLLOWING ARE INPUTS:

COMMON /A/, W, PI, A, Q, GAM, ETA, XK.

B = outer disk radius.
H = the length of wire part of attachment mode.
NL = number of intervals for trapezoidal rule integration over the disk monopole.
ZS = complex wire surface impedance.
ALFD = the angle between the axis of the wire monopole of the attachment mode and the line perpendicular to the disk plane. If ALFD = 0.0, the wire is perpendicular to the disk.

THE FOLLOWING ARE OUTPUTS:

\[ Z \]  = the self impedance of the attachment mode.

NOTES:

The self impedance of an attachment with a perfectly conducting wire is the sum of four partial impedances, i.e.,

\[ Z = Z_{dd} + Z_{dw} + Z_{wd} + Z_{ww} \]

where \( Z_{dd} \) is the disk/disk impedance, \( Z_{dw} \) is the disk/wire impedance, \( Z_{wd} \) is the wire/disk impedance and \( Z_{ww} \) is the wire/wire impedance. Those partial impedances are evaluated by a trapezoidal rule of integration using surface testing and expansion filaments. Each filament-to-filament impedance is evaluated by GGMM1 or GGS1.

23. SUBROUTINE PDPZ1

PURPOSE:

Subroutine PDPZ1 calculates the mutual impedance between a disk monopole parallel to a non-rectangular surface patch monopole.

GENERAL FORM:

\[ \text{PDPZ1}(XMO,YMO,ZMO,K,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,} \\
\[ \text{ZN4,IACN,IN12,INTP,ERVSR,IAT,RMINK,DRK, ZMN,DIST)} \] .

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

157
PO = (XMO, YMO, ZMO) = x, y, z coordinates of the disk monopole center.

K = wire attachment mode 1.

PN1 = (XN1, YN1, ZN1), PN2 = (XN2, YN2, ZN2), PN3 = (XN3, YN3, ZN3),
PN4 = (XN4, YN4, ZN4) = x, y, z coordinates the four expansion surface patch monopole corners.

IACN = monopole shape indicator for identifying the type of surface patch monopoles of the expansion mode.

IN12 = polarity indicator for the direction of current flow on the surface patch expansion monopole.

INTP = the number of Simpson's rule integration intervals used in integrating over the expansion surface patch monopole.

ERVSR(K, JJ) = array containing values of the radial component of the electric field of disk monopole K versus the radial distance \( p \).

IAT = dimension indicator for the maximum number of wire attachments used in array ERVSR(IAT, 400).

RMINK = the minimum value of \( p \) corresponding to ERVSR(K, 1).

DRK = the increment in the value of \( p \), i.e., \( p = RMINK + DRK*JJ \).

DIST = the distance between the disk monopole and surface patch monopole planes.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the disk monopole and the surface patch monopole.
NOTES:

Subroutine PDPZ1 uses array ERVSR to interpolate the value of the \( p \) component of the disk monopole electric field \( E \) and then evaluates \( Z_{MN} = \int E \cdot J \), where \( J \) is the current density on the expansion surface patch monopole, by an NPE interval double Simpson's rule integration.

24. SUBROUTINE PDPZ

PURPOSE:

Subroutine PDPZ calculates the mutual impedance between a disk monopole parallel to a rectangular surface patch monopole. The current density \( J_n \) on the rectangular surface patch monopole is given by Equation (2.12).

GENERAL FORM:

\[
PDPZ(XMO,YMO,ZMO,K,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP, ERVSR,IAT,RMIN,DR,ZMN,DIST).
\]

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV, PI, A ?, GAM, ETA, XK .

\( P_0 = (XMO,YMO,ZMO) = x,y,z \) coordinates of disk monopole center.

\( K = \) wire attachment mode \( K \).

\( P_{N1} = (XN1,YN1,ZN1), P_{N2}=(XN2,YN2,ZN2), P_{N3}=(XN3,YN3,ZN3) = x,y,z \) coordinates of three consecutive corners of the expansion rectangular patch monopole.
IN12 = polarity indicator for the direction of current flow on
the expansion monopole.

INTP = number of Simpson's rule integration intervals used in
integrating over the surface patch monopole. This
integration is implemented in the DO 10 loop.

ERVSR(K,JJ) = array that contains values of the radial
component of the electric field of disk monopole k
versus the radial distance p.

IAT = dimension indicator for maximum number of attachments,
used in array ERVSR(IAT,400).

RMIN = the minimum value of p, corresponding to ERVSR(NAT,1).

DR = the increment in the value of p, i.e., p = RMIN +
    DR*KJ.

DIST = distance between the planes of the disk monopole and
the rectangular surface patch monopole.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the disk monopole and the
rectangular surface patch monopole.

NOTES:

Subroutine PDPZ uses array ERVSR to interpolate the values of the p
component of the disk test monopole electric field Em. Then ZMN, which
is ∫Em†n, is evaluated by an INTP interval Simpson's rule integration.
25. SUBROUTINE ERDSK

PURPOSE:

Subroutine ERDSK calculates the near zone electric field of a disk monopole in the x-y plane with a current density given by Equation (2.16). The electric field has a $\rho$ and $z$ component.

GENERAL FORM:

ERDSK(A,B,X,Z,ETA,WV,NNPTS,EX).

THE FOLLOWING ARE INPUTS:

A = inner radius of disk monopole.
B = outer radius of disk monopole.
X = the radial distance from the origin ($\rho$) where $E_\rho$ is to be evaluated.
Z = the $z$ coordinate of the field observation point.
ETA = complex impedance of free space.
WV = wavelength.
NNPTS = number of Simpson's rule intervals used in the $\phi$ integration over the disk monopole.

THE FOLLOWING ARE OUTPUTS:

EX = the value of the $\rho$ component of the disk monopole electric field in ohms/meter.
NOTES:

Subroutine ERDSK represents the disk monopole by NNPTS radial filaments. The near zone field of every filament is calculated by using its closed form expression. This formula is described in reference [1]. The total near zone field is the Simpson's rule weighted sum of all the filamentary near zone fields.

26. SUBROUTINE COUPLE

PURPOSE:

Subroutine COUPLE finds the coupling between two ports of the wire structure of the antenna.

GENERAL FORM

COUPLE(ZT,ZTF,M1,M2,SN1,SN2,I12,V,NT,IFIL).

The following parameters are inputs:

\( ZT(K) = \) complex impedance array used when \( IFIL = 0 \).
\( ZTF(M,N) = \) complex impedance matrix used when \( IFIL = 1 \).
\( M1 = \) wire segment "location" of first feed port.
\( M2 = \) wire segment "location" of second feed port.
\( SN1 = \) sign factor for first feed port. If port 1 is not by the terminal point of a wire dipole mode, then \( SN1 = 1.0 \). Otherwise \( SN1 = -1.0 \).
SN2 = sign factor for second feed port. If port 1 is not by the terminal point of a wire dipole mode, then SN2 = 1.0. Otherwise SN2 = -1.0.

II2 = matrix inversion indicator. SQROT or CROUT solves the matrix equation \([Z][I]=[V]\). In doing so, \([Z]\) is transformed to an effective inverse. If II2 = 2, this indicates that \([Z]\) is already the effective inverse. If II2 = 1, this indicates that \([Z]\) is the original matrix.

\([V]\) = dummy column vector.

NT = the total number of modes.

IFIL = indicator for choosing between full surface patch test monopoles (IFIL = 0) or filamentary surface patch test monopoles (IFIL = 1).

NOTES:

Let \(ZT\) or \(ZTF\) be the impedance matrix of the problem and \([Z][I] = [V]\) or \([Y][V] = [I]\) where \(Z = ZT\) or \(ZTF\). If we set the \(V\) column vector to zero except for the entries corresponding to the two ports of interest, we can reduce the \([Y][V] = [I]\) matrix equation to

\[
\begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
\begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}
=
\begin{bmatrix}
I_1 \\
I_2
\end{bmatrix}
\]
where the subscripts 1,2 do not represent actual locations in the original Y matrix but ports 1 and 2. The $Y_{11}$, $Y_{12}$, $Y_{21}$ and $Y_{22}$ are determined as follows:

Let the $[V]$ column vector of the original matrix be equal to zero except for the entry corresponding to port 1 which is set to $(1.0,0.0)$. Then the $[V]$ column vector is the input to SQROT or CROUT and comes out as the "induced" current vector $[V]$; $Y_{11} = V(M1)$ and $Y_{12} = V(M2)$. To find the other two parameters the same procedure is used except that the entry corresponding to port 2 is set to $(1.0,0.0)$. Again $Y_{21} = V(M1)$ and $Y_{22} = V(M2)$.

Finally, the admittance matrix is inverted to obtain the impedance matrix $Z$ relating the two ports. COUPLE also finds the maximum coupling between the two ports.

27. SUBROUTINE ANTV

PURPOSE:

Subroutine ANTV evaluates the current vector $I$ of the matrix equation $[Z][I] = [V]$ (see Equation 2.8) for an antenna problem, i.e., when the excitation is due to a delta gap generator. It also calculates the input impedance of the antenna, the power dissipated by the wire structure and the efficiency of the antenna.
GENERAL FORM:

ANTV(I1, I2, I3, IA, IB, IWR, JA, JB, NM, ZT, IFIL, ICC, ZTF, CJ, VG, Y11, Z11, NWR, 
NPL, NAT, VGA, PIN, AM, CMM, D, DISS, GAM, SGD, ZLD, ZS, ZLDA, INM, MD, ND, NSA)

THE FOLLOWING ARE INPUTS:

I1(I) = endpoint 1 of wire dipole I.
I2(I) = terminal point of wire dipole I.
I3(I) = endpoint 2 of wire dipole I.
IA(J) = endpoint A of wire segment J.
IB(J) = endpoint B of wire segment J.

IWR = write indicator used by subroutine SQROT. If IWR = 1, 
SQROT writes out the modal currents.

JA(I) = segment number of the first segment of wire dipole I.
JB(I) = segment number of the second segment of wire dipole I.

NM = total number of wire segments.

ZT(K) = complex impedance array used when IFIL = 0.

IFIL = indicator for choosing either full surface patch test 
modes (IFIL = 0) or filamentary surface patch 
test modes (IFIL = 1).

ICC = dimension indicator for array ZTF(ICC, ICC).

ZTF(M, N) = complex impedance matrix used when IFIL = 1.

VG(II) = complex voltage generator at wire "location" II.

NWR = total number of wire modes.

NPL = total number of plate modes.

NAT = total number of wire attachment dipole modes.

VGA(K) = complex voltage generator at wire attachment K.
AM = wire radius.

CMM = wire conductivity in megamhos/meter. CMM = -1.0 implies a perfect conductor.

D(J) = length of wire segment J.

GAM = complex free space propagation constant defined by Equation (4.1).

SGD(J) = sinh(GAM*D(J))

ZLD(II) = complex impedance loading at wire "location" II.

ZS = complex surface impedance of the wire.

ZLDA(K) = complex impedance loading at wire attachment K.

INM = dimension indicator for array MD(INM,4).

MD(J,L) = list of wire dipoles sharing wire segment J.

ND(J) = total number of wire dipoles sharing wire segment J.

NSA(K) = wire segment "location" of attachment K.

THE FOLLOWING ARE OUTPUTS:

CJ(I) = magnitude of current of mode I.

CG(I) = magnitude of current at wire segment "location" I.

PIN = time average power input to the wire structure.

DISS = time average power dissipated by the wire structure.

Y11 = complex input admittance(mohms).

Z11 = complex input impedance(ohms).
NOTES:

Consider the matrix equation $[Z][I]=[V]$. ANTV finds all the entries $CJ$ of the $V$ column and stores them in array $CG$ before it calls $SQROT$ or $CROUT$. Upon entry to $SQROT$ or $CROUT$, $CJ$ is the excitation column $V$. On exit, the solution vector $I$ is stored in array $CJ$.

In the DO 80 loop, ANTV calculates the admittance $Y_{11}$ of the antenna using the formula $Y_{11} = \text{Current} \times \text{Conjugate(voltage)}$. Actually $Y_{11}$ is the input admittance only if the antenna is fed by a single one-volt generator.

Finally, ANTV calls AGDISS to evaluate the power dissipated in antenna structure and ARITE to find the branch currents, i.e., the currents at each wire "location".

28. SUBROUTINE AGDISS

PURPOSE:

Subroutine AGDISS calculates the time average power dissipated by the imperfectly conducting wire structure of the antenna and its loads.

GENERAL FORM:

AGDISS(AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS,ZLDA,NAT,NSA)

THE FOLLOWING ARE INPUTS:

$AM = \text{wire radius.}$

$CG(I) = \text{magnitude of current at wire "location" I.}$
CMM = wire conductivity in megamhoms/meter. If CMM = -1.0, the wire is a perfect conductor.

D(I) = length of wire mode I.

GAM = complex free space propagation constant.

NM = total number of wire segments.

SGD = sinh(GAM*D)

ZLD(I) = complex impedance load at wire "location" I.

ZS = complex wire surface impedance.

ZLDA(K) = complex impedance load at wire attachment K.

NAT = total number of wire attachments.

NSA(K) = wire "location" of attachment K.

THE FOLLOWING ARE OUTPUTS:

NOTES:

DISS = the time average power dissipated in the wire and the loads.

Subroutine AGDISS uses Poynting's theorem to calculate the time average power dissipated in the imperfect wire in the DO 100 loop. The power dissipated in the loads is calculated in the DO 140 loop using the formula Power = Impedance * (Current)^2.
29. SUBROUTINE ARITE

PURPOSE:

Subroutine ARITE generates a list of branch currents using the loop currents.

GENERAL FORM:

ARITE(IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,CJ,CG,NSA,NWR,NPLTM,NAT).

THE FOLLOWING ARE INPUTS:

IA(J) = endpoint A of wire segment J.
IB(J) = endpoint B of wire segment J.
INM = dimension indicator for array MD(INM,4).
IWR = indicator for writing out the induced modal currents.
I1(J) = endpoint 1 of wire dipole mode J.
I2(J) = terminal point of wire dipole mode J.
I3(J) = endpoint 2 of wire dipole mode J.
MD(J,L) = list of wire dipoles sharing wire segment J.
ND(J) = total number of wire dipoles sharing wire segment J.
NM = total number of wire segments.
CJ(I) = amplitude of the current of mode I.
NSA(K) = wire "location" of attachment K.
NWR = total number of wire modes.
NPLTM = total number of plate modes.
NAT = total number of wire attachments.
THE FOLLOWING ARE OUTPUTS:

\[ CG(J) = \text{amplitude of the current at wire "location" } J. \]

30. SUBROUTINE CROUT

PURPOSE:

Consider the matrix equation \( [Z][I] = [V] \) which represents a system of simultaneous linear equations. Given the excitation vector \( V \) and the matrix \( Z \) subroutine, CROUT solves for vector \( I \).

GENERAL FORM:

\[ \text{CROUT}(S, ICC, C, ISYM, IWR, I12, N) \]

THE FOLLOWING ARE INPUTS:

\[ S(I) = \text{complex array containing the excitation vector } V. \]

\[ ICC = \text{dimension indicator for array } C(I,J). \]

\[ C(I,J) = \text{complex matrix.} \]

\[ ISYM = \text{symmetry indicator. If } ISYM = 0, \text{ array } C(I,J) \text{ is not symmetric and if } ISYM = 1, \text{ array } C(I,J) \text{ is symmetric.} \]

\[ IWR = \text{write indicator for printing the solution vector.} \]

\[ I12 = \text{matrix inversion indicator. CROUT solves the matrix equation } [Z][I] = [V]. \text{ In doing so, } [Z] \text{ is transformed to an effective inverse. If } I12 = 2, \text{ this indicates that } [Z] \text{ is already the effective inverse. If } I12 = 1, \text{ this indicates that } [Z] \text{ is the original matrix.} \]
THE FOLLOWING ARE OUTPUTS:

S(K) = complex array containing the solution vector I.
C(I,J) = effective inverse of input matrix C(I,J).

31. SUBROUTINE SORTB

PURPOSE:

Subroutine SORTB calculates the far zone electric field for an antenna problem or the backscattered electric field and various cross sections for a scattering problem.

GENERAL FORM:

SORTB(IA,IB,11,12,13,NWR,NM,A,CGD,SGD,FHZ,D,IWRSQ,12,ISCAT,ZTF,ZT, IFIL,ICC,X,Y,Z,NPL,NAT,PA,PB,NSA,NPLA,PCN,BDSK,IQUAD, NPLTM,IPL,IPLM,CJP,CJT,ETTS,EPPS,ETPS,THETA,PHI,JA,JB, SCSP,SCST,SPPM,SPTM,STTM,IMAGE,ICN,NONPLT).

THE FOLLOWING ARE INPUTS:

IA(I) = endpoint A of wire segment I.
IB(I) = endpoint B of wire segment I.
11(J) = endpoint 1 of wire dipole mode J.
12(J) = terminal point of wire dipole mode J.
13(J) = endpoint 2 of wire dipole mode J.
NWR = total number of wire modes.
NM = total number of wire segments.
A = wire radius.
\[ D(J) = \text{length of wire segment J.} \]
\[ CGD(J) = \cosh(GAM \cdot D(J)) \]
\[ SGD(J) = \sinh(GAM \cdot D(J)) \]

\[ \text{FHZ} = \text{frequency in Hertz.} \]
\[ \text{IWRSQ} = \text{write indicator used in SQROT(CROUT). If IWRSQ} = 1, \]
\[ \text{then SQROT(CROUT) writes out the induced modal currents.} \]

\[ \text{IL2} = \text{matrix ZT or ZTF inversion indicator. SQROT or CROUT} \]
\[ \text{solves the matrix equation } [Z][I]=[V]. \text{ In doing so, } [Z] \text{ is} \]
\[ \text{transformed to an effective inverse. If IL2} = 2, \text{this} \]
\[ \text{indicates that } [Z] \text{is already the effective inverse. If IL2} = 1, \text{this indicates that } [Z] \text{is the original matrix.} \]

\[ \text{ISCAT = indicator for choosing between antenna calculations} \]
\[ \text{(ISCAT} = 0), \text{backscattering (ICAT} = 1) \text{and bistatic scattering} \]
\[ \text{(ISCAT} = 2). \]

\[ \text{ZTF(M,N)} = \text{complex impedance matrix used when IFIL} = 1. \]
\[ \text{ZT(K)} = \text{complex impedance array used when IFIL} = 0. \]

\[ \text{IFIL = indicator for choosing either full surface patch test} \]
\[ \text{modes (IFIL} = 0) \text{or filamentary surface patch plate modes} \]
\[ \text{(IFIL} = 1). \]

\[ \text{ICC = dimension indicator for matrix ZTF(ICC,ICC).} \]

\[ X(I), Y(I), Z(I) = \text{x,y,z coordinates of point I of the antenna wire} \]
\[ \text{structure.} \]

\[ \text{NPL} = \text{total number of plates.} \]
\[ \text{NAT} = \text{total number of attachments.} \]
PA(I,J,K) = x,y,z coordinates (K=1,2,3) of the J-th corner of monopole A of the I-th surface patch dipole mode.

PB(I,J,K) = x,y,z coordinates (K=1,2,3) of the J-th corner of monopole B of the I-th surface patch dipole mode.

NSA(K) = the wire segment "location" of attachment K.

NPLA(K) = number of plate where wire attachment K is located.

PCN(I,J,K) = x,y,z coordinates (I=1,2,3) of the J-th corner of the K-th plate.

BDSK(K) = outer radius of disk monopole of attachment dipole mode K.

IQUAD(I) = monopole shape indicator for identifying the type of monopoles of plate mode I.

IQUAD(I) = -3 implies both monopoles of plate dipole mode I are rectangular.

IQUAD(I) = 0 implies that either or both monopoles of plate dipole mode I are quadrilateral surface patches.

NPLTM = total number of plate modes.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(3,4,IPLM) and PB(3,4,IPLM).

CJP(I) = array containing the magnitude of the induced modal current on mode I due to a φ polarized incident wave. This array is an input from subroutine ANTV only when ISCAT = 0.
CJT(I) = same as CJP(I) but for θ polarization.

THETA = elevation angle of observation point.

PHI = azimuth angle of observation point.

JA(I) = segment number of the first segment of wire dipole mode I.

JB(I) = segment number of the second segment of wire dipole mode I.

IMAGE = incident wave image indicator.

IMAGE = 0 implies no image incident wave is desired.

IMAGE = 1 implies the image of the incident wave is desired.

The image plane is the x-y plane. The program automatically includes the image of the incident wave when IMAGE = 1 but the user has to define the image geometry.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NDNPLT(I) = total number of plate modes through plate I.

THE FOLLOWING ARE OUTPUTS:

When ISCAT = 0 (radiation problem) the only outputs are:

ETTS = θ component of far zone radiated electric field.

EPPS = φ component of far zone radiated electric field.

When ISCAT = 1 or 2 (scattering or backscattering problem) the outputs are:

ETTS = θ component of far zone scattered electric field due to a θ polarized incident wave.

EPPS = φ component of far zone scattered electric field due to a φ polarized incident wave.
ETPS = \( \theta \) component of far zone scattered electric field due to a \( \phi \) polarized incident wave.

EPTS = \( \phi \) component of far zone scattered electric field due to a \( \theta \) polarized incident wave.

ECSP = extinction cross section due to a \( \phi \) polarized incident wave.

ECST = extinction cross section due to a \( \theta \) polarized incident wave.

SCSP = scattering cross section due to a \( \phi \) polarized incident wave.

SCST = scattering cross section due to a \( \theta \) polarized incident wave.

SPPM = echo area of \( \phi \) polarized scattered electric field due to a \( \phi \) polarized incident wave.

SPTM = echo area of \( \phi \) polarized scattered electric field due to a \( \theta \) polarized incident wave.

STPM = echo area of \( \theta \) polarized scattered electric field due to a \( \phi \) polarized incident wave.

STTM = echo area of \( \theta \) polarized scattered electric field due to a \( \theta \) polarized incident wave.

CJP(I) = array containing the magnitude of the induced current on mode I for a \( \phi \) polarized incident wave. It is an output only when ISCAT = 1 or 2.

CJT(I) = array containing the magnitude of the induced current on mode I for a \( \theta \) polarized incident wave. It is an output only when ISCAT = 1 or 2.
NOTES:

For ISCAT = 0, subroutine SORTB calculates the far zone electric field radiated by the antenna structure. This is done by a superposition of the far zone electric fields of all the expansion modes. The field of a particular mode is obtained by calling the appropriate far zone subroutine (GFF for a wire expansion, SURMFF for a surface patch, DSKFF for a disk monopole of an attachment mode) which outputs the E field as ET (E_θ) and EP (E_ϕ). ET and EP are weighted by the corresponding loop current CJP, which is evaluated in ANTV, to give the actual fields ETTS (E_θ) and EPPS (E_ϕ) of that particular mode. All this is done within the DO 160 loop.

For ISCAT = 1, subroutine SORTB calculates the backscattered far zone electric field from the scatterer. This is done by a superposition of the backscattered electric field of all expansion modes. The backscattered field of a particular mode is evaluated as follows: the ET and EP components of the mode are calculated as for ISCAT = 0 and stored in the arrays ETT and EPP, respectively. The excitation voltages arrays CJP and CJT, described by Equation (10-1), are obtained by multiplying CJI with EP and ET, respectively, and are used by SQROT to obtain the induced loop current arrays CJP and CJT. (Arrays CJP and CJT contain the excitation voltages upon entry to SQROT(CROUT) and upon exit from SQROT(CROUT) they contain the induced loop currents). Finally, ETT and EPP are weighted by CJT and CJP to give the actual backscattered E field of the mode.
An ISCAT = 2 is the same as ISCAT = 1 except that the subroutine does not have to evaluate the excitation voltage arrays CJT and CJP. These are calculated by a preceding call to SORTB with ISCAT = 1.

32. SUBROUTINE SURMFF

PURPOSE:

Subroutine SURMFF calculates the far zone electric field of a rectangular surface patch monopole in the y-z plane. The current density on the monopole is given by Equation (2.12).

GENERAL FORM:

SURMFF(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,I12,TH,PH,ETH,EPH,WVL) .

THE FOLLOWING ARE INPUTS:

P1 = (X1,Y1,Z1), P2=(X2,Y2,Z2), P3=(X3,Y3,Z3) = x,y,z coordinates of three consecutive corners of the rectangular surface patch monopole.

I12 = polarity indicator for the direction of current flow on the expansion surface patch monopole.

TH = elevation angle of observation point.

PH = azimuth angle of observation point.

WVL = wavelength in meters.
THE FOLLOWING ARE OUTPUTS:

\[ \text{ETH} = \theta \text{ component of far-zone electric field } E. \]
\[ \text{EPH} = \phi \text{ component of far-zone electric field } E. \]

NOTES:

This subroutine is an implementation of Equation (103) of [10].

33. SUBROUTINE SURFFP

PURPOSE:

Subroutine SURFFP calculates the far zone electric field electric field of a polygonal surface patch monopole in the y-z plane. The current density on the monopole is given by Equation (2.13).

GENERAL FORM:

\[
\text{SURFFP}(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,NPLS,GAM,ETA,XK,}
\text{FHZ,THR,PHR,IN12,ETH,EPH).}
\]

THE FOLLOWING ARE INPUTS:

\[ \text{PN1} = (XN1,YN1,ZN1), \text{PN2} = (XN2,YN2,ZN2), \text{PN3} = (XN3,YN3,ZN3), \]
\[ \text{PN4} = (XN4,YN4,ZN4) = x,y,z \text{ coordinates of the four corners of the quadrilateral surface patch monopole.} \]
\[ \text{NPLS} = \text{number of Simpson's rule integration intervals used in integrating over the surface patch monopole.} \]

178
GAM = complex propagation constant of free space.
ETA = complex impedance of free space.
XK = k, the free space wave number.
FHZ = frequency in Hertz.
THR = elevation angle of observation point.
PHR = azimuth angle of observation point.
IN12 = polarity indicator for the direction of current flow on the surface patch monopole.

THE FOLLOWING ARE OUTPUTS:
ETH = \theta component of far-zone electric field E.
EPH = \phi component of far-zone electric field E.

NOTES:

Subroutine SURFFP represents the polygonal surface patch monopole with NPLS filaments and calls subroutine GFF to evaluate the far zone electric field of each filament. Then it sums all those filamentary fields using a Simpson's rule weighting to obtain the total far zone electric field of the surface patch monopole.

34. SUBROUTINE DSKFF

PURPOSE:

Subroutine DSKFF calculates the far zone electric field of a disk monopole located in the x-y plane. The surface current density is given by Equation (2.16).
GENERAL FORM:

DSKFF(X0,Y0,Z0,X1,Y1,Z1,X2,Y2,Z2,TH,PH,A,B,WVL,ETH,EPH).

THE FOLLOWING ARE INPUTS:

PO = (X0,Y0,Z0) = x,y,z coordinates of disk monopole center.

P1 = (X1,Y1,Z1), P2 = (X2,Y2,Z2) = x,y,z coordinates of two points on the disk monopole plane.

TH = elevation angle of the observation point.

PH = azimuth angle of the observation point.

A = inner radius of the disk monopole.

B = outer radius of the disk monopole.

WVL = wavelength in meters.

THE FOLLOWING ARE OUTPUTS:

ETH = \theta component of far-zone electric field E.

EPH = \phi component of far-zone electric field E.

NOTES:

DSKFF is an implementation of Equation (113) of [10].
CHAPTER V

SUMMARY

The description of a general purpose computer code based on the Moment Method (MM) solution for electromagnetic radiation and scattering problems has been presented. The MM formulation as implemented by the program was discussed in brief. The program inputs were described and several examples illustrating their use were given. Finally, all the program subroutines were described.

The program can handle geometries consisting of wire, polygonal plates, wire/plate junctions and multiple junctions. Since the computation time is proportional to the square of the number of modes, the program is limited to low frequencies or, equivalently, it is practical up to frequencies that do not make the number of modes prohibitively large. The major advantages of the program are accuracy, flexibility and the simplicity of the input format.
REFERENCES


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A
APPENDIX 1

OUTPUT FOR DESIGN EXAMPLE 1

INPUT DATA

FREQ. (MHz) = 150.000  WAVELEN (M) = 2.000  WIRE RADIUS (M) = 0.001000
INT  = 10  INT  = 18  INT  = 4  INT  = 1

WIRE CONDUCTIVITY = 38.00 MEGAMS/H

GEOMETRY FOR THE 1 PLATE

<table>
<thead>
<tr>
<th>PLATE NUMBER</th>
<th>1 (RECTANGULAR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER OF CORNERS</td>
<td>4</td>
</tr>
<tr>
<td>MAXIMUM SEGMENT SIDE (METERS)</td>
<td>0.1250</td>
</tr>
<tr>
<td>POLARIZATION INDEX</td>
<td>3</td>
</tr>
<tr>
<td>GENERATED SIDE INDEX</td>
<td>0</td>
</tr>
</tbody>
</table>

X,Y,Z COOR. (METERS) OF CORNER

- CORNER 1 = (-0.50000, -0.50000, 0.00000)
- CORNER 2 = (0.50000, -0.50000, 0.00000)
- CORNER 3 = (0.50000, 0.50000, 0.00000)
- CORNER 4 = (-0.50000, 0.50000, 0.00000)

COORD (METERS) OF 24 NODES ON THIS PLATE

<table>
<thead>
<tr>
<th>MONPOLE NO.</th>
<th>X1</th>
<th>Y1</th>
<th>X2</th>
<th>Y2</th>
<th>X3</th>
<th>Y3</th>
<th>X4</th>
<th>Y4</th>
<th>X5</th>
<th>Y5</th>
<th>X6</th>
<th>Y6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 1</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
</tr>
<tr>
<td>A 2</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
</tr>
<tr>
<td>A 3</td>
<td>0.50000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
</tr>
<tr>
<td>A 4</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
</tr>
<tr>
<td>A 5</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.50000</td>
<td>0.00000</td>
</tr>
<tr>
<td>A 6</td>
<td>0.25000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
</tr>
<tr>
<td>B 2</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
</tr>
<tr>
<td>B 3</td>
<td>0.50000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
</tr>
<tr>
<td>B 4</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.50000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
</tr>
<tr>
<td>B 5</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.25000</td>
<td>0.00000</td>
<td>0.50000</td>
<td>-0.50000</td>
<td>0.00000</td>
</tr>
<tr>
<td>B 6</td>
<td>0.25000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
<td>0.00000</td>
<td>0.25000</td>
</tr>
</tbody>
</table>
APPENDIX 2
OUTPUT FOR DESIGN EXAMPLE 2

INPUT DATA

FREQ. (MHz) = 300.000  MAX. DO = 1.000  WIRE RADIUS(R) = 0.0010000
DINT = 10  INT = 18  INT = 4  IFIL = 1

WIRE CONDUCTIVITY = -1.00 MEGAWH/N

GEOMETRY FOR THE 2 PLATES

PLATE NUMBER 1 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIDE INDICATOR = 0
X,Y,Z COOR. (METERS) OF CORNER 1 = 0.00000  0.00000  0.00000
X,Y,Z COOR. (METERS) OF CORNER 2 = 0.00000  0.00000  1.00000
X,Y,Z COOR. (METERS) OF CORNER 3 = 0.50000  0.00000  1.00000
X,Y,Z COOR. (METERS) OF CORNER 4 = 0.50000  0.00000  0.00000

COORD. (METERS) OF 10 NODES ON THIS PLATE

PLATE NUMBER 2 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIDE INDICATOR = 0
X,Y,Z COOR. (METERS) OF CORNER 1 = 0.00000  0.00000  0.00000
X,Y,Z COOR. (METERS) OF CORNER 2 = 0.00000  0.00000  1.00000
X,Y,Z COOR. (METERS) OF CORNER 3 = 0.00000  0.50000  1.00000
<table>
<thead>
<tr>
<th>X.Y.Z (COORD. METERS) OF CORNER 4 =</th>
<th>0.000000</th>
<th>0.500000</th>
<th>0.000000</th>
</tr>
</thead>
</table>

**COORD. (METERS) OF 10 NODES ON THIS PLATE**

**COORD. (METERS) OF 4 OVERLAP NODES BETWEEN PLATE 1, SIDE 1 AND PLATE 2, SIDE 1**

**LISTING OF LOADS AND GENERATORS**

- **NOR** = NUMBER OF WIRE NODES = 0
- **NBLK** = NUMBER OF BLACK NODES = 24
- **NRC** = NUMBER OF ATTACHMENT NODES = 0

**BACKSCATTERING, EXCIT = 1**

<table>
<thead>
<tr>
<th>*** CROSS SECTION (DB/ME*2) ***</th>
<th>*********</th>
<th>PHASE (DEG)</th>
<th>*********</th>
</tr>
</thead>
<tbody>
<tr>
<td>THERMAL RESISTANCE (R)</td>
<td>POWER (W)</td>
<td>STPM</td>
<td>STPM</td>
</tr>
<tr>
<td>90.0</td>
<td>0.0</td>
<td>-7.370</td>
<td>4.744</td>
</tr>
<tr>
<td>90.0</td>
<td>3.0</td>
<td>-9.530</td>
<td>5.166</td>
</tr>
<tr>
<td>90.0</td>
<td>6.0</td>
<td>-11.182</td>
<td>5.473</td>
</tr>
<tr>
<td>90.0</td>
<td>9.0</td>
<td>-10.420</td>
<td>5.673</td>
</tr>
<tr>
<td>90.0</td>
<td>12.0</td>
<td>-7.204</td>
<td>5.774</td>
</tr>
<tr>
<td>90.0</td>
<td>15.0</td>
<td>-4.046</td>
<td>5.785</td>
</tr>
<tr>
<td>90.0</td>
<td>18.0</td>
<td>-1.446</td>
<td>5.716</td>
</tr>
<tr>
<td>90.0</td>
<td>21.0</td>
<td>0.660</td>
<td>5.581</td>
</tr>
<tr>
<td>90.0</td>
<td>24.0</td>
<td>2.370</td>
<td>5.394</td>
</tr>
<tr>
<td>90.0</td>
<td>27.0</td>
<td>3.755</td>
<td>5.172</td>
</tr>
<tr>
<td>90.0</td>
<td>30.0</td>
<td>4.876</td>
<td>4.935</td>
</tr>
<tr>
<td>90.0</td>
<td>33.0</td>
<td>5.757</td>
<td>4.706</td>
</tr>
<tr>
<td>90.0</td>
<td>36.0</td>
<td>6.424</td>
<td>4.500</td>
</tr>
<tr>
<td>90.0</td>
<td>39.0</td>
<td>6.891</td>
<td>4.340</td>
</tr>
<tr>
<td>90.0</td>
<td>42.0</td>
<td>7.170</td>
<td>4.218</td>
</tr>
<tr>
<td>90.0</td>
<td>45.0</td>
<td>7.266</td>
<td>4.203</td>
</tr>
<tr>
<td>90.0</td>
<td>48.0</td>
<td>7.180</td>
<td>4.239</td>
</tr>
<tr>
<td>90.0</td>
<td>51.0</td>
<td>6.912</td>
<td>4.342</td>
</tr>
<tr>
<td>90.0</td>
<td>54.0</td>
<td>6.456</td>
<td>4.503</td>
</tr>
<tr>
<td>90.0</td>
<td>57.0</td>
<td>5.802</td>
<td>4.708</td>
</tr>
<tr>
<td>90.0</td>
<td>60.0</td>
<td>4.938</td>
<td>4.940</td>
</tr>
<tr>
<td>90.0</td>
<td>63.0</td>
<td>3.840</td>
<td>5.177</td>
</tr>
<tr>
<td>90.0</td>
<td>66.0</td>
<td>2.478</td>
<td>5.399</td>
</tr>
<tr>
<td>90.0</td>
<td>69.0</td>
<td>0.805</td>
<td>5.587</td>
</tr>
<tr>
<td>90.0</td>
<td>72.0</td>
<td>-1.248</td>
<td>5.772</td>
</tr>
<tr>
<td>90.0</td>
<td>75.0</td>
<td>-3.770</td>
<td>5.791</td>
</tr>
<tr>
<td>90.0</td>
<td>78.0</td>
<td>-6.815</td>
<td>5.780</td>
</tr>
<tr>
<td>90.0</td>
<td>81.0</td>
<td>-9.977</td>
<td>5.480</td>
</tr>
<tr>
<td>90.0</td>
<td>84.0</td>
<td>-11.258</td>
<td>5.400</td>
</tr>
<tr>
<td>90.0</td>
<td>87.0</td>
<td>-8.716</td>
<td>5.173</td>
</tr>
<tr>
<td>90.0</td>
<td>90.0</td>
<td>-7.624</td>
<td>4.752</td>
</tr>
</tbody>
</table>
APPENDIX 3

OUTPUT FOR DESIGN EXAMPLE 3

INPUT DATA

FREQ (HZ) = 300.000  WAVE (H) = 1.000  WIRE DIAMETER (M) = 0.0010000
INT= 10  INTD= 18  INT = 4  IFIL= 1

WIRE CONDUCTIVITY = -1.00 MEGOHMS/M

GEOMETRY FOR THE 2 PLATES

PLATE NUMBER 1 (RECTANGULAR)

NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 0
GENERATING SIZE INDICATOR = 0

X,Y,Z COR (METERS) OF CORNER 1 = 0.00000  0.00000  0.00000
X,Y,Z COR (METERS) OF CORNER 2 = 0.00000  0.00000  1.00000
X,Y,Z COR (METERS) OF CORNER 3 = 0.50000  0.00000  1.00000
X,Y,Z COR (METERS) OF CORNER 4 = 0.50000  0.00000  0.00000

COORD (METERS) OF 10 NODES ON THIS PLATE

PLATE NUMBER 2 (RECTANGULAR)

NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 0
GENERATING SIZE INDICATOR = 0

X,Y,Z COR (METERS) OF CORNER 1 = 0.00000  0.00000  0.00000
X,Y,Z COR (METERS) OF CORNER 2 = 0.00000  0.00000  1.00000
X,Y,Z COR (METERS) OF CORNER 3 = 0.00000  0.50000  1.00000
### CROSS SECTION (DA/DM/E+2) ###

<table>
<thead>
<tr>
<th>DECA (DEG)</th>
<th>PHA (DEG)</th>
<th>STM</th>
<th>SPM</th>
<th>STM</th>
<th>SPM</th>
<th>STM</th>
<th>SPM</th>
<th>STM</th>
<th>SPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.0</td>
<td>0.0</td>
<td>1.267</td>
<td>1.252</td>
<td>-50.366</td>
<td>-50.361</td>
<td>27.28</td>
<td>-70.30</td>
<td>-136.83</td>
<td>-47.99</td>
</tr>
<tr>
<td>90.0</td>
<td>3.0</td>
<td>2.923</td>
<td>1.588</td>
<td>-50.720</td>
<td>-50.568</td>
<td>32.00</td>
<td>-65.68</td>
<td>-131.42</td>
<td>-47.19</td>
</tr>
<tr>
<td>90.0</td>
<td>9.0</td>
<td>3.407</td>
<td>2.227</td>
<td>-50.51</td>
<td>-50.715</td>
<td>40.63</td>
<td>-57.38</td>
<td>-123.10</td>
<td>-46.22</td>
</tr>
<tr>
<td>90.0</td>
<td>12.0</td>
<td>4.018</td>
<td>2.524</td>
<td>-50.450</td>
<td>-50.760</td>
<td>44.51</td>
<td>-53.70</td>
<td>-118.22</td>
<td>-46.02</td>
</tr>
<tr>
<td>90.0</td>
<td>15.0</td>
<td>4.777</td>
<td>2.802</td>
<td>-50.424</td>
<td>-50.745</td>
<td>49.08</td>
<td>-50.13</td>
<td>-113.57</td>
<td>-45.98</td>
</tr>
<tr>
<td>90.0</td>
<td>18.0</td>
<td>5.063</td>
<td>3.057</td>
<td>-50.416</td>
<td>-50.728</td>
<td>51.14</td>
<td>-47.28</td>
<td>-109.16</td>
<td>-46.08</td>
</tr>
<tr>
<td>90.0</td>
<td>21.0</td>
<td>5.318</td>
<td>3.291</td>
<td>-50.409</td>
<td>-50.716</td>
<td>54.27</td>
<td>-44.54</td>
<td>-105.01</td>
<td>-46.31</td>
</tr>
<tr>
<td>90.0</td>
<td>24.0</td>
<td>5.540</td>
<td>3.500</td>
<td>-50.408</td>
<td>-50.715</td>
<td>56.87</td>
<td>-42.12</td>
<td>-101.11</td>
<td>-46.64</td>
</tr>
<tr>
<td>90.0</td>
<td>27.0</td>
<td>6.299</td>
<td>3.694</td>
<td>-50.738</td>
<td>-50.716</td>
<td>59.13</td>
<td>-40.02</td>
<td>-97.54</td>
<td>-47.06</td>
</tr>
<tr>
<td>90.0</td>
<td>30.0</td>
<td>6.585</td>
<td>3.841</td>
<td>-50.942</td>
<td>-50.719</td>
<td>61.06</td>
<td>-38.24</td>
<td>-94.24</td>
<td>-47.34</td>
</tr>
<tr>
<td>90.0</td>
<td>33.0</td>
<td>6.828</td>
<td>3.970</td>
<td>-50.908</td>
<td>-50.739</td>
<td>62.63</td>
<td>-36.78</td>
<td>-91.25</td>
<td>-48.05</td>
</tr>
<tr>
<td>90.0</td>
<td>36.0</td>
<td>7.018</td>
<td>4.071</td>
<td>-50.540</td>
<td>-50.754</td>
<td>63.86</td>
<td>-35.65</td>
<td>-88.57</td>
<td>-48.55</td>
</tr>
<tr>
<td>90.0</td>
<td>39.0</td>
<td>7.154</td>
<td>4.144</td>
<td>-50.446</td>
<td>-50.742</td>
<td>64.74</td>
<td>-34.03</td>
<td>-86.22</td>
<td>-49.00</td>
</tr>
<tr>
<td>90.0</td>
<td>42.0</td>
<td>7.237</td>
<td>4.188</td>
<td>-60.435</td>
<td>-58.018</td>
<td>65.27</td>
<td>-34.35</td>
<td>-84.20</td>
<td>-49.35</td>
</tr>
<tr>
<td>90.0</td>
<td>45.0</td>
<td>7.266</td>
<td>4.203</td>
<td>-61.017</td>
<td>-58.338</td>
<td>65.45</td>
<td>-34.18</td>
<td>-82.52</td>
<td>-49.55</td>
</tr>
<tr>
<td>90.0</td>
<td>48.0</td>
<td>7.262</td>
<td>4.189</td>
<td>-61.703</td>
<td>-58.693</td>
<td>65.27</td>
<td>-34.35</td>
<td>-81.19</td>
<td>-49.53</td>
</tr>
<tr>
<td>90.0</td>
<td>51.0</td>
<td>7.164</td>
<td>4.145</td>
<td>-62.311</td>
<td>-59.072</td>
<td>64.74</td>
<td>-34.84</td>
<td>-80.23</td>
<td>-49.26</td>
</tr>
<tr>
<td>90.0</td>
<td>54.0</td>
<td>7.012</td>
<td>4.073</td>
<td>-61.461</td>
<td>-59.460</td>
<td>63.86</td>
<td>-35.65</td>
<td>-79.67</td>
<td>-48.67</td>
</tr>
<tr>
<td>90.0</td>
<td>57.0</td>
<td>6.848</td>
<td>3.972</td>
<td>-64.583</td>
<td>-59.818</td>
<td>62.62</td>
<td>-36.79</td>
<td>-79.56</td>
<td>-47.76</td>
</tr>
<tr>
<td>90.0</td>
<td>60.0</td>
<td>6.610</td>
<td>3.843</td>
<td>-65.920</td>
<td>-60.184</td>
<td>61.04</td>
<td>-38.25</td>
<td>-80.01</td>
<td>-46.52</td>
</tr>
<tr>
<td>90.0</td>
<td>63.0</td>
<td>6.320</td>
<td>3.687</td>
<td>-67.516</td>
<td>-60.476</td>
<td>59.12</td>
<td>-40.03</td>
<td>-81.17</td>
<td>-45.03</td>
</tr>
<tr>
<td>90.0</td>
<td>66.0</td>
<td>5.977</td>
<td>3.504</td>
<td>-69.537</td>
<td>-60.949</td>
<td>56.85</td>
<td>-42.13</td>
<td>-83.42</td>
<td>-43.39</td>
</tr>
<tr>
<td>90.0</td>
<td>69.0</td>
<td>5.582</td>
<td>3.295</td>
<td>-72.108</td>
<td>-60.827</td>
<td>54.25</td>
<td>-44.55</td>
<td>-87.60</td>
<td>-41.77</td>
</tr>
<tr>
<td>90.0</td>
<td>72.0</td>
<td>5.136</td>
<td>3.062</td>
<td>-75.593</td>
<td>-60.874</td>
<td>51.32</td>
<td>-47.29</td>
<td>-96.20</td>
<td>-40.34</td>
</tr>
<tr>
<td>90.0</td>
<td>75.0</td>
<td>4.638</td>
<td>2.806</td>
<td>-80.413</td>
<td>-60.841</td>
<td>48.07</td>
<td>-50.34</td>
<td>-118.61</td>
<td>-39.28</td>
</tr>
<tr>
<td>90.0</td>
<td>78.0</td>
<td>4.089</td>
<td>2.529</td>
<td>-82.856</td>
<td>-60.748</td>
<td>44.50</td>
<td>-53.71</td>
<td>-176.74</td>
<td>-38.70</td>
</tr>
</tbody>
</table>

**NOTES:**
- **NAP** = NUMBER OF ATTACHMENT NODES = 0
- **NPLT** = NUMBER OF PLATE NODES = 24
- **NAT** = NUMBER OF ATTACHMENT NODES = 0

**BISTATIC SCATTERING, LOOK = 2**

**THETA DEC. (DEG.) = 90.0**

**PHI DEC. (DEG.) = 45.0**
| 90,0 | 255,0 | 0.537 | 1.067 | -65.240 | -68.008 | -86.99 | 92.26 | -79.31 | -157.72 |
| 90,0 | 258,0 | 3.977 | 1.771 | -65.640 | -69.424 | -81.29 | 95.29 | -77.57 | -157.10 |
| 90,0 | 261,0 | 3.366 | 1.662 | -66.572 | -70.970 | -79.36 | 98.56 | -75.14 | -158.56 |
| 90,0 | 264,0 | 2.705 | 1.546 | -67.454 | -72.566 | -75.14 | 102.05 | -71.03 | -162.25 |
| 90,0 | 267,0 | 1.990 | 1.406 | -68.311 | -74.015 | -70.62 | 105.76 | -67.79 | -169.02 |
| 90,0 | 270,0 | 1.299 | 1.259 | -69.773 | -75.002 | -65.83 | 109.69 | -62.62 | -178.52 |
| 90,0 | 273,0 | 0.415 | 1.098 | -71.380 | -75.276 | -60.79 | 113.82 | -56.30 | -181.44 |
| 90,0 | 276,0 | -0.455 | 0.926 | -73.057 | -74.917 | -55.41 | 118.15 | -45.43 | -163.69 |
| 90,0 | 279,0 | -1.366 | 0.731 | -75.057 | -74.239 | -50.04 | 122.67 | -31.01 | -159.59 |
| 90,0 | 282,0 | -2.333 | 0.546 | -78.826 | -73.903 | -44.40 | 127.19 | -21.31 | -158.34 |
| 90,0 | 285,0 | -3.353 | 0.342 | -77.676 | -72.852 | -38.67 | 132.30 | 19.80 | -161.01 |
| 90,0 | 288,0 | -4.426 | 0.130 | -76.642 | -72.315 | -32.92 | 137.40 | 48.20 | -165.17 |
| 90,0 | 291,0 | -5.555 | 0.086 | -74.693 | -71.895 | -27.26 | 142.70 | 69.06 | -170.97 |
| 90,0 | 294,0 | -6.741 | 0.035 | -72.705 | -71.565 | -21.83 | 148.19 | 83.52 | -178.13 |
| 90,0 | 297,0 | -7.940 | 0.023 | -70.949 | -71.291 | -16.85 | 153.88 | 94.23 | -173.56 |
| 90,0 | 300,0 | -9.276 | 0.006 | -69.486 | -71.032 | -12.61 | 159.78 | 102.82 | -164.29 |
| 90,0 | 303,0 | -10.601 | 0.000 | -68.158 | -70.744 | -9.52 | 165.88 | 110.20 | -154.30 |
| 90,0 | 306,0 | -11.910 | 0.000 | -66.057 | -70.390 | -6.11 | 172.18 | 116.08 | -143.83 |
| 90,0 | 309,0 | -13.213 | 0.000 | -66.478 | -69.942 | -4.95 | 178.67 | 121.12 | -133.20 |
| 90,0 | 312,0 | -13.995 | 0.000 | -65.226 | -69.393 | -4.30 | 186.64 | 129.11 | -122.77 |
| 90,0 | 315,0 | -14.460 | 0.000 | -64.471 | -68.738 | -3.16 | 197.18 | 134.94 | -113.64 |
| 90,0 | 318,0 | -14.765 | 0.000 | -63.877 | -68.017 | -2.00 | 206.78 | 140.69 | -103.63 |
| 90,0 | 321,0 | -15.122 | 0.000 | -62.151 | -67.275 | -2.98 | 216.66 | 146.40 | -95.26 |
| 90,0 | 324,0 | -15.438 | 0.000 | -62.644 | -66.492 | -2.81 | 226.46 | 152.11 | -87.79 |
| 90,0 | 327,0 | -15.632 | 0.000 | -62.148 | -65.709 | -2.71 | 237.24 | 157.84 | -81.17 |
| 90,0 | 330,0 | -9.367 | 1.534 | -61.697 | -64.938 | -24.39 | 132.03 | 163.50 | -75.38 |
| 90,0 | 333,0 | -8.050 | 1.390 | -61.285 | -64.130 | -20.44 | 124.89 | 169.36 | -70.33 |
| 90,0 | 336,0 | -6.765 | 1.201 | -60.909 | -63.471 | -15.69 | 117.80 | 175.16 | -65.96 |
| 90,0 | 339,0 | -5.539 | 1.023 | -60.563 | -62.837 | -10.86 | 111.04 | 179.03 | -62.19 |
| 90,0 | 342,0 | -4.414 | 0.810 | -60.247 | -62.130 | -5.97 | 104.40 | 173.17 | -58.96 |
| 90,0 | 345,0 | -3.346 | 0.618 | -59.957 | -61.510 | 0.63 | 98.00 | 167.34 | -56.21 |
| 90,0 | 348,0 | -2.316 | 0.410 | -59.692 | -60.926 | 6.23 | 91.87 | 161.52 | -53.88 |
| 90,0 | 351,0 | 1.340 | 0.226 | -59.500 | -60.437 | 11.75 | 86.03 | 155.74 | -51.94 |
| 90,0 | 354,0 | 2.450 | 0.056 | -58.252 | -59.875 | 17.12 | 80.48 | 150.01 | -50.36 |
| 90,0 | 357,0 | 3.540 | 0.010 | -57.038 | -59.394 | 22.31 | 75.24 | 144.37 | -49.02 |
| 90,0 | 360,0 | 1.267 | 1.252 | -58.866 | -59.961 | 27.28 | 70.30 | 138.63 | -47.99 |

CPU RUN TIME FOR RUN 1 GEOMETRY 1 = 186.45 SECONDS

TOTAL CPU RUN TIME = 186.96 SECONDS
APPENDIX 4
OUTPUT FOR DESIGN EXAMPLE 4

INPUT DATA

FREQ. (MHz) = 300.000  WAVE NO. = 1.000  WIRE DIAM. (μm) = 0.0010000

WIRE CONDUCTIVITY = 1.00 MEGAWH/M

GEOMETRY FOR THE 3 PLATES

PLATE NUMBER 1 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIZE INDICATOR = 0

X, Y, Z COORD. (METERS) OF CORNER 1 = 0.50000 -0.50000 -1.00000
X, Y, Z COORD. (METERS) OF CORNER 2 = 0.50000 -0.50000 0.00000
X, Y, Z COORD. (METERS) OF CORNER 3 = -0.50000 -0.50000 0.00000
X, Y, Z COORD. (METERS) OF CORNER 4 = -0.50000 -0.50000 -1.00000

COORD. (METERS) OF 24 NODES ON THIS PLATE

PLATE NUMBER 2 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIZE INDICATOR = 0

X, Y, Z COORD. (METERS) OF CORNER 1 = 0.50000 -0.50000 0.00000
X, Y, Z COORD. (METERS) OF CORNER 2 = 0.50000 0.50000 0.00000
X, Y, Z COORD. (METERS) OF CORNER 3 = -0.50000 0.50000 0.00000
2 2 3 0.125000e+00

GEOMETRY FOR THE 1 ATTACHMENT JOINTS
1 SEGMENT END PLATE B(D)
1 1 0 2 0.20000

LISTING OF LOADS AND GENERATORS
0.100000e+01  0.000000e+00 VOLTS AT ATTACHMENT 1

NME = NUMBER OF WIRE NODES = 1
NMTN = NUMBER OF PLATE NODES = 64
NMT = NUMBER OF ATTACHMENT NODES = 1

INPUT ADmittANCE(NODE) = 0.015069 J -0.000726
INPUT INDUCANCE(NODE) = 52.348 J  1.994
EFFICIENCY(PERCENT) = 100.000

CPU RUN TIME FOR RUN 1 GEOMETRY 1 = 342.63 SECONDS

INPUT DATA
FREQ. (HZ) = 300.000  WAVE(N) = 1.000  WIRE RADIES(N) = 0.0010000
INT= 4  INTD= 18  INT = 4  IPIL= 1

WIRE CONDUCTIVITY = -1.00 MEGOHM/K

GEOMETRY FOR THE 3 PLATES

PLATE NUMBER 1(RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIDE INDICATOR = 0
X,Y,Z (COORD. (METERS)) OF CORNER 1 = 0.50000  -0.50000  -1.00000
X,Y,Z (COORD. (METERS)) OF CORNER 2 = 0.50000  -0.50000  0.00000
X,Y,Z (COORD. (METERS)) OF CORNER 3 = -0.50000  -0.50000  0.00000
X,Y,Z (COORD. (METERS)) OF CORNER 4 = -0.50000  -0.50000  -1.00000
\[ x, y, z \text{ (COORD. METERS)} \text{ of corner } 4 = \begin{bmatrix} -0.5000 \\ -0.5000 \\ 0.0000 \end{bmatrix} \]

COORD. (METERS) OF 24 NODES ON THIS PLATE

PLATE NUMBER: 3 (RECTANGULAR)

NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (METERS) = 0.25000
POLARIZATION INDICATOR = 3
GENERATING SIDE INDICATOR = 0

\[
\begin{align*}
  x, y, z \text{ (COORD. METERS)} \text{ of corner } 1 &= \begin{bmatrix} 0.2500 \\ 0.5000 \\ 0.0000 \end{bmatrix} \\
  x, y, z \text{ (COORD. METERS)} \text{ of corner } 2 &= \begin{bmatrix} -0.2500 \\ 0.5000 \\ 0.0000 \end{bmatrix} \\
  x, y, z \text{ (COORD. METERS)} \text{ of corner } 3 &= \begin{bmatrix} -0.2500 \\ 0.5000 \\ 1.0000 \end{bmatrix} \\
  x, y, z \text{ (COORD. METERS)} \text{ of corner } 4 &= \begin{bmatrix} 0.2500 \\ 0.5000 \\ 1.0000 \end{bmatrix}
\end{align*}
\]

COORD. (METERS) OF 10 NODES ON THIS PLATE

COORD. (METERS) OF 4 OVERLAP NODES BETWEEN PLATE 1, SIDE 2 AND PLATE 2, SIDE 4

COORD. (METERS) OF 2 OVERLAP NODES BETWEEN PLATE 2, SIDE 2 AND PLATE 3, SIDE 1

<table>
<thead>
<tr>
<th>POINTS ON THE WIRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X ) (1) ( Y ) (1) ( Z ) (1)</td>
</tr>
<tr>
<td>1 ( 0.0000e+00 ) ( 0.0000e+00 ) ( 0.0000e+00 )</td>
</tr>
<tr>
<td>2 ( 0.0000e+00 ) ( 0.0000e+00 ) ( 0.1250e+00 )</td>
</tr>
<tr>
<td>3 ( 0.0000e+00 ) ( 0.0000e+00 ) ( 0.2500e+00 )</td>
</tr>
</tbody>
</table>

NODES ON THE WIRE STRUCTURE

MAXIMUM NUMBER OF NODES AT ONE POINT = 1
MINIMUM NUMBER OF NODES AT ONE POINT = 1
NUMBER OF WIRE NODES = 1

<table>
<thead>
<tr>
<th>SEGMENTS ON THE WIRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J ) ( J_{A(J)} ) ( B(J) ) ( D(J) )</td>
</tr>
<tr>
<td>1 1 2 ( 0.125000e+00 )</td>
</tr>
</tbody>
</table>
COORD. (X, Y, Z) OF 24 HOLES ON THIS PLATE

PLATE NUMBER 3 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (INCHES) = 0.25000
POLARIZATION AUGMENT = 3
GENERATING SIDE AUGMENT = 0
X, Y, Z COORDINATES (INCHES) OF CORNER 1 = 0.50000  -0.50000  0.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 2 = -0.50000  0.50000  0.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 3 = -0.50000  -0.50000  0.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 4 = 0.50000  0.50000  0.00000

COORD. (X, Y, Z) OF 24 HOLES ON THIS PLATE

PLATE NUMBER 3 (RECTANGULAR)
NUMBER OF CORNERS = 4
MAXIMUM SEGMENT SIZE (INCHES) = 0.25000
POLARIZATION AUGMENT = 3
GENERATING SIDE AUGMENT = 0
X, Y, Z COORDINATES (INCHES) OF CORNER 1 = 0.25000  0.50000  0.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 2 = -0.25000  0.50000  0.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 3 = -0.25000  -0.50000  1.00000
X, Y, Z COORDINATES (INCHES) OF CORNER 4 = 0.25000  0.50000  1.00000

COORD. (X, Y, Z) OF 10 HOLES ON THIS PLATE

COORD. (X, Y, Z) OF 4 OVERLAP HOLES BETWEEN PLATE 1, SIDE 2 AND PLATE 2, SIDE 4

COORD. (X, Y, Z) OF 2 OVERLAP HOLES BETWEEN PLATE 2, SIDE 2 AND PLATE 3, SIDE 1

3 POINTS ON THE WIRE:

<table>
<thead>
<tr>
<th></th>
<th>X (1)</th>
<th>Y (1)</th>
<th>Z (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>0.300000</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>0.000000</td>
<td>0.300000</td>
<td>0.125000</td>
</tr>
<tr>
<td>3</td>
<td>0.000000</td>
<td>0.300000</td>
<td>0.250000</td>
</tr>
</tbody>
</table>
MODES ON THE WIRE STRUCTURE

MAXIMUM NUMBER OF NODES AT ONE POINT = 1
MINIMUM NUMBER OF NODES AT ONE POINT = 1
NUMBER OF WIRE NODES = 1

2 SEGMENTS ON THE WIRE

<table>
<thead>
<tr>
<th>J</th>
<th>IA(j)</th>
<th>IB(j)</th>
<th>D(j)</th>
<th>Q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.125000</td>
<td>00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0.125000</td>
<td>00</td>
</tr>
</tbody>
</table>

GEOMETRY FOR THE 1 ATTACHMENT POINTS

1 SEGMENT END PLATE B=00

| 1 | 1 | 0 | 2 | 0.20000 |

LISTING OF LOADS AND GENERATORS

0.100000=1 0.000000=00 VALUES AT ATTACHMENT 1

NG = NUMBER OF WIRE NODES = 1
NL/IW = NUMBER OF PLATE NODES = 64
NAT = NUMBER OF ATTACHMENT NODES = 1

DRUT ADMITTANCE (A/HES) = 0.015355 J -0.012869
DRUT IMPEDANCE (OHMS) = 158.174 J 31.623
EFFICIENCY (PERCENT) = 100.000

CPU RUN TIME FOR RUN 1 GEOMETRY 2 = 89.03 SECONDS

TOTAL CPU RUN TIME = 432.05 SECONDS
APPENDIX 5

OUTPUT FOR DESIGN EXAMPLE 5

INPUT DATA

FREQ. (kHz) = 300.000  WAVE GU = 1.000  WIRE RADIUS(GU) = 0.0010000
INT= 10  INTA= 18  INT = 4  IPFL= 1

WIRE CONDUCTIVITY = -1.00 MWDM/Hz

GEOMETRY FOR THE 1 PLATES

PLATE NUMBER 1 (POLARIZED)

NUMBER OF CORNERS = 8
MAXIMUM SEGMENT SIZE (METERS) = 0.250000
POLARIZATION DEGREE = 3
GEOGRAPHICAL SIZE INDICATOR = 0

X,Y,Z COORD. (METERS) OF CORNER 1 = 0.00000  0.390000  0.00000
X,Y,Z COORD. (METERS) OF CORNER 2 = 0.21200  0.212000  0.00000
X,Y,Z COORD. (METERS) OF CORNER 3 = 0.31000  0.00000  0.00000
X,Y,Z COORD. (METERS) OF CORNER 4 = 0.21200 -0.21200  0.00000
X,Y,Z COORD. (METERS) OF CORNER 5 = 0.00000 -0.390000  0.00000
X,Y,Z COORD. (METERS) OF CORNER 6 = -0.21200 -0.21200  0.00000
X,Y,Z COORD. (METERS) OF CORNER 7 = -0.31000  0.00000  0.00000
X,Y,Z COORD. (METERS) OF CORNER 8 = -0.21200  0.21200  0.00000

COORD. (METERS) OF 12 NODES ON THIS PLATE

NODE    X1   Y1   X2   Y2   X3   Y3   X4   Y4
A       1    0.24133 0.14133 0.00000 0.21200 0.21200 0.00000 0.30000 0.00000 0.07067 0.12933 0.00000
B       1    0.24133 0.14133 0.00000 0.27067 0.07067 0.00000 0.14133 -0.04133 0.00000 0.07067 0.12933 0.00000
A       2    0.27067 0.07067 0.00000 0.24133 0.14133 0.00000 0.07067 0.12933 0.00000 0.14133 -0.04133 0.00000
B       2    0.27067 0.07067 0.00000 0.31000 0.00000 0.00000 0.21200 -0.21200 0.00000 0.14133 -0.04133 0.00000
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.30000</td>
<td>0.00000</td>
<td>-0.21200</td>
<td>0.21200</td>
<td>0.00000</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.14133</td>
<td>-0.04133</td>
<td>0.00000</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td>-0.07067</td>
<td>-0.12933</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td>-0.21200</td>
<td>0.21200</td>
<td>0.00000</td>
<td>-0.30000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.27067</td>
<td>-0.07067</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.07067</td>
<td>-0.12933</td>
<td>0.00000</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td>-0.27067</td>
<td>-0.07067</td>
<td>0.00000</td>
<td>-0.24333</td>
<td>-0.14133</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.00000</td>
<td>0.30000</td>
<td>0.00000</td>
<td>0.21200</td>
<td>0.21200</td>
<td>0.00000</td>
<td>0.24133</td>
<td>0.14133</td>
<td>0.00000</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.21200</td>
<td>0.21200</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.30000</td>
<td>0.00000</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td>0.24133</td>
<td>0.14133</td>
<td>0.00000</td>
<td>0.27067</td>
<td>0.07067</td>
<td>0.00000</td>
<td>0.14133</td>
<td>-0.04133</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-0.14133</td>
<td>0.04133</td>
<td>0.00000</td>
<td>0.07067</td>
<td>0.12933</td>
<td>0.00000</td>
<td>0.14133</td>
<td>-0.04133</td>
<td>0.00000</td>
<td>-0.07067</td>
<td>-0.12933</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.14133</td>
<td>-0.04133</td>
<td>0.00000</td>
<td>0.27067</td>
<td>0.07067</td>
<td>0.00000</td>
<td>0.30000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.21200</td>
<td>-0.21200</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-0.07067</td>
<td>-0.12933</td>
<td>0.00000</td>
<td>0.14133</td>
<td>-0.04133</td>
<td>0.00000</td>
<td>0.21200</td>
<td>-0.21200</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.30000</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Listing of Loads and Comparisons**

<table>
<thead>
<tr>
<th>N1</th>
<th>M1</th>
<th>N2</th>
<th>M2</th>
<th>N3</th>
<th>M3</th>
<th>N4</th>
<th>M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>N2</td>
<td>N3</td>
<td>N4</td>
<td>M1</td>
<td>M2</td>
<td>M3</td>
<td>M4</td>
</tr>
</tbody>
</table>

**Number of WIDE Holes = 0**

**Number of PLATE Holes = 12**

**Number of ATTACHMENT HOLES = 0**

**Total CPU Run Time = 3.63 Seconds**
APPENDIX 6

SUBROUTINE PLPLCK

SUBROUTINE PLPLCK(PCN, IPL, ICN, N, TOUCH, NP, IOK)

DIMENSION PCN(3,ICN, IPL)

IF (N .EQ. 3) RETURN

DS1 = SQRT((PCN(1,1,NP) - PCN(1,1,NP))^2 + (PCN(2,1,NP) - PCN(2,1,NP))^2 + (PCN(3,1,NP) - PCN(3,1,NP))^2)

DSN = SQRT((PCN(1,1,NP) - PCN(1,1,NP))^2 + (PCN(2,1,NP) - PCN(2,1,NP))^2 + (PCN(3,1,NP) - PCN(3,1,NP))^2)

CAX = (PCN(1,2,NP) - PCN(1,1,NP)) / DS1

CBX = (PCN(2,2,NP) - PCN(2,1,NP)) / DS1

CGX = (PCN(3,2,NP) - PCN(3,1,NP)) / DS1

CAT = (PCN(1,N,NP) - PCN(1,1,NP)) / DSN

CBT = (PCN(2,N,NP) - PCN(2,1,NP)) / DSN

CGT = (PCN(3,N,NP) - PCN(3,1,NP)) / DSN

CC = CAX*CAT + CBX*CBT + CGX*CGT

IF (ABS(CC) .LT. 0.99999) GOTO 15

100

WRITE (6,100) N, NP

100 FORMAT (*** WARNING, CORNERS 1 AND ,13, OF PLATE',13

6, ' ARE PARALLEL')

RETURN

15

SS = SQRT(1. - CC*CC

CAY = (CBX*CGT - CBT*CGX) / SS

CBY = (CGX*CAT - CGT*CAX) / SS

CGY = (CAX*CBT - CAT*CBX) / SS

CAZ = CBX*CGT - CBT*CGX

CBZ = CGX*CAX - CGY*CAX

CGZ = CAX*CBT - CAT*CBX

DO 20 IC = 3, N - 1

XP = CAX*(PCN(1,IC,NP) - PCN(1,1,NP)) + CBX*(PCN(2,IC,NP) - PCN(2,1,NP))

6 + CGX*(PCN(3,IC,NP) - PCN(3,1,NP))

YP = CAY*(PCN(1,IC,NP) - PCN(1,1,NP)) + CBY*(PCN(2,IC,NP) - PCN(2,1,NP))

6 + CGY*(PCN(3,IC,NP) - PCN(3,1,NP))

ZP = CAZ*(PCN(1,IC,NP) - PCN(1,1,NP)) + CBZ*(PCN(2,IC,NP) - PCN(2,1,NP))

6 + CGZ*(PCN(3,IC,NP) - PCN(3,1,NP))

IF (ZP .EQ. 0.) GOTO 20

IF (XP - GP .GT. TOUCH/2.) GOTO 25

PCN(1,IC,NP) = CA*X*XP + CAY*YP + PCN(1,1,NP)

PCN(2,IC,NP) = CB*X*XP + CBY*YP + PCN(2,1,NP)

PCN(3,IC,NP) = CG*X*XP + CGY*YP + PCN(3,1,NP)

GOTO 20

25

IOX = 0

WRITE (6,105) NP, IC, EP, N

105 FORMAT (*** WARNING, PLATE',13, ', CORNER',13, ', IS'

6, 'E10.3', ' METERS OUT FOR THE PLANE FORMED BY CORNERS 1, 2, AND',13)

20 CONTINUE

END

205
APPENDIX 7
SUBROUTINE PLATE3

SUBROUTINE PLATE3 (PC, NC, ICN, NP, NDNPLT, PA, PB, IPLM, SEGM, & IQAD, WV, IRE, IP, MPL1, MPL2, IOK, NM12, NM23, IGS)
C THIS SUBROUTINE BREAKS UP A POLYGONAL PLATE INTO QUADRATLATERAL MODES.
C PC = COORDINATES OF PLATE
C NC = NUMBER OF CORNERS IN PLATE
C NP = PLATE NUMBER
C WV = WAVELENGTH IN METERS
C MPL1 = NUMBER OF MODES WITH FIRST POLARIZATION
C MPL2 = TOTAL NUMBER OF MODES ON PLATE
C IOK = STATUS INDICATOR; = 0 IF RUN IS TO ABORT
C IGS = GENERATING SIDE NUMBER, = 0 IF GENERATING SIDE TO BE CHOSEN BY PLATE3
C MDME = MAXIMUM MODE DIFFERENCE FOR POLYGON WITH EVEN NUMBER OF SIDES
C MDMO = MAXIMUM MODE DIFFERENCE FOR POLYGON WITH ODD NUMBER OF SIDES
C EVSD = .TRUE. IF POLYGON HAS EVEN NUMBER OF SIDES

DIMENSION PC(3,ICN), NDNPLT(1), PA(IPLM,4,3), PB(IPLM,4,3)
DIMENSION GPT(50,2,3), CA(3), CB(3), CG(3), CD(3), CE(3)
LOGICAL EVSD
C SET MAXIMUM MODE DIFFERENCE PARAMETERS
MDME=5
MDMO=2
MPL1=0
MPL2=0
NM12=0
NM23=0
SEGM*WV
C SET EVEN SIDE INDICATOR
EVSD=.TRUE.
IF(NC. NE. 2* (NC/2) )EVSD=.FALSE.
IF(IP. NE. 0) GOTO 4
NDNPLT(NP)=0
IF(NP. NE. 1) NDNPLT(NP)=NDNPLT(NP-1)
RETURN
C SET REFERENCE DIRECTION FOR INTERNAL ANGLE CHECK
4 IC=(NC+1)/2
100 DA=SQR((PC(1,IC)-PC(1,1))**2+(PC(2,IC)-PC(2,1))**2
&+(PC(3,IC)-PC(3,1))**2)
DB=SQR((PC(1,2)-PC(1,1))**2+(PC(2,2)-PC(2,1))**2
&+(PC(3,2)-PC(3,1))**2)
DO 5 IQ=1,3
CA(IQ)=(PC(IQ,IC)-PC(IQ,1))/DA
206
5 \[ CB(1Q) = (PC(IQ,2) - PC(IQ,1))/DB \]
\[ CC = CA(1)*CB(1) + CA(2)*CB(2) + CA(3)*CB(3) \]
\[ SS = \sqrt{1 - CC^2} \]
\[ IF(SS \geq 0.02) \text{GOTO 105} \]

105 \[ ISD = 0 \]
\[ IPD = 0 \]
\[ CG(1) = (CB(2)*CA(3) - CA(2)*CB(3))/SS \]
\[ CG(2) = (CB(3)*CA(1) - CA(3)*CB(1))/SS \]
\[ CG(3) = (CB(1)*CA(2) - CA(1)*CB(2))/SS \]

C CHECK FOR INTERNAL ANGLES > 180 DEGREES; FIND MAXIMUM LENGTH SIDE

DO 10 ICO = 1, NC
IC = ICO + 1
IF(ICO.EQ.NC) IC = 1
IC2 = ICO + 1
DB = SQRT((PC(1,ICO2) - PC(1,ICO))^2 + (PC(2,ICO2) - PC(2,ICO))^2 + (PC(3,ICO2) - PC(3,ICO))^2)
IF(DB.LE.DMX) GOTO 15
DMX = DB
ISMX = IC

15 DO II IQ = 1, 3
CA(IQ) = CB(IQ)
11 \[ CB(IQ) = (PC(IQ,ICO2) - PC(IQ,ICO))/DB \]
\[ CC = CA(1)*CB(1) + CA(2)*CB(2) + CA(3)*CB(3) \]
\[ SS = \sqrt{1 - CC^2} \]
\[ IF(SS \geq 0.02) \text{GOTO 110} \]

C INTERNAL ANGLE = 180 DEGREES

IFD = IPD + 1
GOTO 10

110 CD(1) = (CB(2)*CA(3) - CA(2)*CB(3))/SS
CD(2) = (CB(3)*CA(1) - CA(3)*CB(1))/SS
CD(3) = (CB(1)*CA(2) - CA(1)*CB(2))/SS
DP = CG(1)*CD(1) + CG(2)*CD(2) + CG(3)*CD(3)
IF(DP.GT.0.) ISD = ISD + 1
IF(DP.LT.0.) IDC = IC

10 CONTINUE
IF(ISD+IPD.EQ.NC) GOTO 20
IF(ISD.EQ.1 .OR. NC-ISD-IPD.EQ.1) GOTO 95
WRITE(6,500) NP
500 FORMAT('**** ERROR *** PLATE, I3, ' HAS MORE THAN ONE IN
'Ternal ANGLE GREATER THAN 180 DEGREES')
NDNFLT(NP) = 0
IF(NP.NE.1) NDNFLT(NP) = NDNFLT(NP-1)
I0K = 0
RETURN

20 CONTINUE
C NO INTERNAL ANGLES > 180 DEGREES; SET ISMX = GENERATING SIDE
IF(NC.EQ.4 .AND. IP.EQ.1) ISMX = 1
IF(NC.EQ.4 .AND. IP.EQ.2) ISMX = 2
IF(IGS.NE.0) ISMX = IGS
IC2 = ISMX
IC = ISMX + 1
IF(ISMX.EQ.NC) IC = 1
IIS = 0
GOTO 200

95 CONTINUE
C INTERNAL ANGLE AT CORNER IDC > 180 DEGREES; SET GENERATING SIDE
   IF(ISD.NE.1)GOTO 97
   DO 98 IQ=1,3
98   CG(IQ)=CG(IQ)
   IC2=ICD+1
   IF(IDC.EQ.NC)IC2=1
   IF(ISD.EQ.1)IC2=2
   IC=IC2+1
   IF(IC2.EQ.NC)IC=1
   IIS=0
C SET GRID POINTS STARTING AT ENDPOINTS OF GENERATING SIDE
200 IF(EVSD).MDM=MDE
   IF(.NOT.EVSD).MDM=MDO
C INCREMENT SIDE 2; CHECK FOR COMPLETION
   IC2=IC2+1
   IF(IC2.EQ.NC)IC2=1
   IF(IC2.EQ.IC.AND.ISD.EQ.0)GOTO 205
   IIS=IIS+1
C INCREMENT SIDE 1
   IC=IC-1
   IF(IC.EQ.0)IC=NC
   IF(IC.EQ.IC2)GOTO 210
C REMAINING PIECE IS TRIANGLE; PUT EXTRA CORNER AT MIDPOINT OF
C LONGEST REMAINING SIDE
   DA=SQRT((PC(1,IC)-GPT(IIS-1,1,1))**2+(PC(2,IC)
   &-GPT(IIS-1,1,2))**2+(PC(3,IC)-GPT(IIS-1,1,3))**2)
   DB=SQRT((PC(1,IC2)-GPT(IIS-1,2,1))**2+(PC(2,IC2)
   &-GPT(IIS-1,2,2))**2+(PC(3,IC2)-GPT(IIS-1,2,3))**2)
   IF(DA.GT.DB)GOTO 215
   DO 220 IQ=1,3
   GPT(IIS,1,IQ)=PC(IQ,IC)
220   GPT(IIS,2,IQ)=(PC(IQ,IC2)+GPT(IIS-1,2,IQ))/2.
   IC2=IC2-1
   IF(IC.EQ.0)IC=NC
   GOTO 225
215 DO 230 IQ=1,3
   GPT(IIS,1,IQ)=PC(IQ,IC)+GPT(IIS-1,1,IQ))/2.
230   GPT(IIS,2,IQ)=PC(IQ,IC2)
   IC2=IC2-1
   IF(IC.EQ.0)IC=NC
   GOTO 225
C REMAINING PIECE IS AT MINIMUM A QUADRILATERAL
210 DO 325 IQ=1,3
   GPT(IIS,1,IQ)=PC(IQ,IC)
235   GPT(IIS,2,IQ)=PC(IQ,IC2)
225 IF(IIS.EQ.1)GOTO 200
C CHECK SIDES FOR EXTRAS NEEDED GRID POINT
   DA=SQRT((GPT(IIS,1,1)-GPT(IIS-1,1,1))**2+
   &(GPT(IIS,1,2)-GPT(IIS-1,1,2))**2+(GPT(IIS,1,3)-GPT(IIS-1,1,3))**2)
   DB=SQRT((GPT(IIS,2,1)-GPT(IIS-1,2,1))**2+
   &(GPT(IIS,2,2)-GPT(IIS-1,2,2))**2+(GPT(IIS,2,3)-GPT(IIS-1,2,3))**2)
   N=DA/SEG+0.99
   NBS=DB/SEG+0.99
   NES=MAX0(NAS,NBS)
   IF(NC.EQ.4.AND.IP.EQ.3)NES=MAX0(2,NES)
   IF(NES.EQ.1)GOTO 200
C SET DIRECTIONAL COSINES FOR SIDES
   DO 115 IQ=1,3

208
CA(IQ) = (GPT(IIS,1,IQ) - GPT(IIS-1,1,IQ))/DA
CB(IQ) = (GPT(IIS,2,IQ) - GPT(IIS-1,2,IQ))/DB
C CHECK NUMBER OF MODES AGAINST MAXIMUM MODE DIFFERENCE PARAMETER
IF(NAS-NBS.LT.MDM) GOTO 120

C SHORTEN SIDE 1; TOGGLE EVSD
DO 125 IQ=1,3
125 GPT(IIS,1,IQ) = GPT(IIS-1,1,IQ) + DB*CA(IQ)
   DA=DB
   NAS=NBS
   NES=NBS
   IC=IC+1
   IF(IC.GT.NC) IC=1
   EVSD=.NOT.EVSD
   GOTO 130
120 IF(NBS-NAS.LT.MDM) GOTO 130

C SHORTEN SIDE 2; TOGGLE EVSD
DO 135 IQ=1,3
135 GPT(IIS,2,IQ) = GPT(IIS-1,2,IQ) + DA*CB(IQ)
   DB=DA
   NAS=NBS
   NES=NBS
   IC2=IC2+1
   IF(IC2.EQ.0) IC2=NC
   EVSD=.NOT.EVSD
130 IF(NES.EQ.1) GOTO 200

C CHECK FOR MODES WITH INTERNAL ANGLES > 180 DEGREES
DD=SQRT(GPT(IIS,1,1)-GPT(IIS,2,1))**2 +
    & (GPT(IIS,1,2)-GPT(IIS,2,2))**2 + (GPT(IIS,1,3)-GPT(IIS,2,3))**2)
   DO 240 IQ=1,3
240 CD(IQ) = (GPT(IIS,1,IQ) - GPT(IIS,2,IQ))/DD
   GPT(IIS+NES-1,1,IQ) = GPT(IIS,1,IQ)
   GPT(IIS+NES-1,2,IQ) = GPT(IIS,2,IQ)
   CC = CA(1)*CD(1) + CA(2)*CD(2) + CA(3)*CD(3)
   SS = SQRT(1.-CC**CC)
   IF(SS.LT.0.02) GOTO 244
   CE(1) = (CA(2)*CD(3)-CA(3)*CD(2))/SS
   CE(2) = (CA(3)*CD(1)-CA(1)*CD(3))/SS
   CE(3) = (CA(1)*CD(2)-CA(2)*CD(1))/SS
   DP = CG(1)*CE(1) + CG(2)*CE(2) + CG(3)*CE(3)
   IF(DP.GT.0.) GOTO 244
   WRITE(6,501)NP
501 FORMAT('**** ERROR **** PLATE',I3, ' HAS MODES WITH ',
    & 'INTERNAL ANGLES GREATER THAN 180 DEGREES')
   CONTINUE

C ADD EXTRA NEEDED GRID POINTS
   DO 245 IR=1,NES
245 GPT(IIS+IR-1,1,IQ) = GPT(IIS-1,1,IQ) + IR*DA/NES*CA(IQ)
   GPT(IIS+IR-1,2,IQ) = GPT(IIS-1,2,IQ) + IR*DB/NES*CB(IQ)
   CONTINUE

C FIND LONGEST DISTANCE BETWEEN GRID POINTS
   DMX=0.
   DO 205 IS=1,IIS
205 DA=SQR((GPT(IS,2,1)-GPT(IS,1,1))**2 + (GPT(IS,2,2)-GPT(IS,1,2))**2)
   IF(DA.GT.DMX) DMX=DA
   CONTINUE
70 CONTINUE
C SET NUMBER OF MODES ALONG AND ORTHONAL TO GENERATING SIDE
NMP=IIS-1
NMO=DMX/SEG+0.99
IF(IRES.EQ.1.AND.(ISMX.EQ.1.OR.ISMX.EQ.3))NMI2=NMO
IF(IRES.EQ.1.AND.(ISMX.EQ.1.OR.ISMX.EQ.3))NMI3=NMO
IF(IRES.EQ.1.AND.(ISMX.EQ.2.OR.ISMX.EQ.4))NMI2=NMO
IF(IRES.EQ.1.AND.(ISMX.EQ.2.OR.ISMX.EQ.4))NMI3=NMO
NDN=0
IF(NP.NE.1)NDN=NDN+1
MPL1=NMP*(NMO-1)
MPL2=MPL1+NMO*(NMP-1)

C GENERATE MODES IN DIRECTION ALONG GENERATING SIDE
DO 75 I=1,NMP
DA=SQR((GPT(1,2,1)-GPT(I,1,1))**2+(GPT(1,2,2)-GPT(I,1,2))**2
&+(GPT(1,2,3)-GPT(I,1,3))**2)
DB=SQR((GPT(I+1,2,1)-GPT(I+1,1,1))**2+(GPT(I+1,2,2)-GPT(I+1,1,2))**2
&+(GPT(I+1,2,3)-GPT(I+1,1,3))**2)
DO 76 IQ=1,3
CA(IQ)=(GPT(1,2,IQ)-GPT(I,1,IQ))/DA
CB(IQ)=(GPT(I+1,2,IQ)-GPT(I+1,1,IQ))/DB
DO 80 J=1,NMO-1
NDN=NDN+1
IF(IRES.EQ.1)IQAD=-3
IF(IRES.EQ.0)IQAD=0
DO 80 IQ=1,3
PA(NDN,IQ)=GPT(I,1,IQ)+J*DDA*CA(IQ)
PB(NDN,IQ)=GPT(I,1,IQ)+(J-1)*DDA*CA(IQ)
PA(NDN+1,IQ)=GPT(I+1,1,IQ)+J*DDB*CB(IQ)
PB(NDN+1,IQ)=GPT(I+1,1,IQ)+(J-1)*DDB*CB(IQ)
80 CONTINUE
75 CONTINUE
NDNPLT(NP)=NDN
IF(NC.EQ.4.AND.IP.NE.3)RETURN

C GENERATE MODES ORTHOGONAL TO GENERATING SIDE
DO 85 I=1,NMO
DO 90 J=1,NMP-1
NDN=NDN+1
IF(IRES.EQ.1)IQAD=-3
IF(IRES.EQ.0)IQAD=0
DA=SQR((GPT(I,2,1)-GPT(J,1,1))**2+(GPT(I,2,2)-GPT(J,1,2))**2
&+(GPT(I,2,3)-GPT(J,1,3))**2)
DB=SQR((GPT(J+1,2,1)-GPT(J+1,1,1))**2+(GPT(J+1,2,2)-GPT(J+1,1,2))**2
&+(GPT(J+1,2,3)-GPT(J+1,1,3))**2)
DG=SQR((GPT(J+2,2,1)-GPT(J+2,1,1))**2+(GPT(J+2,2,2)-GPT(J+2,1,2))**2
&+(GPT(J+2,2,3)-GPT(J+2,1,3))**2)
DDA=DA/NMO
DDB=DB/NMO
DDG=DG/NMO
DO 90 IQ=1,3
CA(IQ)=(GPT(J,2,IQ)-GPT(J,1,IQ))/DA
90 CONTINUE
210
<table>
<thead>
<tr>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB(IQ) = ((\text{GPT}(J+1, 2, IQ) - \text{GPT}(J+1, 1, IQ)) / DB))</td>
</tr>
<tr>
<td>CG(IQ) = ((\text{GPT}(J+2, 2, IQ) - \text{GPT}(J+2, 1, IQ)) / DG))</td>
</tr>
<tr>
<td>PA(NDN, 1, IQ) = (\text{GPT}(J+2, 1, IQ) + (I-1) * DDB * CB(IQ))</td>
</tr>
<tr>
<td>PA(NDN, 2, IQ) = (\text{GPT}(J, 1, IQ) + (I-1) * DDA * CA(IQ))</td>
</tr>
<tr>
<td>PA(NDN, 3, IQ) = (\text{GPT}(J, 1, IQ) + I * DDA * CA(IQ))</td>
</tr>
<tr>
<td>PA(NDN, 4, IQ) = (\text{GPT}(J+1, 1, IQ) + I * DDB * CB(IQ))</td>
</tr>
<tr>
<td>PB(NDN, 1, IQ) = (\text{PA}(NDN-1, 10) - \text{PA}(NDN-1, 10))</td>
</tr>
<tr>
<td>PB(NDN, 2, IQ) = (\text{GPT}(J+2, 1, IQ) + (I-1) * DDG * CG(IQ))</td>
</tr>
<tr>
<td>PB(NDN, 3, IQ) = (\text{GPT}(J+2, 1, IQ) + I * DDG * CG(IQ))</td>
</tr>
<tr>
<td>PB(NDN, 4, IQ) = (\text{PA}(NDN, 4, IQ))</td>
</tr>
</tbody>
</table>

90 CONTINUE
85 CONTINUE
NDNPLT(NP) = NDN
RETURN
END
APPENDIX 8

SUBROUTINE POPLOV

SUBROUTINE POPLOV(NPLTS, PCN, NCNRS, TOUCH
2,SEG1, PA, PB, NOVT, NPLTS, I1, I1M, ICN
3, IOVT, DOVL, ITK, NOPL, IQUAD, W, NDNPLT, OVEP)
DIMENSION PCN(3,ICN, IM), NCNRS(1), SEGM(1)
2, PA(I1M,4,3), PB(I1M,4,3), IOVT(I1M,4), DOVL(1)
3, ITK(1), IQUAD(1), NDNPLT(1), OVEP(I1M,3,2)
DIMENSION IBC(2), OE(3,2), CE(3)
2, CA(3), CB(3), CA1(3), CA2(3), CB1(3), CB2(3), OMEP(3,4,2)
DIMENSION OEI(3,2), OEQ(3,2), OER(3,2), VDM(3)
IF(NPLTS.LT.2) RETURN
TPSI=0.258819
SPSI=SQRT(1.-TPSI**2)
CLEN=WV/25.
NOVT=0
NOPL=0
C CHECK FOR TOUCHING PLATES
DO 100 NPA=1,NPLTS-1
DO 100 NPB=NPA+1,NPLTS
DO 110 ISA=1,NCNRS(NPA)
C COMPUTE DIRECTIONAL COSINES OF SIDE ISA, PLATE NPA
isa=ISA+1
IF(ISA.EQ.NCNRS(NPA))ISA=1
CXA=PCN(1,ISA,NPA)-PCN(1,ISA,NPA)
CYA=PCN(2,ISA,NPA)-PCN(2,ISA,NPA)
CZA=PCN(3,ISA,NPA)-PCN(3,ISA,NPA)
DA=SQR(T(CXA*CXA+CYA*CYA+CZA*CZA)
CXA=CXA/DA
CYA=CYA/DA
CZA=CZA/DA
ITCH=0
Do 115 ICB=1,NCNRS(NPB)
C COMPUTE DISTANCE BETWEEN CORNER ICB, PLATE NPB AND
C SIDE ISA, PLATE NPA
XAB=PCN(1,ICB,NPB)-PCN(1,ISA,NPA)
YAB=PCN(2,ICB,NPB)-PCN(2,ISA,NPA)
ZAB=PCN(3,ICB,NPB)-PCN(3,ISA,NPA)
DSQ=XAB*XAB+YAB*YAB+ZAB*ZAB-(XAB*CXA+YAB*CYA+CZA*ZAB)**2
IF(DSQ.GT.TOUCH**2)GO TO 115
ITCH=ITCH+1
IBC(ITCH)=ICB

212
CONTINUE
IF(ITCH.NE.2)GOTO 110
C CORNERS OF PLATE NPB TOUCHES SIDE ISA OF PLATE NPA
C FIND ENDPOINTS OF OVERLAP SEGMENT
ISB=IBC(1)
IF(IBBC(1).EQ.1 .AND. IBC(2).EQ.CNRS(NPB))ISB=CNRS(NPB)
IOV=0
DO 120 J=1,2
ICB=IBC(J)
DCIJ=(PCN(1,ISA,NPA)-PCN(1,ICB,NPB))**2
2+(PCN(2,ISA,NPA)-PCN(2,ICB,NPB))**2
3+(PCN(3,ISA,NPA)-PCN(3,ICB,NPB))**2
DCJH=(PCN(1,ISA,NPA)-PCN(1,ICB,NPB))**2
2+(PCN(2,ISA,NPA)-PCN(2,ICB,NPB))**2
3+(PCN(3,ISA,NPA)-PCN(3,ICB,NPB))**2
IF(DCJL+DCJH-DA*DA.GT.2.*TOUCH**2)G0
120 IOV=IOV+1
DO 130 IQ=1,3
130 OE(IQ,IOV)=PCN(IQ,ICB,NPB)
120 CONTINUE
IF(IOV.EQ.2)GOTO 140
ICB=IBC(1)
ICB=IBC(2)
DBS=(PCN(1,ICB,NPB)-PCN(1,ICB1,NPB))**2
2+(PCN(2,ICB,NPB)-PCN(2,ICB1,NPB))**2
3+(PCN(3,ICB,NPB)-PCN(3,ICB1,NPB))**2
DO 150 J=1,2
ICA=ISA
IF(J.EQ.2)ICA=ISA
DCIJ=(PCN(1,ICB,NPB)-PCN(1,ICA,NPA))**2
2+(PCN(2,ICB,NPB)-PCN(2,ICA,NPA))**2
3+(PCN(3,ICB,NPB)-PCN(3,ICA,NPA))**2
DCJH=(PCN(1,ICB,NPB)-PCN(1,ICA,NPA))**2
2+(PCN(2,ICB1,NPB)-PCN(2,ICA,NPA))**2
3+(PCN(3,ICB1,NPB)-PCN(3,ICA,NPA))**2
IF(DCJL+DCJH-DBS.GT.2.*TOUCH**2)GOTO 150
IOV=IOV+1
DO 160 IQ=1,3
160 OE(IQ,IOV)=PCN(IQ,ICA,NPA)
IF(IOV.NE.2)GOTO 150
DOV=SQRT((OE(1,1)-OE(1,2))**2+(OE(2,1)-OE(2,2))**2
2+(OE(3,1)-OE(3,2))**2)
IF(DOV.GT.TOUCH)GOTO 165
IOV=1
150 CONTINUE
IF(IOV.NE.2)GOTO 110
140 DOV=SQRT((OE(1,1)-OE(1,2))**2+(OE(2,1)-OE(2,2))**2
2+(OE(3,1)-OE(3,2))**2)
IF(DOV.GT.TOUCH)GOTO 165
110 CONTINUE
GOTO 100
C WRITE OVERLAB MODES INTO TABLE
165 NOPL=NOPL+1
IOVT(NOPL,1)=NPA
IOVT(NOPL,2)=ISA
IOVT(NOPL,3)=NPB
IOVT(NOPL,4)=ISB
DOVL(NOPL)=DOV
DO 166 IQ=1,3
OPEP(NOPL,IQ,1)=OE(IQ,1)
OPEP(NOPL,IQ,2)=OE(IQ,2)
CALL PGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEG)
4,NDNPLT,PA,PB, IPLM, WTVUCH.TPSI, SPSI
6, OMSPI, ITK(NOPL))
100 CONTINUE
IF (NOPL.EQ.0) RETURN
IF (NOPL.LE.2) GOTO 200
C SEARCH OVERLAP TABLE, REMOVING UNNECESSARY MODES
DO 170 I=1,NOPL-2
NPAI=IOVT(I,1)
ISAI=IOVT(I,2)
NPBI=IOVT(I,3)
ISBI=IOVT(I,4)
DOVI=DOVL(I)
DII=DOVI/IABS(ITK(I))
DO 171 IQ=1,3
OEI(IQ,1)=OVEP(I,IQ,1)
OEI(IQ,2)=OVEP(I,IQ,2)
171 CE(IQ)=(OEI(IQ,2)-OEI(IQ,1))/DOVI
DO 175 J=I+1,NOPL-1
NPAJ=IOVT(J,1)
ISAJ=IOVT(J,2)
NPBJ=IOVT(J,3)
ISBJ=IOVT(J,4)
IF (NPAI.NE.NPAJ .OR. ISAI.NE.ISAJ) GOTO 175
DOVJ=DOVL(J)
DDJ=DOVJ/IABS(ITK(J))
DO 176 IQ=1,3
OEJ(IQ,1)=OVEP(J,IQ,1)
OEJ(IQ,2)=OVEP(J,IQ,2)
176 VDM(IQ)=(OEJ(IQ,2)-OEJ(IQ,1))/DOVJ
IF (CE(1)*VDM(1)+CE(2)*VDM(2)+CE(3)*VDM(3).GT.0.) GOTO 177
DO 178 J=1,3
OE(IQ,1)=OEJ(IQ,1)
OE(IQ,2)=OEJ(IQ,2)
178 OE(IQ,1)=OE(IQ,1)
177 CONTINUE
DO 180 K=J+1,NOPL
NPAK=IOVT(K,1)
ISAK=IOVT(K,2)
NPBK=IOVT(K,3)
ISBK=IOVT(K,4)
IF (NPAJ.NE.NPAK .OR. ISAJ.NE.ISAK .OR. NPBJ.NE.NPBK .OR.
2ISBJ.NE.ISBK) GOTO 180
DOVK=DOVL(K)
DDK=DOVK/IABS(ITK(K))
DO 181 IQ=1,3
OEK(IQ,1)=OVEP(K,IQ,1)
OEK(IQ,2)=OVEP(K,IQ,2)
181 VDM(IQ)=(OEK(IQ,2)-OEK(IQ,1))/DOVK
IF (CE(1)*VDM(1)+CE(2)*VDM(2)+CE(3)*VDM(3).GT.0.) GOTO 182
DO 183 IQ=1,3
OE(IQ,1)=OEK(IQ,1)
OE(IQ,2)=OEK(IQ,2)
183 OE(IQ,1)=OE(IQ,1)
182 CONTINUE
IF(ABS(DOVI-DOVJ).GT.CLEN .OR. ABS(DOVI-DOVK).GT.CLEN .OR.
&ABS(DOVJ-DOVK).GT.CLEN) GOTO 400
ICK=0
IF(IABS(ITK(I)).EQ.IABS(ITK(J)))ICK=1
IF(IABS(ITK(I)).EQ.IABS(ITK(K)))ICK=ICK+2
IF(IABS(ITK(J)).EQ.IABS(ITK(K)))ICK=ICK+4
IF(ICK.NE.7) GOTO 405
ITK(K)=-ITK(K)
GOTO 175
410 ITK(J)=IABS(ITK(J))
GOTO 175
405 IF(ICK.EQ.1)ITK(K)=IABS(ITK(K))
IF(ICK.EQ.2)ITK(J)=-IABS(ITK(J))
IF(ICK.EQ.4)ITK(I)=IABS(ITK(I))
GOTO 175
400 ICK=0
IF(ABS(DDI-DDJ).LE.CLEN/MAXO(IABS(ITK(I)),IABS(ITK(J))))ICK=1
IF(ABS(DDI-DDK).LE.CLEN/MAXO(IABS(ITK(I)),IABS(ITK(K))))ICK=ICK+2
IF(ICK.EQ.7) GOTO 415
DOV=AMIN1 (DOVK,DOVJ,DOVI)
IF(ICK.EQ.4 .AND. ABS(DOVI-DOV) .GT.CLEN) GOTO 175
C CHECK GROUP I, PLATE A, ENDPOINT 1 = GROUP J, PLATE A, ENDPOINT 1
415 DIJ=SQRT((OEI(1,1)-OEJ(1,1))**2+(OEI(2,1)-OEJ(2,1))**2
&+(OEI(3,1)-OEJ(3,1))**2)
IF(DIJ.LE.CLEN) GOTO 440
DO 420 IQ=1,3
420 VDM(IQ)=(OEI(IQ,1)-OEJ(IQ,1))/DIJ
IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).LT.0.) GOTO 425
NN=DIJ/DDJ+0.5
DO 430 IQ=1,3
430 VDM(IQ)=OEJ(IQ,1)+NN*DDJ*CE(IQ)
DIJ=(OEI(1,1)-VDM(1))**2+(OEI(2,1)-VDM(2))**2
&+(OEI(3,1)-VDM(3))**2
IF(DIJ.LE.CLEN**2) GOTO 440
GOTO 175
425 NN=DIJ/DDI+0.5
DO 435 IQ=1,3
435 VDM(IQ)=OEI(IQ,1)+NN*DDI*CE(IQ)
DIJ=(OEI(1,1)-VDM(1))**2+(OEI(2,1)-VDM(2))**2
&+(OEI(3,1)-VDM(3))**2
IF(DIJ.LE.CLEN**2) GOTO 440
GOTO 175
C CHECK GROUP I, PLATE A, ENDPOINT 2 = GROUP J, PLATE A, ENDPOINT 2
440 DIJ=SQRT((OEI(1,2)-OEJ(1,2))**2+(OEI(2,2)-OEJ(2,2))**2
&+(OEI(3,2)-OEJ(3,2))**2)
IF(DIJ.LE.CLEN) GOTO 465
DO 445 IQ=1,3
445 VDM(IQ)=(OEI(IQ,2)-OEJ(IQ,2))/DIJ
IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).GT.0.) GOTO 450
NN=DIJ/DDI+0.5
DO 455 IQ=1,3

215
455  
VDM(IQ) = OEJ(IQ, 2) - NN*DDJ*CE(IQ) 
DIJ = (OEI(I, 2) - VDM(I))**2 + (OEI(2, 2) - VDM(2))**2
+ (OEI(3, 2) - VDM(3))**2

IF (DIJ.LT.CLEN**2) GOTO 465
GOTO 175

460  
NN = DIJ/DDI + 0.5
DO 460 IQ = 1, 3

470  
VDM(IQ) = OEJ(IQ, 2) - NN*DDJ*CE(IQ)
DIJ = (OEI(I, 2) - VDM(I))**2 + (OEI(2, 2) - VDM(2))**2
+ (OEI(3, 2) - VDM(3))**2

IF (DIJ.LT.CLEN**2) GOTO 465
GOTO 175

C CHECK GROUP I, PLATE B, ENDPOINT 1 = GROUP K, PLATE A, ENDPOINT 1

465  
DIJ = SQRT((OEI(I, 1) - OKE(I, 1))**2 + (OEI(2, 1) - OKE(2, 1))**2
+ (OEI(3, 1) - OKE(3, 1))**2)

IF (DIJ.LE.CLEN) GOTO 490
DO 470 IQ = 1, 3

475  
VDM(IQ) = OEJ(IQ, 1) - OKE(IQ, 1) / DIJ

480  
VDM(IQ) = OKE(IQ, 1) - NN*DDK*CE(IQ)
DIJ = (OEI(I, 1) - VDM(I))**2 + (OKE(2, 1) - VDM(2))**2
+ (OKE(3, 1) - VDM(3))**2

IF (DIJ.LT.CLEN**2) GOTO 490
GOTO 175

C CHECK GROUP I, PLATE B, ENDPOINT 2 = GROUP K, PLATE A, ENDPOINT 2

490  
DIJ = SQRT((OEI(I, 1) - OKE(I, 1))**2 + (OEI(2, 1) - OKE(2, 1))**2
+ (OEI(3, 1) - OKE(3, 1))**2)

IF (DIJ.LE.CLEN) GOTO 515
DO 495 IQ = 1, 3

495  
VDM(IQ) = OEJ(IQ, 2) - OKE(IQ, 2) / DIJ

500  
VDM(IQ) = OKE(IQ, 2) - NN*DDK*CE(IQ)
DIJ = (OEI(I, 2) - VDM(I))**2 + (OKE(2, 2) - VDM(2))**2
+ (OKE(3, 2) - VDM(3))**2

IF (DIJ.LT.CLEN**2) GOTO 515
GOTO 175

C CHECK GROUP J, PLATE B, ENDPOINT 1 = GROUP K, PLATE B, ENDPOINT 1

515  
DIJ = SQRT((OEJ(I, 1) - OKE(I, 1))**2 + (OEJ(2, 1) - OKE(2, 1))**2
+ (OEJ(3, 1) - OKE(3, 1))**2)

IF (DIJ.LE.CLEN) GOTO 540

216
DO 520 IQ=1,3
  VDM(IQ) = OGE(IQ,1) - ODK(IQ,1) / DIJ
  IF (VDM(1) * CE(1) + VDM(2) * CE(2) + VDM(3) * CE(3) .LT. 0.) GOTO 525
  NN = DIJ / DDJ + 0.5
  DO 530 IQ=1,3
  VDM(IQ) = ODK(IQ,1) + NN * DDK * CE(IQ)
  DIJ = OGE(IQ,1) - VDM(1)**2 + (OGE(IQ,2) - VDM(2))**2
 &+ (OGE(IQ,3) - VDM(3))**2
  IF (DIJ .LT. CLEN**2) GOTO 525
  NN = DIJ / DDJ + 0.5
  DO 535 IQ=1,3
  VDM(IQ) = OGE(IQ,1) + NN * DDJ * CE(IQ)
  DIJ = ODK(IQ,1) - VDM(1)**2 + (OGE(IQ,2) - VDM(2))**2
 &+ (OGE(IQ,3) - VDM(3))**2
  IF (DIJ .LT. CLEN**2) GOTO 540
C CHECK GROUP J, PLATE B, ENDPOINT 2 = GROUP K, PLATE B, ENDPOINT 2
  DIJ = SQRT((OGE(IK,1,1) - ODK(IK,1,1))**2 + (OGE(IK,2,1) - ODK(IK,2,1))**2
 &+ (OGE(IK,3,1) - ODK(IK,3,1))**2
  IF (DIJ .LT. CLEN**2) GOTO 540
C CONSTRUCT OVERLAP MODES
DO 205 IV=1,NOPL
  IF (ITK(IV) .GT. 0) GOTO 201
  MP = ITK(IV) - 0
  GOTO 205
201  IF (ITK(IV) .GT. 0) GOTO 205
  ITK(IV) = 0
  GOTO 205
C CONSTRUCT OVERLAP MODES
DO 206 10=1,3
  OE(IQ,1) = OVEP(IV, IQ, 1)
  OE(IQ,2) = OVEP(IV, IQ, 2)
  OE(IQ,3) = OVEP(IV, IQ, 3)
217
CE(IQ)=(OE(IQ,2)-OE(IQ,1))/DOV
CALL PGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCRNS,PCN,ICN,IPL,SEG)
6,NDNPLT,PA,IB,PLM,IV,TOSI,PSI
6,OMSP,MOV

DA=SQRT((OMSP(1,3,1)-OMSP(1,4,1))**2+OMSP(2,3,1)-OMSP(2,4,1))

DB=SQRT((OMSP(1,3,2)-OMSP(1,4,2))**2+OMSP(2,3,2)-OMSP(2,4,2))

DSA=DA/NOV
DSB=DB/NOV
DSE=DOV/NOV

DO 285 IQ=1,3

CA(IQ)=(OMSP(IQ,3,1)-OMSP(IQ,4,1))/DA

DO 281 IQ=1,3
CA1(IQ)=OMSP(IQ,3,1)-OMSP(IQ,4,1)
CA2(IQ)=OMSP(IQ,3,1)-OMSP(IQ,4,1)
CB1(IQ)=OMSP(IQ,4,2)-OMSP(IQ,1,2)
CB2(IQ)=OMSP(IQ,3,2)-OMSP(IQ,2,2)

DA1=SQRT(CA1(1)**2+CA1(2)**2+CA1(3)**2)

DB1=SQRT(CB1(1)**2+CB1(2)**2+CB1(3)**2)

TA1=CE(1)*CA1(1)+CE(2)*CA1(2)+CE(3)*CA1(3)

TB1=(CE(1)*CB1(1)+CE(2)*CB1(2)+CE(3)*CB1(3))

TA2=(CE(1)*CA2(1)+CE(2)*CA2(2)+CE(3)*CA2(3))

TB2=(CE(1)*CB2(1)+CE(2)*CB2(2)+CE(3)*CB2(3))

C FILL IN OVERLAP MODES

DO 330 I=1,NOV
DO 335 IQ=1,3
NN=NDTLM+NOVT+I
PA(NN,1,IQ)=OMSP(IQ,1,1)+(I-1)*DSE*CE(IQ)
PB(NN,1,IQ)=PA(NN,1,IQ)
PA(NN,2,IQ)=OMSP(IQ,4,1)+(I-1)*DSA*CA(IQ)
PB(NN,2,IQ)=OMSP(IQ,4,1)+(I-1)*DSA*CA(IQ)
PA(NN,3,IQ)=OMSP(IQ,4,1)+I*DSA*CA(IQ)
PB(NN,3,IQ)=OMSP(IQ,4,1)+I*DSA*CA(IQ)
PA(NN,4,IQ)=OMSP(IQ,4,1)+I*DSA*CA(IQ)

PB(NN,4,IQ)=PA(NN,4,IQ)

IQUAD(NN)=0

IF (CE(1)*CA(1)+CE(2)*CA(2)+CE(3)*CA(3).LT.0.999) GOTO 340
IF (ABS(TA1).LT.0.03 .AND. ABS(TA2).LT.0.03) IQUAD(NN)=1

IQUAD(NN)=2

218
APPENDIX 9
SUBROUTINE FGPOV

SUBROUTINE FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEGM,NDNPLT,PA,PB,IPLM,WV TOUCH,TPSI,SPSI

DIMENSION OE(3,2),NCNRS(1),PCN(1,ICN,IPL),SEGM(1)
DIMENSION CE(3),CA(3),CB(3),CA1(3),CA2(3),CB1(3),CB2(3),CP(3)
ICB=ISB
ICBl=ISB+1
ICA=ISA
ICA=ISA+1
ELSE IF(ISA.EQ.NCNRS(NPB))ICA=ICA+1
ELSE IF(ISB.EQ.NCNRS(NPB))ICBl=ICBl+1
DAS=(PCN(1,ICA,NPA)-PCN(1,ICA,NPA))**2
+PCN(2,ICA,NPA)-PCN(2,ICA,NPA))**2
3+(PCN(3,ICA,NPA)-PCN(3,ICA,NPA))**2
DBS=(PCN(1,ICBl,NPB)-PCN(1,ICBl,NPB))**2
+PCN(2,ICBl,NPB)-PCN(2,ICBl,NPB))**2
3+(PCN(3,ICBl,NPB)-PCN(3,ICBl,NPB))**2
C SET UP MAXIMUM SEGMENT SIZES
SEGA=SEGM(NPA)*W
SEGB=SEGM(NPB)*W
210 NOVE=DOV/AMIN1(SEGA,SEGB)+0.99
C CALCULATE DIRECTIONAL COSINES OF COMMON SIDE
DO 241 IQ=1,3
241 CE(IQ)=(OE(IQ,2)-OE(IQ,1))/DOV
C INDEX CORNERS OF PLATES
IF(CE(1)*(PCN(1,ICA,NPA)-PCN(1,ICA,NPA))
2+CE(2)*(PCN(2,ICA,NPA)-PCN(2,ICA,NPA))
3+CE(3)*(PCN(3,ICA,NPA)-PCN(3,ICA,NPA)).LT.0.)
GOTO 231
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
ICA=ICA+1
GOTO 232
231 ICA2=ISA

219
```plaintext
ICAI = ICA2 + 1
IF (ICAI .EQ. NCNRS(NPA)) ICA1 = 1
ICA0 = ICA1 + 1
IF (ICAO .EQ. NCNRS(NPA)) ICAO = 1
ICA3 = ICA2 - 1
IF (ICA2 .EQ. NCNRS(NPA)) ICA3 = NCNRS(NPA)

232
IF (CE(1) * (PCN(1,ICB1,NPB) - PCN(1,ICB,NPB))
2 + CE(2) * (PCN(2,ICB1,NPB) - PCN(2,ICB,NPB))
3 + CE(3) * (PCN(3,ICB1,NPB) - PCN(3,ICB,NPB)) .LT. 0.)
GOTO 236

ICB1 = ISB
ICB2 = ICB1 + 1
IF (ICB1 .EQ. NCNRS(NPB)) ICB2 = 1
ICB3 = ICB2 + 1
IF (ICB2 .EQ. NCNRS(NPB)) ICB3 = 1
ICB0 = ICB1 + 1
IF (ICB1 .EQ. NCNRS(NPB)) ICB0 = NCNRS(NPB)
GOTO 237

236
ICB2 = ISB
ICB1 = ICB2 + 1
IF (ICB2 .EQ. NCNRS(NPB)) ICB1 = 1
ICB0 = ICB1 + 1
IF (ICB1 .EQ. NCNRS(NPB)) ICB0 = 1
ICB3 = ICB2 - 1
IF (ICB2 .EQ. NCNRS(NPB)) ICB3 = NCNRS(NPB)

C  FIND PRELIMINARY OVERLAP MODE REGIONS
237
DO 400 IQ = 1, 3
Omsp(IQ,1,1) = PCN(IQ,ICAI,NPA)
Omsp(IQ,1,2) = PCN(IQ,ICAO,NPA)
Omsp(IQ,1,3) = PCN(IQ,ICAO,NPA)
CALL FMDC (NDNPLT, PCN, ICN, IPL, PA, PB, IPLM, ICA2, ICA3, 3, NPA
&1, CE, TOUCH, OMSW, WV, NPB)
CALL FMDC (NDNPLT, PCN, ICN, IPL, PA, PB, IPLM, ICA1, ICA0, 4, NPA
&1, CE, TOUCH, OMSW, WV, NPB)
CALL FMDC (NDNPLT, PCN, ICN, IPL, PA, PB, IPLM, ICB2, ICB3, 3, NPB
&2, CE, TOUCH, OMSW, WV, NPB)
CALL FMDC (NDNPLT, PCN, ICN, IPL, PA, PB, IPLM, ICB1, ICB0, 4, NPB
&2, CE, TOUCH, OMSW, WV, NPB)
NCA1 = 1
NCA2 = 1
NCB1 = 1
NCB2 = 1
Dovas = (Omsp(1,1,1) - Omsp(1,2,1)) ** 2 + (Omsp(2,1,1) - Omsp(2,2,1)) ** 2
& + (Omsp(3,1,1) - Omsp(3,2,1)) ** 2
Dovas = (Omsp(1,1,1) - Omsp(1,2,2)) ** 2 + (Omsp(2,1,1) - Omsp(2,2,2)) ** 2
& + (Omsp(3,1,1) - Omsp(3,2,2)) ** 2
IF (Dovas .GT. Dovbs - 2 .EQ. SQRT(Dovas * Dovbs) .LE. TOUCH**2) GOTO 500
IF (Dovas .LT. Dovbs) NCB1 = 0
DO 500 IQ = 1, 3
IF (Dovas .LT. Dovbs) Omsp(IQ,1,2) = Omsp(IQ,2,1)
IF (Dovas .GT. Dovbs) Omsp(IQ,1,2) = Omsp(IQ,2,2)
CONTINUE

500
Dovas = (Omsp(1,1,1) - Omsp(1,1,1)) ** 2 + (Omsp(2,2,1) - Omsp(2,1,1)) ** 2
& + (Omsp(3,2,1) - Omsp(3,2,1)) ** 2
Dovas = (Omsp(1,2,1) - Omsp(1,1,1)) ** 2 + (Omsp(2,2,1) - Omsp(2,1,1)) ** 2
& + (Omsp(3,2,1) - Omsp(3,1,1)) ** 2

220
```
IF (DOVAS + DOVBS - 2.0 * SQRT (DOVAS * DOVBS)) .LE. TOUCH**2) GOTO 510
IF (DOVAS .LT. DOVBS) NCB1 = 0
IF (DOVAS .GT. DOVES) NCA1 = 0
DO 510 IQ = 1, 3
IF (DOVAS .LT. DOVBS) OMSQP(IQ, 1, 2) = OMSQP(IQ, 1, 1)
IF (DOVAS .GT. DOVES) OMSQP(IQ, 1, 1) = OMSQP(IQ, 1, 2)
510 CONTINUE
DOV = SQRT (AMN1 (DOVAS, DOVBS))
DO 410 IQ = 1, 3
CA(IQ) = OMSQP(IQ, 3, 1) - OMSQP(IQ, 4, 1)
CB(IQ) = OMSQP(IQ, 3, 2) - OMSQP(IQ, 4, 2)
DA = SQRT (CA(1) * CA(1) + CA(2) * CA(2) + CA(3) * CA(3))
DB = SQRT (CB(1) * CB(1) + CB(2) * CB(2) + CB(3) * CB(3))
DO 420 IQ = 1, 3
CA(IQ) = CA(IQ) / DA
CB(IQ) = CB(IQ) / DB
C CALCULATE DIRECTIONAL COSINES OF ADJACENT SIDES
DO 242 IQ = 1, 3
CA1(IQ) = PCN(IQ, ICA0, NPA) - PCN(IQ, ICA1, NPA)
CA2(IQ) = PCN(IQ, ICA1, NPA) - PCN(IQ, ICA2, NPA)
CB1(IQ) = PCN(IQ, ICB0, NPB) - PCN(IQ, ICB1, NPB)
CB2(IQ) = PCN(IQ, ICB1, NPB) - PCN(IQ, ICB2, NPB)
DA1 = SQRT (CA1(1)**2 + CA1(2)**2 + CA1(3)**2)
DA2 = SQRT (CA2(1)**2 + CA2(2)**2 + CA2(3)**2)
DB1 = SQRT (CB1(1)**2 + CB1(2)**2 + CB1(3)**2)
DB2 = SQRT (CB2(1)**2 + CB2(2)**2 + CB2(3)**2)
DO 243 IQ = 1, 3
CA1(IQ) = CA1(IQ) / DA1
CA2(IQ) = CA2(IQ) / DA2
CB1(IQ) = CB1(IQ) / DB1
CB2(IQ) = CB2(IQ) / DB2
C FIND PLATE CORNER ANGLES
TA1 = CE(1) * CA1(1) + CE(2) * CA1(2) + CE(3) * CA1(3)
TA2 = CE(1) * CA2(1) + CE(2) * CA2(2) + CE(3) * CA2(3)
TB1 = CE(1) * CB1(1) + CE(2) * CB1(2) + CE(3) * CB1(3)
TB2 = CE(1) * CB2(1) + CE(2) * CB2(2) + CE(3) * CB2(3)
SA1 = SQRT (1.0 - TA1**2)
SA2 = SQRT (1.0 - TA2**2)
SB1 = SQRT (1.0 - TB1**2)
SB2 = SQRT (1.0 - TB2**2)
C DETERMINE FINAL OVERLAP MODE REGIONS
C CALCULATE DIRECTIONAL COSINES OF NORMAL VECTOR, PLATE NPA
CP(1) = CE(2) * CA1(3) - CA1(2) * CE(3) / SA1
CP(2) = CE(1) * CA1(3) - CA1(3) * CE(1) / SA1
CP(3) = CE(1) * CA1(2) - CA1(1) * CE(2) / SA1
CN(1) = CP(2) * CE(3) - CE(2) * CP(3)
CN(2) = CP(3) * CE(1) - CE(3) * CP(1)
CN(3) = CP(1) * CE(2) - CE(1) * CP(2)
C FIND CORNER 1, PLATE NPA
DO 425 IQ = 1, 3
CMD(IQ) = CP(IQ) * TPSI - CE(IQ) * TPSI
425 CMT(IQ) = OMSQP(IQ, 4, 1) - OMSQP(IQ, 1, 1)
DMT = SQRT (CMT(1)**2 + CMT(2)**2 + CMT(3)**2)
TAL = (CMT(1) * CA(1) + CMT(2) * CA(2) + CMT(3) * CA(3)) / DMT
SAL = SQRT (1.0 - TAL**2)
IF (NCA1 .EQ. 1) GOTO 245
TB1 = CMT(1) * CMD(1) + CMT(2) * CMD(2) + CMT(3) * CMD(3) / DMT
SBE = SQRT (1.0 - TBE**2)
221
TBEP = (CMT(1) * CE(1) + CMT(2) * CE(2) + CMT(3) * CE(3)) / DMT
IF (TBEP .GE. -TPSI) GOTO 245
DML = DMT * SAL / (SAL * TBE - TAL * SBE)
IF (DML .GT. SEG) GOTO 426
DO 427 IQ = 1, 3
427 OA(IQ, 1) = OMS(IQ, 1, 1) + DML * CMD(IQ)
DD1 = (OA(1, 1) - OMS(1, 4, 1)) ** 2 + (OA(2, 1) - OMS(2, 4, 1)) ** 2
2 + (OA(3, 1) - OMS(3, 4, 1)) ** 2
DD2 = (OA(1, 1) - OMS(1, 3, 1)) ** 2 + (OA(2, 1) - OMS(2, 3, 1)) ** 2
2 + (OA(3, 1) - OMS(3, 3, 1)) ** 2
IF (DD1 + DD2 - DADA .GT. 2 * TOUCH * 2) GOTO 245
GOTO 250
DML = DTM * TAL
DML2 = SEG ** 2 - (DMT * SAL) ** 2
IF (DML2 .GT. 0.) DML = DML - SQRT(DML2)
DO 428 IQ = 1, 3
428 OA(IQ, 1) = OMS(IQ, 4, 1) + DML * CA(IQ)
GOTO 250
245 IF (DMT .GT. SEG) GOTO 426
DO 430 IQ = 1, 3
430 OA(IQ, 1) = OMS(IQ, 4, 1)
C FIND CORNER 2, PLATE NPA
250 DO 435 IQ = 1, 3
435 CMD(IQ) = CN(IQ) * SPSI + CE(IQ) * TPSI
CMT(IQ) = OMS(IQ, 3, 1) - OMS(IQ, 2, 1)
DMT = SQRT (CMT(1) ** 2 + CMT(2) ** 2 + CMT(3) ** 2)
TAL = (CMT(1) * CA(1) + CMT(2) * CA(2) + CMT(3) * CA(3)) / DMT
SAL = SQRT (1 - TAL * TAL)
IF (NCA2 .EQ. 1) GOTO 255
TBE = (CMT(1) * CMD(1) + CMT(2) * CMD(2) + CMT(3) * CMD(3)) / DMT
SBE = SQRT (1 - TBE * TBE)
TBEP = (CMT(1) * CE(1) + CMT(2) * CE(2) + CMT(3) * CE(3)) / DMT
IF (TBEP .LT. TPSI) GOTO 255
DML = DML - SAL / (SAL * TBE + TAL * SBE)
IF (DMT .GT. SEG) GOTO 436
DO 437 IQ = 1, 3
437 OA(IQ, 2) = OMS(IQ, 2, 1) + DML * CMD(IQ)
DD1 = (OA(1, 2) - OMS(1, 4, 1)) ** 2 + (OA(2, 2) - OMS(2, 4, 1)) ** 2
2 + (OA(3, 2) - OMS(3, 4, 1)) ** 2
DD2 = (OA(1, 2) - OMS(1, 3, 2)) ** 2 + (OA(2, 2) - OMS(2, 3, 1)) ** 2
2 + (OA(3, 2) - OMS(3, 3, 1)) ** 2
IF (DD1 + DD2 - DADA .GT. 2 * TOUCH * 2) GOTO 255
GOTO 260
DML = DMT * TAL
DML2 = SEG ** 2 - (DMT * SAL) ** 2
IF (DML2 .GT. 0.) DML = DML - SQRT(DML2)
DO 438 IQ = 1, 3
438 OA(IQ, 2) = OMS(IQ, 3, 1) - DML * CA(IQ)
GOTO 260
255 IF (DMT .GT. SEG) GOTO 436
DO 440 IQ = 1, 3
440 OA(IQ, 2) = OMS(IQ, 3, 1)
C CALCULATE DIRECTIONAL COSINES "NORMAL VECTOR, PLATE NPB
260 CP(I) = CE(2) * CB1(3) - CB1(2) * CE(3) / SB1
CP(2) = CE(3) * CB1(1) - CB1(3) * CE(1) / SB1
CP(3) = CE(1) * CB1(2) - CB1(1) * CE(2) / SB1
CN(I) = CP(I) * CE(3) - CE(I) * CP(3)
CN(2) = CP(2) * CE(3) - CE(2) * CP(3)
CN(3) = CP(3) * CE(1) - CE(3) * CP(1)

222
FIND CORNER 1, PLATE NB

DO 445 IQ=1,3
CMD(IQ)=CN(IQ)*SPSI-CE(IQ)*TPSI
CMT(IQ)=OMSP(IQ,4,2)-OMSP(IQ,1,2)
DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
TAL=(CMT(1)*CB(1)+CMT(2)*CB(2)+CMT(3)*CB(3))/DMT
SAL=SQRT(1.-TAL*TAL)
IF(NCB1.EQ.1)GOTO 265

TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
SBE=SQRT(1.-TBE*TBE)
TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
IF(TBEP.GE.TPSI)GOTO 265

DML=DMT*SAL/(SAL*TBE-TAL*SBE)
IF(DML.GT.SEGB)GOTO 446

DO 447 IQ=1,3

OB(IQ,1)=OMSP(IQ,1,2)+DML*CMD(IQ)
DD1=OB(IQ,1)-OMSP(1,4,2)**2+OB(2,1)-OMSP(2,4,2)**2
2+OB(3,1)-OMSP(3,4,2)**2

DD2=OB(IQ,1)-OMSP(1,3,2)**2+OB(2,1)-OMSP(2,3,2)**2
2+OB(3,1)-OMSP(3,3,2)**2
IF(DD1+DD2-DB*DB.GT.2.*TOUCH**2)GOTO 265

GOTO 270

446 DML=DMT*TAL
DML2=SEGB**2-(DMT*SAL)**2
IF(DML2.GT.0.)DML=DML-SQRT(DML2)
DO 448 IQ=1,3

OB(IQ,1)=OMSP(IQ,4,2)+DML*CB(IQ)
GOTO 270

265 IF(DML.GT.SEGB)GOTO 446

DO 450 IQ=1,3

450 OB(IQ,1)=OMSP(IQ,4,2)

FIND CORNER 2, PLATE NB

DO 455 IQ=1,3
CMD(IQ)=CN(IQ)*SPSI+CE(IQ)*TPSI
CMT(IQ)=OMSP(IQ,3,2)-OMSP(IQ,2,2)
DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
TAL=(CMT(1)*CB(1)+CMT(2)*CB(2)+CMT(3)*CB(3))/DMT
SAL=SQRT(1.-TAL*TAL)
IF(NCB2.EQ.1)GOTO 275

TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
SBE=SQRT(1.-TBE*TBE)
TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
IF(TBEP.LE.TPSI)GOTO 275

DML=DMT*SAL/(SAL*TBE-TAL*SBE)
IF(DML.GT.SEGB)GOTO 456

DO 457 IQ=1,3

OB(IQ,2)=OMSP(IQ,2,2)+DML*CMD(IQ)
DD=(OB(1,2)=OMSP(1,4,2)**2+(OB(2,2)-OMSP(2,4,2)**2
2+(OB(3,2)-OMSP(3,4,2)**2

2+OB(1,2)-OMSP(1,3,2)**2+(OB(2,2)-OMSP(2,3,2)**2
2+(OB(3,2)-OMSP(3,3,2)**2
IF(DD1+DD2-DB*DB.GT.2.*TOUCH**2)GOTO 275

GOTO 280

456 DML=DMT*TAL
DML2=SEGB**2-(DMT*SAL)**2
IF(DML2.GT.0.)DML=DML-SQRT(DML2)
DO 458 IQ=1,3

223
OB(IQ,2) = OMS(IQ,3,2) - DML*CB(IQ)
GOTO 280

275 IF(DMT.GT.SEGB) GOTO 456
DO 460 IQ = 1, 3
460 OB(IQ,2) = OMS(IQ,3,2)
C DETERMINE NUMBER OF OVERLAP MODES
280 DA = SQRT((OA(1,2) - OA(1,1))**2 + (OA(2,2) - OA(2,1))**2)
NOVA = DA / SEGA + .99
DB = SQRT((OB(1,2) - OB(1,1))**2 + (OB(2,2) - OB(2,1))**2)
NOVB = DB / SEGB + .99
NOV = JMAX(NOVA, NOVA, NOVB)
DO 515 IQ = 1, 3
OMSP(IQ,3,1) = OA(IQ,2)
OMSP(IQ,4,1) = OA(IQ,1)
OMSP(IQ,3,2) = OB(IQ,2)
OMSP(IQ,4,2) = OB(IQ,1)
515 RETURN
END
SUBROUTINE FMDC

SUBROUTINE FMDC(NDNPLT, PCN, ICN, IPL, PA, PB, IPLM, NC, NAC, MC, NP & IAB, CE, TOUCH, ONSP, W, NPO)
C SUBROUTINE FMDC FINDS THE OVERLAP REGION POINT ONSP(I, MC, IAB)
DIMENSION NDNPLT(1), PCNC3, ICN, IPL), PA(IPLM, 4, 3), PB(IPLM, 4, 3)
C, CE(3), ONSP(3, 4, 2), VM(3)
TCHS=TOUCH**2
DMN=WV*W
MAB=0
M=1
IF(NP.NE.1)M=M+1
M=M+1
IF(M.EQ.4)GOTO 100
DO 110 M=M, 4
C CHECK MONOPOLE A OF MODE M
DO 120 IQ=1, 3
VM(IQ)=PA(M, 3, IQ)-PA(M, 2, IQ)
DES=VM(1)*VM(1)+VM(2)*VM(2)+VM(3)*VM(3)-(VM(1)*CE(1)
2+VM(2)*CE(2)+VM(3)*CE(3))**2
IF(DES.GT.TCHS)GOTO 130
C CHECK MONOPOLE A, CORNER 2
DCS=(PCN(1, NC, NP)-PA(M, 2, 1))**2+(PCN(2, NC, NP)-PA(M, 2, 2))**2
2+(PCN(3, NC, NP)-PA(M, 2, 3))**2
IF(DCS.LE.TCHS)GOTO 145
IF(DMN.LT.DCS)GOTO 130
DMN=DCS
MAB=M
M=1
MCO=2
GOTO 140
145 DO 150 IQ=1, 3
OMSP(IQ, MC, IAB)=PA(M, 1, IQ)
RETURN
C CHECK MONOPOLE A, CORNER 3
DCS=(PCN(1, NC, NP)-PA(M, 3, 1))**2+(PCN(2, NC, NP)-PA(M, 3, 2))**2
2+(PCN(3, NC, NP)-PA(M, 3, 3))**2
IF(DCS.LE.TCHS)GOTO 155
IF(DMN.LT.DCS)GOTO 130
DMN=DCS
MAB=M
MAB=1
MCO=3
GOTO 130
155 DO 160 IQ=1,3
160 OMSP(IQ,MC,1AB)=PA(M,4,IQ)
RETURN
C CHECK MONOPOLE B OF MODE M
130 DO 170 IQ=1,3
170 VM(IQ)=PB(M,3,IQ)-PB(M,2,IQ)
DESM=VM(1)*VM(1)+VM(2)*VM(2)+VM(3)*VM(3)-(VM(1)*CE(1)
2+VM(2)*CE(2)+VM(3)*CE(3))**2
IF(DES.GT.TCHS)GOTO 110
C CHECK MONOPOLE B, CORNER 2
180 DCS=(PCN(1,NC,NP)-PB(M,2,1))**2+(PCN(2,NC,NP)-PB(M,2,2))**2
2+(PCN(3,NC,NP)-PB(M,2,3))**2
IF(DCS.LE.TCHS)GOTO 195
IF(DMN.LT.DCS)GOTO 185
DMN=DCS
MDN=M
MAB=2
MCO=2
GOTO 180
185 DO 190 IQ=1,3
190 OMSP(IQ,MC,1AB)=PB(M,1,IQ)
RETURN
C CHECK MONOPOLE B, CORNER 3
180 DCS=(PCN(1,NC,NP)-PB(M,3,1))**2+(PCN(2,NC,NP)-PB(M,3,2))**2
2+(PCN(3,NC,NP)-PB(M,3,3))**2
IF(DCS.LE.TCHS)GOTO 110
IF(DMN.LT.DCS)GOTO 195
DMN=DCS
MDN=M
MAB=2
MCO=3
GOTO 110
195 DO 200 IQ=1,3
200 OMSP(IQ,MC,1AB)=PB(M,4,IQ)
RETURN
110 CONTINUE
IF(MAB.EQ.0)GOTO 100
C CORNER MCO OF MONOPOLE MAB OF MODE MDN IS CLOSEST TO CORNER NC
C OF PLATE NP
1 IF(DMN/WV/VT.GT.0.125)WRITE(6,1)NP,NPO
1 FORMAT('** POSSIBLE PROBLEM WITH OVERLAP',
2 MCO2=3
C IF(MCO.EQ.3)MCO2=4
M2=1
C IF(MC.EQ.3)MCO2=2
DO 310 IQ=1,3
C IF(MAB.EQ.2)GOTO 300
OMSP(IQ,MC2,1AB)=PA(MDN,MCO,IQ)
OMSP(IQ,MC,1AB)=PA(MDN,MCO2,IQ)
GOTO 310
300 OMSP(IQ,MC2,1AB)=PB(MDN,MCO,IQ)
OMSP(IQ,MC,1AB)=PB(MDN,MCO2,IQ)
310 CONTINUE
RETURN
100 DO 210 IQ=1,3
210 OMSP(IQ,MC,1AB)=PCN(IQ,NAC,NP)
RETURN
END
APPENDIX 11

SUBROUTINE MPLOT

SUBROUTINE MPLOT(NCNRS, PCN, NPL, ICN, IPL, IPLM, NPL11,
1 NPL22, NDNPLT, PA, PB, IPN)
DIMENSION PCN(3, ICN, IPL), X(20), Y(20), Z(20), PHI(20), IPN(1),
1 SX(20), RZ(20), SXZ(20), APH(20), APHI(20), PHIY(20)
2 ,NCNRS(1), NPL11(1), NPL22(1), NDNPLT(1), PA(IPLM, 4, 3),
3 PB(IPLM, 4, 3), XS(10), YS(10), ZS(10), TNPL(20)
SIZE=2.5
NP=NPL
NXPL=NPL
NC=NCNRS(NPL)
PI=3.141592
IF(NPL.EQ.1) TOT=NDNPLT(NPL)
IF(NPL.NE.1) TOT=NDNPLT(NPL)-NDNPLT(NPL-1)
DO 1 I=1, NC
X(I)=PCN(1, I, NPL) -PCN(1, 1, NPL)
Y(I)=PCN(2, I, NPL) -PCN(2, 1, NPL)
Z(I)=PCN(3, I, NPL) -PCN(3, 1, NPL)
CONTINUE
1 CX=-Y(2)*Z(3)+Z(2)*Y(3)
CY=X(2)*Z(3)-X(3)*Z(2)
CZ=X(2)*Y(3)+X(3)*Y(2)
CMAG=(CX**2+CY**2+CZ**2)**0.5
THC=ACOS(CZ/CMAG)
IF(CX.EQ.0.0.AND.CY.EQ.0.0) GO TO 800
IF(CX.EQ.0.0.AND.CY.NE.0.0) GO TO 70
IF(CX.NE.0.0) GO TO 12
11 MY(I)/((Y(I)**2)**0.5)
PHI(I)=(PI/2)*M-PHC
GO TO 10
12 PHI(I)=ATAN2(Y(I), X(I))
PHI(I)=PHI(I)-PHC
GO TO 10
13 CONTINUE
10 CONTINUE
DO 20 I=2,NC
IF(X(I).EQ.0.0.AND.Y(I).EQ.0.0)GO TO 14
GO TO 15
14 X(I)=0.0
Y(I)=0.0
15 X(I)=SXY(I)*COS(PHI(I))
Y(I)=SXY(I)*SIN(PHI(I))
SXZ(I)=(X(I)**2+Z(I)**2)**0.5
IF(Z(I).EQ.0.0.AND.ABS(X(I)).LT.0.0005)GO TO 30
GO TO 31
30 X(I)=0.0
Z(I)=0.0
GO TO 20
31 PHIY(I)=ATAN2(X(I),Z(I))
PHIY(I)=PHIY(I)-TIC
X(I)=SXZ(I)*SIN(PHIY(I))
Y(I)=Y(I)
Z(I)=SXZ(I)*COS(PHIY(I))
20 CONTINUE
800 X1=0.0
Y1=0.0
Z1=0.0
DO 2 I=1,NC
X1=X1+X(I)
Y1=Y1+Y(I)
Z1=Z1+Z(I)
2 CONTINUE
X1=X1/NC
Y1=Y1/NC
Z1=Z1/NC
C CALL VPLOTS(0,0,0)
SM=1.0
DO 110 I=1,NC
DO 120 J=1,NC
IF(I.EQ.J)GO TO 70
120 SL=SQR(T((X(I)-X(J))**2+(Y(I)-Y(J))**2)
IF(SL.GT.SM)SM=SL
110 CONTINUE
FAC=SIZE/SM
DO 130 I=1,NC
X(I)=(X(I)-X1)*FAC
Y(I)=(Y(I)-Y1)*FAC
Z(I)=(Z(I)-Z1)*FAC
130 CONTINUE
CMAX=999.0
DO 16 I=1,NC
IF(X(I).GT.CMAX)CMAX=X(I)
16 CONTINUE
CALL PLOT(3.0,0.75,-3)
CALL SYMBOL(-1.5,-0.5,0.2,20HTOTAL MODES ON PLATE,0.0,20)
CALL NUMBER(-2.30,-0.5,0.2,TOT,0.0,-1)
CALL NUMBER(2.9,-0.5,0.2,XNPL,0.0,-1)
NPS=NC+1
X(NPS)=X(I)
Y(NPS)=Y(I)
DO 1100 K=1,2
YY=3.0
XX1 = 2.0
XX2 = 1.5
IF (K.EQ.1) YY = 1.5
YY1 = 0.3
CALL PLOT (0.0, YY, -3)
CALL SYMBOL (XX1, YY1, 0.2, 5, MOD10, 0.0, 5)
DO 100 N = 1, NPL
IU = 3
IF (N.GT.1) IU = 2
1999 FORMAT (1X, 'X(N)', F6.2, 2X, 'Y(N)', F6.2)
100 CALL PLOT (X(N), Y(N), IU)
IF (NPL.EQ.1) GO TO 444
GO TO 445
444 IF (K.EQ.1) I0 = 1
IF (K.EQ.1) I2 = NPL11 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL22 (NPL) - NPL11 (NPL)
IF (K.EQ.1 OR K.EQ.2) CALL NUMBER (XX2, YY1, 0.2, AM, 0.0, -1)
IF (K.EQ.2) AM = NPL11 (NPL) + NPL11 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL22 (NPL) - NPL11 (NPL)
IF (K.EQ.3) AM = NPL11 (NPL) - NPL22 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL11 (NPL) - NPL22 (NPL)
IF (K.EQ.3) AM = NPL11 (NPL) - NPL22 (NPL)
445 IF (K.EQ.1) I0 = NPL11 (NPL) - 1
IF (K.EQ.1) I2 = NPL11 (NPL) - 1 + NPL11 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL11 (NPL) - NPL11 (NPL)
IF (K.EQ.3) AM = NPL11 (NPL) - NPL11 (NPL)
IF (K.EQ.1) AM = NPL22 (NPL) - NPL11 (NPL)
IF (K.EQ.2) AM = NPL22 (NPL) - NPL11 (NPL)
IF (K.EQ.3) AM = NPL22 (NPL) - NPL11 (NPL)
IF (I = NPL (NPL), EQ. 0) AM = 0
IF (K.EQ.1 OR K.EQ.2) CALL NUMBER (XX2, YY1, 0.2, AM, 0.0, -1)
IF (K.EQ.2) AM = NPL11 (NPL) + NPL11 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL22 (NPL) - NPL11 (NPL)
IF (K.EQ.3) AM = NPL22 (NPL)
IF (K.EQ.1) AM = NPL11 (NPL)
IF (K.EQ.2) AM = NPL11 (NPL)
IF (K.EQ.3) AM = NPL11 (NPL)
446 IF (I = NPL (NPL), EQ. 0) GO TO 1100
DO 1000 I = 10, 12
IF (I = NPL (NPL), EQ. 0) GO TO 1000
DO 1002 IAB = 1, 2
1001 J = 1, 4
IF (IAB.EQ.2) GO TO 501
XS (J) = PA (I, J, 1) - PCN (1, I, NPL)
YS (J) = PA (I, J, 2) - PCN (2, I, NPL)
ZS (J) = PA (I, J, 3) - PCN (3, I, NPL)
GO TO 1001
501 XS (J) = PB (I, J, 1) - PCN (1, I, NPL)
YS (J) = PB (I, J, 2) - PCN (2, I, NPL)
ZS (J) = PB (I, J, 3) - PCN (3, I, NPL)
1001 CONTINUE
IF (CX.EQ.0.0.AND.CY.EQ.0.0) Go to 900

DO 1003 M=1,4
SXY(M)=SQRT(XS(M)**2+YS(M)**2)
IF (XS(M).EQ.0.0.AND.YS(M).EQ.0.0) Go to 35
IF (XS(M).EQ.0.0.AND.YS(M).NE.0.0) Go to 36
PHI(M)=ATAN2(XS(M),YS(M))-PHC
Go to 400

36 MI=YS(M)/((YS(M)**2)**0.5)
PHI(M)=(PI/2)*MI-PHC
Go to 400

35 XS(M)=XS(M)
YS(M)=YS(M)
ZS(M)=ZS(M)
Go to 1003

400 XS(M)=SXY(M)*COS(PHI(M))
YS(M)=SXY(M)*SIN(PHI(M))
ZS(M)=ZS(M)

1003 Continue

DO 1004 JL=1,4
SXZ(JL)=SQRT(XS(JL)**2+ZS(JL)**2)
IF (ZS(JL).EQ.0.0.AND.ABS(XS(JL)).LT.0.0005) Go to 37
PHIY(JL)=ATAN2(XS(JL),ZS(JL))--THC
Go to 401

37 XS(JL)=0.0
ZS(JL)=0.0
YS(JL)=YS(JL)
Go to 1004

38 XS(JL)=XS(JL)
ZS(JL)=0.0
YS(JL)=YS(JL)
Go to 1004

401 XS(JL)=SXZ(JL)*SIN(PHIY(JL))
ZS(JL)=SXZ(JL)*COS(PHIY(JL))
YS(JL)=YS(JL)

1004 Continue

900 DO 1111 JK=1,4
XS(JK)=(XS(JK)-X1)*FAC
YS(JK)=(YS(JK)-Y1)*FAC

1111 ZS(JK)=(ZS(JK)-Z1)*FAC
KPS=5
XS(KPS)=XS(1)
YS(KPS)=YS(1)
ZS(KPS)=ZS(1)

334 FORMAT(1X,'THE FINAL COOR. ARE:',/) DO 1005 II=1,5
IU=3
IF (II.GT.1) IU=2

1005 CALL PLOT(XS(II),YS(II),IU)
XAB=0.0
YAB=0.0
DO 1006 L=1,4
XAB=XAB+XS(L)
YAB=YAB+YS(L)

1006 Continue

XAB=XAB/4
YAB=YAB/4
DX=(XS(1)+XS(4))/2
DY=(YS(1)+YS(4))/2

230
DXCD=DX-XAB
DYCD=DY-YAB
CPX=XAB+0.1*DXCD
CPY=YAB+0.1*DYCD
CALL PLOT(CPX,CPY,3)
CALL PLCT(DX,DY,2)
IF(IAB.EQ.2)GO TO 700
GO TO 1002
700
DIFX=DX-CPX
DIFY=DY-CPY
DIF=SQR(DIFX**2+DIFY**2)
IF(ABS(DIFX).LT.0.005)THETA=(PI/2)*DIFY/ABS(DIFY)
IF(ABS(DIFX).GE.0.005)THETA=ATAN2(DIFY,DIFX)
CPE=0.30*DIF
222
FORMAT(1X,'THETA IS = ',F12.6)
EX1=CPX+CPE*COS(THETA-PI/6)
EY1=CPY+CPE*SIN(THETA-PI/6)
EX2=CPX+CPE*COS(THETA+PI/6)
EY2=CPY+CPE*SIN(THETA+PI/6)
CALL PLOT(CPX,CPY,3)
CALL PLOT(EX1,EY1,2)
CALL PLOT(CPX,CPY,3)
CALL PLOT(EX2,EY2,2)
1002
CONTINUE
1000
CONTINUE
1100
CONTINUE
CALL PLOT(0.0,0.0,-999)
RETURN
END
APPENDIX 12

SUBROUTINE MOPLOT

SUBROUTINE MOPLOT(PCN, NCNRS, IPL, ICN, PA, PB, IPLM)
DIMENSION PCN(3,ICN,IPL), NCNRS(IPL), PA(IPLM,4,3)
2, PB(IPLM,4,3), IOVT(IPLM,4), ITK(IPLM), OE(3,2)
IF (NOPL.EQ.0) RETURN
ICT=0
SIZ=2.5
DO 10 IV=1,NOPL
IF (ITK(IV).EQ.0) GOTO 10
CALL PLOT(4.0,5.25,-3)
NPA=IOVT(IV,1)
ISA=IOVT(IV,2)
NPB=IOVT(IV,3)
ISB=IOVT(IV,4)
ICA1=ISA
ICA2=ICA1+1
IF (ICA1.EQ. NCNRS(NPA)) ICA2=1
ICA0=ICA1-1
IF (ICA1.EQ.1) ICA0=NCNRS(NPA)
ICBl=ISB
ICB2=ICBl+1
IF (ICBl.EQ. NCNRS(NPB)) ICB2=1
ICBl=ICBl-1
IF (ICBl.EQ.1) ICBl=NCNRS(NPB)
DAS=(PCN(1,ICA2,NPA)-PCN(1,ICA1,NPA))**2
2+(PCN(2,ICA2,NPA)-PCN(2,ICA1,NPA))**2
3+(PCN(3,ICA2,NPA)-PCN(3,ICA1,NPA))**2
DBS= (PCN(1,ICB2,NPB)-PCN(1,ICB1,NPB))**2
2+(PCN(2,ICB2,NPB)-PCN(2,ICB1,NPB))**2
3+(PCN(3,ICB2,NPB)-PCN(3,ICB1,NPB))**2
IOV=0
DO 40 IC=1,2
ICB=ICBl
IF (IC.EQ.2) ICB=ICB2
DCJL=(PCN(1,ICA1,NPA)-PCN(1,ICB,NPB))**2
2+(PCN(2,ICA1,NPA)-PCN(2,ICB,NPB))**2
3+(PCN(3,ICA1,NPA)-PCN(3,ICB,NPB))**2
DCJH= (PCN(1,ICA2,NPA)-PCN(1,ICB,NPB))**2
2+(PCN(2,ICA2,NPA)-PCN(2,ICB,NPB))**2
3+(PCN(3,ICA2,NPA)-PCN(3,ICB,NPB))**2

232
IF (DCJL+DCJH-DAS.GT.1.E-5) GOTO 40
IOV=IOV+1
DO 50 IQ=1,3
40
CONTINUE

IF (IOV.EQ.2) DOV=SQRT((OE(1,2)-OE(1,1)**2
+OE(2,2)-OE(2,1)**2+OE(3,2)-OE(3,1)**2)
+IOV.EQ.2) GOTO 60
DO 70 IC=1,2
ICA=ICA1
IF (IC.EQ.2) ICA=ICA2

DCJL=(PCN(1,ICB1,NPB)-PCN(1,ICA,NPA))**2
+PCN(2,ICB1,NPB)-PCN(2,ICA,NPA))**2
+PCN(3,ICB1,NPB)-PCN(3,ICA,NPA))**2

DCJH=(PCN(1,ICB2,NPB)-PCN(1,ICA,NPA))**2
+PCN(2,ICB2,NPB)-PCN(2,ICA,NPA))**2
+PCN(3,ICB2,NPB)-PCN(3,ICA,NPA))**2

IF (DCJL+DCJH-DAS.GT.1.E-5) GOTO 70
IOV=IOV+1
DO 80 IQ=1,3
80
CONTINUE

IF (IOV.EQ.2) DOV=SQRT((OE(1,2)-OE(1,1)**2
+OE(2,2)-OE(2,1)**2+OE(3,2)-OE(3,1)**2)
+IOV.GT.1.E-5) GOTO 60
IOV=1

CONTINUE

DO 60 IC=1,NCNRS(NPA)
SL=(PCN(1,IC,NPA)-TPX)**2+(PCN(2,IC,NPA)-TPY)**2
+PCN(3,IC,NPA)-TPZ)**2
IF (SL.GT. SG) SG=SL
60
CONTINUE

FAK=SQRT(SG)
CXX=(OE(1,2)-OE(1,1))/DOV
CYX=(OE(2,2)-OE(2,1))/DOV
CZX=(OE(3,2)-OE(3,1))/DOV

DA=SQRT(DAS)
AXX=(PCN(1,ICA2,NPA)-PCN(1,ICA1,NPA))/DA
AYX=(PCN(2,ICA2,NPA)-PCN(2,ICA1,NPA))/DA
AZX=(PCN(3,ICA2,NPA)-PCN(3,ICA1,NPA))/DA

IF (AXX*CXX+AYX*CYX+AZX*CZX.GT.0) GOTO 90
ICA1=ICA2
ICA2=ISA
IF (ICA1.EQ.NCNRS(NPA)) ICA0=1
90
DB=SQRT(DBS)
BXX=(PCN(1,ICB2,NPB)-PCN(1,ICB1,NPB))/DB
BYX=(PCN(2,ICB2,NPB)-PCN(2,ICB1,NPB))/DB
BZX=(PCN(3,ICB2,NPB)-PCN(3,ICB1,NPB))/DB

233
IF (BXX*CXX+BYX*CYX+BZX*CZX.GT.0.) GOTO 100
ICB1=ICB2
ICB2=ISB
ICB0=ICB1+1
IF (ICB1 .EQ. NCNRS (NPB)) ICB0=1
100
PAX=PCN (1, ICAL, NPA)=PCN (1, ICA0, NPA)
PAY=PCN (2, ICAL, NPA)=PCN (2, ICA0, NPA)
PZ=PCN (3, ICAL, NPA)=PCN (3, ICA0, NPA)
PA=SQRS (PA*X+PAY*PAY+PAZ*PAZ)
PAX=PAX/PAS
PAY=PAY/PAS
PAZ=PAZ/PAS
AXZ=PAX*CZX-PAZ*CYX
AYZ=PAY*CXX-PAX*CZX
AZZ=PAY*CYX-PAX*CXX
A=SQRS (AXZ*AXZ+AYZ*AYZ+AZZ*AZZ)
AXZ=AXZ/A
AYZ=AYZ/A
AZZ=AZZ/A
axy=AYZ*CXX-AZZ*CYX
AYY=AYZ*CXX-AXX*CXz
AYX=AXX*CYX-AZZ*CXZ
PBX=PCN (1, ICB0, NPB)-PCN (1, ICB1, NPB)
PBY=PCN (2, ICB0, NPB)-PCN (2, ICB1, NPB)
PZ=PCN (3, ICB0, NPB)-PCN (3, ICB1, NPB)
PBS=SQRS (PBX*PBX+PBY*PBY+PBZ*PBZ)
PAX=PBX/PBS
PBY=PBY/PBS
PBZ=PBZ/PBS
BXZ=PBY*CZX-PBZ*CYX
BYZ=PBY*CXX-PBX*CZX
BZ=PBX*CYX-PBY*CXX
B=SQRS (BXZ*BXZ+BYZ*BYZ+BZ*BZ)
BXZ=BXZ/B
BYZ=BYZ/B
BZ=BZ/B
BXZ=BYZ*CZX-BZ*CXY
BYZ=BZ*CXX-BXZ*CZX
BZ=BXZ*BYZ-CXX
IU=3
DO 110 IC=1,NCNRS (NPA)+1
IF (IC. NE. 1) IU=2
JC=1
IF (IC. EQ. NCNRS (NPA)+1) JC=1
X=(CXX*PCN (1, JC, NPA)-TPX)+CYX*PCN (2, JC, NPA)-TPY
2+CXX*PCN (3, JC, NPA)-TPY) *FAC
Y=AXY*PCN (1, JC, NPA)-TPX)+AYY*PCN (2, JC, NPA)-TPY
2+AYY*PCN (3, JC, NPA)-TPY) *FAC
CALL PLOT (X, Y, IU)
IU=3
DO 120 IC=1,NCNRS (NPB)+1
IF (IC. NE. 1) IU=2
JC=1
IF (IC. EQ. NCNRS (NPB)+1) JC=1
X= (CXX*PCN (1, JC, NPB)-TPX)+CYX*PCN (2, JC, NPB)-TPY
2+CXX*PCN (3, JC, NPB)-TPY) *FAC
Y= (BXY*PCN (1, JC, NPB)-TPX)+BYY*PCN (2, JC, NPB)-TPY
2+BYY*PCN (3, JC, NPB)-TPY) *FAC
234
CALL PLOTCX,Y,IU
DO 130 IM=1,ITK(IV)
II=NPMT-NOVT+ICT+IM
IU=3
DO 140 IC=1,5
JC=IC
IF (IC.EQ.5) JC=1
IF (IC.NE.1) IU=2
X=(CXX*(PA(II,JC,1)-TPX)+CYX*(PA(II,JC,2)-TPY)
2+CXX*(PA(II,JC,3)-TPX)+CYX*(PA(II,JC,2)-TPY)
2+AZY*(PA(II,JC,3)-TPX)+AZY*(PA(II,JC,2)-TPY)
140 CALL PLOTCX,Y,IU
IU=3
DO 150 IC=1,5
JC=IC
IF (IC.EQ.5) JC=1
IF (IC.NE.1) IU=2
X=(CXX*(PB(II,JC,1)-TPX)+CYX*(PB(II,JC,2)-TPY)
2+CXX*(PB(II,JC,3)-TPX)+CYX*(PB(II,JC,2)-TPY)
2+AZY*(PB(II,JC,3)-TPX)+AZY*(PB(II,JC,2)-TPY)
150 CALL PLOTCX,Y,IU
XMA=(PA(II,1,1)+PA(II,2,1)+PA(II,3,1)+PA(II,4,1))/4.
YMA=(PA(II,1,2)+PA(II,2,2)+PA(II,3,2)+PA(II,4,2))/4.
ZMA=(PA(II,1,3)+PA(II,2,3)+PA(II,3,3)+PA(II,4,3))/4.
XMB=(PB(II,1,1)+PB(II,2,1)+PB(II,3,1)+PB(II,4,1))/4.
YMB=(PB(II,1,2)+PB(II,2,2)+PB(II,3,2)+PB(II,4,2))/4.
ZMB=(PB(II,1,3)+PB(II,2,3)+PB(II,3,3)+PB(II,4,3))/4.
XME=(PB(II,1,1)+PB(II,2,1)+PB(II,3,1)+PB(II,4,1))/2.
YME=(PB(II,1,2)+PB(II,2,2)+PB(II,3,2)+PB(II,4,2))/2.
ZME=(PB(II,1,3)+PB(II,2,3)+PB(II,3,3)+PB(II,4,3))/2.
X=(CXX*(XMA-TPX)+CYX*(YMA-TPY)+CZX*(ZMA-TPZ))*PAC
Y=(AXY*(XMA-TPX)+AYX*(YMA-TPY)+AZX*(ZMA-TPZ))*PAC
CALL PLOT(X,Y,3)
X=(CXX*(XME-TPX)+CYX*(YME-TPY)+CZX*(ZME-TPZ))*PAC
CALL PLOT(XE,0.0,2)
X=(CXX*(XMB-TPX)+CYX*(YMB-TPY)+CZX*(ZMB-TPZ))*PAC
Y=(AXY*(XMB-TPX)+AYX*(YMB-TPY)+AZX*(ZMB-TPZ))*PAC
CALL PLOT(XB,YB,2)
DP=SQRT((XB-XE)**2+YB*YB)
CXP=(XB-XE)/DP
CYP=YB/DP
X=DP*(CXP=0.74+CYP*0.15)+XE
Y=DP*(CYP=0.74-CXP*0.15)
CALL PLOT(X,Y,2)
CALL PLOT(XB,YB,3)
X=DP*(CXP=0.74-CYP*0.15)+XE
Y=DP*(CYP=0.74+CXP*0.15)
CALL PLOT(X,Y,2)
CONTINUE
130 IC=ICT+ITK(IV)
CALL NUMBER(-3.25,-3.6,.2,FLOAT(ITK(IV)),0.,-1)
CALL SYMBOL(-2.65,-3.6,.2,\"OVERLAP MODES BETWEEN\",0.,21)
CALL SYMBOL(-3.25,-3.95,.2,\"PLATE SIDE AND\",0.,21)
CALL NUMBER(-2.05,-3.95,.2,FLOAT(NPA),0.,-1)
CALL NUMBER(-2.5,-3.95,.2,FLOAT(ISB),0.,-1)
CALL SYMBOL(-3.25,-4.3,.2,\"PLATE SIDE\",0.,0.14)
CALL NUMBER(-2.05,-4.3,.2,FLOAT(NPB),0.,-1)
CALL NUMBER(-2.5,-4.3,.2,FLOAT(ISB),0.,-1)
CALL PLOT(0.,0.,-999)
CONTINUE
RETURN
END
APPENDIX 13

SUBROUTINE GPLOT2

SUBROUTINE GPLOT2(NM, NP, X, Y, Z, IA, IB, NPLTS, PCN, IPL, NWR, NPLTM, NAT, NW, ICN, NCNRS)
DIMENSION X(1), Y(1), Z(1), IA(1), IB(1), PCN(3, ICN, IPL)
DIMENSION NCNRS(1)
XNTOT = NWR + NPLTM + NAT
XNWR = NWR
XNPLTM = NPLTM
XNAT = NAT
DMX = -1.0
XMN = 1.0E10
YMN = XMN
ZMN = XMN
DO110 I = 1, NP
  IF (X(I) .LT. XMN) XMN = X(I)
  IF (Y(I) .LT. YMN) YMN = Y(I)
  IF (Z(I) .LT. ZMN) ZMN = Z(I)
DO110 CONTINUE
DO160 J = 1, NPLTS
  IF (NPLTS .EQ. 0) GO TO 160
  DO390 NC = 1, NCNRS(J)
    XM = PCN(1, NC, J)
    YM = PCN(2, NC, J)
    ZM = PCN(3, NC, J)
    IF (XM .LT. XMN) XMN = XM
    IF (YM .LT. YMN) YMN = YM
    IF (ZM .LT. ZMN) ZMN = ZM
  DO390 CONTINUE
  DO410 I = 1, NP
    IF (NP .EQ. 0) GO TO 410
    DIJ = SQRT((X(I) - XM)**2 + (Y(I) - YM)**2 + (Z(I) - ZM)**2)
    IF (DIJ .GT. DMX) DMX = DIJ
  DO410 CONTINUE
DO420 K = J, NPLTS
  IF (NPLTS .EQ. 0) GO TO 420
  DO430 MC = 1, NCNRS(K)
    XM = PCN(1, MC, K)
    YM = PCN(2, MC, K)
    ZM = PCN(3, MC, K)
    DIJ = SQRT((XM - XN)**2 + (YM - YN)**2 + (ZM - ZN)**2)
  DO430 CONTINUE
236
IF (DIJ.GT.DMX) DMX=DIJ
CONTINUE

430
CONTINUE
420
CONTINUE
390
CONTINUE
160
CONTINUE

F=3.0/DMX
FW=1.0/(FW*FW)
DO100 IV=1,4
IF (IV.EQ.1) CALLPLOT(0.75,1.5,-3)
IF (IV.EQ.2) CALLPLOT(6.5,0.0,-3)
IF (IV.EQ.3) CALLPLOT(-6.5,7.5,-3)
IF (IV.EQ.4) CALLPLOT(3.5,-3.0,-3)
IF (IV.EQ.1) CALLSYMBOL(0.0,-0.5,0.2,11HAXIS VIEW,0.0,11)
IF (IV.EQ.2) CALLSYMBOL(-3.0,-0.5,0.2,11HY AXIS VIEW,0.0,11)
IF (IV.EQ.3) CALLSYMBOL(0.0,-3.5,0.2,11HZ AXIS VIEW,0.0,11)
IF (IV.EQ.4) CALLSYMBOL(0.0,0.0,0.2,11HATTACH. MODES,0.0,11)

120
CONTINUE
XPI=-(XI-XMN)*F
YP1=-(Z1-ZMN)*F
XP2=-(Y2-YMN)*F
YP2=-(Z2-ZMN)*F
GOTO140

400
CONTINUE
IF (NM.LE.0) GOTO210
DO200 I=1,NM
N1=IA(I)
N2=IB(I)
XI=X(N1)
Y1=Y(N1)
Z1=Z(N1)
X2=X(N2)
Y2=Y(N2)
Z2=Z(N2)

IF (IV.EQ.2) GOTO120
IF (IV.EQ.3) GOTO130

XPI=(Y1-YMN)*F
YP1=(Z1-ZMN)*F
XP2=(Y2-YMN)*F
YP2=(Z2-ZMN)*F
GOTO140

237
GOTO140
CONTINUE
XP1=(Y1-YMN)*F
YP1=-(X1-XMN)*F
XP2=(Y2-YMN)*F
YP2=-(X2-XMN)*F
CONTINUE
FORMAT(1X,4E15.3)
CALLSYMBOL(XP1,YP1,0.05,1,0.0,-1)
CALLPLOT(XP2,YP2,2)
CALLSYMBOL(XP2,YP2,0.05,1,0.0,-1)
CONTINUE
DO230K=KK
DO220J=1,NPLTS
KK=NCNRS(J)+1
DO230K=1,KK
L=K
IF(K.EQ.KK)L=1
X1=PCN(1,L,J)
Y1=PCN(2,L,J)
Z1=PCN(3,L,J)
IU=2
IF(K.EQ.1)IU=3
IF(IU.EQ.2)GOTO270
IF(IU.EQ.3)GOTO280
XP=(Y1-YMN)*F
YP=(Z1-ZMN)*F
GOTO290
CONTINUE
XP=-(X1-XMN)*F
YP=(Z1-ZMN)*F
GOTO290
CONTINUE
XP=(Y1-YMN)*F
YP=-(X1-XMN)*F
CONTINUE
CALLPLOT(XP,YP,IU)
CONTINUE
CONTINUE
CONTINUE
CALLPLOT(0.,0.,999)
RETURN
END
APPENDIX 14

SUBROUTINE ZTOT

SUBROUTINE ZTOT(IA, IB, INM, ISC, IL, I2, I3, JA, JB, MD, NWR, ND, NM,
2NP, CGD, SGD, D, X, Y, Z, ZLD, NPL, NAT, ZS, ZLDA, PA, PB,
3NSA, NPLA, PCN, IPL, IPLM, BSK, ZT, ZTF, NM12N, NM23N, ICN,
4NDNPL, NOVT, INT, INTM, CM, ERSR, RMIN, DR, IAT, IPN,
5IQUAD, NCNS, IFIL, IREC, ICC)

DIMENSION IA(1), IB(1), ISC(1), IL(1), I2(1), I3(1), JA(1), JB(1),
2MD(INM, 4), ND(1), D(1), X(1), Y(1), Z(1), IPN(1), DRD(1)
DIMENSION PA(IPLM, 4, 3), PB(IPLM, 4, 3), NSA(1), NPLA(1), PCN(3, ICN, IPL)
DIMENSION BSK(1), NM12N(1), NM23N(1), NDNPLT(1), RMIN(1), DR(1)
DIMENSION PDIST(6), IQUAD(1), ERSR(1), NCNS(1), IREC(1)
COMPLEX ZLDA(1), SGD(1), CGD(1), ZLD(1), ZTF(ICC, ICC), ZT(1)
COMPLEX ES, KX, ETA, GAM, ZMN, EP3, ERSR(IAT, 400), EX, EY, EZ, DZMN
COMMON /A/ WV, PI, A, Q, GAM, ETA, XK

AB(I, A, B) = (I - 1) * B + (2 - 1) * A
IJ(I, J, NT) = (1 - J) * NT - (J * J - J) / 2 + 1
IWG = 0
IWZ = 0
ICAL = 1
IF((ING + INZ) GT 0) WRITE(6, 365) IRDZM

365 FORMAT(//3X, 'Z MATRIX READ OPTION: IRDZM = ', I3)
IF (ING.EQ. 0) WRITE(6, 360) NWR, NPL, NAT
E0 = 8.85E-12
EP3 = CMPLX(0, 0.0)
NTOT = NWR + NDNPLT(NPL) + NOVT + NAT
WRITE(6, *) NTOT, NWR, NDNPLT(NPL), NOVT, NAT
NDT = 18
NDE = 18
IFGD = 0
C
C EVALUATE THE IMPEDANCE ARRAY ZT(MN) (IF IFIL = 0) OR
C IMPEDANCE MATRIX ZTF(M, N) (IF IFIL = 1)
C
C (COUNTING ACROSS)
C REGION TEST EXP
C 1 W W
C 2 W P
C 3 W A
C 4 P W
C 5 P P
C 6 P A

239
C 7 A W
C 8 A P
C 9 A A

CALCULATE REGION 1 (WIRE/WIRE).

IF(NMR.EQ.0)GOTO100
IF(IRDZM.EQ.1)GOTO100
FHZ=3.E8/WV
AEFAC=1.0
AEQ=AEFAC*A
CALLSGANT(IA,IB,INM,INT,ISC,II,13,JA,JB,MD,NWR,ND,NN,NP,AEQ,
2 A,ZT,CGD,CMMD,ETA,EP3,ETA,FHZ,GAM,SGD,X,Y,Z,ZLD,ZS)
DO110J=1,NWR
JJ=NWR-J+1
DO110I=J,J
II=NWR-I+1

C Regular surface patch test modes are used (IFIL = 0). Then
C the impedances calculated by SGANT are stored in the one
C dimensional array ZT.

C*************************************************************************
C IF(IFILEQ.0)THEN
IJJN=IJJ(II,JJ,NTOT)
K=IJJ(II,JJ,NWR)
ZT(IJJN)=ZT(K)
END IF
C*************************************************************************
C Filamentary surface patch test modes are used (IFIL = 1). Then
C the impedances calculated by SGANT are stored in the two
C dimensional array ZTF.

C*************************************************************************
K=IJJ(II,JJ,NWR)
ZTF(JJJJ)=ZT(K)
ZTF(IJJ)=ZT(JJJ)
END IF
C*************************************************************************
C IF(INWZ.EQ.1)WRITE(6,390)I,J,IJJN,ZT(IJJN)
390 FORMAT(2X,214,16,215.5)
110 CONTINUE
100 CONTINUE
C******
C IF(NPL*NAT.EQ.0)RETURN
C
C CALCULATE REGIONS 2 THROUGH 9.
C
DO130N=1,NTOT
C*************************************************************************
C If regular surface patch test modes are used (IFIL = 0) the
C resulting impedance matrix is symmetric; i.e. only the lower
C triangular part of it is calculated and the entries are stored
C in array ZT. If filamentary surface patch modes are used (IFIL = 1)
C the whole impedance matrix is calculated and stored in array
C ZTF(M,N).
C*************************************************************************
DO140M=NN,NTOT
INDI=0

240
IF (IFIL.EQ.0) MN=IJ(M, N, NTOT)
IF (M.LE.NWR .AND. N.LE.NWR) GO TO 140
IF (M.LE.NWR) GO TO 115
IF (N.LE.NWR) GO TO 115
IF (N.GT.NWR+NDNPLT(NPL)) GO TO 115
IF (M.GT.NWR+NDNPLT(NPL)) GO TO 115

C*************************************************************************
C Determine what, if any, plate we are on.
C*************************************************************************
DO 20 I=1,NPL
IF (M-NWR.GT. NDNPLT(I)) GO TO 20
GOTO 25
20 CONTINUE
GOTO 115
25 DO 30 J=1,NPL
IF (N-NWR.GT. NDNPLT(J)) GO TO 30
GOTO 35
30 CONTINUE
GOTO 115
35 IF (I.NE. J) GO TO 115
C*************************************************************************
C We are on plate I=J
C*************************************************************************
IF (IREC(I).EQ.0) GO TO 115
C*************************************************************************
C Calculation of the mutual impedance between two
C modes on the same rectangular plate using the
C TOEPLITZ properties.
C*************************************************************************
NM12=NMl2N(I)
NM23=NM23N(I)
NDNI=0
IF (I.GT.1) NDNI=NDNPLT(I-1)
IF (IPN(I).EQ.2) NDNI=NDNI-(NM12-1)*NM23
K=M-NWR-NDNI
L=N-NWR-NDNI
IF (K.LT.L) GO TO 140
IF (L.EQ.1 .OR. L.EQ. NM23*(NM12-1)+1) THEN
INDI=1
GO TO 115
END IF
CALL TOPO(NM12, NM23, K, L, MT, NTSGN)
K=KT+NWR+NDNI
L=LT+NWR+NDNI
C*************************************************************************
C Regular surface patch test modes are used (IFIL = 0).
C*************************************************************************
IF (IFIL.EQ.0) THEN
KL=IJ(K, L, NTOT)
ZT(MN)=ZT(KL)*SGN
END IF
C*************************************************************************
C Filamentary surface patch test modes are used (IFIL = 1).
C*************************************************************************
IF (IFIL.EQ.1) THEN
ZTF(M, N)=ZTF(K, L)*SGN
ZTF(N, M)=ZTF(M, N)
END IF
C The polarity indicators IM12 and IN12 are defined below.

C***********************
IM12=(-1)**I
IN12=(-1)**J

C DETERMINE TEST MODE TYPE.

C IF(M.GT.NWR+NDNPLT(NPL)+NOVT) GOTO 190
IF(M.GT.NWR) GOTO 180

C TEST MODE IS A WIRE

C***********************

K=M
I1K=I2(K)
I2K=I1(K)
IF(I.EQ.2) I2K=I3(K)
KWS=JA(K)
IF(I.EQ.2) KWS=JB(K)
XM1=X(I1K)
YM1=Y(I1K)
ZM1=Z(I1K)
XM2=X(I2K)
YM2=Y(I2K)
ZM2=Z(I2K)
IOP=3
IIOP=0
IF(IWG.EQ.1) WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2
360 FORMAT(2X,5I4,12F8.3)
GOTO 280

180 CONTINUE

C TEST MODE IS A PLATE.

C K=M-NWR

C******************************

C The geometry of the test plate mode is defined below.

C***********************

IF(IFIL.EQ.0) IOP=1
IF(IFIL.EQ.1) IOP=3
IACM=IQUAD(K)
XM1=AB(I,PA(K,1,1),PB(K,1,1))
YM1=AB(I,PA(K,1,2),PB(K,1,2))
ZM1=AB(I,PA(K,1,3),PB(K,1,3))
XM2=AB(I,PA(K,2,1),PB(K,2,1))
YM2=AB(I,PA(K,2,2),PB(K,2,2))
ZM2=AB(I,PA(K,2,3),PB(K,2,3))
XM3=AB(I,PA(K,3,1),PB(K,3,1))
YM3=AB(I,PA(K,3,2),PB(K,3,2))
ZM3=AB(I,PA(K,3,3),PB(K,3,3))
XM4=AB(I,PA(K,4,1),PB(K,4,1))
YM4=AB(I,PA(K,4,2),PB(K,4,2))
ZM4=AB(I,PA(K,4,3),PB(K,4,3))

C If IFIL = 1 the endpoints of the filamentary
C test surface patch monopole are defined
C below.
C***************************************************
IF(IFIL.EQ.1) THEN
  XM1=(XM1+XM4)/2.0
  YM1=(YM1+YM4)/2.0
  ZM1=(ZM1+ZM4)/2.0
  XM2=(XM2+XM3)/2.0
  YM2=(YM2+YM3)/2.0
  ZM2=(ZM2+ZM3)/2.0
  GO TO 280
END IF

IF(IWG.EQ.1) WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2,
X3,YM3,ZM3,X4,YM4,ZM4
GOTO280
190 CONTINUE
C
C TEST MODE IS AN ATTACHMENT MODE.
C
K=M-MR-NDNFLT(NFLT)-NOVT
IM12=1
IF(I.EQ.2) GOTO 200
C***************************************************
C The geometry of the disk monopole of the attachment
C test mode is defined below.
C***************************************************
IOP=2
IIOP=0
NAS=NSA(K)
IF(NAS.GT.NM) GOTO 210
NAP=IA(NAS)
GOTO 220
210 CONTINUE
NAS=NAS-NM
NAP=IB(NAS)
220 CONTINUE
BM=BD5K(K)
XM1=X(NAP)
YM1=Y(NAP)
ZM1=Z(NAP)
NPLK=NPLA(K)
XM2=PCN(1,1,NPLK)
YM2=PCN(2,1,NPLK)
ZM2=PCN(3,1,NPLK)
XM3=PCN(1,2,NPLK)
YM3=PCN(2,2,NPLK)
ZM3=PCN(3,2,NPLK)
IF(IWG.EQ.1) WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2,
X3,YM3,ZM3
GOTO 280
200 CONTINUE
C***************************************************
The geometry of the wire monopole of the attachment test mode is defined below.

**C** The geometry of the wire monopole of the attachment test mode is defined below.

**C**

**IO=3**

**II=0**

**NAS**=**NSA**(**K**)

**IF**(**NAS**.GT.**NM**) **GOTO** **230**

**NPP**=**IB**(**NAS**)**GOTO** **240**

**230 CONTINUE**

**NAS**=**NAS**.-**NM**

**NPP**=**IA**(**NAS**)**GOTO** **240**

**CONTINUE**

**XM=K**(**NPP**)**Y=K**(**NPP**)**Z=Z**(**NPP**)**IF**(**IWG**.EQ.1) **WRITE** (**6,**360)**M, **N, **I, **L, **X**,**Y**,**Z**

**GOTO** **280**

**280 CONTINUE**

**C**

**DETERMINE EXPANSION MODE TYPE.**

**C**

**IF** (**N**.GT.**NWR**+**NDNPLT**(NPL**)+**NOVT**) **GOTO** **270**

**IF** (**N**.GT.**NWR**) **GOTO** **260**

**C**

**EXPRESSION MODE IS A WIRE.**

**L=N**

**JOP=3**

**IF** (**IRDZM**.NE.1) **GOTO** **262**

**IF** (**IOP**.GE.2 .AND. **JOP**.GE.2) **GOTO** **140**

**262 IF** (**TRBZM**.NE.2) **GOTO** **263**

**IF** (**IOP**.EQ.1 .AND. **JOP**.EQ.1) **GOTO** **140**

**263 CONTINUE**

**C**

**The geometry of the expansion wire monopole is defined below.**

**C**

**IF** (**I+J**.EQ.2) **THEN**

**IF** (**IFIL**.EQ.0) **ZT**(M,N)**=(0,0,0)**

**IF** (**IFIL**.EQ.1) **ZTF**(M,N)**=(0,0,0)**

**END IF**

**I=2**(**L**)**I2=I**(**L**)**I3=I**(**L**)**

**IF** (**J**.EQ.2) **I2**=**I3**(**L**)**XN**=**X**(**I**)**YN**=**Y**(**I**)**ZN**=**Z**(**I**)**XN**2**X**(**I2**)**YN**2**Y**(**I2**)**ZN**2**Z**(**I2**)**IF** (**IWG**.EQ.1) **WRITE** (**6,**360)**M, **N, **L, **JOP, **INL**,**XN**,**YN**,**ZN**,**XN**2,**YN**2,**ZN**2

**261 IF** (**ICAL**.NE.1) **GO TO** **6001**

**IF** (**IOP**.EQ.3) **IOP=3**

**C**

**Test monopole is a wire.**

**C**

**IF** (**IOP**.EQ.3) **CALL** **ZWTWE**(XM,YM,ZM,XM2,YM2,ZM2,IM12,
&XN1, YN1, ZN1, XN2, YN2, ZN2, IN12, ZMN, 1)

C Test monopole is a disk.
C

IF (IOP.EQ.2) CALL DSKTST (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, 
& 3, XN1, YN1, ZN1, XN2, YN2, ZN2, XM3, YM3, ZM3, IN12, 0,
& INTD, BM, BN, ZMN)
IF (IOP.NE.1) GO TO 6001

C Test monopole is a surface patch.
C

CXMN = (XN1 + XN2) / 2.
CYMN = (YN1 + YN2) / 2.
CZMN = (ZN1 + ZN2) / 2.
CXM4 = (XM1 + XM2 + XM3 + XM4) / 4.
CYM4 = (YM1 + YM2 + YM3 + YM4) / 4.
CZM4 = (ZM1 + ZM2 + ZM3 + ZM4) / 4.
C
C DISMN = SQRT ( (XM4 - XMN) **2 + (YM4 - YMN) **2 + (ZM4 - ZMN) **2 )
IF (DISMN.LE.0.25*WV) NPT = 8
IF (DISMN.GT.0.25*WV. AND. DISMN.LE.0.35*WV) NPT = 4
IF (DISMN.GT.0.35*WV) NPT = 2
IF (IACM.NE.-3) CALL PLTST2 (XM4, YM4, ZM4, XM1, YM1, ZM1, XM2, 
& YM2, ZM2, XM3, YM3, ZM3, IN12, JOP, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YM3, ZN3, 
& YM3, ZM3, IN12, JOP, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, NPT,
& 0, BN, ZMN)
6001 IF (IFIL.EQ.1) ZTF (M, N) = ZTF (M, N) + ZMN
IF (IFIL.EQ.0) ZT (MN) = ZT (MN) + ZMN
GOTO 160

260 CONTINUE
C
C EXPANSION MODE IS A PLATE.
C
C JOP = 1
IF (IRDZM.NE.1) GO TO 264
IF (IOP.EQ.2. AND. JOP.GE.2) GO TO 140
264 IF (IRDZM.NE.2) GO TO 265
IF (IOP.EQ.3) GO TO 140
IF (IOP.EQ.1. AND. JOP.EQ.1) GO TO 140
265 CONTINUE
L = N = NMRR

C The geometry of the expansion surface patch
C monopole is defined below.
C

IACN = IQUAD (L)
XN1 = AB (J, PA (L, 1, 1), PB (L, 1, 1))
YN1 = AB (J, PA (L, 1, 2), PB (L, 1, 2))
ZN1 = AB (J, PA (L, 1, 3), PB (L, 1, 3))
XN2 = AB (J, PA (L, 2, 1), PB (L, 2, 1))
YN2 = AB (J, PA (L, 2, 2), PB (L, 2, 2))
ZN2 = AB (J, PA (L, 2, 3), PB (L, 2, 3))
XN3 = AB (J, PA (L, 3, 1), PB (L, 3, 1))
YN3 = AB (J, PA (L, 3, 2), PB (L, 3, 2))
ZN3 = AB (J, PA (L, 3, 3), PB (L, 3, 3))
XN4 = AB (J, PA (L, 4, 1), PB (L, 4, 1))
YN4 = AB(J, PA(L, 4, 2), PB(L, 4, 2))
ZN4 = AB(J, PA(L, 4, 3), PB(L, 4, 3))
IF(IOP.EQ.3) IOP = 3
IF(I+J.EQ.2) THEN
IF(IFIL.EQ.0) ZTF(M,N) = (0.0,0.0)
IF(IFIL.EQ.1) ZTF(M,N) = (0.0,0.0)
END IF
IF(IW.G.EQ.1) WRITE (6,360) M,N,L,JOP, IN12, XN1, YN1, ZN1, XN2, YN2, ZN2,
& XN3, YN3, ZN3
IF (ICAL .NE. 1) GOTO 6002
XM4D = (XN1+XN2+XN3+XN4) / 4.0
YM4D = (YN1+YN2+YN3+YN4) / 4.0
ZM4D = (ZN1+ZN2+ZN3+ZN4) / 4.0
IF(IOP.EQ.1) GO TO 431
C Test monopole is a surface patch.
C***************************************
XM4D = (XN1+XN2+XN3+XN4) / 4.0
YM4D = (YN1+YN2+YN3+YN4) / 4.0
ZM4D = (ZN1+ZN2+ZN3+ZN4) / 4.0
DISMN = SQRT((XM4D-XMDN)**2+(YM4D-YMDN)**2+(ZM4D-ZMDN)**2)
IF (DISMN.LE.0.25*WV) NPT = 8
IF (DISMN.GT.0.25*WV .AND. DISMN.LE.0.35*WV) NPT = 4
IF (DISMN.GT.0.35*WV) NPT = 2
IF (IACM.EQ.-3 .AND. IDISMN.EQ.0.6*WV) NPT = 1
IF (IACM.EQ.-3 .AND. IACN.EQ.-3) GO TO 492
GO TO 479
492 CONTINUE
CALL PLTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,XM12,1,
& XM1,YM1,ZM1,XM2,YM2,ZM2,XN3,YM3,ZM3,XN12,NPT,NPT,BN,
& ZMN)
GO TO 6002
479 CONTINUE
IF(IOP.EQ.1) CALL PLTST2(XM4,YM4,ZM4,
& XM1,YM1,ZM1,XM2,YM2,ZM2,
& XM3,YM3,ZM3,XM12,JOP,YM4,ZM4,XN3,YM3,ZM3,XN12,NPT,NPT,BN,
& ZMN)
GO TO 6002
431 CONTINUE
IF(IOP.NE.2) GO TO 458
C***************************************
C Test monopole is a disk
C***************************************
IF(L.GT.NDNP(L)) GO TO 5100
C CHECK FOR PARALLEL PLATE-DISK
C***************************************
PX = (YN2-YN1)*(ZN3-ZN2)-(YN3-YN2)*(ZN2-ZN1)
PY = (XN3-XN2)*(ZN2-ZN1)-(XN2-XN1)*(ZN3-ZN2)
PNZ = (XN2-XN1)*(YN3-YN2)-(XN3-XN2)*(YN2-YN1)
DNL2 = SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
DN23 = SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
DNX = (YN2-YN1)*(ZN3-ZN2)-(YN3-YN2)*(ZN2-ZN1)
DNY = (XN2-XN1)*(ZN3-ZN2)-(XN3-XN2)*(ZN2-ZN1)
DNZ = (XN2-XN1)*(YN3-YN2)-(XN3-XN2)*(YN2-YN1)
COSTH = (PX*DNX+PY*DNY+PNZ*DNZ)/SQRT((PX)**2+...
& PNY*2+PNZ*2) *(DNX**2+DNY**2+DNZ**2))
IF(ABS(COSTH).LT. .997) GOTO 5100

C**************************************
C PLATE AND DISK ARE PARALLEL
C CHECK FOR FIRST COLUMN OF PLATE
C**************************************
IF(J.EQ.2) GOTO 5005
KPL=1
IF(I.EQ.1) GOTO 5002
DO 5001 I=2,NPL
KPL=I
IF(I.EQ.NDPLT(I-1)+1) GOTO 5002
5001 CONTINUE
C**************************************
C NOT ON A FIRST PLATE COLUMN
C*******************************************************************************
5005 RMINK=RMIN(K)
DRK=DR(K)
DIST=PDIST(K)
IF(IACN.EQ.3) CALL PDPL2(XM1,YM1,ZN1,X1,Y1,ZN1,ZN2, XN3,YN3,ZN3,IN12,INTP,ERVSR,IAT,RMINK,DRK,ZMN,DIST)
IF(IACN.NE.3) CALL PDPZI (XM1,YN14141,K,XN4,YN4,ZN4,XN2,YN2,ZN2,XN3,YN3,ZN3,IACN,IN12,INTPERVSR,IAT,RMINK,DRK,ZMN, & DIST)
GOTO 6002
C*******************************************************************************
C ON A FIRST PLATE COLUMN
C FIND RMAX,RMIN
C*******************************************************************************
5002 IF(IREC(KPL).EQ.0) GOTO 5003
PX0=(PCN(1,1,KPL)+PCN(1,3,KPL))/2.
PY0=(PCN(2,1,KPL)+PCN(2,3,KPL))/2.
PZ0=(PCN(3,1,KPL)+PCN(3,3,KPL))/2.
DIAG=.5*SQRT((PCN(1,1,KPL)-PCN(1,3,KPL))**2+
&(PCN(2,1,KPL)-PCN(2,3,KPL))**2+(PCN(3,1,KPL)-PCN(3,3,KPL))**2)
& R=SQRT((PX0-XM1)**2+(PY0-Y1)**2+(PZ0-ZM1)**2)
RMAX=BDK5(K)+R+DIAG
RMIN(K)=R-BDK5(K)-DIAG
IF(RMIN(K).LT.0.) RMIN(K)=0.
GOTO 5006
5003 RMIN(K)=0.0
DIAG=0.0
PX0=0.0
PY0=0.0
PZ0=0.0
DO 5004 IKC=1,NCNRS(KPL)
PX0=PX0+PCN(1,IKC,KPL)
PY0=PY0+PCN(2,IKC,KPL)
PZ0=PZ0+PCN(3,IKC,KPL)
DCC=SQRT((XM1-PCN(1,IKC,KPL))**2+(YM1-PCN(2,IKC,KPL))**2+
& (ZN1-PCN(3,IKC,KPL))**2)
IF(DCC.GT.DIAG) DIAG=DCC
5004 CONTINUE
PX0=PX0/NCNRS(KPL)
PY0=PY0/NCNRS(KPL)
PZ0=PZ0/NCNRS(KPL)
R=SQRT((PX0-XM1)**2+(PY0-Y1)**2+(PZ0-ZM1)**2)
RMAX=R+DIAG+BDK5(K)

247
\[ NINA = 2 \times (60 \times (RMAX - RMIN(K)) / \text{WV}) \]
\[ NINA = NINA + 1 \]
\[ \text{IF}(NINA \geq 400) \text{ WRITE}(6, 5111) M, N, NINA \]
\[ \text{DR}(K) = (RMAX - RMIN(K)) / NINA \]

---

C Compute the distance between the plane of the disk monopole and the plane of the surface.
C the patch.
C*******************************************************************************
C PND = -PNX*XM1 - PNY*YM1 - PNZ*ZN1
C DIST = (PNX*XM1 + PNY*YM1 + PNZ*ZN1 + PND) / SQRT(PNX**2 + PNY**2 + PNZ**2)
C DIST = ABS(DIST)
C PDIST(K) = DIST
C*******************************************************************************
C FILL ARRAY ZVSR
C*******************************************************************************
DO 5010 JJ = 1, NINA
R = RMIN(K) + (JJ - 1) * DR(K)
CALL ERDSK(A, BDSK(K), R, DIST, ETA, WV, 100, EX)
5010 ERSR(K, JJ) = EX
GOTO 5005
C*******************************************************************************
C Disk monopole is not parallel to the surface.
C the patch.
C*******************************************************************************
5100 DISMN = SQRT((XM1 - XM2)**2 + (YM1 - YM2)**2 + (ZN1 - ZN2)**2)
C IF(DISMN.LE.0.35*WV) THEN
  INTP = 4
  NDT = 8
  END IF
C IF(DISMN.GT.0.35*WV.AND.DISMN.LE.0.6*WV) THEN
  INTP = 4
  NDT = 4
  END IF
C IF(DISMN.GT.0.6*WV) THEN
  INTP = 2
  NDT = 4
  END IF
C IF(IACN.NE.3) CALL DSKTSX(XM1, YM1, XM2, YM2, XM3, YM3, ZM3, IN12, INTP, NDT, BM, BN, ZMN)
C IF(IACN. EQ. -3) CALL DSKTST(XM1, XM2, YM2, XM3, YM3, ZM3, IN12, INTP, NDT, BM, BN, ZMN)
GOTO 6002
458 IF(IOP.NE.3) GO TO 453
C*******************************************************************************
C Expansion monopole is a wire.
C*******************************************************************************
KINT = 0
DM = SQRT((XM2 - XM1)**2 + (YM2 - YM1)**2 + (ZN2 - ZN1)**2)
XMMD = (XM1 + XM2) / 2.
YMMD = (YM1 + YM2) / 2.
ZNMD = (ZN1 + ZN2) / 2.
DISMN = SQRT((XMMD - XM1)**2 + (YMMD - YM1)**2 + (ZNMD - ZN1)**2)
C IF(DISMN.LE.0.25*WV) NPT = 8
IF(DISMN.GT.0.25*WV.AND.DISMN.LE.0.35*WV) NPT=4
IF(DISMN.GT.0.35*WV) NPT=2
IF(IACN.NE.-3.AND.DISMN.GE.0.6*WV) NPT=1
CONTINUE
IF(IACN.NE.-3.AND.DISM.N.GE.0.6*WV) CALL ZWTPE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM1,2,
1 XM1,YM1,ZM1,XM2,YM2,ZM2,IM1,2,
IF(IFGD.EQ.1.AND.IOP.EQ.3)CALL PLOT(0.0,0.999)
IF(IWZ.EQ.1)WRITE(6,*)M,N,ZT(MN),ZMN
IF(IWZ.EQ.1)WRITE(6,410)M,N,I,J,LT,NPT,NPE,CPU
410 FORMAT(1X,' M ',13,2X,' N ',13,2X,' I = ',12,2X,' J = ',12,2X,
1 ' NPT = ',12,1' NPE = ',12,2X,' CPU = ',F12.6)
GO TO 160
CONTINUE
C EXPANSION MODE IS AN ATTACHMENT MODE.
C The geometry of the disk monopole of the expansion
C attachment mode is defined below.
C******************************************************
NAS=NSA(L)
IF(NAS.GE.NM) GOTO 300
NAP=IA(NAS)
NAP2=IB(NAS)
GOTO 310
300 CONTINUE
NAS=NAS+NM
NAP=IA(NAS)
NAP2=IB(NAS)
310 CONTINUE
BN=BDSK(L)
XN1=X(NAP)
YN1 = Y(NAP)
ZN1 = Z(NAP)
XN2 = X(NAP2)
YN2 = Y(NAP2)
ZN2 = Z(NAP2)
NPLL = NPLA(L)
YN3 = Y(NAP2)
ZN3 = Z(NAP2)

IF(ING.EQ.1) WRITE(6,360) M,N,L,JOP,IN12,XNI,YN1,ZN1,XN2,YN2,ZN2,
2XN3,YN3,ZN3
GOTO 335
290 CONTINUE
C**************************
C The wire monopole of the attachment mode is
C defined below.
C**************************
NAS = NSA(L)
IF(NAS.GT.NM) GOTO 320
NPP = IB(NAS)
GOTO 330
320 CONTINUE
NAS = NAS - NM
NPP = IA(NAS)
330 CONTINUE
XN2 = X(NPP)
YN2 = Y(NPP)
ZN2 = Z(NPP)
JOP = 3
IF(IOP.NE.3.OR.JOP.NE.3) GOTO 341
IF(NAS .EQ. KWS) GOTO 341
IF(11.MK.EQ.NAP) GOTO 342
DZN1 = (GAM* D(NAS) * CGD(NAS) - SGD(NAS)) * ZS / (4. * PI * GAM * SQD(NAS)**2)
GOTO 343
342 DZN2 = (SGD(NAS) * CGD(NAS) - GAM * D(NAS)) * ZS / (4. * PI * GAM * SQD(NAS)**2)
343 ZMN = ZMN + DZN1*112*1N12
341 CONTINUE
IF(IWG.EQ.1) WRITE(6,360) M,N,L,JOP,IN12,XNI,YN1,ZN1,XN2,YN2,ZN2
340 IF(MIAL .NE. 1) GOTO 336
C Test monopole is a wire.
C**************************
IF(IOP.EQ.3) CALL ZW'IME(XM1,YM1,ZM1,XN2,YM2,ZM2,IN12,
2XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN)
C**************************
C Test monopole is a disk.
C**************************
IF(IOP.EQ.2) CALL DSKTST(XM1,YM1,ZM1,XM2,YM2,ZM2,IN12,
2XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN)
C**************************
C Test monopole is a surface patch.
C**************************
IF(IOP.EQ.1 .AND. IACM.EQ.-3) CALL PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,IN12,
2XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN)
250
IF (IOP. EQ. 1 .AND. IACM. NE. -3) CALL PLTST2 (XM4, YM4, ZM4, XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IN12, JOP, XM4, YM4, ZM4, XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IN12, NPT, NUE, BN, IACM, IACN, ZMN)
GOTO 336

335 IF (IOP. NE. 2) GOTO 531
IF (K. NE. 1) GOTO 531
C COMPUTE ATTACHMENT TO ATTACHMENT (SELF) IMPEDANCE
C***************************************
D = (NASS)
DXW = (XNT-XNI) / D = (NASS)
DYW = (YNT-YNI) / D = (NASS)
DZW = (ZNT-ZN1) / D = (NASS)
DPL = SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
DXP = (XN2-XN1) / DPL
DYP = (YN2-YN1) / DPL
DZP = (ZN2-ZN1) / DPL
DPL = SQRT((XN3-XN1)**2+(YN3-YN1)**2+(ZN3-ZN1)**2)
DXQ = (XN3-XN1) / DPL
DYQ = (YN3-YN1) / DPL
DZQ = (ZN3-ZN1) / DPL
DXN = DYP*DZQ-DYQ*DZP
DYN = DXQ*DZP-DXP*DZQ
DZN = DXP*DYP-DYQ*DQP
COSA = DXW*DXN+DYW*DYN+DZW*DZN
COSA = ABS COSA)
PSI = ACOS (COSA)
PSI = PSI * 180.0 / PI
CALL ZATAT2 (BM, D = (NASS), ZMN, 40, ZS, PSI)
IF (IFIL. EQ. 0) ZT (MN) = ZMN + ZLDA (K)
IF (IFIL. EQ. 1) ZTF (M, N) = ZMN + ZLDA (K)
IF (IN2. EQ. 1) WRITE (6-370) M, N, ZMN
GOTO 1073
C***************************************
C EXPANSION MONOPOLE IS A DISK.
C***************************************
531 IF (ICAL. NE. 1) GOTO 336
C***************************************
C Test monopole is a wire.
C***************************************
IF (IOP. EQ. 0) CALL ZWTDE (XM1, YM1, ZM1, XM2, YM2, ZM2, IM12, 6, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, BN, ZMN)
C***************************************
C Test monopole is a disk.
C***************************************
IF (IOP. EQ. 2) CALL DSKTST (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, 6, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, INTD, 6, BM, BN, ZMN)
C***************************************
C Test monopole is a surface patch.
C***************************************
IF (IOP. EQ. 1 .AND. IACM. EQ. 3) CALL PLTST (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, 6, ZM3, IN12, 1, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, INTD, 6, BN, ZMN)
IF (IOP. EQ. 1 .AND. IACM. NE. -3) CALL PLTST2 (XM4, YM4, ZM4, XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IN12, JOP, XM4, YM4, ZM4, XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IN12, NPT, 0, BN, IACM, IACN, ZMN)

251
IF (IFIL.EQ.0) ZT(M,N) = ZT(M,N) + ZMN
IF (IFIL.EQ.1) ZTF(M,N) = ZTF(M,N) + ZMN
IF (IWZ.EQ.1) WRITE (6, 370) M, N, ZT(M,N), ZMN
370 FORMAT (6X, 2I4, 4E12.3)
GOTO 160

160 CONTINUE

150 CONTINUE
IF (IWG.EQ.1) WRITE (6, 380)
380 FORMAT (/)

1073 CONTINUE

140 CONTINUE

130 CONTINUE
CALL GETCP(III1)
CPU1 = (III1 - III) / 100.

9973 CONTINUE
RETURN
END
APPENDIX 15

SUBROUTINE TOPO

C CHECK TOEPLITZ PROPERTIES
SGN=1.
IF (K .GT. NM23*(NM12-1)) GOTO 20
C X POLARIZATIONS
NRM= (K-1)/(NM12-1)+1
NCM= (NM12-1)/(NM12-1)+1
NRN= (L-1)/(NM12-1)+1
NCN= (NM12-1)/(NM12-1)+1
NU= NRM-NCN
NO= IABS (NCN-NCN)
MT= NU*(NM12-1)+NO+1
NT= 1
RETURN
20 IF (L .LE. NM23*(NM12-1)) GOTO 30
C Y POLARIZATIONS
MM= K-NM23*(NM12-1)
NN= L-NM23*(NM12-1)
NRMM= (MM-1)/(NM23-1)+1
NCMM= (NM12-1)/(NM23-1)+1
NRNN= (NM12-1)/(NM23-1)+1
NCNN= (NM12-1)/(NM23-1)+1
NU= NRMM-NCMM
NO= IABS (NCMM-NCNN)
MT= NU*(NM12-1)+NO+1+NM23*(NM12-1)
NT= 1+NM23*(NM12-1)
RETURN
30 CONTINUE
C XY POLARIZATIONS
MM= K-NM23*(NM12-1)
NSOM= 2*((MM-1)/(NM23-1))+1
NSUM= 2*((MM-1)/(NM23-1)+1)
NSUN= 2*((L-1)/(NM23-1)+1)
NSUN= 2*((L-1)/(NM23-1)+1)
NU= NSUN-NSUM
NO= NSUN-NSUM
IF (NU .GT. 0) SGN= -SGN
IF (NU .GT. 0) NU= -NU
IF (NO .GT. 1) SGN= -SGN
IF (NO .GT. 1) NO= NO
MT= (1-NO)/2*(NM23-1)+(1-NU)/2+NM23*(NM12-1)
NT= 1
RETURN
END
SUBROUTINE PLTTST

SUBROUTINE PLTTST (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IM12, 
& JOP, X11, Y11, Z11, XJ2, Y12, Z12, X13, Y13, Z13, IN12, INTPT, NINT, 
& BN, ZMN)

COMPLEX Z, ZMN, GAM, ETA

COMMON /A/ W1V, PI, A, Q, GAM, ETA, XK

IF (JOP .NE. 1) GO TO 100

XN3-XN2
YN3=YN2
ZN3=ZN2

100 CONTINUE

D=SQRT((XM1+XM3-XN1-XN3)**2+(YM1+YM3-YN1-YN3)**2+
& (ZM1+ZM3-ZN1-ZN3)**2)/2.

NPT=INTPT

IF (D.GT. .25*W1V) NPT=2*(INTPT/6)

IF (NPT.LT. 2) NPT=2

ZMN=(0., 0J)

AM=XM3-XM2
BM=YM3-YM2
CM=ZN3-ZM2

DM23=SQRT(AM*AM+BM*BM+CM*CM)

IF (JOP .NE. 1) GOTO 6

C CHECK FOR PARALLEL LINE SOURCES

COSTH=(XM2-XM1)*(XM2-XM1)+COSTH=(YM2-YM1)+(YM2-YM1)+
& (ZN2-ZN1)

DM12=SQRT((XM2-XM1)**2+(YM2-YM1)**2+(ZN2-ZN1)**2)

DN12=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)

COSTH=COSTH/(DM12*DN12)

IF (ABS(COSTH).GE. .997) GOTO 4

C CHECK FOR PARALLEL LINE SOURCE TRANSVERSE VECTORS

AN=XN3-XN2
BN=YN3-YN2
CN=ZN3-ZN2

DN23=SQRT(AN*AN+BN*BN+CN*CN)

COSTH=(AN*AN+BN*BN+CN*CN)/(DN23*DM23)

IF (ABS(COSTH).GE. .997) GOTO 4

NPT=MAXO (2* (INTPT/11), 2)

GOTO 6

4 CALL PPLTS (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IM12, 
& XN1, YN1, ZN1, XJ2, Y12, Z12, X13, Y13, Z13, IN12, NPT, ZMN)

RETURN
D23=DM23
NPI=NPT+1
DX=AM/NPT
DY=BM/NPT
DZ=CM/NPT
DH=D23/NPT
DO 10 I=1,NPI
W=3+(-1)**I
X1=XMI+DX*(I-1)
Y1=YMI+DY*(I-1)
Z1=ZM1+DZ*(I-1)
X2=XMI+DX*(I-1)
Y2=YMI+DY*(I-1)
Z2=ZM1+DZ*(I-1)
IF(JOP.EQ.1)CALL ZWTPE(X1,Y1,Z1,X2,Y2,Z2,1N12,XN1,YN1,ZN1,
& XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NINT,Z)
IF(JOP.EQ.2)CALL ZWDE(X1,Y1,Z1,X2,Y2,Z2,IN12,XN1,YN1,ZN1,
& XN2,YN2,ZN2,XN3,ZN3,NINT,BN,Z)
IF(JOP.EQ.3)CALL ZWWE(X1,Y1,Z1,X2,Y2,Z2,IN12,XN1,YN1,ZN1,
& XN2,YN2,ZN2,IN12,Z,0)
10
ZMN=ZMN+ZM*COS(XK*(D23/2.-(I-1)*DH))
ZMN=ZMN*XX*DH/(6.*SIN(XK*D23/2.))
RETURN
END
APPENDIX 17
SUBROUTINE PPLTS

SUBROUTINE PPLTS (XMi, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IM12, & XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, IN12, NPT, ZMN)
COMPLEX Z, ZMN, GAM, ETA
COMPLEX EGD, CGDS, SGDS, SGDT, DUM, ZVSD(123)
COMMON /A/ WV, PI, A, Q, GAM, ETA, XK
C PLATE CURRENT DIRECTIONS ARE ORTHOGONAL TO A COMMON LINE
C USE FAST INTERPOLATION METHOD AND LN SINGULARITY TERM
ZMN=(0., 0.)
D23=SQRT ((XM3-XM2)**2 + (YM3-YM2)**2 + (ZM3-ZM2)**2)
E23=SQRT ((XN3-XN2)**2 + (YN3-YN2)**2 + (ZN3-ZN2)**2)
AM=XM2-XM1
BM=YM2-YM1
CM=ZM2-ZM1
AT=XN2-XN1
BT=YN2-YN1
CT=ZN2-ZN1
DM12=SQRT (AM*AM+BM*BM+CM*CM)
DN12=SQRT (AT*AT+BT*BT+CT*CT)
AM=AM/DM12
BM=BM/DM12
CM=CM/DM12
AT=AT/DN12
BT=BT/DN12
CT=CT/DN12
33 FORMAT('ERROR - NPK IN PPLTS GT 120')
EGD=CEXP (GAM*DM12)
CGDS=(EGD-1./EGD)/2.
SGDS=(EGD-1./EGD)/2.
EGD=CEXP (GAM*DN12)
SGDT=(EGD-1./EGD)/2.
DH=AM*DL1(D23,E23)/NPT
NPI=2*IFIX(.5+.5*D23/DH)
NPJ=2*IFIX(.5+.5*E23/DH)
DH=D23/NPI
EH=E23/NPJ
NPI1=NPI+1
NPJ1=NPJ+1
DD=(DH+EH)/2.
DX=(XM3-XM2)/NPI
DY=(YM3-YM2)/NPI

256
\[
DZ = \frac{(Z_3 - Z_2)}{NPJ}
\]
\[
EX = \frac{(X_3 - X_2)}{NPJ}
\]
\[
EY = \frac{(Y_3 - Y_2)}{NPJ}
\]
\[
EZ = \frac{(Z_3 - Z_2)}{NPJ}
\]

C COMPUTE DMIN, DMAX

\[
DMIN = 1000.
\]
\[
DMAX = 1000.
\]

DO 30 I = 1, NPJ

XM = XM1 + DX * (I - 1)

YM = YM1 + DY * (I - 1)

ZM = ZM1 + DZ * (I - 1)

DO 30 J = 1, NPJ

XN = XN1 + EX * (J - 1)

YN = YN1 + EY * (J - 1)

ZN = ZN1 + EZ * (J - 1)

D = DIST(XM, YM, ZM, AM, BM, CM, XN, YN, ZN, AT, BT, CT)

IF (D.GT.DMAX) DMAX = D

IF (D.GT.DMIN) GOTO 30

DMIN = D

30 CONTINUE

C FILL ZVSD WITH IMPEDANCES BETWEEN DMIN DMAX

ZN = \[(XN1 - XM1) * AM + (YN1 - YM1) * BM + (ZN1 - ZM1) * CM\]

ZNN = \[(XN2 - XM1) * AM + (YN2 - YM1) * BM + (ZN2 - ZM1) * CM\]

IF (ABS(AM * AT + BM * BT + CM * CT) .LT. .995) GOTO 100

C PARALLEL FILAMENT CASE

\[
XN = 0.
\]

XNN = 0.

GOTO 200

C PARALLEL TRANSVERSE VECTOR CASE

100 DN01 = \[(XN1 - XM1)^2 + (YN1 - YM1)^2 + (ZN1 - ZM1)^2\]

DN02 = \[(XN2 - XM1)^2 + (YN2 - YM1)^2 + (ZN2 - ZM1)^2\]

DMIN = DIST(XM1, YM1, ZM1, AM, BM, CM, XN1, YN1, ZN1, AT, BT, CT)

AM2 = (XM1 - XM2) / D23

BM2 = (YM1 - YM2) / D23

CM2 = (ZM1 - ZM2) / D23

AM3 = BM * CM2 - BM2 * CM

BM3 = CM * AM2 - AM * CM2

CM3 = AM * BM2 - BM * AM2

XN = \[(XN1 - XM1) * AM3 + (YN1 - YM1) * BM3 + (ZN1 - ZM1) * CM3\]

XNN = \[(XN2 - XM1) * AM3 + (YN2 - YM1) * BM3 + (ZN2 - ZM1) * CM3\]

NPK = IFIX(1.1 + (DMAX - DMIN) / DD)

IF (NPK.GT.120) WRITE(6,33)

DO 40 K = 1, NPK

D = DMIN + (K - 1) * DD

IF (D.LT.DD) D = DMIN + DD / 2.

CALL ZGSM4M(Z0, 0, 0, 0, Z0, 1M12, XN, D, ZN, XN, D, ZN, Q, DM12, & CGDS, SGDS, DN12, SGDT, Z)

40 ZVSD(K) = IM12 * IN12 * Z

IF (DMIN.GE.DD) GOTO 45

C TAKE CARE OF LOGARITHMIC SINGULARITY

RZ = REAL(ZVSD(1))

X1 = AIMG(ZVSD(1))

X2 = AIMG(ZVSD(2))

C2 = (X2 - X1) / ALOG(DMIN + DD) / (DMIN + DD / 2.)

C1 = X1 - C2 * ALOG(DMIN + DD / 2.)

AIZ = C1 + C2 * ALOG(DMIN + DD) + (DMIN + DD) / 2. * C2 * DMIN

ATAN2(DD, DMIN) / DD = (C1 + C2 * ALOG(SQRT(DMIN * DMIN + DD * DD)))

ZVSD(1) = CMPLX(RZ, AIZ)
DO 90 I=1,NPI1
   W=W*(-1)**I
   XM=XM1+DX*(I-1)
   YM=YM1+DY*(I-1)
   ZM=ZM1+DZ*(I-1)
DO 90 J=1,NPJ1
   V=V*(-1)**J
   IF(J.EQ.1 .OR. J.EQ.NPJ1) V=V/2.
   XN=XN1+EX*(J-1)
   YN=YN1+EY*(J-1)
   ZN=ZN1+EZ*(J-1)

C COMPUTE DISTANCE BETWEEN THE TWO MONOPOLES

D=DIST(XM,YM,ZM,AM,BM,CM,XN,YN,ZN,AT,BT,CT)
N=ABS((D-DMIN)/DD+1)
Z=ZVSD(N)+(ZVSDCN+1)-ZVSD(N))/DD*(D-DMIN-(N-1)*DD)
Z=Z*W*COS(XM*(D23/2.-I-1)*DH)*XK*DH/(6.*SIN(XK*D23/2.))
Z=Z*V*COS(YM*(E23/2.-(J-1)*EH))*XK*EH/(6.*SIN(XK*E23/2.))

90 ZMN=ZMN+Z
RETURN
END

FUNCTION DIST(X1,Y1,Z1,A1,B1,C1,X2,Y2,Z2,A2,B2,C2)

C DISTANCE BETWEEN LINE IN DIRECTION(A1,B1,C1) THROUGH P1 TO LINE IN DIRECTION(A2,B2,C2) THROUGH P2

A3=B1*C2-B2*C1
B3=C1*A2-A1*C2
T1=(Y2-Y1)*C1-(Z2-Z1)*B1
T2=(Z2-Z1)*A1-(X2-X1)*C1
T3=(X2-X1)*B1-(Y2-Y1)*A1
D=A3*B3+C3*C3
IF(D.GT.1.E-6) GOTO 10

C PARALLEL LINES

DIST=SQRT(T1*T1+T2*T2+T3*T3)
RETURN
C NOT PARALLEL

10 DIST=ABS(A2*T1+B2*T2+C2*T3)/SQRT(D)
RETURN
END
APPENDIX 18

SUBROUTINE PLTST2

SUBROUTINE PLTST2(XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, 
& ZM3, XM4, YM4, ZM4, XM12, JOP, XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, 
& YM3, ZM3, XM4, YM4, ZM4, IN12, NPT, NINT, IACM, 
& IACN, ZMN)
C

THIS ROUTINE ONLY WORKS WHEN DISTANCES BETWEEN POINTS 1 AND 2
C ARE GREATER THAN ZERO.
DIMENSION SX(125), SY(125), SY1(125), SY2(125), SY3(125), SX(125)
DIMENSION SW(125)
DIMENSION SYR(125)
COMPLEX ZVSD1, ZVSD2
COMPLEX Z, ZMN, GAM, ETA, RIT, EGD
COMMON /A/ %V, FI, A, Q, GAM, ETA, X
IWE=0
JSDTCH=0
MPT=NPT
MINT=NINT
KINT=0
IF (NPT.GT.1) GO TO 15
X1=(XM1+XM2)/2.0
Y1=(YM1+YM2)/2.0
Z1=(ZM1+ZM2)/2.0
X2=(XM3+XM4)/2.0
Y2=(YM3+YM4)/2.0
Z2=(ZM3+ZM4)/2.0
D12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
GO TO 79
15 CONTINUE
ZMN=(0., 0.)
DM43=SQR((XM3-XM4)**2+(YM3-YM4)**2+(ZM3-ZM4)**2)
DM12=SQR((XM2-XM1)**2+(YM2-YM1)**2+(ZM2-ZM1)**2)
DM23=SQR((XM3-XM2)**2+(YM3-YM2)**2+(ZM3-ZM2)**2)
DM14=SQR((XM4-XM1)**2+(YM4-YM1)**2+(ZM4-ZM1)**2)
DMMX=AMAX1(DM14, DM23)
62 CONTINUE
NPL=MPT+1
DXM=(XM2-XM1)/MPT
DYM=(YM2-YM1)/MPT
DZM=(ZM2-ZM1)/MPT
DXM=(XM3-XM4)/MPT
DYE= (YM3-YM4)/MPT
DZE= (ZM3-ZM4)/MPT
DHT= DM12/MPT
FM= 0.
DO 10 IDO= 1, NP1
I= IDO
WT= 3.0+(-1)**I
IF (I.EQ.1.OR. I.EQ. NP1) WT= WT/2.0
SK(I) = (I-1)*DHT
X1= XM1+DXT*(I-1)
Y1= YM1+DYT*(I-1)
Z1= ZM1+DZT*(I-1)
X2= XM4+DXT*(I-1)
Y2= YM4+DYT*(I-1)
Z2= ZM4+DZT*(I-1)
D12= SQRT( (X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
IF (D12.GT.Q) GO TO 49
Z= 1.0
GO TO 200
49 CONTINUE
CONTINUE
IF (JOP.EQ.1) CALL ZWTE2 (X1, Y1, Z1, X2, Y2, Z2, D12, 1,
1 XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3,
2 XM4, YM4, ZM4, XM1, MINT, IACN, Z, KINT)
IF (NPT.GT.1.OR.JOP.NE.1) GO TO 104
XNC=(XM1+XM2+XM3+XM4)/4.0
YNC=(YM1+YM2+YM3+YM4)/4.0
ZN= (ZM1+ZM2+ZM3+ZM4)/4.0
DMINC= SQRT ((XNC-XM1)**2+(YNC-YN1)**2+(ZNC-ZM1)**2)
DM4NC= SQRT ((XNC-XM4)**2+(YNC-YN4)**2+(ZNC-ZM4)**2)
D1INC= SQRT ((XNC-X1)**2+(YNC-Y1)**2+(ZNC-Z1)**2)
D2INC= SQRT ((XNC-X2)**2+(YNC-Y2)**2+(ZNC-Z2)**2)
D3INC= SQRT ((XNC-X3)**2+(YNC-Y3)**2+(ZNC-Z3)**2)
DAV1= 0.5*(D1INC+D4INC)
DAV2= 0.5*(D2INC+D3INC)
EDG= 1.0/(CEXP(GAM*DAVG)*DAVG)
RIT= 1.0/(CEXP(GAM*DAV1)*DAV1)+4.0*EDG+1.0/(CEXP(GAM*DAV2)*DAV2)
ZMN= Z*RIT/(6.0*EDG)
RETURN
104 CONTINUE
IF (JOP.NE.3) GO TO 206
DS= SQRT ((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
CAS= (X2-X1)/DS
CBS= (Y2-Y1)/DS
CGS= (Z2-Z1)/DS
S21= (XM1-X1)*CAS+(YM1-Y1)*CBS+(ZM1-Z1)*CGS
S22= (XM2-X2)*CAS+(YM2-Y2)*CBS+(ZM2-Z2)*CGS
RHO1= SQRT ((XM1-X1-S21*CAS)**2+(YN1-Y1-S21*CBS)**2
S+ (ZM1-Z1-S21*CGS)**2)
RHO2= SQRT ( (XM2-X2-S22*CAS)**2+(YN2-Y2-S22*CBS)**2
S+ (ZM2-Z2-S22*CGS)**2)
RHO= AMIN1 (RHO1, RHO2)
SIGN= 1.0
IF (RHOM.GT.0.1*DHT) GO TO 160
XCL= XM1
YCL= YM2
110 CONTINUE
X1=XM1+(I-1) *DXT +SIGN*0.5* DXT
Y1=YM1+(I-1) *DYT +SIGN*0.5* DYT
Z1=ZM1+(I-1) *DZT +SIGN*0.5* DZT
X2=XM4+(I-1) *DXE +SIGN*0.5* DXE
Y2=YM4+(I-1) *DYE +SIGN*0.5* DYE
Z2=ZM4+(I-1) *DZE +SIGN*0.5* DZE
CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IML2,XN1,YN1,ZN1,
6XN2,YN2,ZN2,INL2,ZVSD2,0)
SIGN=1.0
110 CONTINUE
X1=XM1+(I-1) *DXT+SIGN*0.5*DXT
Y1=YM1+(I-1) *DYT+SIGN*0.5* DYT
Z1=ZM1+(I-1) *DZT+SIGN*0.5* DZT
X2=XM4+(I-1) *DXE+SIGN*0.5* DXE
Y2=YM4+(I-1) *DYE+SIGN*0.5* DYE
Z2=ZM4+(I-1) *DZE+SIGN*0.5* DZE
CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IML2,XN1,YN1,ZN1,
6XN2,YN2,ZN2,INL2,ZVSD2,0)
X2=REAL(ZVSD1)
X2=AIMAG(ZVSD1)
X22=AIMAG(ZVSD2)
C2=(X22-X21)/ALOG(2.0)
C1=X21-C2*ALOG(DHT/2.)
AAZ=2.*C1+2.*C2*(ALOG(DHT)-1.)-XZ2
SY(I)=AAZ
SYR(I)=REAL(Z)
GO TO 10
160 CONTINUE
CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IML2,XN1,YN1,ZN1,
2XN2,YN2,ZN2,INL2,ZVSD2,0)
ZVSD2=Z
200 CONTINUE
SY(I)=AIMAG(Z)
SYR(I)=REAL(Z)
206 ZMN=ZMN+Z*WT
FM=FM+WT
10 CONTINUE
ZMN=ZMN/FM
IF(JOP.NE.3)RETURN
ZVSR=SPLINT(SX,SYR,NNP1,0,DM12,SYL,SY2,SY3,SE,IND)
6/DM12
ZVS=SPLINT(SX,SY,NNP1,0,DM12,SY1,SY2,SY3,SE,IND)
6/DM12
ZMN=CMPLX(ZVSR,ZVS)
RETURN
END
261
APPENDIX 19

SUBROUTINE ZWTPE

SUBROUTINE ZWTPE(XMI, YN1, Z41, XM2, Y142, ZM2, IN12, XN1, YN1, ZN1, XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, INTP, ZMN)

COMPLEX CGDM, SGDM, SGDNEGD, ETA, GAM, ZMN, P11, DUM
COMMON /A/ WV, PI, A, Q, GAM, ETA, XK

DM = SQRT((XMI - XM2) ** 2 + (YN1 - YM2) ** 2 + (ZMI - ZM2) ** 2)
EGD = CEXP (GAM * DM)
CGDM = (EGD + 1./EGD) / 2.
SGDM = (EGD - 1./EGD) / 2.

WN = SQRT((XN3 - XN2) ** 2 + (YN3 - YN2) ** 2 + (ZN3 - ZN2) ** 2)
DN = SQRT((XN2 - XN1) ** 2 + (YN2 - YN1) ** 2 + (ZN2 - ZN1) ** 2)
EGD = CEXP (GAM * DN)
SGDN = (EGD - 1./EGD) / 2.

C CHECK DIST BETWEEN TEST & EXP
D = SQRT(((XMI + XM2 - XN3 - XN1) ** 2 + (YM1 + YM2 - YN3 - YN1) ** 2 + (ZM1 + ZM2 - ZN3 - ZN1) ** 2) / 4.)
DD = (DM + SQRT(WN * WN + DN * DN)) / 1.8

NPLS = INTP
IF(DGT.DD) NPLS = MAX0(2*(INTP/11), 2)

DX = (XN3 - XN2) / NPLS
DY = (YN3 - YN2) / NPLS
DZ = (ZN3 - ZN2) / NPLS
NP1 = NPLS + 1
ZMN = (0., 0.)

DO 10 I = 1, NP1

W = 3 + (-1) ** I
IF(I.EQ.1 OR I.EQ.NP1) W = W / 2.

XI = XN1 + (I-1) * DX
YI = YN1 + (I-1) * DY
ZI = ZN1 + (I-1) * DZ
X2 = XN2 + (I-1) * DX
Y2 = YN2 + (I-1) * DY
Z2 = ZN2 + (I-1) * DZ

CALL ZGSMM(XMI, YN1, Z41, XM2, Y142, ZM2, XI, Y1, Z1, X2, Y2, Z2, Q, DM, CGDM, SGDM, DN, SGDNEGD, P11)

CONTINUE

ZMN = ZMN * W / (3. * NPLS) * IN12 * IM12 * XK / (2. * SIN(XK * WN / 2.))

RETURN

END
APPENDIX 20

SUBROUTINE ZWTDE

SUBROUTINE ZWTDE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
& XNO,YNO,ZNO,XN1,YN1,ZN1,XN2,YN2,ZN2,INTD,B,ZMN)

COMPLEX ZMN,P11,DUM,EGD,ETA,GAM,SGDM,CGDM,SGDN
REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
COMMON /A/ WV,PI,A,Q,GAM,ETA,XK

C DEFINE PRIME COORDS ON PLANE OF DISK
DNR=SQRT((XN2-XNO)**2+(YN2-YNO)**2+(ZN2-ZN0)**2)
L1=(XN2-XNO)/DNR
M1=(YN2-YNO)/DNR
N1=(ZN2-ZN0)/DNR
L3=(YN1-YNO)*((XN2-ZN0)-(YN2-YNO)*(ZM1-ZM0)
M3=(ZN1-ZN0)*((XN2-XN0)-(ZN2-ZN0)*(XN1-XN0)
N3=(XN1-XN0)*(YN2-YNO)-(XN2-XN0)*(YN1-YNO)
RN=SQRT(L3*L3+M3*M3+N3*N3)
L3=L3/RN
M3=M3/RN
N3=N3/RN
L2=M3*N1-N3*M1
M2=L1*N3-N1*L3
N2=L3*M1-L1*M3

C DEFINE PARAMETERS FOR GGS
DM=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
DN=B-A
EGD=CEXP(GAM*DM)
SGDM=(EGD-1./EGD)/2.
CGDM=(EGD+1 ./EGD)/2.
EGD=CEXP(GAM*DN)
SGD=(EGD-1./EGD)/2.

C CHECK DIST BETWEEN TEST AND EXP
D=SQRT(((XM1+XM2)/2.-XNO)**2+((YM1+YM2)/2.-YNO)**2
& +((ZM1+ZM2)/2.-ZNO)**2)
DD=B+DM/1.8
NDLS=INTD
IF(D.GT.DD) NDLS=2*(INTD/6)
IF(NDLS.LT.2)NDLS=2
NP1=NDLS+1
DPH=2. *PI/NDLS
ZN=(0.,0.)
DO 10 I=1,NP1
W=3+(-1)**I

263
IF (I.EQ.1 .OR. I.EQ.NP1) W=W/2.
P=PI/(I-I)*DPH
XP1=A*COS(PH)
YP1=A*SIN(PH)
XP2=B*COS(PH)
YP2=B*SIN(PH)
C TRANSFORM COORDS TO ORIGINAL SYSTEM
X1=L1*XP1+L2*YP1+XNO
Y1=M1*XP1+M2*YP1+YNO
Z1=N1*XP1+N2*YP1+ZNO
X2=L1*XP2+L2*YP2+XNO
Y2=M1*XP2+M2*YP2+YNO
Z2=N1*XP2+N2*YP2+ZNO
CALL ZGSMM(XM1, YM1, ZM1, XM2, YM2, ZM2, X1, Y1, Z1, X2, Y2, Z2, & Q, DM, CGDM, SGDM, DN, SGDN, P1, P2)
ZMN=ZM1-P*P1
10 CONTINUE
ZMN=ZM*N*IM*DPH/(6.*PI)
RETURN
END

264
APPENDIX 21
SUBROUTINE ZWTWE

SUBROUTINE ZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IN12,IN12,ZMN,IWW)
COMPLEX ZMN,P11.DUM,E,SGDM,CGDM,SGDN,ETA,GAM
COMMON /A/ W,V,F,A,Q,GAM,ETA,XK
DM=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
DN=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
EGD=CEXP(GAM*DM)
SGDM=(EGD-1./EGD)/2.
CGDM=(EGD+1./EGD)/2.
SGDN=(EGD-1./EGD)/2.
AA=Q
IF(IWW.EQ.1) AA=A
CALL ZGSMM(XM1,YM1,ZM1,XM2,YM2,ZM2,IN12,IN12,ZMN,IN12,IN12,ZMN,AA,DM,CGDM,SGDM,SN,SGDN,P11)
ZMN=P11*IN12*IN12
RETURN
END
APPENDIX 22

SUBROUTINE ZWTPE2

SUBROUTINE ZWTPE2(XM1, YM1, ZM1, XM2, YM2, ZM2, DM,
                XM4, YM4, ZM4, IN12PNPLS, IACN, ZMNKINT)

DIMENSION SX(125), SY(125), SY1(125), SY2(125), SY3(125)
DIMENSION SW(125), SE(125), SYR(125)
COMPLEX ZMN, P11, DUM, EGD, ETA, GAM, SGDM, CGDM, SGDN, ZPI1
COMPLEX ZVSD1, ZVSD2, ZVSD3, RIT, YP11
COMMON /A/ WV, PI, A, Q, GAM, ETA, XK
CAM=(XM2-XM1)/DM
CBM=(YM2-YM1)/DM
CGM=(2M2-ZM1)/DM
CDKS=COS(XK*DM)
SDKS=SIN(CXK*DM)
WNT=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
WNE=SQRT((XN3-XN4)**2+(YN3-YN4)**2+(ZN3-ZN4)**2)
CAST=(XN2-XN1)/WNT
CBST=(YN2-YN1)/WNT
CGST=(ZN2-ZN1)/WNT
CASE=(XN3-XN4)/WNE
CBSE=(YN3-YN4)/WNE
CGSE=(ZN3-ZN4)/WNE
1=0
2=1
NP1=1
HHT=WNT/2.0
HHE=WNE/2.0
Hf=FHT
ILS=1
ISM=C
i=1
GO TO 31
31 CONTINUE
DN14=SQRT((XN4-XN1)**2+(YN4-YN1)**2+(ZN4-ZN1)**2)
DN23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
DNMX=AMAX1(DN14, DN23)
HHT=WNT/NPLS
HHE=WNE/NPLS
Hf=FHT
NP1=NPLS+1
DD=FHT
SLIMT=WNT
IF(NPI.GT.90)WRITE(11,16)NPI
16 FORMAT(/5X,***"WARNING NPI IN SUB. ZWTHE IS TOO LARGE:',
   & NPT=',15,***/')
ZMN=(.0,0)
111=0
DO 12 I=1,NPI
112 ILS=I-1
GXH=HHT
GYH=HHE
SX(I)=ILS*HH
IF(111.GT.0)GO TO 8
119 X1=XN1+(ILS*HHT)*CAST
Y1=YN1+(ILS*HHT)*CBST
Z1=ZN1+(ILS*HHT)*CGST
X2=XN4+(ILS*HHE)*CASE
Y2=YN4+(ILS*HHE)*CBSE
Z2=ZN4+(ILS*HHE)*CGSE
DN12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
DN12=DN12
IF(DN12.GT.Q)GO TO 215
P11=(.0,.0)
GO TO 8
215 CONTINUE
DDMN=0.0
IF(NPI.NE.1)GO TO 22
DMCNC1=SQRT((0.5*(XM1+XM2-XN1-XN4))**2+(0.5*(YM1+YM2-YN1-YN4))
   & **2+(0.5*(ZM1+ZM2-ZN1-ZN4))**2)
DMCNC2=SQRT((0.5*(XM1+XM2-XN2-XN3))**2+(0.5*(YM1+YM2-YN2-YN3))
   & **2+(0.5*(ZM1+ZM2-ZN2-ZN3))**2)
IF(DMCNC1.GT.DMCNC2)GO TO 25
DMCNC=DMCNC1
DM11=SQRT((XM1-XN1)**2+(YM1-YN1)**2+(ZM1-ZN1)**2)
DM12=SQRT((XM1-XN4)**2+(YM1-YN4)**2+(ZM1-ZN4)**2)
DM21=SQRT((XM2-XN1)**2+(YM2-YN1)**2+(ZM2-ZN1)**2)
DM22=SQRT((XM2-XN4)**2+(YM2-YN4)**2+(ZM2-ZN4)**2)
GO TO 26
25 CONTINUE
DMCNC=DMCNC2
DM11=SQRT((XM1-XN2)**2+(YM1-YN2)**2+(ZM1-ZN2)**2)
DM12=SQRT((XM1-XN3)**2+(YM1-YN3)**2+(ZM1-ZN3)**2)
DM21=SQRT((XM2-XN2)**2+(YM2-YN2)**2+(ZM2-ZN2)**2)
DM22=SQRT((XM2-XN3)**2+(YM2-YN3)**2+(ZM2-ZN3)**2)
26 DDMN=AMIN1(DMCNC,DM11,DM12,DM21,DM22)
DM1NC=SQRT((XM1-0.5*(X1+X2))**2+(YM1-0.5*(Y1+Y2))
   & **2+(ZM1-0.5*(Z1+Z2))**2)
DM2NC=SQRT((XM2-0.5*(X1+X2))**2+(YM2-0.5*(Y1+Y2))
   & **2+(ZM2-0.5*(Z1+Z2))**2)
DAV1=0.5*(DM1NC+DM2NC)
DM1NA=SQRT((XM1-0.5*(XM1+XM2))**2+(YM1-0.5*(YM1+YM2))
   & **2+(ZM1-0.5*(ZM1+ZM2))**2)
DM2NA=SQRT((XM2-0.5*(XM1+XM2))**2+(YM2-0.5*(YM1+YM2))
   & **2+(ZM2-0.5*(ZM1+ZM2))**2)
DAV1=0.5*(DM1NA+DM2NA)
DM1NB=SQRT((XM1-0.5*(XM1+XM2))**2+(YM1-0.5*(YM1+YM2))
   & **2+(ZM1-0.5*(ZM1+ZM2))**2)
DM2NB=SQRT((XM2-0.5*(XM1+XM2))**2+(YM2-0.5*(YM1+YM2))
   & **2+(ZM2-0.5*(ZM1+ZM2))**2)
DAV2=0.5*(DM1NB+DM2NB)

267
GO TO 87
22 CONTINUE
DM1N1=SQRT((X1M-X1)**2+(Y1M-Y1)**2+(Z1M-Z1)**2)
DM1N2=SQRT((X1M-X2)**2+(Y1M-Y2)**2+(Z1M-Z2)**2)
DM2N1=SQRT((X2M-X1)**2+(Y2M-Y1)**2+(Z2M-Z1)**2)
DM2N2=SQRT((X2M-X2)**2+(Y2M-Y2)**2+(Z2M-Z2)**2)
DMCN=SQRT((0.5*(XM2+XM1-X2-X1))**2+(0.5*(YM2+YM1-Y2-Y1))**2+(0.5*(ZM2+ZM1-Z2-Z1))**2)
DDMN=DMIN1(DM1N1,DM1N2,DM2N1,DM2N2)
CAN=(X2-X1)/DN12
CBN=(Y2-Y1)/DN12
CGN=(Z2-Z1)/DN12
DPM1=ABS(CAM*CAN+CBM*CBN+CGM*CGN)
85 CONTINUE
C CHECK DIST BETWEEN TEST AND EXP
INT=0
IF((KINT.EQ.1))GO TO 86
IF(DMCN.GT.0.5*(DM+DN12)+0.15*WV)GO TO 710 89
SZDA=(X1M-X1M)*CAM+(Y1M-Y1M)*CBM+(Z1M-Z1M)*CGM
XZDA=X1M-X1M-SZDA*CAM
YZDA=Y1M-Y1M-SZDA*CBM
ZZDA=Z1M-Z1M-SZDA*CGM
RHODA=SQRT((X1M-XZDA+YZDA*Y1M+Z1M-ZZDA)
SZDB=(X2M-X1M)*CAM+(Y2M-Y1M)*CBM+(Z2M-Z1M)*CGM
SZC1=(X1M-X1M)*CAN+(YM1-Y1M)*CBN+(Z1M-Z1M)*CGN
SZC2=(XM1-X1M)*CAN+(YM1-Y1M)*CBN+(Z1M-Z1M)*CGM
IF(DPM1.LT.0.999)GO TO 156
IF((D1.12.DT.DM))GO TO 145
IF((SZDA.GT.0.0.AND.SZDA.LT.DM))GO TO 155
IF((SZDB.GT.0.0.AND.SZDB.LT.DM))GO TO 156
145 IF((SZC1.GT.0.0.AND.SZC1.LT.DM))GO TO 155
IF((SZC2.GT.0.0.AND.SZC2.LT.DM))GO TO 156
155 IF((RHODA.GT.0.12*WV))GO TO 86
IF((RHODA.LT.0.20*WV))GO TO 87
GO TO 89
156 CONTINUE
XZDB=X2M-X1M-SZDB*CAM
YZDB=Y2M-Y1M-SZDB*CBM
ZZDB=Z2M-Z1M-SZDB*CGM
RHODB=SQRT((XZDB*XZDB+YZDB*YZDB+ZZDB*ZZDB)
XZC1=X1M-X1M-SZC1*CAM
YZC1=Y1M-Y1M-SZC1*CBM
ZZC1=Z1M-Z1M-SZC1*CGM
RHOC1=SQRT((XZC1*XZC1+YZC1*YZC1+ZZC1*ZZC1)
XLC1=X1M-X1M-SZC1*CAM
YLC1=Y1M-Y1M-SZC1*CBM
ZLC1=Z1M-Z1M-SZC1*CGM
RHOC2=SQRT((XLC2*XLC2+YLC2*YLC2+ZLC2*ZLC2)
DTDADB=XZDA*XZDB+YZDA*YZDB+ZZDA*ZZDB
DTC1C2=XZC1*XZC2+YZC1*YZC2+ZZC1*ZZC2
IF((DTDADB.GT.0.0.OR.DTC1C2.GT.0.0))GO TO 105
RHOMIN=DMIN1(RHODA,RHODB,RHOC1,RHOC2)
IF((RHOMIN.LE.0.05*WV))GO TO 86
CAP=CBN*CGM-CGM*CBM
CSP=CAM*CAM+CBN*CBM
CGP=CAM*CBN-CBM*CAM
268
DCP=SQRT(CAP*CAP+CBP*CBP+CGP*CGP)
DPP=ABS((XM1-X1)*CAP+(YM1-Y1)*CBP+(ZM1-Z1)*CGP)/DCP
IF(DPP.LE.0.05*WV)GO TO 86
IF(DPP.LE.0.1*WV)GO TO 87
INT=2
GO TO 86
105 CONTINUE
IF(DTDADB.GT.0.0)GO TO 107
RHONP=AMN1(RHOC1,RHOC2)
GO TO 109
107 IF(DTC12.GT.0.0)GO TO 108
RHONP=AMN1(RHODA,RHODB)
GO TO 109
108 RHONP=DDMN
109 CONTINUE
IF(RHONP.LE.0.05*WV)GO TO 86
IF(RHONP.LE.0.1*WV)GO TO 87
89 INT=2
GO TO 86
87 INT=4
86 CONTINUE
SDKT=SIN(XK*DN12)
SZM1=(X1-XM1)*CAG+(Y1-Y1)*CBM+(Z1-ZM1)*CGM
CALL GGS1(XM1,YM1,ZM1,XM2,YM2,ZM2,X1,Y1,Z1,X2,Y2,Z2,XK,2
DM,DKS,DKS,DKS,DKT,INT,ETA,GAM,P11,DUM,DUM,DUM)
EGD=1.0/(CEXP(GAM*DAVG)*DAVG)
RIT=1.0/(CEXP(GAM*DAV1)*DAV1)+4.0*EGD+1.0/(CEXP(GAM*DAV2)*DAV2)
ZMN=P11*IM12*IN12*RIT/(6.0*EGD)
RETURN
97 CONTINUE
ZVSD2=P11
8 CONTINUE
SYR(I)=REAL(P11)
SYI(I)=AIMAG(P11)
111=0
12 CONTINUE
ZVSR=SPLINT(SX,SYR,NP1.,0,SLIMIT,SY1,SY2,SY3,SE,IND)
&/SLIMIT
ZVST=SPLINT(SX,SY,NP1.,0,SLIMIT,SY1,SY2,SY3,SE,IND)
&/(SLIMIT)
ZMN=IM12*IN12*CMPLX(ZVSR,ZVST)
RETURN
END
APPENDIX 23
SUBROUTINE DSKTST

SUBROUTINEDSKTST(XM0,YM0,ZM0,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,
&XM1,YM1,ZM1,XN2,YM2,ZN2,XN3,YM3,ZN3,N12,NINT,INTD,
&BM,BN,ZMN)

COMPLEX Z,ZMN,GAM,ETA
REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
COMMON /A/ WV,PI,A,Q,GAM,ETA,XX
DNR=SQRT((XM2-XM0)**2+(YM2-YM0)**2+(ZM2-ZM0)**2)
L1=(XM2-XM0)/DNR
M1=(YM2-YM0)/DNR
N1=(ZM2-ZM0)/DNR
L3=(YM1-YM0)*(ZM2-ZM0)-(YM2-YM0)*ZM1-ZM0)
M3=(ZM1-ZM0)*(XM2-XM0)-(ZM2-ZM0)*(XM1-XM0)
N3=(XM1-XM0)*(YM2-YM0)-(XM2-XM0)*(YM1-YM0)
RN=SQRT(L3*L3+M3*M3+N3*N3)
L3=L3/RN
M3=M3/RN
N3=N3/RN
L2=L3*N1-N3*M1
M2=L1*N3-M1*L3
N2=L3*M1-L1*M3
C CHECK DISTANCE BETWEEN TEST AND EXP
IF(JOP.NE.3)GO TO 100
XM3=XM2
YM3=YM2
ZN3=ZN2
100 CONTINUE
D=SQRT(((XM1+XM3)/2.-XM0)**2+(YM1+YM3)/2.-YM0)**2
&+(ZM1+ZN3)**2+2*INTD/6)
NDT=INTD
IF(D.GT.5*WV)NDT=2*(INTD/6)
IF(NDT.LT.2)NDT=2
DPH=2.*PI/NDT
ZMN=0.,0.
DO 10 I=1,NDT
W=3+(-1)**I
PH=(I-1)*DPH
XP1=A*COS(PH)
YP1=A*SIN(PH)
XP2=BM*COS(PH)
YP2=BM*SIN(PH)
10 CONTINUE
C TRANSFORM COORDS TO ORIGINAL SYSTEM

\[
\begin{align*}
X1 &= L1 \cdot X1P1 + L2 \cdot YP1 + XM0 \\
Y1 &= M1 \cdot X1P1 + M2 \cdot YP1 + YM0 \\
Z1 &= N1 \cdot X1P1 + N2 \cdot YP1 + ZM0 \\
X2 &= L1 \cdot X2P2 + L2 \cdot YP2 + XM0 \\
Y2 &= M1 \cdot X2P2 + M2 \cdot YP2 + YM0 \\
Z2 &= N1 \cdot X2P2 + N2 \cdot YP2 + ZM0 \\
\end{align*}
\]

IF (JOP.EQ.1) CALL ZWTPE (X1, Y1, Z1, X2, Z2, -1, XN1, YN1, ZN1, \\
& XN2, YN2, ZN2, XN3, YN3, ZN3, IN12, NINT, 2)

IF (JOP.EQ.2) CALL ZWTDE (X1, Y1, Z1, X2, Z2, -1, XN1, YN1, ZN1, \\
& XN2, YN2, ZN2, XN3, YN3, ZN3, NINT, BN, 2)

IF (JOP.EQ.3) CALL ZWTWE (X1, Y1, Z1, X2, Z2, -1, XN1, YN1, ZN1, \\
& XN2, YN2, ZN2, IN12, Z, 0)

ZMN = ZMN + Z*W

10 CONTINUE

ZMN = ZMN / (3. * NDT)

RETURN

END
APPENDIX 24

SUBROUTINE DSKTS2

SUBROUTINEDSKTS2(XM0, YM0, ZM0, XM1, YM1, ZM1, XM2, YM2, ZM2, JOP, NDT, BM, BN, ZMN)

REAL XM1, YM1, ZM1, XM2, YM2, ZM2
COMMON /A/ WV, PI, A, Q, GAM, ETA, XK

L1 = (XM2-XM0)/DNR
M1 = (YM2-YM0)/DNR
N1 = (ZM2-ZM0)/DNR

L3 = (YM1-YM0)*(ZM2-ZM0) - (YM2-YM0)*(ZM1-ZM0)
M3 = (ZM1-ZM0)*(XM2-XM0) - (ZM2-ZM0)*(XM1-XM0)
N3 = (XM1-XM0)*(YM2-YM0) - (XM2-XM0)*(YM1-YM0)
RN = SQRT(L3*L3 + M3*M3 + N3*N3)

L3 = RN
M3 = RN
N3 = RN

L2 = M3*N1 - N3*M1
M2 = L1*N3 - N1*L3
N2 = M1*L3 - M3*L1

PHI = 2.*PI/NDT

IF(JOP.NE.1)GO TO 5

DMR1 = SQRT(XM1-XM0)**2 + (YM1-YM0)**2 + (ZM1-ZM0)**2
DMR2 = SQRT(XM2-XM0)**2 + (YM2-YM0)**2 + (ZM2-ZM0)**2
STM = SQRT((YM1-YM0)*(ZM2-ZM0) - (YM2-YM0)*(ZM1-ZM0))**2 +
2*((XM1-XM0)*(ZM1-ZM0) - (XM1-XM0)*(ZM2-ZM0))**2 +
3*((XM1-XM0)*(YM2-YM0) - (XM2-XM0)*(YM1-YM0))**2 /
(DMR1*DMR2)

STM = SQRT((YM2-YM0)*(ZM3-ZM2) - (YM3-YM2)*(ZM2-ZM1))**2 +
2*((XM2-XM1)*(ZM2-ZM1) - (XM2-XM1)*(ZM3-ZM2))**2 +
3*((XM2-XM1)*(YN3-YN2) - (XM3-XN2)*(YN2-YN1))**2 /
(DN12*DN23)

CAS = ((YN2-YN1)*(ZN3-ZN2) - (YN3-ZN2)*(YN2-YN1))*((YN1-YM0) -
2*(ZM2-ZM0) - (YM2-YM0)*(ZM1-ZM0)) /
(DMR1*DMR2*DN12*DN23*STM*STM)

CAS = ((YN2-YN1)*(ZN3-ZN2) - (YN3-ZN2)*(YN2-YN1))*((YN1-YM0) -
2*(ZM2-ZM0) - (YM2-YM0)*(ZM1-ZM0)) /
(DMR1*DMR2*DN12*DN23*STM*STM)

272
CGS = ((XN2-XN1) *(YN3-YN2) - (XN3-XN2) *(YN2-YN1)) * ((XM1-XMO) * 2 *(YN2-YMO) - (XM2-XMO) *(YN1-YMO)) / (DMR1*DMR2*DM12*DN12*STM*STM)
IF (ABS(CAS+CBS+CGS) .LT. 0.999) GO TO 5
DK = (XM1-XN1) *CAS+ (YN1-YN1) *CBS+ (ZM1-ZMO) *CGS
IF (ABS(DK) .LT. Q) IQ = 1
5 CONTINUE
ZMN = (-0.,0)
14 CONTINUE
DO 10 IDO = 1, NDT
I = IDO
18 CONTINUE
W = 3. + (-1)**I
PH = (I-1) *DPH
XP1 = A*COS(PH)
YP1 = A*SIN(PH)
XP2 = BM*COS(PH)
YP2 = BM*SIN(PH)
C TRANSFORM COORDS TO ORIGINAL SYSTEM
IF (IQEQ.0) GO TO 8
X1 = L1*XPI1 + L2*YP1 + L3*Q + XM0
Y1 = M1*XP1 + M2*YP1 + M3*Q + YM0
Z1 = N1*XP1 + N2*YP1 + N3*Q + ZM0
X2 = L1*XP2 + L2*YP2 + L3*Q + XM0
Y2 = M1*XP2 + M2*YP2 + M3*Q + YM0
Z2 = N1*XP2 + N2*YP2 + N3*Q + ZM0
GO TO 9
8 CONTINUE
X1 = L1*XPI1 + L2*YP1 + XM0
Y1 = M1*XP1 + M2*YP1 + YM0
Z1 = N1*XP1 + N2*YP1 + ZM0
X2 = L1*XP2 + L2*YP2 + XM0
Y2 = M1*XP2 + M2*YP2 + YM0
Z2 = N1*XP2 + N2*YP2 + ZM0
9 CONTINUE
20 CONTINUE
24 CONTINUE
ISDTCH = 0
D12 = SQRT((X2-X1)**2 + (Y2-Y1)**2 + (Z2-Z1)**2)
KINT = 0
IPPR = 0
IF (JOP = 1) CALL ZWTPE2(X1, Y1, Z1, X2, Y2, Z2, D12, 2, 2-1, XM1, YM1, ZM1, 3, XM2, YM2, ZM2, XM3, YM3, ZM3, XM4, YM4, ZM4, IN12, 4, NINT, IACN, Z, KINT)
IF (JOP = 2) CALL ZWTDE(X1, Y1, Z1, X2, Y2, Z2, -1, XM1, YM1, ZM1, 5, XM2, YM2, ZM2, XM3, YM3, ZM3, NINT, BN, Z)
IF (JOP = 3) CALL ZWTE(X1, Y1, Z1, X2, Y2, Z2, -1, XM1, YM1, ZM1, 6, XM2, YM2, ZM2, IN12, 2, 0)
IF (IPGD = 4) WRITE (6, *) I, Z
2MN = 2MN + Z
10 CONTINUE
2MN = 2MN / (3.*NDT)
RETURN
END
SUBROUTINE ZATAT2(B,H,Z,NL,ZS,ALFD)
COMPLEX Z,CGDS,SGDS,SGDW,CGDW,ETA,GAMI1DUM,EGD
COMPLEX ZWW,ZWD,ZIJW,ZDD
COMMON /A/ WV,PI,A,Q,GAM,ETA,XR
C COMPUTE D/D,D/W,W/D
DS=B-A
EGD=CEXP(GAM*DS)
CGDS=(EGD+1./EGD)/2.
SGDS=(EGD-1./EGD)/2.
EGD=CEXP(GAM*H)
SGDW=(EGD-1./EGD)/2.
CGDW=(EGD+1./EGD)/2.
N=2*((NL+1)/2)
DPH=2.0*PI/N
ALF=ALFD*PI/180.0
ZDD=(0.,0.)
ZDW=(0.,0.)
ZWD=(0.,0.)
ZWW=(0.,0.)
DO 10 I=1,N
PH=I-.5)*DPH
CPSI=COS(PH)
CALL GGMNL(A,B,A,B,Q,CGDS,SGDS,CPSI,ETA,GAM, & P11,DUM,DUM,DUM)
ZDD=ZDD+P11
XA=A*COS(PH)
YA=A*SIN(PH)
ZA=0.0
XB=B*COS(PH)
YB=B*SIN(PH)
ZB=0.0
X1=A
Y1=0.0
Z1=0.0
X2=A
Y2=H*SIN(ALF)
Z2=H*COS(ALF)
CALLGGSI(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,Q,DS,CGDS, & SGDS,H,SGDW,0.,ETA,GAM,P11,DUM,DUM,DUM)
ZDW=ZDW-P11
CALL GGS1(X1,Y1,Z1,X2,Y2,Z2,XA,YA,ZA,XB,YB,ZB,Q,H,CGDW,
2SGDW,DS,SGDS,0,ETA,GAM,P11,DUM,DUM,DUM)
ZWD=ZWD-P11
D=A*SQR(2.-2.*COS(PH))
CALL GMML(0.,H,0.,H,0.,0.,CGDW,SGDW,SGDW,1.,ETA,GAM,
&P11,DUM,DUM,DUM)
10 CONTINUE
ZDD=ZDD/N
ZDW=ZDW/N
ZWW=ZWW/N
Z=Z+(SGDW*CGDW-GAM*H)*Z/4*A**2
RETURN
END
SUBROUTINE PDPZ

SUBROUTINE PDPZ:XMO,YMO,ZMO,NAT,XN1,YN1,ZN1,XN2,YN2,ZN2,
& XN3,YN3,ZN3,INTP,ERVSR,IAT,RMIN,DR,ZMN,DIST
COMPLEX ERVSR(IAT,400),ZMN,GAM,ETA,ER
COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
D=SQRT(((XN1+XN3)/2.-XMO)**2+((YN1+YN3)/2.-YMO)**2
&+((ZN1+ZN3)/2.-ZMO)**2)
NPE=INTP
IF(D.GT. .5*WV) NPE=2*(NPE/6)
IF(NPE.LT.2) NPE=2
DX12=(XN2-XN1)/NPE
DY12=(YN2-YN1)/NPE
DZ12=(ZN2-ZN1)/NPE
DX23=(XN3-XN2)/NPE
DY23=(YN3-YN2)/NPE
DZ23=(ZN3-ZN2)/NPE
WN=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
HN=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
DW=WN/NPE
DH=HN/NPE
NPE1=NPE+1
ZMN=(0.,0.)
DO 10 I=1,NPE1
W=W+(-1)**I
YP=W/NPE2.
V=V+(-1)**J
DO 10 J=1,NPE1
ZM=ZM+ER*W*V*SIN(XR*(EN/2.-ZP)) *COS(XR*YP)
10 CONTINUE
ZMN=-ZMN*DH*DW*XK/(18.*SIN(XK*HN)*SIN(XK*WN/2.))
RETURN
END

APPENDIX 26
SUBROUTINE PDPZ
SUBROUTINE PDPZ1

SUBROUTINE PDPZ1(XN0,YM0,ZN0,NAT,XN1,YN1,ZN1,XN2,YN2,ZN2,
& XN3,YN3,ZN3,XN4,YN4,ZN4,IACN,IN12,NPE,ERVSR,IAT,RMIN,
& DR,ZMN,DIST)

COMMON /A/ WV,PI,A,Q,GAM,ETA,ER
IPLOT=0
IF(IPLOT.LE.0)GO TO 15
CALL PLOTS(0,0,0)
CALL PLOT(4.25,5.0,-3)
CALL PLOT(XN1,YN1,3)
CALL PLOT(XN2,YN2,2)
CALL PLOT(XN3,YN3,2)
CALL PLOT(XN4,YN4,2)
CALL PLOT(XN1,YN1,2)
CALL SYMBOL(XM0,YM0,0.25,10.0,0,-1)
15 CONTINUE

D=SQR((((XN1+XN3)/2.-XMO)**2+((YN1+YN3)/2.-YMO)**2
& +((ZN1+ZN3)/2.-ZMO)**2)
& IF(D.GT.0.5*WV)NPE=2*(NPE/6)
& IF(NPE.LT.2)NPE=2
& IF(IPLOT.LE.0)GO TO 30
DN14=SQR((XN4-XN1)**2+(YN4-YN1)**2+(ZN4-ZN1)**2)
DN23=SQR((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
DMNX=AMAX1(DN14,DN23)
30 CONTINUE

DX12=(XN2-XN1)/NPE
DY12=(YN2-YN1)/NPE
DZ12=(ZN2-ZN1)/NPE
DX43=(XN3-XN4)/NPE
DY43=(YN3-YN4)/NPE
DZ43=(ZN3-ZN4)/NPE
WN=SQR((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
DZ=WN/NPE
NPE1=NPE+1
ZMN=(0.,0.,0.)
FWT=0.0
DO 10 I=1,NPE1
W=W*(I-1)**I
XA=XN1+(I-1)*DX12
YA=YN1+(I-1)*DY12
ZA=ZN1+(I-1)*DZ12
XB=XN4+(I-1)*DX43
YB=YN4+(I-1)*DY43
ZB=ZN4+(I-1)*DZ43
HN=SQRTR((XB-XA)**2+(YB-YA)**2+(ZB-ZA)**2)
SINHN=SIN(XH*HN)
DH=HN/NPE
DXAB=(XB-XA)/NPE
DYAB=(YB-YA)/NPE
DZAB=(ZB-ZA)/NPE
ZMNL=(0,0,0)
DO 9 J=1,NPE
V=3+(-1)**J
IF(J.EQ.1 .OR. J.EQ.NPE1) V=V/2.
ZP=HN/2.+(J-1)*DH
X=XA+(J-1)*DXAB
Y=YA+(J-1)*DYAB
Z=ZA+(J-1)*DZAB
IF(I.PLOT.GT.0)CALL SYMBOL(X,Y,0.1,11,0.0,-1)
RCAP=SQRTR((XMO-X)**2+(YMO-Y)**2+(ZMO-Z)**2+Q**2)
R=SQRTR(ABS(RCAP**2-DIST**DIST))
IF(R.LT.1.E-10) GO TO 9
N=ABS(R-RMIN)/DR+1
ER=ERVSR(NAT,N)+(ERVSR(NAT,N+1)-ERVSR(NAT,N))/DR*(R-RMIN-(N-1)*DR)
ER=ER*((X-XMO)*DXAB+(Y-YMO)*DYAB+(Z-ZMO)*DZAB)*NPE
/R*HN/
ZMNL=ZMNL+ER*V*SIN(XH*HN/2.)/WN
CONTINUE
9
CONTINUE
ZMN=ZMN+ZMNL*W/SINHN
FWT=FWT+W
10
CONTINUE
ZMN=(ZMN/2.)*INL2*DW/(3.*FWT)
IF(I.PLOT.GT.0) CALL PLOT(0.,0.,999)
RETURN
END
SUBROUTINE ERDSK

COMPLEX EX, J, EL, E2, ELR, E2R, C, CERL, ELL, ETA
J = (0.0, 1.0)
PI = 3.14159
D = B - A
XK = 2.0 * PI / WV
DK = D * XK
SDK = SIN (DK)
CDK = COS (DK)
C1 = 1.0 / (2.0 * PI)
C = - J * ETA / (4.0 * PI * SDK)
NP1 = NNPTS + 1
R = SQRT (X * X + Z * Z)
EX = (0.0, 0.0, 0.0)
1 = 2.0 * PI / NNPTS
DO 100 N = 1, NP1
PH = H / 2.0 + (N - 1) * H
CPH = COS (PH)
SPH = SIN (PH)
RH = X * CPH
XM = RH * CPH
YM = RH * SPH
DM = SQRT ((X - XM) ** 2 + (YM) ** 2 + Z ** 2)
XA = A * CPH
YA = A * SPH
R1 = SQRT ((X - XA) ** 2 + (YA) ** 2 + Z ** 2)
XB = B * CPH
YB = B * SPH
R2 = SQRT ((X - XB) ** 2 + (YB) ** 2 + Z ** 2)
CT1 = (RH - A) / R1
CT2 = (RH - B) / R2
DX = (X - XM) / DM
EL = CEXP (- J * XK * CPH)
E2 = CEXP (- J * XK * RPH)
ELR = EL / R1
E2R = E2 / R2
ERL = (J * EL * SDK + EL * CDK * CT1 - E2 * CT2) / DM
ELL = E2R - ELR * CDK
F = 3 + (-1) ** N
IF (N .EQ. 1 .OR. N .EQ. NP1) F = F / 2.
EX = EX + P * (ERL * DX + ELL * CPH)
CONTINUE
EX = C1 * CPH * EX / 3.
RETURN
END
SUBROUTINE ZGSMH

SUBROUTINE ZGSMN(XA,YA,ZA,XB,YB,ZB,XI,Y1,Z1,X2,Y2,Z2,A,D1oCGD.SGD.D2,SGD2,Zl2)

COMMON /A/ NV,PI,AAA,Q,GAM,ETA,XK

COMPLEX E3Rz12,P11,Pl2,P21,P22GAM,ETA

COMPLEX CGDI,SGDSGD2,FFF,FT,CST,T1,T2,T3,EJKT,FR


INT=0
IF(R.LT. .15*WV) GOTO 50

RK=R*XK
CAT=(X2-X1)/D2
CBT=(Y2-Y1)/D2
CGT=(Z2-Z1)/D2

CAS=(XB-XA)/D1
CBS=(YB-YA)/D1
CGS=(ZB-ZA)/D1

CAR=(X1+X2-XA-XB)/(2.*R)
CBA=(Y1+Y2-YA-YB)/(2.*R)
CGA=(Z1+Z2-ZA-ZB)/(2.*R)

SDT=CAT*CBS*CBT*CGT
CTH1=CAT*CAR*CBS*CGT
CTH2=(CAT*CAR+CBS*CBT)*CGT

SS1=1.-CTH1*CTH1
SS2=1.-CTH2*CTH2

AB=ABS(SS1*SS2)
IF(AB.LT. .8 .AND. R.LT. 1.2*(D1+D2)) GOTO 50

INT=2
GOTO 50

CONTINUE

SDK1=AIMAG(SGD1)
SDK2=AIMAG(SGD2)
CDK1=REAL(CGD1)
CDK2=SQRT(1.-SDK2*SDK2)
R2=R*KK
EJKR=CMPLX(COS(RK),-SIN(RK))
IF(AB.GT. .001) GOTO 30

Z12=FR*EJKR
GOTO 60

T1=(0.,1.)*CTH1*SDK1-CDK1
EJKT=CEXP((0.,1.)*RK*D1*CTH1/2.)

APPENDIX 29

SUBROUTINE ZGSMM

SUBROUTINE ZGSMM(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2)

COMMON /A/ NV,PI,AAA,Q,GAM,ETA,XK

COMPLEX E3Rz12,P11,Pl2,P21,P22GAM,ETA

COMPLEX CGDI,SGDSGD2,FFF,FT,CST,T1,T2,T3,EJKT,FR


INT=0
IF(R.LT. .15*WV) GOTO 50

RK=R*XK
CAT=(X2-X1)/D2
CBT=(Y2-Y1)/D2
CGT=(Z2-Z1)/D2

CAS=(XB-XA)/D1
CBS=(YB-YA)/D1
CGS=(ZB-ZA)/D1

CAR=(X1+X2-XA-XB)/(2.*R)
CBA=(Y1+Y2-YA-YB)/(2.*R)
CGA=(Z1+Z2-ZA-ZB)/(2.*R)

SDT=CAT*CBS*CBT*CGT
CTH1=CAT*CAR*CBS*CGT
CTH2=(CAT*CAR+CBS*CBT)*CGT

SS1=1.-CTH1*CTH1
SS2=1.-CTH2*CTH2

AB=ABS(SS1*SS2)
IF(AB.LT. .8 .AND. R.LT. 1.2*(D1+D2)) GOTO 50

INT=2
GOTO 50

CONTINUE

SDK1=AIMAG(SGD1)
SDK2=AIMAG(SGD2)
CDK1=REAL(CGD1)
CDK2=SQRT(1.-SDK2*SDK2)
R2=R*KK
EJKR=CMPLX(COS(RK),-SIN(RK))
IF(AB.GT. .001) GOTO 30

Z12=FR*EJKR
GOTO 60

T1=(0.,1.)*CTH1*SDK1-CDK1
EJKT=CEXP((0.,1.)*RK*D1*CTH1/2.)

280
T2=(0.,-1.)*((XK*D1/2.*T1+SDX1))*E.*JKT4(0.,1.)*XK*D1/(2.*EJET)
T2=T2/(2.*SDK1)
T3=(0.,1.)*((XK*XK*D1*SDK1/2.+XK*XK*D1*1/4.+XK**3*D1**3/8.*T1)
T3=T3*EJKT*(0.,1.)*XK**3*D1**3/(8.*EJKT)
T3=T3/(2.*SDK1)
T3=(0.,0.)
FR=(0.,-1.)*XK*T2+(0.,1.)*T3/(XK*R*R)
FR=FR*(0.,-60.)*EJKR/(R2)*FFF(SS2,CTH2,XK,D2,SDK2,CDK2)
PT=FFF(SS1,CTH1,XK,D1,SDK1,CDK1)*FFF(SS2,CTH2,XK,D2,SDK2,CDK2)
PT=PT*(SDK1+CTH1*CTH2)
CST=30.*XK*XK*EJKR/RK
$12=CST*PT*CMPLX(1./RK,1.-1./R2)
$12=$12-CTH2*FR
GOTO 60
50 CONTINUE
102 FORMAT(' INT/IN= ',I2)
CALL GGS1(XA,YA,ZA,XE,YD,ZB,X1,Y1,Z1,X2,Y2,Z2,
& A,D1,SDK1,D2,SDK2,INT,ETA,GAM,P11,P12,P21,P22)
$12=P11
60 RETURN
END
FUNCTION FFF(SS1,CTH1,XK,D1,SDK1,CDK1)
COMPLEX FFF,EJK
EJK=CEXP((0.,1.)*XK*CTH1*D1/2.)
IF (SS1.LT. .001) GOTO 10
FFF=EJK*(((0.,1.)*CTH1*SDK1-CDK1)+1./EJK
FFF=FFF/(XK*SS1*SDK1)
RETURN
10 FFF=(0.,1.)*((XK*D1/EJK-SDK1/EJK)/(2.*XK*SDK1))
RETURN
END
SUBROUTINE GGS1

SUBROUTINE GGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AN,CX2-X1) /DT
CB- (Y2-Y1) /DT
CG-CZ2-Z1) /DT
q CABU (XB-XA) /DS
CBS- (YB-YA) /Ds
CGS- (ZB-ZA) /DS
CC=CA*CAS+CB*CBS+CG*CGS
IF(ABS(CC).GT..997)GO TO 200
S= (X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
IF(INT.LE.0)GO TO 300
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)
AMB=AM*AM
SGB=1.
DO 100 IN=1.IP
Z11=SZ
Z22=S2-DS
XXS=X1+T*CA-XA-SZ*CAS
YYS=Y1+T*CB-YA-SZ*CBS
ZZZ=X1+T*CG-XA-SZ*CGS
RS=XXI**2+YX**2+ZZZ**2
R1=SQRT(RS+Z11**2)
EJA=CEXP(-GAM*R1)
EJ1=EJA/X1
R2=SQRT(RS+Z22**2)
EJB=CEXP(-GAM*R2)
EJ2=EJB/R2
ER1=EJ*A*SGDS+II*EJ1*CGDS-II*A*EJ1
ER2=EJ*B*SGDS+II*EJ2*CGDS-II*A*EJ1
FAC=0.0
IF(RS.GT.ANS)FAC=(CA*XXI+CB*YYI+CG*ZZI)/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.1.P)C=1.
BGD=CEXP(GAM*(DT-6))
C1=C*(BGD-1./BGD)/2.
BGD=CEXP(GAM*T)
C2=C*(BGD-1./BGD)/2.
P11=P11+ET1*C1
P12=P12+ET2*C2
P21=P21+ET1*C1
P22=P22+ET2*C2
T=T+DELT
S1=S2+DS2
100 SGN=-SGN
CST=ETA*DELT/(3.*PP*SGDS*SGDT)
P11=CST*P11
P12=CST*P12
P21=CST*P21
P22=CST*P22
RETURN
200 S11=(XI-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
S21=S21+DT*CC
RH1=SQRT((XI-XA-S21*CAS)**2+(Y1-YA-S21*CBS)**2+(Z1-ZA-S21*CGS)**2)
S22=S22+DT*CC
RH2=SQRT((X2-XA-S22*CAS)**2+(Y2-YA-S22*CBS)**2+(Z2-ZA-S22*CGS)**2)
DDD=(RH1+RH2)/2.
IF(DDD.GT.20.*AN.AND.INT.GT.0)GO TO 20
IF(DDD.LT.AN)DDD=AN
CALL GGM (.0DS,SZ,SZ2DDDoCGSSGDSsGDTI
2,ETA,GAM,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=(XI-XA)*CAD+(Y1-YA)*CBD+(Z1-ZA)*CGD
DK=ABS(DK)
IF(DK.LT.AN)DK=AN
XI=XA+S2*CAS
YI=YA+S2*CBS
ZI=ZI+S2*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-X1)+CBP*(YP1-Y1)+CGP*(ZP1-Z1)
T1=P1/SS
S1=T1*CC-S2
CALL GGM(.0DS,S1,S1+DS,T1,T1+DT,DK,CGDS,SGDS,SGDT,CC,ETA,GAM
2,P11,P12,P21,P22)
RETURN
END

283
SUBROUTINE COUPLE

SUBROUTINE COUPLE(ZT,ZTP,M1,N2,SN1,SN2,I12,V,NT,IFIL,ICC)
COMPLEX * (1), V(1), D, Y11, Y12, Y21, Y22, Z11, Z12, Z21, Z22

DO 100 I = 1, 2
!!!= 1
M = M1
IF (I .EQ. 2) M = M2
DO 10 K = 1, NT
V(K) = (0.0, 0.0)
V(N) = (1.0, 0.0)
IF (IFIL .EQ. 0) CALL SQRZOT(ZT, V, 0, I12, NT)
IF (IFIL .EQ. 1) CALL CROUT(ZT, V, ICC, 0, 0, I12, NT)
IF (I .EQ. 2) GOTO 120
Y11 = V(M1)
Y12 = V(M2)
GOTO 100
100 CONTINUE
Y11 = Y11 * SN1 * SN1
Y21 = Y21 * SN1 * SN2
Y22 = Y22 * SN2 * SN2
Y12 = Y12 * SN2 * SN1
WRITE (6, *) Y11, Y12
WRITE (6, *) Y21, Y22
D = Y11 * Y22 - Y12 * Y21
$11 = Y22 / D
$12 = - Y12 / D
$21 = Y21 / D
$22 = Y11 / D
WRITE (6, 130)
130 FORMAT (/ 3X, 'TWO PORT IMPEDANCE MATRIX (OHMS)' )
WRITE (6, 140) $11, $12, $21, $22
EL = ABS(Y11) / (2.0 * REAL(Y11) * REAL(Y22) - REAL(Y12 * Y21))
EL = ABS(EL)
GMAX = EL / 2.0
IF (ABS(EL) .GT. 0.003) GMAX = (1.0 - SQRT(1.0 - EL * EL)) / EL
GMAX = 10.0 * ALOG10(GMAX)
WRITE (6, 200) GMAX
200 FORMAT (3X, 'MAX. COUPLING IN DB = ', F10.3/)
TYPE*, GMAX
RETURN
END
SUBROUTINE ANTV

SUBROUTINE ANTV(I1, I2, I3, IA, IB, IWR, JA, JB, NM,
IZT, CJ, CG, VG, Y11, Z11, NWR, NFL, NAT, VGA, PIN,
ZAN, COMD, D, DSS, GAM, SGD, ZLD, ZS, ZLDA, INM, MD, ND, NSA)
COMPLEX CJ(1), VG(1), Y11, Z11, CG(1)
COMPLEX VGA(1), ZT(1), GAM, SGD(1), ZS, ZLDA(1), ZLD(1)
DIMENSION I1(1), I2(1), I3(1), IA(1), IB(1), JA(1), JB(1)
DIMENSION(D(1), MD(INM, 4), ND(1), NSA(1))
IJ(I, J, N) = (J-1) * N - (J*J-J) / 2 + I
N=I1=NWR+NPL+NAT
DO 50 I=1, N
CJ(I) = (0, 0)
IF(I.GT.NWR) GOTO 50
K = JA(I)
DO 40 K = 1, 2
KA = IA(K)
KB = IB(K)
JJ = K
FI = 1.
IF(KB.EQ.I2(I)) GOTO 36
IF(KB.EQ.I1(I)) FI = 1.
CJ(I) = CJ(I) + FI*VG(JJ)
GO TO 40
36 IF(KA.EQ.I3(I)) FI = -1.
JJ = K + NM
CJ(I) = CJ(I) + FI*VG(JJ)
40 K = JB(J)
50 CONTINUE
IF(NAT.EQ.0) GOTO 89
DO 90 I = 1, NAT
K = NWR + NFL + I
90 IF(NAT.EQ.0) GOTO 89
DO 90 I = 1, NAT
K = NWR + NFL + I
90 CONTINUE
60 FORMAT(/3X, 'LISTING OF GENERATORS'/)
DO 55 I = 1, NAT
IF(CABS(CJ(I)).EQ.0.) GOTO 55
IF(1.GT.NWR + NFL) GOTO 65
WRITE(6, 70) CJ(I), I
70 FORMAT(3X, 2F8.3, 2X, 'VOLT GENERATOR IN WIRE MODE', 13)
GOTO 65
55 I = I + NFL - NWR

APPENDIX 32

SUBROUTINE ANTV

SUBROUTINE ANTV(I1, I2, I3, IA, IB, IWR, JA, JB, NM,
IZT, CJ, CG, VG, Y11, Z11, NWR, NFL, NAT, VGA, PIN,
ZAN, COMD, D, DSS, GAM, SGD, ZLD, ZS, ZLDA, INM, MD, ND, NSA)
COMPLEX CJ(1), VG(1), Y11, Z11, CG(1)
COMPLEX VGA(1), ZT(1), GAM, SGD(1), ZS, ZLDA(1), ZLD(1)
DIMENSION I1(1), I2(1), I3(1), IA(1), IB(1), JA(1), JB(1)
DIMENSION(D(1), MD(INM, 4), ND(1), NSA(1))
IJ(I, J, N) = (J-1) * N - (J*J-J) / 2 + I
N=I1=NWR+NPL+NAT
DO 50 I=1, N
CJ(I) = (0, 0)
IF(I.GT.NWR) GOTO 50
K = JA(I)
DO 40 K = 1, 2
KA = IA(K)
KB = IB(K)
JJ = K
FI = 1.
IF(KB.EQ.I2(I)) GOTO 36
IF(KB.EQ.I1(I)) FI = 1.
CJ(I) = CJ(I) + FI*VG(JJ)
GO TO 40
36 IF(KA.EQ.I3(I)) FI = -1.
JJ = K + NM
CJ(I) = CJ(I) + FI*VG(JJ)
40 K = JB(J)
50 CONTINUE
IF(NAT.EQ.0) GOTO 89
DO 90 I = 1, NAT
K = NWR + NFL + I
90 IF(NAT.EQ.0) GOTO 89
DO 90 I = 1, NAT
K = NWR + NFL + I
90 CONTINUE
60 FORMAT(/3X, 'LISTING OF GENERATORS'/)
DO 55 I = 1, NAT
IF(CABS(CJ(I)).EQ.0.) GOTO 55
IF(1.GT.NWR + NFL) GOTO 65
WRITE(6, 70) CJ(I), I
70 FORMAT(3X, 2F8.3, 2X, 'VOLT GENERATOR IN WIRE MODE', 13)
GOTO 65
55 I = I + NFL - NWR

285
C

WRITE(6,75) CJ(I),II
75 FORMAT(3X,2P8.3,2X,'VOLT GENERATOR IN ATTACHMENT MODE',I3)
55 CG(I)=CJ(I)
102 CALL SQROT(ST,CJ,IWR,1,NTOT)
   Y11=(0.,0.)
   DO 80 I=1,NTOT
80 Y11=Y11+CJ(I)*CONJG(CG(I))
   Z11=1./Y11
   PIN=REAL(Y11)
   CALLARITE(IA,IB,INM,0.11,12,13,ND,NN,CJ,CG,NSA,NWR,NPL,NAT)
   CALLAGDISS(AN,CG,CM,D,DISS,GAM,NN,SGD,ZLD,ZS,ZLDA,NAT,NSA)
   PRAD=PIN-DISS
   EFF=100.0*PRAD/PIN
   WRITE(6,513) Y11,Z11,EFF
513 FORMAT(3X,'INPUT ADMITTANCE(MHOS) = ',F10.6,' J ',F10.6/
23X,'INPUT IMPEDANCE(OMS) = ',F10.3,' J ',F10.3/,33X,'EFFICIENCY(PERCENT) = ',F7.3/)
RETURN
END
APPENDIX 33

SUBROUTINE CROUT

SUBROUTINE CROUT(C,S,ICC,ISYM,IMR,IL2,N)
COMPLEX C(1CC,ICC),S(I)
COMPLEX F,F,SS,T
2 FORMAT(1X,115,1F10.3,1F15.7,1F10.0)
5 FORMAT(1HO)
IF(II2.NE.1)GO TO 22
IF(N.EQ.1)S(l)-S(l)/C(1,1)
IF(N.EQ.2)GO TO 100
IF(ISYM.NE.0)GO TO 8
DO 6 I=1,N
DO 6 J=1,N
6 C(J,1)=C(I,J)
8 F=C(1,1)
DO 10 L=2,N
10 C(l,L)=C(I,L)/F
DO 20 L=2,N
LLL=L-1
DO 20 I=1,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.1)GO TO 20
F=C(L,L)
IF(ISYM.EQ.0)GO TO 15
F=C(L,1)
DO 12 K=1,LLL
12 F=F-C(L,K)*S(R)
287
7.
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(R)

287
30 S(I)=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(INR.LE.0) GO TO 100
   WRITE(6,5)
   CNOR=0.
   DO 40 I=1,N
   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
APPENDIX 34

SUBROUTINE AGDISS

SUBROUTINE AGDISS (AM, CG, CM, D, DISS, GM, NM, SG, ZLD, ZS, ZLDA, NT, NSA)
COMPLEX CG(1), SG(1), ZLD(1), CJ, CJB, GM, MS
COMPLEX ZLDA(1), ZJ, ZK
DIMENSION D(1), NSA(1)
DATA PI/3.14159/
DISS=0
IF (CM, LE. 0.) GO TO 120
ALPH=REAL (GM)
BETA=AIMAG (GM)
RH=REAL (ZS)/(4.*PI*AN4)
DO 100 K=1,NM
DK=D(K)
DEN=CABS (SGD(K))**2
EAD=EXP (ALPH*DK)
CAD=(EAD+1./EAD)/2.
CBD=COS (BETA*DK)
SAD=DK
IF (ALPH, NE. 0.) SAD=(EAD-1./EAD)/(2.*ALPH)
SBD=DK
IF (BETA, NE. 0.) SBD=Sin (BETA*DK)/BETA
FA=RH*(SAD*CAD-SBD*CBD)/DEN
FB=2.*RH*(CAD*SBD-SAD*CBD)/DEN
CJ=CJG (K)
L=K+NM
CJB=CG(L)
D0 100 J=1,NM
D0 120 J=1,NM
DO 140 J=1,NM
K=J+NM
ZJ=ZLD(J)
ZK=ZLD(K)
IF (NT, EQ, 0.) GOTO 150
DO 160 NA=1, NAT
IF (NSA(NA), EQ, J) ZJ=ZJ+ZLDA(NA)
IF (NSA(NA), EQ, K) ZK=ZK+ZLDA(NA)
160 CONTINUE
CONTINUE
140 DISS=DISS+REAL(ZJ)*(CABS(CG(J))**2)+REAL(ZK)*(CABS(CG(K))**2)
2 +REAL(ZK)*(CABS(CG(K))**2)
RETURN
END
SUBROUTINE ARITE

SUBROUTINE ARITE(IA, IB, INM, IWR, I1, I2, I3, MD, ND, NM, CJ, NS,

COMPLEX CJ(1), CG(1), CJA, CJB
DIMENSION IA(1), IB(1), I1(1), I2(1), I3(1), MD(INM,4), NS(1)

FORMAT (1X, 15.2G10.3, 2F10.1, 4G15.6)

DO 100
100 CONTINUE

DO 40 NN=1, ND
IF (KB.EQ.I2(I)) GO TO 36
IF (KB.EQ.I3(I)) GO TO 40
CJB=CJB+FI*CJ(I)
40 CONTINUE

CG(K)=CJA
ACJ=CABS(CJA)
BCJ=CABS(CJB)
IF (ACJ.GT.ANAX) ANAX=ACJ
IF (BCJ.GT. ANAX) ANAX=BCJ

RETURN

IF (IWR.GT.0) GO TO 110

CONTINUE
110 IF (AMAX.LE.0.) AMAX=1.
    DO 200 K=1,NM
    CJA=CG(K)
    KK=K+NM
    CJB=CG(KK)
    ACJ=CABS(CJA)/AMAX
    BCJ=CABS(CJB)/AMAX
    PA=.0
    PB=.0
    IF (ACJ.GT.0.) PA=57.29578*ATAN2(AIMAG(CJA),REAL(CJA))
    IF (BCJ.GT.0.) PB=57.29578*ATAN2(AIMAG(CJB),REAL(CJB))
200 WRITE (6,2) K, ACJ, BCJ, PA, PB, CJA, CJB
    WRITE (6,5)
    RETURN
END
SUBROUTINE SORTB

SUBROUTINE SORTB(IA, IB, I1, I2, I3, NWR, NM, A, CGD, SGD, PHZ, D,
&wrsq, I12, ISCAT, ZTF, ZT, IFIL, ICC, ETT, EPP,
&X, Y, Z, NPL, NAT, PA, PB, NSA, NPLA, PCN, BDSK, IQUAD,
&NPLTM, IPL, IPLM, CJP, CJT, ETT, EPS, ETPS, ETTS, EPS, ETPS, EPTST, THETA, PHI, JA, JB,
&SCSP, SCST, SCPM, SPM, STM, STM, IMAGE, ICN, NDNPLT)

DIMENSION IA(1), IB(1), I1(1), I2(1), I3(1),
2D(1), X(1), Y(1), Z(1), IPN(1), IQUAD(1)

DIMENSION IA(IPLM, 4), PB(IPLM, 4), NSA(1), NPLA(1), PCN(3), NPLT, IPLM)

DIMENSION NDNPLT(1), BDSK(1), NM12N(1), NM23N(1), JA(1), JB(1)

COMPLEX SGD(1), CGD(1), ETTS, EPS, ETPS, EPTST, THETA, EPP, ETA

COMPLEX CJP(1), CJT(1), ZTP(ICC, ICC), SGD, CGDN

COMPLEX CSUMP, CSUMT

DATA PI, TP/3.1415926, 6.283185/

AB(I, A, B) = (I-1) * B + (2-I) * A

NTOT = NWR + NPLTM + NAT

XJ = CMPLX(0.0, 1.0)

WV = 2.99828/FHZ

XX = 2.0 * PI * WV

ETA = CMPLX(120.0 * PI, 0.0)

GAM = XJ * XX

CJI = 4.0 * PI / (ETA * GAM)

GGG = REAL(1.0 / ETA)

ETTS = (0.0, 0.0)

ETPS = (0.0, 0.0)

IWR = 0

IF (ISCAT.NE.1) GO TO 15

DO 10 I = 1, NTOT

CJP(I) = (0.0, 0.0)

CJT(I) = (0.0, 0.0)

10 IF (ISCAT.EQ.0) GO TO 20

DO 12 I = 1, NTOT

ETT(I) = (0.0, 0.0)

12 EPP(I) = (0.0, 0.0)

CONTINUE

IAT = I + IMAGE

DO 200 IAS = 1, IAT

C

20 CONTINUE
IF (IAS .EQ. 2) GOTO 210
TH = THETA
CTH = COS (THETA)
STH = SIN (THETA)
CPH = COS (PHI)
SPH = SIN (PHI)
PFAC = 1.0
GOTO 220
CONTINUE
TH = PI - THETA
CTH = COS (PI - THETA)
STH = SIN (PI - THETA)
PFAC = -1.0
GOTO 220
CONTINUE
DO 130 N = 1, MTOT
DO 160 J = 1, 2
C DETERMINE EXPANSION MODE TYPE.
C IF (N .GT. NWR + NPLTM) GOTO 270
IF (N .GT. NWR) GOTO 260
C EXPANSION MODE IS A WIRE
L = N
II2 = I2 (L)
I2L = I1 (L)
IF (J .EQ. 2) I2L = I3 (L)
XN1 = X (II2)
YN1 = Y (II2)
ZN2 = Z (II2)
XN2 = X (I2L)
YN2 = Y (I2L)
ZN2 = Z (I2L)
IN12 = (-1)**J
II2 = II2 + J
IF (J .EQ. 2) LL = JB (L)
DN = D (LL)
CGDN = CGD (LL)
SGDN = SGD (LL)
CALL GPP (XN1, YN1, ZN1, XN2, YN2, ZN2, DN, CGDN, SGDN, CTH, STH, CPH,
& SPH, GAM, ETA, ET, DUN, EP, DUN)
EP = EP * PFAC
IF (ISCAT .NE. 0) GO TO 50
IF (INWR .EQ. 2) WRITE (6, *) N, J, XN1, YN1, ZN1, XN2, YN2, ZN2,
ET = ET + IN12
EP = EP * IN12
IF (INWR .GE. 1) WRITE (6, *) N, J, ET, EP, CJ P (N)
ETTS = ETTS + ET * CJ P (N)
EPFS = EPFS + EP * CJ P (N)
GOTO 160
50  IF (IAS .EQ. 1) ETT (N) = ETT (N) + ET * IN12
IF (IAS .EQ. 1) EPF (N) = EPF (N) + EP * IN12
IF (ISCAT .EQ. 2) GO TO 160
CPJ (N) = CPJ (N) + EP * CJ * IN12
CJT (N) = CJT (N) + ET * CJ * IN12
GO TO 160
GOTO 160
260 CONTINUE
C EXPANSION MODE IS A PLATE.

293
L=N-NWR
IACN=1QUAD(L)
IN12=(-1)**J
XN1=AB(J,PA(L,1,1),PB(L,1,1))
YN1=AB(J,PA(L,1,2),PB(L,1,2))
ZN1=AB(J,PA(L,1,3),PB(L,1,3))
XN2=AB(J,PA(L,2,1),PB(L,2,1))
YN2=AB(J,PA(L,2,2),PB(L,2,2))
ZN2=AB(J,PA(L,2,3),PB(L,2,3))
XN3=AB(J,PA(L,3,1),PB(L,3,1))
YN3=AB(J,PA(L,3,2),PB(L,3,2))
ZN3=AB(J,PA(L,3,3),PB(L,3,3))
XN4=AB(J,PA(L,4,1),PB(L,4,1))
YN4=AB(J,PA(L,4,2),PB(L,4,2))
ZN4=AB(J,PA(L,4,3),PB(L,4,3))
IF(IACN.NE.3)GO TO 11
GO TO 14

11 NPLS=10
IF(L.GT.NDMLT(NPL,1))NPLS=6
CALL SURFPP(XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,
1XN4,YN4,ZN4)
IF(INR.EQ.2)WRITE(6,*)N,J,ET,EP
EP=EP*PFAC
IF(ISCAT.NE.0)GO TO 101
CJT(N)=CJP(N)
ETTS=ETTS+ET*CJT(N)
EPPS=EPPS+EP*CJP(N)
GO TO 160

101 IF(IAS.EQ.1)ETT(N)=ETT(N)+ET
IF(IAS.EQ.1)EPP(N)=EPP(N)+EP
IF(ISCAT.EQ.2)GO TO 160
CJP(N)=CJP(N)+EP*CJI
CJT(N)=CJT(N)+ET*CJI
GO TO 160

14 CALL SURFPP(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,
1XN4,YN4,ZN4)
EP=EP*PFAC
IF(ISCAT.NE.0)GO TO 100
IF(INR.EQ.2)WRITE(6,*)N,J,KN1,KX1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3
IF(INR.GE.1)WRITE(6,*)N,J,ET,EP
ETTS=ETTS+ET*CJP(N)
EPPS=EPPS+EP*CJP(N)
GOTO160

100 IF(IAS.EQ.1)ETT(N)=ETT(N)+ET
IF(IAS.EQ.1)EPP(N)=EPP(N)+EP
IF(ISCAT.EQ.2)GO TO 160
CJP(N)=CJP(N)+EP*CJI
CJT(N)=CJT(N)+ET*CJI
GO TO 160

270 CONTINUE
C
C EXPANSION MODE IS AN ATTACHMENT MODE.
C
L=N-NWR-NPLTM
IF(J.EQ.2)GOTO290
C DISK COMPONENT OF ATTACHMENT MODE.
NAB=NSA(L)
IF(NAS.GT.NM)GOTO300
NAP=IA(NAS)
GOTO310
300 CONTINUE
NAB=NAS-NM
NAP=IB(NAS)
310 CONTINUE
XN1=X(NAP)
YN1=Y(NAP)
ZN1=Z(NAP)
NPLL=NPLA(L)
XN2=PCN(1,1,NPLL)
YN2=PCN(2,1,NPLL)
ZN2=PCN(3,1,NPLL)
XN3=PCN(1,2,NPLL)
YN3=PCN(2,2,NPLL)
ZN3=PCN(3,2,NPLL)
B=BDSK(L)
CALL DSKFX(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,TETPHI,A,B,W,E,ET,EP)
EP=EP*PPAC
IF(IWR.EQ.2)WRITE(6,*)N,J,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3
IF(IWR.EQ.1)WRITE(6,*)N,J,ET,EP
IF(ISCAT.NE.0)GOTO100
ETTS=ETTS+ET*PPAC(N)
EPPS=EPPS+EP*PPAC(N)
GOTO 160
290 CONTINUE
C WIRE COMPONENT OF ATTACHMENT MODE.
NAS=NSA(L)
IF(NAS.GT.NM)GOTO320
NPP=IB(NAS)
GOTO330
320 CONTINUE
NAB=NAS-NM
NPP=IA(NAS)
330 CONTINUE
XN2=X(NPP)
YN2=Y(NPP)
ZN2=Z(NPP)
DN=D(NAB)
CGDN=CGD(NAS)
SGDN=SGD(NAS)
CALL GFF(XN1,YN1,ZN1,XN2,YN2,ZN2,ET,EP,DUM)
EP=EP*PPAC
EP=WEP*PPAC
IF(IWR.EQ.2)WRITE(6,*)N,J,XN1,YN1,ZN1,XN2,YN2,ZN2
IF(IWR.EQ.1)WRITE(6,*)N,J,ET,EP
IF(ISCAT.NE.0)GOTO100
ETTS=ETTS+ET*PPAC(N)
EPPS=EPPS+EP*PPAC(N)
GOTO 160
160 CONTINUE
130 CONTINUE
200 CONTINUE
IF(ISCAT.NE.1)GO TO 170
135 FORMAT(//' CURRENTS FOR PHI POLARIZED PLANE WAVE')
136 FORMAT(//' CURRENTS FOR THETA POLARIZED PLANE WAVE')
137 FORMAT (3X,14.2(5X,2E20.5))
138 FORMAT (//'RIGHT-HANDSIDE VECTOR: PHI POLARIZATION',30X,
2'THETA POLARIZATION'/)
132 FORMAT (//'IMPEDEANCE MATRIX: UPPER TRIANGULAR MATRIX'/)
134 FORMAT (2X,2E14.4,4X,2E14.4,4X,2E14.4,4X,2E14.4,4X)
141 FORMAT (//'RIGHT-HANDSIDE VECTOR: BY MULT. ZT*CJP,ZT*CJT')
IF (IWRSQ.LE.0) GO TO 139
133 CONTINUE
172 CONTINUE
139 CONTINUE
IF (IWRSQ.GE.1) WRITE (6,136)
IF (IFIL.EQ.0) CALL SQROT(ZT,CJP,IWRSQ,IT2,NTOT)
IF (IFIL.EQ.1) CALL CROUT(ZTF,CJP,ICC,1,IWRSQ,IT2,NTOT)
IF (IWRSQ.GE.1) WRITE (6,136)
IF (IFIL.EQ.0) CALL SQROT(ZT,CJT,IWRSQ,IT2,NTOT)
IF (IFIL.EQ.1) CALL CROUT(ZTF,CJT,ICC,1,IWRSQ,IT2,NTOT)
PIN=0
TIN=0
DO 164 I=1,NTOT
VP=CJI*EPP(I)
VT=CJI*ETT(I)
PIN=PIN+REAL(VP*CONJG(CJPCI))
164 TIN=TIN+REAL(VT*CONJG(CJT(I)))
ECSF=PIN/GGG
ECTF=TIN/GGG
SCP=ECSP
SCST=ECTF
170 IF (ISCAT.EQ.0) RETURN
EPTS=(.0,.0)
ETPS=(.0,.0)
DO 180 I=1,NTOT
EPS=EPPS+CJP(I)*EPP(I)
ETP=ETPS+CJP(I)*ETT(I)
180 ETPS=ETPS+CJT(I)*EPP(I)
SPPM=2.*TP*(CABS(EPPS)**2)
SPTH=2.*TP*(CABS(ETPS)**2)
STPM=2.*TP*(CABS(ETPS)**2)
STTH=2.*TP*(CABS(ETPS)**2)
RETURN
END
SUBROUTINE SURFFP

SUBROUTINE SURFFP (XM1, YM1, ZM1, XM2, YM2, ZM2, XM3, YM3, ZM3, XM4, YM4, ZM4), NPLS, GAM, ETA, XX, FY, ZN1, THR, PFR, ETH, EPB)

COMPLEX ETH, EPB, ET1, EP1, DUM
COMPLEX EGD, CGD, SGD, GAM, ETA

VW = 2.99838/FH2
DX12 = (XM2 - XM1) / NPLS
DY12 = (YM2 - YM1) / NPLS
DZ12 = (ZM2 - ZM1) / NPLS
DX43 = (XM4 - XM1) / NPLS
DY43 = (YM4 - YM1) / NPLS
DZ43 = (ZM4 - ZM1) / NPLS

RHL = SQRT (DX12**2 + DY12**2 + DZ12**2)
NPL = NPLS + 1

CPH = COS (PFR)
SPH = SIN (PFR)

WNT = SQRT ((XM2 - XM1)**2 + (YM2 - YM1)**2 + (ZM2 - ZM1)**2)

ETH = (0, 0)

EPB = (0, 0)

FF = 0.0

DD14 = SQRT ((XM4 - XM1)**2 + (YM4 - YM1)**2 + (ZM4 - ZM1)**2)

DD23 = SQRT ((XM3 - XM2)**2 + (YM3 - YM2)**2 + (ZM3 - ZM2)**2)

DDW = MAX (DD14, DD23)

DO 10 IS = 1, NPL

W = 3. + (-1)**IS

IF (IS.EQ.1 . OR. IS.EQ.NPL) W = W / 2.0

XI = XM1 + (IS-1) * DX12

YI = YM1 + (IS-1) * DY12

ZI = ZM1 + (IS-1) * DZ12

X2 = XM4 + (IS-1) * DX43

Y2 = YM4 + (IS-1) * DY43

Z2 = ZM4 + (IS-1) * DZ43

16 DD = SQRT ((X2 - XI)**2 + (Y2 - YI)**2 + (Z2 - ZI)**2)

IF (DD.GT.0.00001*WV) GO TO 17

SFB = 1.0

IF (IS.EQ.1) SFB = -1.0

XI = XI - 0.01 * DX12 * SFB

YI = YI - 0.01 * DY12 * SFB

ZI = ZI - 0.01 * DZ12 * SFB
X2=X2-0.01*DX43*SFB
Y2=Y2-0.01*DY43*SFB
Z2=Z2-0.01*DZ43*SFB
GO TO 16

17 CONTINUE
EGD=CEXP(GAM*DD)
SGD=(EGD-1./EGD)/2.0
CGD=(EGD+1./EGD)/2.0
CALL GFF(X1,Y1,X2,Y2,Z2,DD,CGD,SGD,CTH,STH,CPH,SPH)

2 CONTINUE
ETH=ETH+W*ETL*COS(XK*(WNT/2.0-(IS-1)*HH1))
EPH=EPH+W*EP1*COS(XK*(WNT/2.0-(IS-1)*HH1))
PF=PF+COS(XK*(WNT/2.0-(IS-1)*HH1))
ETH=ETH+W*ETL
EPH=EPH+W*EP1
PF=PF+W

10 CONTINUE
ETH=IN12*ETH/FF
EPH=IN12*EPH/FF
RETURN
END
APPENDIX 38

SUBROUTINE SURMFF

SUBROUTINE SURMFF(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,TH,PH,
& ETH,EPH,EWL)
COMPLEX ETH,EPH,EX,EY,EZ,EXP,EPX,EPY
REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
PI=3.14159

C TRANSFORM TO STANDARD COORD SYSTEM

W=5*SQRT((X3-X2)**2+(Y3-Y2)**2+(Z3-Z2)**2)
W=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
X0=(X3+X1)/2.
Y0=(Y3+Y1)/2.
Z0=(Z3+Z1)/2.
L2=(X2-X3)/(2.*W)
M2=(Y2-Y3)/(2.*W)
N2=(Z2-Z3)/(2.*W)
L3=(X2-X1)/H
M3=(Y2-Y1)/H
N3=(Z2-Z1)/H
L1=M2*N3-M3*N2
M1=L3*N2-L2*N3
N1=L2*N3-L3*N2
X=SIN(TH)*COS(PH)
Y=SIN(TH)*SIN(PH)
Z=COS(TH)
XP=L1*X+M1*Y+N1*Z
YP=L2*X+M2*Y+N2*Z
ZP=L3*X+M3*Y+N3*Z
TH=ACOS(.999999*XP/SQRT(XP*XP+YP*YP+ZP*ZP))
PH=0.0
IF(XP**2+YP**2.GT.0.0) PH=ATAN2(YP,XP)
ZP=H/2.

C COMPUTE FOR FIELD OF MONPOLE IN STANDARD COORD SYSTEM

ETA0=376.7
XX=2.*PI/EWL
ETH=(0.,0.)
EPH=(0.,0.)
STH=SIN(THP)
IF (ABS(STH) .LT. .001) RETURN

CTH=COS(THP)
SINX=SIN(XX*W)
COSX=COS(XX*W)
SPH=SIN(PHP)
CPH=COS(PHP)
SKH=SIN(KH+H)
CKH=COS(KH+H)
ETH=(0.,1.)*ETA0/(4.*PI)
IF (ABS(ETH).LT...001 .AND. ABS(CPH).LT...001) GOTO 10
C COMPUTE NORMAL FORM OF ETH
ETH=ETH*(SKM*COS(KH+H)*STH*SPH)-((STH*SPH*CKW*SIN(KH+H)*SIN))
ETH=ETH/((KH+H)*(1.-STH)*STH*SPH)*STH)
ETH=ETH*CEXP((0.,1.)*KH+H)*CEXP)
ETH=ETH*(CEXP((0.,1.)*KH+H)*CTH+((0.,1.)*CTH*SKH
& -CKH)+L.)
GOTO 20
10 ETH=ETH*(K+H*KH+H)*((1.-CKH)/(2.*SKH*SKW)

C TRANSFORM TO ORIGINAL SYSTEM
20 ETH=ETH*CEXP((0.,1.)*KH+H*(X*0.0+Y*0.0)+Z))
EXP=ETH*CTH*CPH
EXP=ETH*CTH*CPH
EXP=ETH*CTH*CPH
EXP=ETH*CTH*CPH
EXP=ETH*CTH*CPH
ETH=EX*COS(TH)*COS(PH)+EY*COS(TH)*SIN(PH)-EZ*SIN(TH)
EY=EX*SIN(PH)+EY*COS(PH)
RETURN
END
SUBROUTINE DSKFF

COMPLEX ETH, EPH, EX, EY, EZ, EXP, EYP, EYP
REAL L1, L2, L3, M1, M2, M3, N1, N2, N3
PI = 3.14159265
XK = -2. *PI / WVL
ETAO = 376.7
XKBA = XK * (B - A)

C TRANSFORM TO STANDARD SYSTEM
D = SQRT((X2 - XO)**2 + (Y2 - Y0)**2 + (Z2 - Z0)**2)
L1 = (X2 - XO) / D
M1 = (Y2 - Y0) / D
N1 = (Z2 - Z0) / D
L3 = (Y1 - Y0) * (X2 - X0) - (Z2 - Z0) * (X1 - X0)
M3 = (Z1 - Z0) * (X2 - X0) - (Z2 - Z0) * (X1 - X0)
N3 = (X1 - X0) * (Y2 - Y0) - (X2 - X0) * (Y1 - Y0)
R = SQRT( (L3 * L3) + (M3 * M3) + (N3 * N3) )
L3 = L3 / R
M3 = M3 / R
N3 = N3 / R
L2 = M3 * N1 - N3 * M1
M2 = L1 * N3 - N1 * L3
N2 = L3 * M1 - L1 * N3
X = SIN(TH) * COS(PH)
Y = SIN(TH) * SIN(PH)
Z = COS(TH)
XP = L1 * X + M1 * Y + N1 * Z
YP = L2 * X + M2 * Y + N2 * Z
ZP = L3 * X + M3 * Y + N3 * Z
THP = ACOS(0.9999 * ZP / SQRT(XP * XP + YP * YP + ZP * ZP))
PHP = ATAN2(YP + 0.0000001, XP + 0.0000001)

C COMPUTE FLOW FIELD OF DISK IN STANDARD FORM
F = -SIN(XKBA) / XK - A * COS(XKBA)
ETH = (-1.0) * COS(THP) * SIN(THP) * F / (8.0 * PI * SIN(XKBA))
ETH = ETH * ETAO / XK

C TRANSFORM TO ORIGINAL SYSTEM
ETH = ETH * CEXP((0.0, 0.0) * XK * (X0 * X + Y0 * Y + Z0 * Z))
EXP = ETH * COS(THP) * COS(PHP)
EYP = ETH * COS(THP) * SIN(PHP)
EXP = -ETH * SIN(THP)
EXP = EXP + L2 * EYP + L3 * EXP
EY = M1 * EXP + M2 * EYP + M3 * EXP
EXP = N1 * EXP + N2 * EYP + N3 * EXP
ETH = ETH * COS(TH) * COS(PH) + ETH * COS(TH) * SIN(PH) - ETH * SIN(TH)
EPH = ETH * SIN(PH) + ETH * COS(PH)
RETURN
END
APPENDIX 40

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
</tr>
<tr>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
</tr>
<tr>
<td>110</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
</tr>
<tr>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td>128</td>
<td>129</td>
</tr>
</tbody>
</table>

302