PRIMER AND ANALYSIS TO EM-TRANAIR CODE EXECUTION

G. S. MESERVE

AERODYNAMIC METHODS GROUP
AERODYNAMICS & AIRFRAME BRANCH
AEROMECHANICS DIVISION

MAY 1990

Approved for public release; distribution is unlimited.

FLIGHT DYNAMICS LABORATORY
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6553
This report was prepared by Gregory S. Meserve of the Aerodynamics and Airframe Branch, Aeromechanics Division, Flight Dynamics Laboratory, Wright Research and Development Center, Wright-Patterson AFB, Ohio, under Project 2404, Aeromechanics Technology, Task 240410, Aerodynamics/Airframe Technology, Unit 240410A1, Aerodynamic Design and Analysis Methods.

A special thanks to the members of the Aerodynamics Methods Group for their help and patience towards aiding in this effort. The input from J.E. Bussoletti used to assist in jumping the hurdles is greatly appreciated. Don Kinsey's guidance and for being a caring sounding board were invaluable towards reaching successful goals. And finally a hats off to Deborah Rice and Cheryl Brown for helping to create this document.

The work reported herein was performed during the period 29 Nov 89 to 31 May 90.

This work has been reviewed and is approved.

G.S. MESER  Capt, USAF
Aeronautics Engineer

DR DON KINSEY
Technical Manager
Aerodynamics Methods Group

DENNIS SEDLOCK
Chief, Aerodynamics & Airframe Branch
Aeromechanics Division

Accession For
NTIS GRA&I □
DTIC TAB □
Unannounced Justification

By
Distribution/
Availability Codes
Avail and/or
Dist Special

A-1
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Input Data/Source Deck</td>
<td>3</td>
</tr>
<tr>
<td>VIINP Input Processor</td>
<td>4</td>
</tr>
<tr>
<td>VISOL Solver</td>
<td>8</td>
</tr>
<tr>
<td>Configuration Execution Processing</td>
<td>10</td>
</tr>
<tr>
<td>Post RCS Results</td>
<td>11</td>
</tr>
<tr>
<td>Conclusion</td>
<td>20</td>
</tr>
<tr>
<td>References</td>
<td>21</td>
</tr>
<tr>
<td>Appendix A Subroutine Listings for Libraries, Input Processor and Solver</td>
<td>22</td>
</tr>
<tr>
<td>Appendix B Error Messages Generated by the CFT77 Compiler on VIINP</td>
<td>26</td>
</tr>
<tr>
<td>Appendix C Original and Revised Initialization for VIINP</td>
<td>38</td>
</tr>
<tr>
<td>Appendix D Original and Revised Conducting Sphere Network Input Data</td>
<td>44</td>
</tr>
<tr>
<td>Appendix E Original and Revised Initialization for VISOL</td>
<td>52</td>
</tr>
<tr>
<td>Appendix F IEQUAL Dummy Subroutine</td>
<td>58</td>
</tr>
</tbody>
</table>
INTRODUCTION

In simplest terms EM-TRANAIR is a computer program for the solution of Maxwell's equations in three dimensions. The ability to execute the code for the Aerodynamic Methods Group marks the beginning of the investigation into low observable technology. EM-TRANAIR was developed by the Boeing Military Airplane Company and delivered to the USAF in May 1987.

"EM-TRANAIR uses a variational principle to characterize Maxwell's field equations and specified boundary conditions as an extremum problem in the calculus of variations. A multiply-preconditioned Krylov space method called GMRES (generalized minimum residual algorithm), with both an extremely powerful sparse solver preconditioner and an exterior Helmholtz solver preconditioner, provides convergence to machine error in ten to twenty iterations."  

Some schemes used to calculate low observable parameters initially determine the point sources on the surface grid generated by the electric or magnetic field. From these point sources the far field parameters are calculated with a numerical analysis. Through a different approach using prescribed mathematical tools, EM-TRANAIR uses sources that are fundamental unknowns for the constant coefficient differential operator governing the far field, rather than the electric or magnetic field at grid points. The solution is peculiar to the numerical analysis of Maxwell's equations.

This report is intended to relate information concerning manipulation and execution of the EM-TRANAIR code. The process of code installation became as involved as the actual theory and mathematical analysis employed to achieve solution. Hopefully the following content will prove to be a useful tool to troubleshoot the EM-TRANAIR software.

EM-TRANAIR consists of two main programs supported by five libraries. The two main programs are an input processor, V1INP, and a solver/post-processor, VISOL. The input processor is extremely extensive in length which makes it difficult to troubleshoot for errors. It's quite time intensive when it comes to gaining familiarity.

The library, input and solver files are first saved as program listings and binary files using UPDATE. "UPDATE is a line oriented test editor for maintaining programs in the form of source code, as well as other types of text data. UPDATE creates and modifies program libraries and produces output that can be used as input to other programs, particularly compilers and assemblers."  

The library files are needed so that pieces may be pulled from them when the source decks for the input processor and the solver are created using SEGLDR. The reason for this methodology is to economize the available central memory during code execution. Use of SEGLDR results in the creation of a nonsegmented executable program. One for the input processor called EMINP and one for the solver called EMSOL will be created. Nonsegmented programs are those that have all of their code continually memory resident. "SEGLDR is an automatic loader for code produced by language processors."
The code delivered by Boeing was compiled using CFT FORTRAN compiler on a CRAY computer with 8 million words of available central memory under the COS operating system. These three characteristics (chosen compiler, size of central memory and chosen operating system) are key factors influencing successful code execution.

Execution problems needed to be resolved to utilize EM-TRANAIR and obtain useful numerical RCS calculations. Compiler problems such as the ability to pass argument values between source programs and subroutines are addressed. Hardware limitation such as the size of the central core memory is an important factor in what configurations would result in converged solutions. Manipulations within the FORTRAN codes such as setting initial values for variables and memory allocation for vector arrays are also discussed. Some familiarity with successful numerical RCS calculations resulted in the presentation of a few "rules of thumb." These and other installation and execution problems are presented to help the reader with EM-TRANAIR.

This document is not a dissertation on theory and code development but a helpful primer to EM-TRANAIR installation at WRDC and a guide to trouble-shooting confronted execution and processing problems.
INPUT DATA/SOURCE DECK

GFLIB, EMLIB, GPLIB, TRLIB and EXSOL are the libraries that support the input processor and the solver. GFLIB calculates the Green's function. EMLIB contains the EM (electromagnetic) subroutines. GPLIB contains the general purpose subroutines. TRLIB contains supplemental subroutines. EXSOL contains the exterior Helmholtz solver. The subroutines within the libraries, the input processor and the solver are contained in Appendix A.

Appendix B shows a listing generated when the input processor V1INP was initially compiled using CFT77. Because of the difference between the CFT and CFT77 compilers a number of bugs were created and needed to be resolved for proper compilation. Equivalence and common statements left a number of errors as can be seen in the listing in Appendix B. The CFT77 compiler didn't like the way variables were being passed from the source code back and forth to the subroutines. The call and subroutine statements involving these passed variables were extended to fully cover all the arguments. In this way the equivalence and common statements could be deleted from the code in certain places. The best and recommended way to find the changes would be to compare the original deck with the executable deck. It may be a time consuming task, but the decks are just too extensive to create listings and the number of changes were numerous. The listing such as in Appendix B will provide the information so one can zero in on an area of interest. Remember, the listing in Appendix B can be generated for any code with ON=H in the CFT77 job control statement.

Another problem encountered with compiling on CFT77 was the dimensioning of arrays. The pointers for the arrays weren't being calculated as integers but real numbers. These problems were fixed by insuring arithemtic operations dealing with array pointers resulted in integers.

The final problem worth mentioning dealt with boolean algebra operations. Only the CFT77 can do boolean manipulations. Makes you wonder how the Boeing folks got around this one using CFT? The bottomline is that an executable deck containing boolean algebra requests can only be compiled using CFT77 to be successful at WRDC.

UNICOS operating system requires the use of the CFT77 compiler. Half the battle is won switching from COS to UNICOS because hopefully most of the bugs have been resolved. The code has been made CFT77 compile successful although being originally compiled on CFT.

Just a little hint on troubleshooting, the $DEBUG and dump files are a great aid in getting traceback maps. These maps help in following paths to that place in the code that tripped the execution error. Using COS or UNICOS, various flags are turned on in the job control language statements to establish symbol files and maps for $DEBUG.

Happy hunting.
The input processor is the heart of EM-TRANAIR. The code contains approximately 27,000 lines making path following to debug problems quite a task. The nonsegmented structure helps in your searches for execution problems because the numerous subroutines plus the libraries subroutines act as blaze markings along the pathway. The imbedded error flags and output messages are also another big plus.

The processor reads in the case test configuration and does extensive presolving tests and comparisons to avoid glitches when it's time to use the solver. "The VIINP program reads a description of a configuration and some incidence conditions describing a plane wave and computes coefficients of a discrete operator on a Cartesian grid which approximates Maxwell's equations and appropriate configuration boundary conditions.

The functional operations of the input processor are:

- Process User Input Data
- Assemble Configuration and Test for Validity
- Define Coupling with Cartesian Grid
- Define Locations of Unknowns
- Compute Discrete Operators
- Communicate Problem Definition to Solver, VISOL [4]

The first problem and probably the most recurring is central memory management. Because the size available was limited to under 2 million words of memory the choice of a test case was highly restrictive. Various cases run by Boeing are discussed in reference [1] and one of these would be a good example to compare against during code installation execution runs. It became fairly apparent that only the sphere would become the likely candidate.

The key driver is choosing a size for the parameter NDUMSS. The default values or really the values set in the delivered code established for NDUMSS and other memory management parameters are listed in Table 1. NDUMSS sets the amount of scratch storage needed for the input processor to do its compilations and manipulations. Using NDUMSS set to 1.1 million words really limits the amount of remaining central memory for the other parameters when less than 2 million words total is available. With this in mind and discussions with Boeing, NDUMSS was set to 800,000 words and the sphere would be the only test case that might fit.

The sphere configuration was still too large to fit on the CRAY using COS. The solution was to pull out every other point of the surface grid. This resulted in an executable test configuration. Using COS and less than 2 million words central memory resulted in a really simple geometry for the initial code installation. Memory management is highly dependent upon input configuration.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MXNETT</td>
<td>225</td>
<td>Maximum number of networks in configuration</td>
</tr>
<tr>
<td>MAXPTS</td>
<td>7500</td>
<td>Maximum number of mesh points.</td>
</tr>
<tr>
<td>NGRD</td>
<td>140481</td>
<td>Maximum number of grid points.</td>
</tr>
<tr>
<td>MAXSRC</td>
<td>14048</td>
<td>Maximum number of sources, any one component</td>
</tr>
<tr>
<td>NDUMSS</td>
<td>1.1M</td>
<td>Total scratch storage</td>
</tr>
<tr>
<td>MAXSRCH</td>
<td>100</td>
<td>Maximum number of search directions (GMRES) any one component</td>
</tr>
<tr>
<td>MAXMAT</td>
<td>100</td>
<td>Maximum number of different materials</td>
</tr>
<tr>
<td>MXLPAN</td>
<td>150</td>
<td>Maximum number of panels in a grid box</td>
</tr>
<tr>
<td>MXNTPN</td>
<td>2000</td>
<td>Maximum number of panels in one network</td>
</tr>
<tr>
<td>MAXNB</td>
<td>14048</td>
<td>Maximum number of B points (not used)</td>
</tr>
<tr>
<td>MXIABT</td>
<td>3200</td>
<td>Maximum of pairwise abutments</td>
</tr>
<tr>
<td>MXNABT</td>
<td>1000</td>
<td>Maximum number of abutments</td>
</tr>
<tr>
<td>MXE1AB</td>
<td>20</td>
<td>Maximum number of edges in an abutment</td>
</tr>
<tr>
<td>MXFDSG</td>
<td>1200</td>
<td>Maximum number of fundamental segments</td>
</tr>
<tr>
<td>MXEMPT</td>
<td>6500</td>
<td>Maximum number of edge mesh points</td>
</tr>
<tr>
<td>MXEDMP</td>
<td>400</td>
<td>Maximum number of edge mesh points per network</td>
</tr>
<tr>
<td>MXNAI</td>
<td>350</td>
<td>Maximum number of abutment intersections</td>
</tr>
<tr>
<td>MXNPEC</td>
<td>100</td>
<td>Maximum number of pts in an eq. class</td>
</tr>
<tr>
<td>MNODMX</td>
<td>100</td>
<td>Maximum number of nodes at abutment intersection</td>
</tr>
<tr>
<td>NSEGGMX</td>
<td>60</td>
<td>Maximum number of segments at abutment intersection</td>
</tr>
<tr>
<td>MAXKMS</td>
<td>19500</td>
<td>Maximum number of micro-edge abutment descriptions</td>
</tr>
</tbody>
</table>

**TABLE 1: EM-TRANAIR Parameter Declarations**
Another area that dictated concern dealt with the initialization of code parameters and variables. For some unknown reason the original decks for the input processor and the solver wouldn't execute properly with the initial value sets for the variables. For the processor and solver, source program AAINPUT and AA3DS respectively, would call up a subroutine containing the initial value sets. Appendix C contains the listing of the initial value set for the input processor.

To get around this execution problem with the passing of the initial values the contents of the initial values subroutine was imbedded into the source program. This way the transfer with the call and return was eliminated. If the BLOCK DATA statement was left in the source program the execution would still stop. So the BLOCK DATA statement was also eliminated.

When the "new" deck with the imbedded initialization values is run a caution is returned during the CPT77 compile operation. The caution is to reveal that with the initialized values imbedded in the source program there's no guarantee that the initialized value will be kept throughout the program. In the original deck the initialize subroutine is only called once so its pretty safe to say no problems are encountered due to initialization imbedded in the program source deck. The imbedded initialization to the input processor VIINP program source deck AAINPUT is also shown in Appendix C.

As stated previously, central core memory management was a recurring problem. After getting over the variable initialization problem the input processor still wouldn't execute too far down into the deck. A bit of time was consumed troubleshooting and the best means of solution was to track the values of various variables using write statements to print out values on FT06. It was discovered that the surface grid network for the conducting sphere had twice as many points as there was space provided for in the vector arrays. There weren't any imbedded helpful error messages in VIINP to point at a fix to the problem. Since the size of central memory was set, the number of points on the surface grid had to be decreased. Every other point was eliminated on the surface grid input data. The original surface network grid and final network grid for the conducting sphere configuration are shown in Appendix D. Using the UNICOS operating system instead of COS because of more available central memory should alleviate any problems with surface grid sizes.

One oversight to avoid when trying to follow tracebacks concerns calls to subroutines that won't be found in the listings of Appendix A. It's not that they don't exist but they are CRAY libraries. A couple of discovered examples where CRAY subroutines are called upon are as follows.

The library deck GPLIB which contains the general purpose subroutines also has a call statement for WOPEN. When the subroutine listings for the libraries in Appendix A are reviewed it will be discovered that there is no listing for WOPEN. This is an internal CRAY library subroutine that opens a word-addressable, random access dataset.
The same can be discovered for the routine SCOPY found in the input processor deck, subroutine ABTIDN. SCOPY, again an internal CRAY library subroutine, is used to copy a real or complex vector into another.

Tracebacks greatly aid in path following troubleshooting. If the path appears to call on an unlisted subroutine not found in Appendix A, it's probably a CRAY internal library subroutine.

The various problems encountered by the author to get the input processor VIINP to execute have been discussed. If any hints or insights were revealed to the next individual working with EM-TRANAIR the reward was well worth the effort. Use this information to work the changes made to the original decks if different test configurations result in unexpected execution errors and unrealistic computational results.
VISOL SOLVER

With the input processor being the heart of EM-TRANAIR, the solver, VISOL is the life support. Extensive preprocessing with the input processor establishes an input configuration that predictably runs through the solver error free. By comparison the input processor contains approximately 27,000 lines of code while the solver contains approximately 3,000 lines of code.

The solver, VISOL, "relied heavily on two special purpose libraries to provide the discrete Green's function and the exterior Helmholtz Solver (GFLIB and EXLIB, respectively)."

"The VISOL program iteratively solves the set of discrete equations which approximate Maxwell's equations and describes the scattering of electromagnetic energy from some material object. The solution process can be divided into the tasks of:

- Initialization
- Data Entry
- Overhead Operations
- The Iterative Solution Process [4],

As with the input processor, the problem of initialization of parameters was also encountered with the solver. Two subroutines, INIT and INICOM (shown in Appendix E), in the original VISOL deck were used to pass parameter initial values into the source program AA3DS. The source program AA3DS would call on INIT which would call on INICOM. As expected, the original deck wouldn't execute. To resolve the problem, subroutine INICOM was imbedded into subroutine INIT which in turn was imbedded into the source program AA3DS.

The results are shown in the second listing of Appendix E. This listing is the revised section of the source program AA3DS that contains the imbedded parameter initialization.

The only other execution error encountered with the VISOL was due to a couple of vector arrays being erased from allocated central memory. The vector array was then needed for manipulation further into the code but was no longer available. This problem occurs in subroutines CLUSS and CLUSM which are found in VISOL and called from the source program AA3DS. The subroutines are listed in Appendix F along with a "dummy" subroutine called IEQUAL.

Vector arrays C3 and KCB are the ones that are inadvertently erased. In subroutine CLUSS, vector array KCB is equated with vector array NC. When this is done vector array KCB becomes unlisted on central memory because that allocation of memory space has become known as NC. This is where the IEQUAL "dummy" subroutine comes in. By passing the vector arrays CB and KCB as arguments their memory allocation remains intact. If this isn't done the computer can't find the arrays because essentially they don't exist. But
with IEQUAL, vector arrays CB and KCB still retain their values and memory allocations.

Why are both vector arrays, CB and KCB, passed as arguments to subroutine IEQUAL? Depending on subroutine CLUSS and CLUSM, there are statements within these subroutines that equate CB and KCB to other vector arrays. Note Appendix F is what code exists in the revised VISOL deck. IEQUAL didn't appear in the original VISOL deck.

Installing the solver was a small hurdle compared to installing the input processor. Hopefully, the insights given to troubleshooting these execution errors will aid in any future troubleshooting. These solutions may be a place to start an investigation for future solver software problems if encountered.
CONFIGURATION EXECUTION PROCESSING

Before the SSD on the CRAY was committed as an external resource for UNICOS a number of configurations where submitted for execution. This section deals with a few "rules-of-thumb" that were discovered with this opportunity to test the code while the COS operating system was still available. These rules-of-thumb are still applicable with the UNICOS operating system because they are peculiar to EN-TRANAIR.

When execution starts in the solver, VISOL, various parameters are computed and output to FT06. Two of these, delta (for X, Y and Z) and the number of mesh points per wavelength, lambda, are especially important. Delta should be within the range

\[ 0.2 < \text{delta} < 5 \]

and the number of mesh points per wavelength, lambda, should be within the range

\[ 5 < \text{points/\lambda} < 10 \]

If you stay within these limits the numerical results are reliable.

The Green's function is calculated within VISOL. Information concerning the Green's function is output to FT06. Because of some bug in the code a flag stays on that causes error tolerances to be printed even if the calculations are within accurate solution boundaries. Note that if the tolerances output to FT06 are zeros; rest assured the Green's function is accurate.

Beware of thin geometries such as disks and flat plates. The delta aspect ratio in the "thin" direction may lead to problems in getting a converged solution to Maxwell's equations.

A "thin" configuration resulting in a solution is greatly influenced by the chosen frequency used for the incidence radiation. If a low frequency is specified, you won't have many mesh points per wavelength, lambda, in the "thin" direction. If a high frequency is specified you'll be playing the aspect ratio deltas against the calculated discrete wave number. A possible solution to "thin" geometries may be to scale up the geometry ($\text{REFerence length parameter}$) so the "thin" dimension is no longer a problem.

Finally, be careful with the $\$\text{BOX}$ definition parameters. Be sure the number of mesh points in the computational grid is of the type

\[ (2^a)(3^b)(5^c)+1 \]

Where a, b and c are integers and the resulting number is odd.

These "rules-of-thumb" will be helpful in achieving converged solutions. Experience with the code is the best medicine but the rules-of-thumb are a good heads up before diving in.
POST RCS RESULTS

Before switching the SSD over to UNICOS a few weeks were given where 4 million words of central memory were available with COS. This was an excellent opportunity to try some EM-TRANAI R runs before having to change to UNICOS. This was the chance to gain some familiarity with the code and become better assured things were executing properly.

Various configurations were tried and some were successful and others ran into problems as were outlined in the previous section concerned with configuration execution processing. Presented here are some actual post RCS graphics that compare results to those conducted by Boeing and an actual comparison to an experimental run taken at one of ASD/WRDC radar ranges. These preliminary results are encouraging and reveal the acceptable reliability concerning the software code EM-TRANAI R.

The first configuration used was a conducting sphere. This configuration is the same as used by Boeing for code validation and a precise definition is presented in reference [1,4]. Figure 1 was taken from reference [1]. Figure 2 shows the results for the same conditions as Figure 1 but being an execution run on WPAFB's CRAY X-MP/216. A total RCS measurement for vertical polarization incidence at 2 degree interval bistatic stations is calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement is excellent.

The frequency of the incident radiation is the driver that regiments acceptable and non-acceptable configuration runs as far as number of mesh grid points and points per wavelength lambda. Numerous configurations were tried to get a better feel for the driving parameters and the number of failures exceeded the number of successes. The size of an acceptable geometry is determined by the incidence radiation frequency. This relationship is the key to an accurate solution of Maxwell's equations using Green's function.

Initial runs were executed at high incident frequencies from 9 to 16 GHz. The geometries being used just weren't compatible to the desired incidence frequencies. Possibly, the problem was being to close and in some cases being involved within the resonance region. Looking over the evidence mostly the Rayleigh region appears to lead to code execution success.

The bottom line for the preliminary lessons learned revealed that staying within the Rayleigh region brought about accurate solutions and this is the norm for the configurations used by Boeing for validation. So more configurations were executed to try to mimic Boeing's results.

Moving up in incidence frequency, Figure 3, taken from reference [1] is Boeing's run with a conducting sphere at a medium frequency as determined by the diameter of the sphere. The execution results on the CRAY X-MP/216 for the same configuration used for Figure 3 are presented in Figure 4. A total RCS measurement for vertical polarization incidence at 2 degree interval bistatic stations was calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement

11
FIGURE 1: Vertical Polarization Bistatic RCS of Low Frequency Sphere (Boeing)

FIGURE 2: Vertical Polarization Bistatic RCS of Low Frequency Sphere (CRAY X-MP/216)
FIGURE 3: Vertical Polarization Bistatic RCS of Medium Frequency Sphere (Boeing)

FIGURE 4: Vertical Polarization Bistatic RCS of Medium Frequency Sphere (CRAY X-MP/216)
looks good with the possible exceptions of the deep troughs. More bistatic calculations, say at every degree interval, would produce a better curve.

Having moved up in incidence frequency, a look was taken to see what type of agreement was found at the same incidence frequency but with horizontal polarization of the incidence electric field. Figure 5 was taken from reference [1]. Figure 6 shows the results for the same conditions as Figure 5 but being an execution run on the CRAY X-MP/216. A total RCS measurement for horizontal polarization incidence at 2 degree interval bistatic stations was calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement is excellent.

Finishing up the conducting sphere configuration successful results, is a look at monostatic RCS measurements. Figure 7 shows the monostatic RCS measurement for a 6-inch diameter conducting sphere being radiated by a vertical polarized electric field at 10 GHz. This actual experimental run is presented to show how the total RCS magnitude is a constant over the surface of the conducting sphere. A good numerical result using EM-TRANAIR to compare to the experimental measurements in Figure 7 was not achieved. A good guess as to the inaccuracies in the Green's function solution of Maxwell's equations may be due to the high frequency incidence radiation and the small diameter geometry. A converged solution couldn't be obtained.

The sphere configuration used as Boeing's validation case was used to obtain a numerical monostatic RCS solution (the low frequency case with ka=3.0). The important comparison between the numerical results and the experimental results in Figure 7 is the constant magnitude of RCS over the sphere surface. Figure 8 shows the total RCS measurement for vertical polarization incidence at 2 degree interval monostatic stations. A fifth order polynomial curve fit was used to create a smooth fit through 45 monostatic stations. Figure 9 shows the total RCS measurement for horizontal polarization incidence at 2 degree interval monostatic stations. Once again, a fifth order polynomial curve fit was used to create a smooth fit through 45 monostatic stations. The monostatic RCS numerical results with EM-TRANAIR definitely possess a constant RCS magnitude over the conducting sphere surface for both a horizontal and vertical polarized incidence electric field.

The last configuration and numerical RCS result to be discussed was an attempt to get RCS information on an 8-inch diameter, .25-inch thick, conducting circular disk. The reason this configuration was chosen is due to the experimental results made available from one of WPAFB radar ranges. These experimental results using an incidence radiation (vertical polarization) at 6 GHz are shown in Figure 10. The EM-TRANAIR numerical results are shown in Figure 11 for the conducting circular disk. A total RCS measurement for vertical polarization incidence at 2 degree interval monostatic stations are calculated. A fifth order polynomial curve fit was used to create a smooth fit through 45 monostatic stations. The Radar Range (Figure 10) angle in degrees from 90 to 180 degrees should be compared to the X-MP/216 (Figure 11) angle in degrees from 0 to 90 degrees. This comparable range runs from a face on monostatic view of the disk revolving to an edge on monostatic view. Magnitude in DB from Figure 10 is the same as Total RCS in Figure 11. The comparison between the experimental and
FIGURE 5: Horizontal Polarization Bistatic RCS of Medium Frequency Sphere (Boeing)

FIGURE 6: Horizontal Polarization Bistatic RCS of Medium Frequency Sphere (CRAY X-MP/216)
FIGURE 7: ASD/WRDC Radar Range 6-inch Diameter Conducting Sphere Test Case
Vertical Polarization Monostatic RCS at 10 GHz
Monostatic RCS of Conducting Sphere
$\kappa_a = 3.0$, VV Polarization
$17 \times 17 \times 17$ Grid

Monostatic RCS of Conducting Sphere
$\kappa_a = 3.0$, HH Polarization
$17 \times 17 \times 17$ Grid

FIGURE 8: Vertical Polarization Monostatic
RCS of Low Frequency Sphere
(CRAY X-MP/216)

FIGURE 9: Horizontal Polarization Monostatic
RCS of Low Frequency Sphere
(CRAY X-MP/216)
FIGURE 10: ASD/WRDC Radar Range 8-inch Diameter 0.25-inch Thick Circular Disk Test Case
Vertical Polarization Monostatic RCS at 6 GHz

FIGURE 11: 8-inch Diameter 0.25-inch Thick
Circular Disk Vertical Polarization
Monostatic RCS at 6 GHz
(CRAY X-MP/216)
numerical results reveals a pretty good correlation between the two when defining the troughs and the nodes. The magnitude of the RCS face on (full side of disk) is in good agreement, but as the measurements near the "thin" edge the similarity falls off. Possibly this is a region that doesn't contain enough mesh points per wavelength, lambda, and/or a good enough aspect ratio to get a good converged solution. In reference [5], user's of EM-TRANAIR at the Naval Weapon Center at China Lake had problems with the numerical solutions on closed cylinders.

"[They]...discovered that solutions for vertically polarized scattering from conducting cylinders differed significantly from experimental measurements and from solutions obtained by other computational techniques." The conducting circular disk, being a closed cylinder may be incurring this inherent problem. More field testing is encouraged.
CONCLUSION

EM-TRANAIR is a computer program for the solution of Maxwell's equations in three dimensions. This primer on EM-TRANAIR installation at WRDC and guide to confronted execution problems will aid in understanding the internals of EM-TRANAIR. Successful pathways for accurate converged numerical solutions are possible.

The architecture of EM-TRANAIR was briefly presented in the discussion of various installation problems encountered when moving from a CFT compiler to a CFT77 compiler. Compilation problems and solutions dealt with passing arguments, dimensioning arrays and boolean algebra manipulations.

Execution problems with the input processor, V1INP and the input solver, V1SOL were presented. Troubleshooting methodologies and working solutions were revealed. The actual problems needing to be surmounted for successful program execution are the examples contained in this document.

Different input configurations were used to establish "rule-of-thumb" parameters to achieve accurate numerical solutions. Parameters such as incidence frequency, number of mesh points and aspect ratios were discussed.

Finally, the area to prove being the most successful in the post-processing examination was the Rayleigh region.

The numerical RCS solutions presented are proof positive that the EM-TRANAIR code is executable and providing good computational RCS results.
REFERENCES


APPENDIX A

SUBROUTINE LISTINGS FOR LIBRARIES, INPUT PROCESSOR AND SOLVER

This appendix is intended to guide the user through the various pathways as the result of CALL SUBROUTINE statements. The listings were generated with UPDATE and the CD output option under COS.
*****APPENDIX A - V1INP SUBROUTINE DECKS**************************

AAINPUT INIT FLDFIL ERRMSG NETMOD OPDEF
P1W4 UNP1W4 BOPDEF EDGPI1 FBHFDN FBPND
FESBLE FL2DEF LPCUT LTCTU PANDQC PANDEF
PANPNT PFBLS1 PIFBCH TADJST TCOORD EDGABT
TIFBCH TRIPNT STRTAU ABTCHK CPETP
DETEQV PROEQV CHGGRD PRONEQ PEADEP
NETWORKS MSHIND EMARK FBXSLT PANYST PANCHB
EPTAU ABTIDN ABTDNC DRMCHK DZBCDE ABTECD
D2LINE DOMVEC ABTAIP ABTDZ0 DZBCCH ABTEQC
ABTDAS ABTDAB ABTSYM ABTDAI ABTDUE
MPTQEQ ABTMGS ABTEND CMPTED EDGIN DASPL
EPOINT NRMESH ABTELS IDNG0 BLCLK BLCCAL
NREDGE INTRNM EDGTAU OUTPKV BMARK REFLOC SIND
PERPNT FILOPN KEYINV OPGEN PEAFOR PROMIN
LSOLVE MINCUT GRDGEN MSPNT1 CAMBER INBC INBC1
CIRC ELLPT GADNET TRINP PEAINP
INBC2 MESHPT QUADNT TRWAKE LATIJK GDDEF
MSHNS INECH0 INSUM IJKLAT ABTPT2 BLCCAL
ASEGL ABTSEG APNTL ABTPT1 BLCAL DASPL
CHCOLP BCONCL SPBCS BCOPT SOPCAL
CMNRGD CONTRL CPABN CPNOR PINSID PDRVR
X Daspl XDASIN EDGFLPL GEOMC TREG PTINT
SING SNGDEL SUBORP SURPRO TCNTRL
TSING ZCADJ ENRCHG GTALAM NRPTH SP
NRPTED OPSTNC DEFPGC OPSET DEFDPT
APTDEL DEFFPT TFILL ADJWK DREGD VOPCAL
MAPINV EQUIVC LSCHOP EIBCH CIFBCCH
TRICAL ECAL FCAL HCAL SOPCAL
GLOCAT VELCAL TBTOP PEQLC PINSID PDRVR
PNGHR INNGBR EXNGBR REQC TREG PTINT
ABTORD VM0MC EMNCTU CMPCTL OPPL CDRVR
DEFFPT SWOMC LINTRI INDCHK PAREA PAREA
SELU1 GNRLM FLDCHK TRINRM PAREA CDRVR
TESTP OPCRCH MATCHX FAOFRT AGNRM CDRVR
CNTD VCA0F VCALA IBSET DEFCSA GCALAA
LNSL CDRVR1 IBSET1 LFCT1 VCALAA LUMPC
LUMP LUMPS LSQST BASEVL BASEVL LUMPC
ABTINT GPHSBN GPHPLK GPLUCK DERIVE SITPNT
THNBN CNTRCAL TREBRT TREBRT KID
FACDC FACTOP TDFEQV ODPEQV ANGORS EDGRDC
PLNEQV PROJDT SOUT WRFAOP WRSTOP CRIND
FREEOP BDYST WVCAL CAMLT2 TZCAL CORIND
FLDSCF PSDCOF HNGHBR MXT OPSTN OPSTN
DMATOP EMOUT OPADD LTDDEF CONDEF CJDEF
SRCDEF RHODEF GHRDFE GAMDEF

******APPENDIX A - GFLIB SUBROUTINE DECKS****************************

INICOR SETC0R GETCOR FRECOR BNUAD BVPGEN
BVPG3D DFNG3D DFNIJK DFNSGV GETGWL CQG3
GQXTND G2ECCOR OUTPKV QGQR12 GSPAT GQRG3
TIMING OMGCON GRNAXS COMGEN LSQSOL
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LSQFB</td>
<td>RDG3D</td>
<td>VIP</td>
<td>ZERO</td>
<td>DCBHTG</td>
<td>OUTLIN</td>
</tr>
<tr>
<td>OUTMAT</td>
<td>OUTVEC</td>
<td>PIW4</td>
<td>UABEND</td>
<td>UNPIW4</td>
<td>SPARSE</td>
</tr>
<tr>
<td>AMCON</td>
<td>SHELLX</td>
<td>SPARSA</td>
<td>SPARSE</td>
<td>SPARSF</td>
<td>ASYG3D</td>
</tr>
<tr>
<td>SPARSP</td>
<td>SPARSS</td>
<td>POLGEN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**APPENDIX A - EMLIB SUBROUTINE DECKS**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FFZER</td>
<td>SPCSCAL</td>
<td>SPDIV</td>
<td>SPGRAD</td>
<td>DIV</td>
<td>CCONT</td>
</tr>
<tr>
<td>CDISP</td>
<td>CFILLI</td>
<td>FCOPI</td>
<td>CAXPYI</td>
<td>CD0TCTI</td>
<td></td>
</tr>
<tr>
<td>CTHRS</td>
<td>WHENNCCT</td>
<td>VHEAD</td>
<td>CASSUM</td>
<td>BUBBLE</td>
<td>QUICK</td>
</tr>
<tr>
<td>DIVIDE</td>
<td>CDSPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**APPENDIX A - GPLIB SUBROUTINE DECKS**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CADD</td>
<td>CFILL</td>
<td>CGATHR</td>
<td>CSCATT</td>
<td>CZERO</td>
<td>RSIGN</td>
</tr>
<tr>
<td>ICONTR</td>
<td>CCONTUR</td>
<td>CONTUR</td>
<td>CSTPRT</td>
<td>CVECWR</td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>DSSPPLT</td>
<td>CONJUG</td>
<td>RMATWR</td>
<td>IMATWR</td>
<td></td>
</tr>
<tr>
<td>CMATWR</td>
<td>PLTCAI</td>
<td>PPLET</td>
<td>FSHEL2</td>
<td>KEYSRT</td>
<td></td>
</tr>
<tr>
<td>THSHELL</td>
<td>RELIST</td>
<td>SETPT2</td>
<td>SETPT1</td>
<td>CPLLOT</td>
<td></td>
</tr>
<tr>
<td>IZERO</td>
<td>LSPARI</td>
<td>PARINT</td>
<td>RFILL</td>
<td>RZERO</td>
<td></td>
</tr>
<tr>
<td>LSTN</td>
<td>IADD</td>
<td>NMRARG</td>
<td>CRXIPY</td>
<td>CRXPYI</td>
<td></td>
</tr>
<tr>
<td>OMATWR</td>
<td>RTENWR</td>
<td>CTENWR</td>
<td>ITENWR</td>
<td>OTENWR</td>
<td></td>
</tr>
<tr>
<td>CROSS</td>
<td>DISTNC</td>
<td>MAG</td>
<td>MMOD</td>
<td>PIDENT</td>
<td></td>
</tr>
<tr>
<td>UKYSRT</td>
<td>UVECT</td>
<td>VADD-CR</td>
<td>VMUL-CR</td>
<td>IFILL</td>
<td></td>
</tr>
<tr>
<td>CXXIPY</td>
<td>CCXXPYI</td>
<td>MODVEC</td>
<td>PLOT</td>
<td>PRCONT</td>
<td>ANSPEK</td>
</tr>
<tr>
<td>CGMRES</td>
<td>CNGRES</td>
<td>CTRXTP</td>
<td>UNIONL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTRSL</td>
<td>SETPTR</td>
<td>SF...</td>
<td>CMXM</td>
<td>CMXMA</td>
<td>RRXIPY</td>
</tr>
<tr>
<td>RRPXYI</td>
<td>RXITY</td>
<td>BSEQR</td>
<td>BSEQW</td>
<td>REVERS</td>
<td></td>
</tr>
<tr>
<td>GWANA</td>
<td>MSNDX</td>
<td>CQLSP</td>
<td>DCHBTX</td>
<td>LSQSFX</td>
<td>DEFVEC</td>
</tr>
<tr>
<td>DCBHT</td>
<td>SGINV</td>
<td>LSQSF</td>
<td>FSIMP</td>
<td>LSQSG</td>
<td></td>
</tr>
<tr>
<td>INCMPR</td>
<td>SCMPKT</td>
<td>SORTAK</td>
<td>TCOF</td>
<td>TRANS</td>
<td></td>
</tr>
<tr>
<td>CPLEOT</td>
<td>NTVCWR</td>
<td>NORCAL</td>
<td>IBRCH</td>
<td>UVECTM</td>
<td></td>
</tr>
<tr>
<td>BINSCH</td>
<td>INT1P</td>
<td>INT2P</td>
<td>QCOF</td>
<td>NFILL</td>
<td>BISEQF</td>
</tr>
<tr>
<td>BISEQW</td>
<td>IIXMAX</td>
<td>IMIN</td>
<td>PCCONT</td>
<td>CHCONT</td>
<td></td>
</tr>
<tr>
<td>ROUND</td>
<td>RREWND</td>
<td>BISEQR</td>
<td>HSMCON</td>
<td>CMUL</td>
<td>HSSFTS</td>
</tr>
<tr>
<td>HSFFT1</td>
<td>HSFFT2</td>
<td>HSFFT3</td>
<td>HSFFT4</td>
<td>HSFFT5</td>
<td></td>
</tr>
<tr>
<td>HSFFT6</td>
<td>HSFFT7</td>
<td>HGERR</td>
<td>MISCAL</td>
<td>MSUM</td>
<td>MCOPY</td>
</tr>
<tr>
<td>M2COPY</td>
<td>MAXPY</td>
<td>M2AXPY</td>
<td>M2SCAL</td>
<td>M2SCOPY</td>
<td></td>
</tr>
<tr>
<td>M2SCPY</td>
<td>M2SBTR</td>
<td>MSBUTR</td>
<td>M2BUTR</td>
<td>M2SHFT</td>
<td>MPSUM</td>
</tr>
<tr>
<td>M2PSUM</td>
<td>MVADD</td>
<td>MSWAP</td>
<td>B2FFTR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MARITH</td>
<td>HISIN</td>
<td>HISSIN</td>
<td>HICOS</td>
<td>PERM2P</td>
<td>RCMUL</td>
</tr>
<tr>
<td>PERM2</td>
<td>PERMS</td>
<td>PERM2S</td>
<td>M2ZERO</td>
<td>MVADSB</td>
<td></td>
</tr>
<tr>
<td>MZER0</td>
<td>FACTOR</td>
<td>MPERM</td>
<td>M2PERM</td>
<td>MFFTS</td>
<td>HFFTP</td>
</tr>
<tr>
<td>H2SIN</td>
<td>H2COS</td>
<td>H2SSIN</td>
<td>MVCMUL</td>
<td>M2SCOS</td>
<td></td>
</tr>
<tr>
<td>VMCOPY</td>
<td>MOUT</td>
<td>MIOUT</td>
<td>PGOUT</td>
<td>H1SCOS</td>
<td></td>
</tr>
<tr>
<td>SSCOPY</td>
<td>ROPEN</td>
<td>RREAD</td>
<td>RWRT</td>
<td>RCLOS</td>
<td></td>
</tr>
<tr>
<td>MS2HT</td>
<td>MSSUM</td>
<td>GETLIN</td>
<td>RTSRCH</td>
<td>TSRCR</td>
<td>RVECWR</td>
</tr>
<tr>
<td>IVECWR</td>
<td>PYCONT</td>
<td>ISMX64</td>
<td>SGBSL</td>
<td>MASUM</td>
<td></td>
</tr>
<tr>
<td>MFILL</td>
<td>MVSADD</td>
<td>SGBL</td>
<td>SGBLT</td>
<td>SGBU</td>
<td>SGBUT</td>
</tr>
<tr>
<td>INDFND</td>
<td>PYCSTRT</td>
<td>CPMAI</td>
<td>IMSHFT</td>
<td>IMFILL</td>
<td></td>
</tr>
<tr>
<td>VNNORM2</td>
<td>REINT</td>
<td>SGBFA</td>
<td>SGBCO</td>
<td>SETRA</td>
<td>GETRA</td>
</tr>
<tr>
<td>IOFKA</td>
<td>GCOL</td>
<td>FRERA</td>
<td>SETC1O</td>
<td>PNDREC</td>
<td></td>
</tr>
<tr>
<td>REDRA</td>
<td>GACRA</td>
<td>ISCATC</td>
<td>ISCATZ</td>
<td>CNTIP</td>
<td>CNTRP</td>
</tr>
<tr>
<td>MAPEQV</td>
<td>LSRUT</td>
<td>TREBRN</td>
<td>TERRUT</td>
<td>KID</td>
<td></td>
</tr>
</tbody>
</table>
*****APPENDIX A - TRLIB SUBROUTINE DECKS

CCALN  COMPIP  CMPSCL  GCPCAL  GRDIND  ISCAL
MRS0TM  PANUNI  SURFIT  XXADJ  KRMOVE  ROT1
PROJ  PNMOM  LCHVAR  EDGLS  LPROJ  R0T2
ROT2  SCALE  TRAN  ZMPROJ  ZMERGE  UNIPAN
UNIPAN  POIL0  POILY  POIL1  TRI2D  TRI2DS
H1SYM  H2SYM  BMUL  BMUL1  GRDWN  GBPRO
GBPRO  GBPRY  GBPRYZ  GDB  GRPLAN  GRADML
GRADML  GYADML  GBRO  GRNASM  GCOMP  GREEN
GREEN  REGGRN  GREEN1  SMGRN  BDYCV1  BSIZE
BDYCV0  BADD  BCP  BADDO  BDYCVY  EXGRN
EXGRN  PERMZ  POISOL  EXPOIO  EXP010  POIF0
POIF0  POIFY  POIF1  THRMS

*****APPENDIX A - EXSOL SUBROUTINE DECKS

H1SIN  H2SIN  H1FFTR  H2FFTR  PERM  PERM1P
PERM2P  PERMS  MADD  SWTSUM  TR13H  CONBOX
TR12H  TRI2R  PERMC  H1COS  H2COS  CONVLU
CONVLU  CONFFT  GBRHE  MVADDN  HELMO  HELMO
HELMO  EXHELM  BDYCVH  HCFFTS  HCFFT1  HCFFT2
HCFFT3  HCFFT4  HCFF5  HCFFT6  HCFFT7  MSHUFL
MSHUFL  M2SHFL  ISIN11  HFIRST  HLAST  MABUTR

*****APPENDIX A - V1SOL SUBROUTINE DECKS

AA3DS  ACAL  ALTFSP  AQCAL  BLDNDX  BLDIEL
CGMRES  DDIVG  DERIVE  DGRAD  DJCOLL  DNSADD
DNSADD  DNSTFL  FLDNS  FOPEN  GREENF  INCFAD
INCFAD  INICOM  INIT  JINIT  JSTORE  JWIRTE
JWIRTE  OQCAL  OUTPUT  OCVWR  READIN  OPGET
OPGET  RADSUB  MDBIAS  ORDER  PROXIM  RADC
RADC  CORIND  REDSET  ADJAC  OPCOUP  CMPY
CMPY  FNDLV  BLDDLW  EQUIVC  MODVEC  FFRC5
FFRC5  BISRC5  GAPARM  XFMMTX  MONRCS  MOHOUT
MOHOUT  EQNRDC  FNDIND  INDFND  INDMEQ  INDSTD
INDSTD  RADELW  RADFND  RADIND  RCDEQN  RESTART
RESTART  RSINDX  PSIEM  LINORD  MOMEQ  P0AORD
P0AORD  PLNORD  NESTED  SELPLN  PSHTSR  CSINIT
CSINIT  CSINST  CSFLTR  CSFLSH  CSFAIL  CSSORT
CSSORT  CSSRTB  CSRADX  CSWM0  CSRGM  CSADDI
CSADDI  CSGNRC  CSTRNF  CSLU  CSDEC  DAMPRB
DAMPRB  FCHRB  INIRB  STORB  SLSNE  SGTHR
SGTHR  CGTHK  CGTHRZ  CSCTR  ROWSCL  INCFP1
INCFP1  IEQUAL

25
APPENDIX B

ERROR MESSAGES GENERATED BY THE CFT77 COMPILER ON V1INP

This appendix contains a listing of the error messages that were returned when the original input processor source deck was compiled using the FORTRAN CFT77 compiler. The code was compiled using a FORTRAN compiler CFT when it was developed by Boeing. Error messages will lead to fatal crashes when decks containing them are tried to execute. Warning messages may lead to fatal crashes and cautions are usually just noteworthy comments.

Use this listing or others that can be generated for the library decks and the solver code setting ON=H in the CFT77 job control statement under COS. These can be compared to the successful executable decks to trace down any changes the author made to the EM-TRANAIR code.
APPENDIX B - OUTPUT GENERATED FROM CFT77 COMPILATION OF THE ORIGINAL INPUT PROCESSOR SOURCE DECK V1INP

1. PROGRAM AAINPUT
   AAINPUT.2

75. BLOCK DATA INIT
    INIT.2

235. SUBROUTINE FLDFIL(NK,K)
    FLDFIL.2

240. SUBROUTINE ERRMSG(IFLAG,INAV,IERR)
    ERRMSG.2

437. SUBROUTINE NETMOD(NNET,NA,NL,K,I)
    NETMOD.2

451. SUBROUTINE OPDEF
    OPDEF.2

468. SUBROUTINE PIW4(IN,OUT,NW)
    PIW4.2

499. SUBROUTINE UNPIW4(IN,OUT,NW)
    UNPIW4.2

*** *** FF169 [warning] < UNPIW4, Line = 9, File = $CPL, line = 507 >:

   Integer arithmetic operation exceeds 46-bit maximum. Result will be computed with 64-bit arithmetic

543. SUBROUTINE BOPDEF(IXB,IZMPT,ITPTL,ITPN,ITPS,NPNTS, NAPTS, BOPDEF.2

*** *** FF622 [warning] < BOPEDEF, Line = 190, File = $CPL, line = 732 >:

   OPTIMIZATION - an exponentiation was replaced with a multiplication. This may cause numerical differences.

777. SUBROUTINE EDGPNT(P1,P2,P,P21,V,Q,D)
    EDGPNT.2

799. SUBROUTINE FFBND(P,P1,DP,DF,D1N,IN,M1N,MAX,MIN,KMAX)
    FFBND.2

824. SUBROUTINE FESBLE(N,A,B,C,CHECK)
    FESBLE.2

853. SUBROUTINE FLDFDE(NX,NY,NZ,P1,DP,MTF,S,KB,ERABRT)
    FLDFDE.2

960. SUBROUTINE LPCUT(Q1,Q2,QP,ICS,DCP,EN,ITR,R,P)
    LPCUT.2

1023. SUBROUTINE LTCUT(Q1,Q2,P1,P2,P3,R,P,CHECK)
    LTCUT.2

1049. SUBROUTINE PANDQC(Z1,Z2,Z3,Z4,CP,ICS,DCP,EN,AR,AJ,ALAM,NP)
    PANDQC.2

1105. SUBROUTINE PANDEF
    PANDEF.2

1238. SUBROUTINE PANPNT(CP,ICS,EN,P,ITR,ISED,Q,D)
    PANPNT.2

1348. SUBROUTINE PFBLST(P1,DP,DF,CP,ICS,NX,NY,NZ,KB)
    PFBLST.2

1390. SUBROUTINE PIFBCH(XYZ,CP,ICS,CHECK)
    PIFBCH.2

1411. SUBROUTINE TADJST(P1,P2,P3,PTOL1,P,Q,ADJST)
    TADJST.2

1436. SUBROUTINE TCOORD(P1,P2,P3,P21,P31,EN,Q)
    TCOORD.2

1468. SUBROUTINE TFIBCH(XYZ,P1,P2,P3,CHECK)
    TFIBCH.2

1495. SUBROUTINE TRIPNT(P1,P2,P3,P,PS,PT,PH,V,Q,D)
    TRIPNT.2

1533. SUBROUTINE STRTAU(NJ,NJ1,TAUJ,TAU2,TAUEMP,SEQ)
    STRTAU.2

1812. SUBROUTINE ABTCHK(K,NK,NNK,NZAK,ISD)
    ABTCHK.2

1690. SUBROUTINE CPETP(NMK,NNK,NZAK,WM,ISD,I1Z1,I1Z2,I1Z)
    CPETP.2

1710. SUBROUTINE EDGABT(NML,NNL,NZAL,JSD,J1Z,J2Z)
    EDGABT.2

1784. SUBROUTINE DETEQV(NPEABT, NOTOK)
    DETEQV.2

1973. SUBROUTINE PROEQV(NPEABT, IECTOT)
    PROEQV.2

2145. SUBROUTINE CHGGRD(NPEABT)
    CHGGRD.2

2382. SUBROUTINE PRNEQV(NPEABT, IECTOT)
    PRNEQV.2

2688. SUBROUTINE PEADET(NPEABT,IE,KZEDG,KNCEDG,KNCINT,KNEDG)
    PEADET.2

2887. SUBROUTINE NETWKS
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2734</td>
<td>SUBROUTINE WSHIND (ISD, IZ, IN, NM, NN, L)</td>
</tr>
<tr>
<td>2791</td>
<td>SUBROUTINE EMARK (LABEL)</td>
</tr>
<tr>
<td>2800</td>
<td>SUBROUTINE FBXLST (PI, DP, DF, IPAN, LPAN, KPN, NPNT, IPHT, LPNT, KEY, NLST)</td>
</tr>
<tr>
<td>2874</td>
<td>SUBROUTINE PADJST (K, L, CP, ICS, DCP, EN, PTOL1, PTOL2, PTOL3, P, Q)</td>
</tr>
<tr>
<td>2995</td>
<td>SUBROUTINE PANGHB (K, L, ISED, NAP, KAP, LAP)</td>
</tr>
<tr>
<td>3136</td>
<td>SUBROUTINE EPTAU (NP, IFDSGN, TAU, TAUTMP, EPSTAU)</td>
</tr>
<tr>
<td>3159</td>
<td>SUBROUTINE ABTDN (NNET, NN, N, Z, NT, CMPDRS, BETAMS, EPSGEO, )</td>
</tr>
<tr>
<td>3788</td>
<td>SUBROUTINE ABTDNC (NNET, NN, N, Z, Q)</td>
</tr>
<tr>
<td>3805</td>
<td>SUBROUTINE DNXCEK (KNET, M, N, Z, Q, HEADER)</td>
</tr>
<tr>
<td>3846</td>
<td>SUBROUTINE WPJNORM (ZI, ZS, ZA, EN)</td>
</tr>
<tr>
<td>3858</td>
<td>SUBROUTINE D2LINE (ZI, ZS, DIST)</td>
</tr>
<tr>
<td>3958</td>
<td>SUBROUTINE DOMVEC (A, V)</td>
</tr>
<tr>
<td>3975</td>
<td>SUBROUTINE ABTAIP (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>4060</td>
<td>SUBROUTINE #3DZ2O (NNET, NN, NN, Z, EPSGEO, NZA, ZSV)</td>
</tr>
<tr>
<td>4093</td>
<td>SUBROUTINE DZ5CH (K, M, N, EPS, Z, Z5V, HEADER)</td>
</tr>
<tr>
<td>4199</td>
<td>SUBROUTINE ABTECD (NNET, NN, NN, Z, NT, CMPDRS, BETAMS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>4444</td>
<td>SUBROUTINE ABTFSD (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>4784</td>
<td>SUBROUTINE ABTDAB (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>4979</td>
<td>SUBROUTINE ABTSYM (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>5098</td>
<td>SUBROUTINE ABTDAB (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>5237</td>
<td>SUBROUTINE ABTDUE (NNET, NN, NN, Z, NT, CMPDRS, BETAMS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>5661</td>
<td>SUBROUTINE ABTAIO (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>5833</td>
<td>SUBROUTINE ABTABO (NNET, NN, NN, Z, NT, CMPDRS, EPSGEO, NSYM, NZA)</td>
</tr>
<tr>
<td>6017</td>
<td>SUBROUTINE FSGCMP (KMP, NEDMPA, NFSGA, KFDSEG, NNET)</td>
</tr>
<tr>
<td>6089</td>
<td>SUBROUTINE ADDIN2 (N, IA, IAX)</td>
</tr>
<tr>
<td>6114</td>
<td>SUBROUTINE ABTEMP (NE, KFDS, KSGN, Z, NZA, NN, NN, NEDMPA, EPSGEO)</td>
</tr>
<tr>
<td>6376</td>
<td>SUBROUTINE ABTEG (KPT, KSGN, NPT, IZ, JX, IJSGN)</td>
</tr>
<tr>
<td>6441</td>
<td>SUBROUTINE WPTEG (KPT, NPT, IZ, JX)</td>
</tr>
<tr>
<td>6412</td>
<td>SUBROUTINE ABTHSG (MSG)</td>
</tr>
<tr>
<td>6504</td>
<td>SUBROUTINE ABTEND (MSG)</td>
</tr>
<tr>
<td>6517</td>
<td>SUBROUTINE CWPIED (KMP, NNET, NEDMPA, NZA, NN, NN, KZ)</td>
</tr>
<tr>
<td>6553</td>
<td>SUBROUTINE EDGIND (ISD, NM, NK, KZEDG, KNCEDG, KNCINT, KNEDG)</td>
</tr>
<tr>
<td>6587</td>
<td>SUBROUTINE EPOINT (IZ, INCZ, NZ, T, ZT)</td>
</tr>
<tr>
<td>6633</td>
<td>SUBROUTINE NRMSH (IZX, ZE, INCE, NE, ISGN, TE, ZNR, INR, DNR)</td>
</tr>
<tr>
<td>6629</td>
<td>SUBROUTINE ABTELS (P1, P2, ZE, INCE, NE, EPSGEO, ABUT, TE1, TE2)</td>
</tr>
<tr>
<td>6670</td>
<td>SUBROUTINE IDNGEO (IZ, ZS, EPSGEO, EPSSEQ)</td>
</tr>
<tr>
<td>6682</td>
<td>SUBROUTINE ABTPOS (IZ, EPSGEO, NSYM, KPDS)</td>
</tr>
<tr>
<td>6691</td>
<td>SUBROUTINE EDGWI (KEDG, IMP, NEDMPA, IEDMP)</td>
</tr>
<tr>
<td>6723</td>
<td>SUBROUTINE NREDGE (P, ZE, INCE, NE, ISGN, TSIGN, PE, TE, DE)</td>
</tr>
<tr>
<td>6822</td>
<td>SUBROUTINE INTNRM (IZX, ZX, Y1, Y2, EN)</td>
</tr>
<tr>
<td>6835</td>
<td>SUBROUTINE EDGTAU (ZK, INCE, NE, DZ, TE, TAU)</td>
</tr>
<tr>
<td>6870</td>
<td>SUBROUTINE OUTPKV (LABEL, N, V)</td>
</tr>
<tr>
<td>6891</td>
<td>SUBROUTINE BMARK (LABEL)</td>
</tr>
<tr>
<td>6900</td>
<td>SUBROUTINE PERPNP (P1, P2, P3, P4, Z, Q, D, RWIN, DS)</td>
</tr>
<tr>
<td>6965</td>
<td>1. SUBROUTINE FIOPN</td>
</tr>
<tr>
<td>6981</td>
<td>1. SUBROUTINE KEYINV(M,K,N,L)</td>
</tr>
<tr>
<td>6989</td>
<td>1. SUBROUTINE OPGEN</td>
</tr>
<tr>
<td>7020</td>
<td>1. SUBROUTINE PEAFOR(NPABT, NOTOK)</td>
</tr>
<tr>
<td>7387</td>
<td>1. SUBROUTINE PROWIN(Z,Q1,V1,D2,R)</td>
</tr>
<tr>
<td>7433</td>
<td>1. SUBROUTINE LSOV2M(N,N,L,A,INFO)</td>
</tr>
<tr>
<td>7446</td>
<td>1. SUBROUTINE MNCUT(P1,P2,IPM,ITKM,IPLM,PM)</td>
</tr>
<tr>
<td>7488</td>
<td>1. SUBROUTINE GRDGEN(DX,OT,0Z,XX,XY,XZ,YX,YY,YZ,ZZ,DX,DY,DZ</td>
</tr>
<tr>
<td>7569</td>
<td>1. SUBROUTINE MSPNT1</td>
</tr>
<tr>
<td>7895</td>
<td>1. SUBROUTINE CAMBER(KN,NPCT,NYST,NCEN,NTRL)</td>
</tr>
<tr>
<td>7917</td>
<td>1. SUBROUTINE CIRC(K)</td>
</tr>
<tr>
<td>7917</td>
<td>1. SUBROUTINE ELLPT(K)</td>
</tr>
<tr>
<td>7987</td>
<td>1. SUBROUTINE GADNET(KN,NROW,NCOL,NCEN)</td>
</tr>
<tr>
<td>8069</td>
<td>1. SUBROUTINE TRINP(IERR)</td>
</tr>
<tr>
<td>9977</td>
<td>1. SUBROUTINE INBC(K,ICA,BET,NBET)</td>
</tr>
<tr>
<td>10425</td>
<td>1. SUBROUTINE INBC1(ICA,K,BET,NBET)</td>
</tr>
<tr>
<td>10490</td>
<td>1. SUBROUTINE INBC2(ICA,K,BET,NBET)</td>
</tr>
<tr>
<td>10553</td>
<td>64. READ(NIN,5070) (RHS(1,L-I),I=1,NS)</td>
</tr>
</tbody>
</table>

```plaintext
| 10560 | 71. RHS(1,L-I)=BET(1,L+M) | INBC2.48 |
```

```plaintext
| 10562 | 73. RHS(1,L+M)=RHS(1,L+N+M+1) | INBC2.50 |
```

```plaintext
| 10566 | 77. CALL SCOPY(W,RHS(1,L-M+1),2,RHS(1,L+M),2) | INBC2.54 |
```

```plaintext
| 10568 | 79. CALL RZERO(M,N,RHS(1,ICA+1),2) | INBC2.56 |
```

```plaintext
| 10569 | 80. IF(AMNSW.EQ.1.) CALL WTRXTP(2,M,N,RHS(1,ICA+1),BET,4+NBET) | INBC2.57 |
```
*** *** FF133 [ warning ] < INBC2, Line = 80, File = $CPL, line = 10569 > :

The number of subscripts is less than the number of declared dimensions

10584 1. SUBROUTINE WESHP(K,IPTER,AMNSW,DMNSW,S,NS)  WESHP.2
10794 1. SUBROUTINE QUADNT(KN,NROW,NCOL)  QUADNT.2
10878 1. SUBROUTINE TRWAKE(KN,XWAKE)  TRWAKE.2
10972 1. SUBROUTINE PEAINP(NPEABT)  PEAINP.2
11182 1. SUBROUTINE MSHNS(ZM,NM,NN,WF,NF,NOUT)  MSHNS.2
11291 1. SUBROUTINE INEO(DIERR)  INEO.2
11335 1. SUBROUTINE INSUM  INSUM.2
11595 1. SUBROUTINE IJKLAT(IJK,I,J,K)  IJKLAT.2
11607 1. SUBROUTINE LATIJK(I,J,K,IJK)  LATIJK.2
11618 1. SUBROUTINE GRODEF  GRODEF.2
11682 1. SUBROUTINE ASEG(NK,NK,IMP,NSMNSW,NSWLST,NSWLST)  ASEG.2
11746 1. SUBROUTINE ABTSEG(NABT,NEDAB,NEDMP,KFDFD,KDFSEG,TAEWMP,  ABTSEG.2
11996 1. SUBROUTINE APNLT(NK,NK,IMP,NSMNSW,NSWLST,NSWLST)  APNLT.2
12002 1. SUBROUTINE ABTPT1(NABT,NEDAB,NEDMP,KFDFD,KDFSEG,TAEWMP,  ABTPT1.2
12301 1. SUBROUTINE ABTPT2(NABINT,NMPNA,NEDMP,KEFKEY,KDFSEG,  ABTPT2.2
12543 1. SUBROUTINE CHCOLP(KDFSEG,KSG1,KSG2,MSEGA,MOLPS,KCOLPS)  CHCOLP.2
12570 1. SUBROUTINE BCONCL  BCONCL.2
12947 378. CALL RFILL(4,RHS(1,K),BET,1)  BCONCL.287

*** *** FF133 [ warning ] < BCONCL, Line = 378, File = $CPL, line = 12947 > :

The number of subscripts is less than the number of declared dimensions

12954 385. CALL RFILL(4,RHS(2,K),BET,1)  BCONCL.294

*** *** FF133 [ warning ] < BCONCL, Line = 385, File = $CPL, line = 12954 > :

The number of subscripts is less than the number of declared dimensions

13417 1. SUBROUTINE SPBSC (KCN,IFN,IFN,ISD, KBASIS, LOCERT,KEYLOC,MAPS)  SPBSC.2
13461 1. SUBROUTINE BCOP  BCOP.2
13779 1. SUBROUTINE BLCAL(BLC,BL)  BLCAL.2
13864 1. SUBROUTINE BLCAL(M,N,N,M,NZ,N,2)  BLCAL.2
14287 1. SUBROUTINE CWMROD (KNC,MCN,NC)  CWMROD.2
14364 1. SUBROUTINE CONTNL (KNC,NT,NM,NM,NC,NCA,NBCA,NMAPCA,NPA)  CONTNL.2
14797 1. SUBROUTINE CPABT (IFN,IFN,KN, KRAMT,KSFG,TAUC,ZNC,NEDG,KS,ICPAMC  CPABT.2
16124 1. SUBROUTINE CPNOR (IP,ZP,ENP)  CPNOR.2
15288 1. SUBROUTINE DASPL (KNET,NK,NM,NN,NA,S,S,S,S,PA,2)  DASPL.2
15778 1. SUBROUTINE XDASPL(KNET,IPAN,IPAN,IND,INDEX,ASTX, NSSAX,IPRT)  XDASPL.2
15993 1. SUBROUTINE XDASIN (ASTX,INDEX,INDEX,INDEX,ASTX, IPRT)  XDASIN.2
16052 1. SUBROUTINE EDGSPX (X,W,W)  EDGSPX.2
16076 1. SUBROUTINE GEOMC (KNC,NM,NN,MMC,NA,ZN,NS,NS)  GEOMC.2
16567 3. SUBROUTINE RELOC (EN,SPAC, A,JAC,RPTNYP, AI)  RELOC.2
16648 1. SUBROUTINE SINCD(2,DS,IC)  SINCD.2
16879 1. SUBROUTINE SING (KNET,NK,NM,NN,NSA,NSSA,NS,SS)  SING.2
17191 1. SUBROUTINE SNGDEL
17503 1. SUBROUTINE SUBPQR(CP, AR, P, ALAM, PP, QQ, RR, IC) SNGDEL.2
17649 1. SUBROUTINE SURPRO(Z, ZP, IC) SUBPQR.2
17771 1. SUBROUTINE TCNTRL SURPRO.2
17984 1. SUBROUTINE TSING TCNTRL.2
18194 1. SUBROUTINE ZCADJ TSING.2
18395 3. SUBROUTINE ENRCHG(K, M, N, Z) ZCADJ.2
18441 1. SUBROUTINE GTALAM(C1, C2, ALAM) ENRCHG.2
18475 1. SUBROUTINE NRPTHP(FTCMPD, CP, ICS, PZ, SVAL, TVAL) GTALAM.2
18602 1. SUBROUTINE INSIDE(Q, ICS, AMN, P, WITHIN) NRPTHP.2
18717 1. SUBROUTINE NRPTED(QM, QP, QNEAR, DIST, TAU) INSIDE.2
18750 1. SUBROUTINE OPSTNC(NX, NY, NZ, IBVH, IBDW) NRPTED.2
18787 1. SUBROUTINE DEFGPF(RDXY2, DXY2, BETAMS, OPVH, OPRT, OPDW, OPDFS, OPDL) OPSTNC.2
19854 1. SUBROUTINE OPSET DEFGPF.2
19904 51. EQUIVALENCE(KB, IPNT), (S, KPAN), (MATF, KPNT), (IFPTL, IPNT) OPSET.2
19904 52. EQUIVALENCE(KB, IPNT), (S, KPAN), (MATF, KPNT), (IFPTL, IPNT) OPSET.21

*** *** FF726 [warning] < OPSET >:
  Array MATF with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning] < OPSET >:
  Array S with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning] < OPSET >:
  Array IFPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning] < OPSET >:
  Array KB with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF430 [error] < OPSET >:
  Array IFPTL is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error] < OPSET >:
  Array KB is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error] < OPSET >:
  Array S is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error] < OPSET >:
  Array MATF is assumed-size but is not a dummy argument or a pointee.

19000 1. SUBROUTINE DEFGPF
19055 58. 2 (IAD, IAPTL(6+MAXSRC+1)), (KAD, ITPN) DEFGPF.2

*** *** FF726 [warning] < DEFGPF >:
  Array IAS with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning] < DEFGPF >:
  Array MATG with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning] < DEFGPF >:
Array IAPIL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array IAD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array ITPL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array IDT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array BUFBC with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array DIVD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array ISPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array ITPN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array KAO with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array KOPT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array IDPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array IZMPT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array S with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array IDN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF726 [warning ] < DEFDP > :
Array KDN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.
*** *** FF728 [warning ] ( DEFDPTr ) :
Array KTPN with length 1 is EQUIVALENCExed to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF728 [warning ] ( DEFDPTr ) :
Array MAP with length 1 is EQUIVALENCExed to a COMMON block - CFT77 assumes there are no out-of-bounds references to the array.

*** *** FF430 [error ] ( DEFDPTr ) :
Array KAD is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IAD is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array KDN is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IDN is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IZVPT is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array ISPTL is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IAS is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IDPTL is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array MDPT is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array CDPT is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IAIPTL is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array MAP is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array BUFBC is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array S is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array DIVD is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IPND is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array KDT is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array IDT is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error ] ( DEFDPTr ) :
Array KTPN is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error] (DEFDPT):
  Array ITPN is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error] (DEFDPT):
  Array ITPTL is assumed-size but is not a dummy argument or a pointee.

*** *** FF430 [error] (DEFDPT):
  Array MATG is assumed-size but is not a dummy argument or a pointee.

19234 1.  SUBROUTINE APTDEL(NPPTS,NDPTS,NAPTS,IDT,KDT,IN,KON),  APTDEL.2
19316 1.  SUBROUTINE DEFFPT(MATG,NPPTS,ITPTL,NFPT,FPTL,KGRD)  DEFFPT.2
19388 1.  SUBROUTINE TFILL(IPT,N,IPT,N,IFT,IFT,IFS)  TFILL.2
19404 1.  SUBROUTINE ADJWK(M,N,Z,NZ,ISIDE,XIN,XP,IPADD)  ADJWK.2
19494 1.  SUBROUTINE DREGD(IPN,NDPTS,DP,DPT,KEY)  DREGD.2
19600 1.  SUBROUTINE VOPCAL(IAD,ID,IA,NA,NDN,KON),  VOPCAL.2
19866 1.  SUBROUTINE MAPINV(M,N,M,M,MAP,MAP,MAP,KEY)  MAPINV.2
19901 1.  SUBROUTINE EQUIVC(M,N,M,N,EQ,EQ,EQ,KEY)  EQUIVC.2
19919 1.  SUBROUTINE LSCHP(KP,DP,P,P,P,P,P,P,T1,T2)  LSCHP.2
19945 1.  SUBROUTINE EIFBCH(KP,DP,P,P,P,P,P,P,T1,T2)  EIFBCH.2
19988 1.  SUBROUTINE CIFBCH(KP,DP,P,P,P,P,P,P,T1,T2)  CIFBCH.2
20054 1.  SUBROUTINE ECAL(KP,P,P,E,E,N,F,F)  ECAL.2
20112 1.  SUBROUTINE FCAL(KP,P,P,E,E,N,F,F)  FCAL.2
20134 1.  SUBROUTINE HCAL(KP,P,P,E,E,N,F,F)  HCAL.2
20151 1.  SUBROUTINE SOPCAL(NPPTS,NDIVD,DIVD,KDT,NS,S)  SOPCAL.2
20296 146. IRHS=RHS(2*L,M,JP)/2  SOPCAL.111

*** *** FF251 [caution] (SOPCAL, Line = 146, File = $CPL, line = 20296):
  A floating point division has been encountered in an expression being converted to integer.

20296 146. IRHS=RHS(2*L,M,JP,NM(KP))/2  SOPCAL.113

*** *** FF251 [caution] (SOPCAL, Line = 146, File = $CPL, line = 20296):
  A floating point division has been encountered in an expression being converted to integer.

20300 160. IRHS=RHS(2*L,M,JP,NM(KP)+1)/2  SOPCAL.115

*** *** FF251 [caution] (SOPCAL, Line = 150, File = $CPL, line = 20300):
  A floating point division has been encountered in an expression being converted to integer.

20302 162. IRHS=RHS(2*L,M,JP,1)/2  SOPCAL.117

*** *** FF251 [caution] (SOPCAL, Line = 152, File = $CPL, line = 20302):
  A floating point division has been encountered in an expression being converted to integer.

20543 1. SUBROUTINE PDRVR(S,VXS)  PDRVR.2
20617 1. SUBROUTINE GLOCAT(IID,IX,YZ)  GLOCAT.2
20629 1. SUBROUTINE VELCAL(VC,IC,IC,IB,IB,IFPM, VELCAL.2
20719 1. SUBROUTINE TBTO(NPNT,IPANT,NSCR,SCR)  TBTO.2
20816 1. SUBROUTINE PEQCL(NP,IPL,NP,NP,IPANT,MAXJPL,JPL)  PEQCL.2
20840 1. SUBROUTINE PINSID(CP,DP,INSIDE)  PINSID.2
SUBROUTINE PNGHR (IPL, INSIDE, MXLST, NSMLST, MAXJPL, NPJ, JPL, IULCON) PNGHR.2
SUBROUTINE INNGBR (NZA, INET, NM, NN, IS, IXP, IYIP, IB, JISID, NPJ, JPL) INNGBR.2
SUBROUTINE EXNGBR (INET, IS, IXP, MXLST, NSMLST, NNSW, NPJ, JPL, NPLAG) EXNGBR.2
SUBROUTINE REQC (NPEQC, LPEQC, IPSEQ, ISEQ, NSEQ, NPI, ISEQC) REQC.2
SUBROUTINE TREG (NPEQC, LPEQC, NSEQC, IEQC) TREG.2
SUBROUTINE PRINT (IP, IT, ITRI, PTRI, P) PRINT.2
SUBROUTINE ABTORD (IABT, NEDABE, NE, IE, IFP, KSEG, KFDKEY, KFDSEG, ABTORD.2
SUBROUTINE VVWOC (H, NH, D, LUMP, HH, EE) VVWOC.2
SUBROUTINE EMNCL (P1, P2, IP1, IT1, IP2, IT2, IPM, ITRV, IUPLM, PW) EMNCL.2
SUBROUTINE CMPCTL (N, IL1, IL2, ISEM, IL2) CMPCTL.2
SUBROUTINE OPPLT (NX, NY, NZ, IFP, NPT, NPFTP, IFPTL, NTHER, NOUT) OPPLT.2
SUBROUTINE DEFPTT (KDT, IFPTL, IDT, IAPT, TGSO, IPTL, SCR) DEFPTT.2
SUBROUTINE SWOMC (NP, N, IFP, D, NPT, IFPTL, NOUT) SWOMC.2
SUBROUTINE LINTRI (P1, P2, P3, C) LINTRI.2
SUBROUTINE INDCOH (K, NL, I, M) INDCOH.2
SUBROUTINE TRICPI (I, CP, PO, CPT) TRICPI.2
SUBROUTINE FPNPLT (P, DP, NP, CP, IC, IDP, K, S, NOUT) FPNPLT.2
SUBROUTINE SELUL (ISE, IP, IT, IUL, EN, ULE, P) SELUL.2
SUBROUTINE GNRNL (KNET, ZM, NM, NN, L, ZN, NOUT) GNRNL.2
SUBROUTINE FLDRCH (MA, N, NY, N2, NPT, ITPTL, KORD) FLDRCH.2
SUBROUTINE TRINRM (I, CP, PO, ENC, ENV) TRINRM.2
SUBROUTINE PAREA (NP, P, EN, ARE) PAREA.2
SUBROUTINE TESTP (C, NP, IAPT, PHE) TESTP.2
SUBROUTINE OPCHK (IOPER) OPCHK.2
DATA TOL = /1.E-9/

** FF726 [warning] < OPCHK >:
Array OPER with length 1 is EQUIVALENCED to a COMMON block - CFT77 assumes there are no out-of-bounds references to
the array.

** FF726 [warning] < OPCHK >:
Array BUF with length 1 is EQUIVALENCED to a COMMON block - CFT77 assumes there are no out-of-bounds references to
the array.

** FF726 [warning] < OPCHK >:
Array IAPTL with length 1 is EQUIVALENCED to a COMMON block - CFT77 assumes there are no out-of-bounds references to
the array.

** FF726 [warning] < OPCHK >:
Array RFLD with length 1 is EQUIVALENCED to a COMMON block - CFT77 assumes there are no out-of-bounds references to
the array.

** FF726 [warning] < OPCHK >:
Array DRS with length 1 is EQUIVALENCED to a COMMON block - CFT77 assumes there are no out-of-bounds references to
the array.

** FF430 [error] < OPCHK >:
Array DRS is assumed-size but is not a dummy argument or a pointer.

** FF430 [error] < OPCHK >:
Array BUF is assumed-size but is not a dummy argument or a pointer.
*** *** FF498 [error ] < EMOUT, Line = 111, File = $CPL, line = 27124 >:
Component of the complex constant is not a real or an integer constant, or, this is an I/O statement which contains illegal parenthesis'.

27124 111. C ((MATH(I,N),DI(I,N)),I=1,NMAT(N)) EMOUT.75

27153 1. SUBROUTINE OPADD (L,KDT,IAD,KAD,IAPTS,NXY23,DIVD) OPADD.2
27200 1. SUBROUTINE LSTDEF (IOPER) LSTDEF.2
27305 1. SUBROUTINE CONDEF (NX,NY,NZ,WATF,NFAK1) CONDEF.2
27322 1. SUBROUTINE CJOEF (NX,NY,NZ,WATF,NCON,ICONL,IRECB) CJOEF.2
27363 1. SUBROUTINE SRCDEF (NX,NY,NZ,WATF,NFAK1,NFPTL,MAXST, MSLST) SRCDEF.2
27406 1. SUBROUTINE RHODEF (NX,NY,NZ,WATF,MAXST,IBIAS) RHODEF.2
27426 1. SUBROUTINE GRHDEF (NX,NY,NZ,WATF,MAXST,IBIAS) GRHDEF.2
27448 1. SUBROUTINE CAMDEF (NX,NY,NZ,WATF,NCON,ICONL,IRECB) CAMDEF.2
APPENDIX C

ORIGINAL AND REVISED INITIALIZATION FOR VIINP

The two decks listed in this appendix were used to initialize variables for the input processor VIINP source program AAINPUT. The first listing is from the original code where a call statement in AAINPUT is used to call up the BLOCK DATA INIT. The second listing is the result of solving an execution error by imbedding the initialization in the source program AAINPUT.
APPENDIX C - INITIALIZATION OF PARAMETERS FROM ORIGINAL INPUT PROCESSOR DECK VIINF

*****

• DECK INIT
  BLOCK DATA INIT
  • CA PARAM
  • CA /CHRCTR/
  • CALL PRNT
  • CALL ABTPRT
  • CALL ABTNEW
  • CA /CRSDAT/
  • CA /FILDAT/
  • CA /GRIDQ/
  • CA /INCFEM/
  • CA /ITERQ/
  • CALL /ARVIS/
  • CA /NUMRCN/
  • CA /PHYSCN/
  • CA /LSCALE/
  • CALL ACASE
  • CALL COMPRS
  • CALL SYMM
  • CALL /OFBOD/
  • CALL KSTMLN
  • CALL FMCOF

C

INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES

C

  /CHRCTR/
  DATA REVDAT/’7/31/87’/
  DATA TITLE /’DEFAULT TITLE’/,
  NAME /’DEFAULT NAME’/,
  1
  IPW /’PW’/, IPOL /’EFLD’/

C

  /PRNT/
  DATA IGEOMP /0/, ISINGP /0/, ICONTP /0/, IBCONF /0/
  DATA IEDGEP /0/, ISINGS /0/, IPRAIC /0/, IPARTP /0/
  DATA IPARTS /0/, NEXDG /0/, IOUTPR /0/, IFMCPR /0/
  DATA ICDSTP /0/, IEXTRP /0/, ISMAP /0/, ICMP /0/
  DATA IBCMAP /0/

C

  /ABTPRT/
  DATA IGEON /1/, IGEOUT /0/, NWXREF /0/, NWPROP /0/, IABUTD /0/
  DATA IABSUM /1/, IFABPR /1/
  DATA IGEOM /0/, IBCND /0/, IOPER /0/, IPANL /0/
  DATA NWDO /0/

C

  /ABTNEW/
  DATA EPSGEO /0.0/

C

  /ACASE/
  DATA ALPHA /4.0.0/, BETA /4.0.0/, FSVM /4.1.0/
  DATA IACASE /1/, NACASE /1/

C

  /COMPRS/
  DATA AMACH /0.0001/, ALPC /0.0/, BETC /0.0/

C

  /SYMM/
  DATA NSYM /0.0/, MISYM /0.0/, MSYM /0.0/

C

  /OFBOD/
  DATA NOF /0.0/

39
C /KSTMLN/
DATA NUMPTS /0/
C /FMCOF/
DATA SREF /1./, BREF /1./, CREF /1./, DREF /1./
DATA XREF /0./, YREF /0./, ZREF /0./
C /FILDAT/
DATA NIN /5/, NOUT /6/, NPRSL /7/, NPRPO /8/
DATA NABFL /11/, NFAKI /12/, NFBLP /13/, NFKP /14/,
   NFVD /15/, NFVC /16/, NFBC /17/, NFGRE /18/, NFGRD /19/
   NFDFW /20/, NFDAW /21/, NFDDW /22/, NFPTL /23/, NFMTL /24/,
   NFSC  /25/
C /CRSDAT/
DATA NABS  /1/, NANGB(1) /1/, ANGB(1,1) /0./, ANGB(2,1) /0./,
   ANGB(3,1) /0./
C /GRIDQ/
DATA NX /NGRD1/, NY /NGRD2/, NZ /NGRD3/
DATA XI /0./, XF /0./, YI /0./, YF /0./, ZI /0./, ZF /0./
C /INCFEM/
DATA FREQ /3.E8/, EHMAG /1., 0./
   NASW  /1/, NANG(1) /1/, ANG(1,1) /0./, ANG(2,1) /0./,
   ANG(3,1) /0./
C /ITERQ/
DATA TOL /1.E-10/, NITER /50/, NSRCH /50/, MSGVL /0/
DATA RLOSS /0./, DROPT /0./, NRELXD /1/
DATA IDMRC  /3,5*0/, IDSWP /2,5*0/
C /ARVIS/
DATA NVIS /1/, RMCUT /0.95/
C /LSCALE/
DATA RLEN /1./
C /NUMRCN/
DATA PI /3.14159265358979/, CONE /1., 1./
C /PHYSCN/
DATA GAMMA /1.4/
DATA VLIGHT /2.997925E8/
END
***** APPENDIX C - INITIALIZATION IMBEDDED IN INPUT PROCESSOR V1INP
SOURCE PROGRAM AAINPUT

*DECK AAINPUT
PROGRAM AAINPUT

C
C DRIVER FOR THE TRANAIR INPUT PROCESSOR
C
*CA /CASE/
*CA /FILDAT/
*CALL ABTPRT
*CALL /CHRCTR/
*CALL /GRIDQ/
*CA PARAM
*CALL PRNT
*CALL ABTNEW
*CA /CRSDAT/
*CA /INCFEM/
*CA /ITERQ/
*CALL /ARVIS/
*CA /NUMRCN/
*CA /PHYSCN/
*CA /LSCALE/
*CALL ACASE
*CALL COMPRS
*CALL SYMM
*CALL /OFBOD/
*CALL KSTMLN
*CALL FMCOF

REAL CARD(10)

C
C INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
C
C /CHRCTR/
DATA REVDAT/'7/31/87'/
DATA TITLE/' DEFAULT TITLE
1     '/, NAME/' DEFAULT NAME
2
3 IPW/'PW '/, IPOL/'EFLD'/
C
/PRNT/
DATA IGEOMP/0/, ISINGP/0/, ICONTP/0/, IBCOMP/0/
DATA IEDGEH/0/, ISINGS/0/, IPRAIC/0/, IPARTP/0/
DATA IPARTS/0/, NEXDCN/0/, IOUTPR/0/, IFMCPR/0/
DATA ICOSTP/0/, IEXTRP/0/, ISPMAP/0/, ICPMAP/0/
DATA IBCMAP/0/
C
/ABTPRT/
DATA IGEOIN/1/, IGEOUT/0/, NWXREF/0/, NWPROP/0/, IABUTD/0/
DATA IABSUM/1/, IFABPR/1/
DATA IGEOM/0/, IBCND/0/, IOPER/0/, IPANL/0/
DATA NWDK/0/
C
/ABTNEW/
DATA EPSGEO/0./
C
/ACASE/
DATA ALPHA/4*0./, BETA/4*0./, FSVM/4*1./
DATA IACASE/1/, NACASE/1/
C
/COMPRS/
DATA AMACH /0.0001/, ALPC /0./, BETC /0./
C /SYMM/
DATA NSYM /0/, MSYM /0/, MJSYM /0/
C /0FBOD/
DATA NOF /0/
C /KSTMLN/
DATA NUMPTS /0/
C /FMCOF/
DATA SREF /1./, BREF /1./, CREF /1./, DREF /1./
DATA XREF /0./, YREF /0./, ZREF /0./
C /FILDAT/
DATA NIN /5/, NOUT /6/, NPRSL /7/, NPRPO /8/
DATA NABFL /11/, NFAKI /12/, NFBLP /13/, NFBKP /14/,
   NFVD/15/, NFVC/16/, NFBC/17/, NFGRE/18/, NFGRD/19/,
   NFDWF/20/, NFDWA/21/, NFDWD/22/, NFPTL/23/, NFMTL/24/,
   NFSRC/25/
C /CRSDAT/
DATA NABS /1/, NANGB(1) /1/, ANGB(1,1) /0./, ANGB(2,1) /0./,
   ANGB(3,1) /0./
C /GRIDQ/
DATA NX /NGRD1/, NY/NGRD2/, NZ /NGRD3/
DATA XI /0./, XF /0./, YI /0./, YF /0./, ZI /0./, ZF /0./
C /INCFEM/
DATA FREQ/3.8E8/, EHMAG/(1.,0.),
   NASW/1/, NANG(1) /1/, ANG(1,1) /0./, ANG(2,1) /0./,
   ANG(3,1) /0./
C /ITERQ/
DATA TOL /1.E-10/, NITER /50/, NSRCH /50/, MSGVVL /0/
DATA RLOSS /0./, DR0PT /0./, NRELXD /1/
DATA IDMRCN /3.5*0/, IDSWP /2.5*0/
C /ARVIS/
DATA NVIS /1/, RMCUT /.95/
C /LSCALE/
DATA RLEN /1./
C /NUMRCN/
DATA PI /3.14159265358979/, CONE /(0.,1.)/
C /PHYSCN/
DATA GAMMA /1.4/
DATA VLIGHT/2.997925E8/
C WRITE(NOUT,6005)
C CALL DERIVE
C CALL VHEAD(NOUT,'VIINP: ',REVDAT)
WRITE(NOUT,6030)
C CALL CSTPRT('INITIAL')
C IERR=0
CALL TRINP(IERR)
C CALL CSTPRT('TRINP')
IF(IERR.EQ.3) THEN
WRITE(NOUT,6060)
CALL ABORT('END-OF-FILE ON INPUT')
ENDIF

C
IF(IERR.NE.0) THEN
WRITE(NOUT,6050) IERR
CALL ABORT
ENDIF
CALL OPDEF
CALL CSTPRT('OPDEF')
DO 130 I=1,4
   IF(IDCHK(I).EQ.-1) GO TO 200
130 CONTINUE
C
CALL SOUT
CALL CSTPRT('SOUT')
C
200 CONTINUE
C
READ(NIN,END=210) CARD
GOTO 200
C
210 CONTINUE
C
REWIND NPRSL
REWIND NPRPO
C
WRITE(NOUT,6040)
C
6005 FORMAT(///10X,'ENTERING INITIALIZATION PHASE')
6030 FORMAT(///10X,'STARTING INPUT PROCESSOR PROGRAM')
6040 FORMAT(///10X,'COMPLETING INPUT PROCESSOR PROGRAM')
6050 FORMAT(///10X,'PROGRAM ABORT DUE TO ERROR',I4)
6060 FORMAT(///10X,'END-OF-FILE ON INPUT FILE')
STOP
END
APPENDIX D

ORIGINAL AND REVISED CONDUCTING SPHERE NETWORK INPUT DATA

This appendix contains the original and revised listings for the conducting sphere configuration. The original network configuration used by Boeing contains twice as many points as the executable network.
APPENDIX D - ORIGINAL CONDUCTING SPHERE SURFACE GRID NETWORK INPUT

**TITLE**
SPHERE - NO PLANES OF SYMMETRY

**CASE #1**

| $SYM | 0. 0. |
|      |      |

| $DAT | 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. |

| $BOX | -1.0000 1.0000 17. |
|      | -1.0000 1.0000 17. |
|      | -1.0000 1.0000 17. |

| $NET | 4. |

| $CIR | 4. 1. 0.0 |

|        | 0.5 1. 0.0 |

**RIGHT FRONT**

| 41. | .8 0.79938323 0.03140785 0.79753387 0.06276728 |
| 41. | .79445477 0.09402992 0.79015067 0.12514757 0.78462822 0.15607226 |
| 41. | .77789594 0.18675629 0.76996419 0.21715236 0.76084521 0.24721360 |
| 41. | .75055307 0.27689365 0.73910363 0.30614675 0.72651454 0.33492779 |
| 41. | .71280522 0.36319240 0.69799681 0.39089699 0.68211213 0.41799885 |
| 41. | .66517569 0.44445619 0.64721360 0.47022820 0.62825355 0.49527516 |
| 41. | .60832477 0.51955844 0.58745801 0.54304060 0.56568543 0.56568543 |
| 41. | .54304060 0.58745801 0.51955844 0.49527516 0.46825355 0.42651454 |
| 41. | .47022820 0.64721360 0.44445619 0.41799885 0.66517569 0.49527516 |
| 41. | .39089699 0.69799681 0.36319240 0.71280522 0.33492779 0.72651454 |
| 41. | .30614675 0.73910363 0.27689365 0.72651454 0.24721360 0.76084521 |
| 41. | .21715236 0.76996419 0.18675629 0.77789594 0.15607226 0.78462822 |
| 41. | .12514757 0.79015067 0.09402992 0.79445477 0.06276728 0.79753387 |
| 41. | .03140785 0.79938323 0.0 |

**LEFT FRONT**

| 41. | .8 |
| 41. | 0.03140785 0.79938323 0.06276728 0.79753387 |
| 41. | 0.09402992 0.79445477 0.12514757 0.79015067 0.15607226 0.78462822 |

45
<p>| 81.    | 90.    | 92.25 | 94.5 | 96.75 | 99.0 | 101.25 | 103.5 | 105.75 | 108.0 | 110.25 | 112.5 | 114.75 | 117.0     | 119.25 | 121.50 | 123.75 | 126.0 | 128.25 | 130.5 | 132.75 | 135.0 | 137.25 | 139.5 | 141.75 | 144.0 | 146.25 | 148.5 | 150.75 | 153.0 | 155.25 | 157.5 | 159.75 | 162.0 | 164.25 | 166.5 | 168.75 | 171.0 | 173.25 | 175.5 | 177.75 | 180.0 | 182.25 | 184.5 | 186.75 | 189.0 | 191.25 | 193.5 | 195.75 | 198.0 | 200.25 | 202.5 | 204.75 | 207.0 | 209.25 | 211.5 | 213.75 | 216.0 | 218.25 | 220.5 | 222.75 | 225.0 | 227.25 | 229.5 | 231.75 | 234.0 | 236.25 | 238.5 | 240.75 | 243.0 | 245.25 | 247.5 | 249.75 | 252.0 | 254.25 | 256.5 | 258.75 | 261.0 | 263.25 | 265.5 | 267.75 | 270.0 | 0.0 |</p>
<table>
<thead>
<tr>
<th></th>
<th>072.0</th>
<th>074.25</th>
<th>076.5</th>
<th>078.75</th>
<th>81.0</th>
<th>83.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.5</td>
<td>87.75</td>
<td>90.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LEFT BACK**

<table>
<thead>
<tr>
<th>41.0</th>
<th>.8</th>
<th>-0.03140785</th>
<th>0.79938323</th>
<th>-0.06276728</th>
<th>0.79753387</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.09402992</td>
<td>0.79445477</td>
<td>-1.2514757</td>
<td>0.79015067</td>
<td>-1.5607226</td>
<td>0.78462822</td>
</tr>
<tr>
<td>-0.18675629</td>
<td>0.77789594</td>
<td>-2.1715236</td>
<td>0.76996419</td>
<td>-2.4721360</td>
<td>0.76084521</td>
</tr>
<tr>
<td>-0.27689365</td>
<td>0.75055307</td>
<td>-3.0614675</td>
<td>0.73910363</td>
<td>-3.3492779</td>
<td>0.72651454</td>
</tr>
<tr>
<td>-0.36319240</td>
<td>0.71280522</td>
<td>-3.9089699</td>
<td>0.69799681</td>
<td>-4.1799885</td>
<td>0.68211213</td>
</tr>
<tr>
<td>-0.44445619</td>
<td>0.66517569</td>
<td>-4.7022820</td>
<td>0.64721360</td>
<td>-4.9527516</td>
<td>0.62825355</td>
</tr>
<tr>
<td>-0.51955844</td>
<td>0.60832477</td>
<td>-5.4304060</td>
<td>0.58745801</td>
<td>-5.6568543</td>
<td>0.56568543</td>
</tr>
<tr>
<td>-0.58745801</td>
<td>0.54304060</td>
<td>-6.0832477</td>
<td>0.51955844</td>
<td>-6.2825355</td>
<td>0.49527516</td>
</tr>
<tr>
<td>-0.64721360</td>
<td>0.47022820</td>
<td>-6.6517569</td>
<td>0.44445619</td>
<td>-6.8211213</td>
<td>0.41799885</td>
</tr>
<tr>
<td>-0.69799681</td>
<td>0.39086999</td>
<td>-7.1280522</td>
<td>0.36319240</td>
<td>-7.2651454</td>
<td>0.33492779</td>
</tr>
<tr>
<td>-0.73910363</td>
<td>0.30614675</td>
<td>-7.5055307</td>
<td>0.27689365</td>
<td>-7.6084521</td>
<td>0.24721360</td>
</tr>
<tr>
<td>-0.76996419</td>
<td>0.21712523</td>
<td>-7.7789594</td>
<td>0.18675629</td>
<td>-7.8462822</td>
<td>0.15607226</td>
</tr>
<tr>
<td>-0.79015067</td>
<td>0.12514757</td>
<td>-7.9445477</td>
<td>0.09402992</td>
<td>-7.9753387</td>
<td>0.06276728</td>
</tr>
<tr>
<td>-0.79938323</td>
<td>0.03140785</td>
<td>-8.0000000</td>
<td>0.0000000</td>
<td>-8.0000000</td>
<td>0.0000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>81.0</th>
<th>270.0</th>
<th>272.25</th>
<th>274.5</th>
<th>276.75</th>
<th>279.0</th>
<th>281.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>283.5</td>
<td>285.75</td>
<td>288.0</td>
<td>290.25</td>
<td>292.5</td>
<td>294.75</td>
<td>297.0</td>
</tr>
<tr>
<td>301.5</td>
<td>303.75</td>
<td>306.0</td>
<td>308.25</td>
<td>310.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>315.0</td>
<td>317.25</td>
<td>319.5</td>
<td>321.75</td>
<td>324.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>328.5</td>
<td>330.75</td>
<td>333.0</td>
<td>335.25</td>
<td>337.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>342.0</td>
<td>344.25</td>
<td>346.5</td>
<td>348.75</td>
<td>351.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>355.5</td>
<td>357.75</td>
<td>360.0</td>
<td>002.25</td>
<td>004.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>009.0</td>
<td>011.25</td>
<td>013.5</td>
<td>015.75</td>
<td>018.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>022.5</td>
<td>024.75</td>
<td>027.0</td>
<td>029.25</td>
<td>031.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>036.0</td>
<td>038.25</td>
<td>040.5</td>
<td>042.75</td>
<td>045.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>049.5</td>
<td>051.75</td>
<td>054.0</td>
<td>056.25</td>
<td>058.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>063.0</td>
<td>065.25</td>
<td>067.5</td>
<td>069.75</td>
<td>072.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>076.5</td>
<td>078.75</td>
<td>081.0</td>
<td>083.25</td>
<td>85.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$ITE</th>
<th>50.5</th>
<th>0.0</th>
<th>0.0</th>
<th>50.0</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$INC

<table>
<thead>
<tr>
<th>HFLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PW</td>
</tr>
<tr>
<td>1.7893</td>
</tr>
</tbody>
</table>

| $RCS |
| 2.0 |
| 91.0 |
| 0.0  |

| $THE |
| 4.0 |
| 1.0 | 2.0 | 0.0 |
| 1.0 | 2.0 | 0.0 |
| 1.0 | 2.0 | 0.0 |
$PRI
1.  0.  0.  0.
1.  0.  0.  0.
$END
/EOF
*****APPENDIX D - EXECUTABLE CONDUCTING SPHERE NETWORK WITH FEWER SURFACE GRID POINTS

$TITLE
SPHERE - NO PLANES OF SYMMETRY
CASE #1
$SYM
0. 0.
$DAT
0. 0. 0. 0. 0. 0.
$BOX
-1.0000 1.0000 17.
-1.0000 1.0000 17.
-1.0000 1.0000 17.
$NET
4.
$CIR
4. 1. 0. 0. RIGHT FRONT
41.
.8 0. .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .79015067 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27683635 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49527516
.60832472 .51955844 .58745801 .54304060 .56568543 .56568543
.54304060 .58745801 .51955844 .60832477 .49527516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72561454
.30614675 .73910363 .27683635 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0 .8
41.
94.5 99.0 103.5 108.0 112.5
117. 121.5 126. 130.5 135.0 139.5
144. 148.5 153. 157.5 162.0 166.5
171. 175.5 180. 184.5 189. 193.5
198. 202.5 207.0 211.5 216.0 220.5
225. 229.5 234.0 238.5 243.0 247.5
252 0 256.5 261.0 265.5 270.0
0.
41.
0. .8 -.03140785 .79938323 -.06276728 .79753387
-.09402992 .79445477 -.12514757 .79015067 -.15607226 .78462822
-.18675629 .77789594 -.21715236 .76996419 -.24721360 .76084521
-.27683635 .75055307 -.30614675 .73910363 -.33492779 .72561454
-.36319240 .71280522 -.39089699 .69799681 -.41799885 .68211213
-.44445619 .66517569 -.47022820 .64721360 -.49527516 .62825355
-.51955844 .60832477 -.54304060 .58745801 -.56568543 .56568543
-.58745801 .54304060 .60832477 .51955844 .62825355 .49527516
-.64721360 .47022820 .66517569 .44445619 .68211213 .41799885
49
- .69799681 .39089699 .71280522 .36319240 .72651454 .33492779
- .73910363 .30614675 .75055307 .27689365 .76084521 .24721360
- .78996419 .21715236 .77789594 .18675629 .78462822 .15607226
- .79015067 .12514757 .79445477 .09402992 .79753387 .06276728
- .79938323 .03140785 .8 0.0

41.
90. 94.5 99.0 103.5 108. 112.5
117. 121.5 126. 130.5 135.0 139.5
144. 148.5 153. 157.5 162.0 166.5
171. 175.5 180. 184.5 189. 193.5
198. 202.5 207.0 211.5 216.0 220.5
225. 229.5 234.0 238.5 243.0 247.5
252.0 256.5 261.0 265.5 270.0

0.

RIGHT BACK
41.
.8 .0 .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27689365 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49327516
.60832477 .51955844 .58745801 .54304060 .5658543 .5658543
.54304060 .58745801 .51955844 .60832477 .49327516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72651454
.30614675 .73910363 .27689365 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0 .8

41.
270. 274.5 279. 283.5 288. 292.5
297. 301.5 306.0 310.5 315. 319.5
324. 328.5 333. 337.5 342. 346.5
351. 355.5 000.0 004.5 009.0 013.5
018.0 022.5 027.0 031.5 036.0 040.5
045.0 049.5 054.0 058.5 063.0 067.5
072.0 076.5 81. 85.5 90.0

LEFT BACK
41.
0. .8 -.03140785 .79938323 -.06276728 .79753387
.09402992 .79445477 .12514757 .79015067 -.15607226 .78462822
.18675629 .77789594 .21715236 .76996419 -.24721360 .76084521
.27689365 .75055307 .30614675 .73910363 -.33492779 .72651454
.36319240 .71280522 -.39089699 .69799681 -.41799885 .68211213
.44445619 .66517569 .47022820 .64721360 .49327516 .62825355
.51955844 .60832477 .54304060 .58745801 -.5658543 .5658543
.58745801 .54304060 .60832477 .51955844 -.62825355 .49327516
.64721360 .47022820 .66517569 .44445619 -.68211213 .41799885
.69799681 .39089699 .71280522 .36319240 -.72651454 .33492779
.73910363 .30614675 .75055307 .27689365 -.76084521 .24721360
.76996419 .21715236 .77789594 .18675629 -.78462822 .15607226
.79015067 .12514757 .79445477 .09402992 -.79753387 .06276728
-.79938323 .03140785 -.8 0.0

41.
270. 274.5 279. 283.5 288. 292.5
297. 301.5 306.0 310.5 315. 319.5
324. 328.5 333. 337.5 342. 346.5
351. 355.5 000.0 004.5 009.0 013.5
018.0 022.5 027.0 031.5 036.0 040.5
045.0 049.5 054.0 058.5 063.0 067.5
072.0 076.5 81. 85.5 90.0

50
<table>
<thead>
<tr>
<th></th>
<th>324.</th>
<th>328.5</th>
<th>333.</th>
<th>337.5</th>
<th>342.</th>
<th>346.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>351.</td>
<td>355.5</td>
<td>000.0</td>
<td>004.5</td>
<td>009.0</td>
<td>013.5</td>
<td></td>
</tr>
<tr>
<td>018.0</td>
<td>022.5</td>
<td>027.0</td>
<td>031.5</td>
<td>036.0</td>
<td>040.5</td>
<td></td>
</tr>
<tr>
<td>045.0</td>
<td>049.5</td>
<td>054.0</td>
<td>058.5</td>
<td>063.0</td>
<td>067.5</td>
<td></td>
</tr>
<tr>
<td>072.0</td>
<td>076.5</td>
<td>81.</td>
<td>85.5</td>
<td>90.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$ITE
50. 5. 0. 0. 50. 1. 0. 0. 0.

$INC
HFLD
PW
1.7893 8. 1.
1. 1.
0. 0. 0. 0. 0. 0.

$RCS
4. 180.
0. 0. 0. 2. 0. 0.
180.
0. -90. 90. 2. 0. 0.
-90.
0. 0. 0. 2. 0. 0.
-90.
0. -90. 90. 2. 0. 0.

$THE
4.
1. 1. 2. 0.
2. 1. 2. 0.
3. 1. 2. 0.
4. 1. 2. 0.

$PRI
1. 1. 1. 1.
1. 1. 1. 1.

$SEND
/EOF
APPENDIX E

ORIGINAL AND REVISED INITIALIZATION FOR VISOL

The two decks listed in this appendix were used to initialize variables for the solver VISOL source program AA3DS. The first listing is from the original code where a call statement in AA3DS calls subroutine INIT which in turn contains a call statement to call up the BLOCK DATA INICOM. The second listing is the result of solving an execution error by imbedding the initialization in the source program AA3DS.
APPENDIX E - INITIALIZATION OF PARAMETERS WITH SUBROUTINES FROM THE ORIGINAL SOLVER DECK V1SOL

*DECK INICOM
  BLOCK DATA INICOM
  BLOCK DATA INITIALIZATION
  CALL FORMARK
  CALL PARAM
  CALL COMMARK
  CALL /CBPARM/
  CALL /CHRDAT/
  CALL /DIAGN/
  CALL /FILDAT/
  CALL /GRIDQ/
  CALL /INCANG/
  CALL /INCFILQ/
  CALL /ITERQ/
  CALL /MATDAT/
  CALL /NUMRCN/
  CALL /PHYSN/

  INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
  /CBPARM/
  /CHRDAT/
    DATA REVDAT/'7/31/87'/
    DATA TITLE /'DEFAULT DEFAULT DEFAULT DEFAULT '/
    DATA NAME /'DEFAULT DEFAULT DEFAULT DEFAULT '/
    DATA CHPOL/"EE",CHPW/"PW"/
  /DIAGN/
    DATA EPS/1.E-8/,LIMIT/O/,NDOUT/6/
    DATA DEBUG/.FALSE./
  /FILDAT/
    DATA NIN/5/,NOUT/6/,NERR/6/,NPRSL/7/,NFCJ/9/
    DATA NFRCS/10/
    DATA NDFOUT/4/
    DATA NFGRN/11/,IFSRCH/12/,IFOP/13/
    DATA IFNJ/14/,IFJ/15/,IFGRH/16/,IFGAM/17/
    DATA IFW/18/,IFDJ/19/
    DATA NFMRCS/29/
    DATA IFQC/20/,IFQB/21/,IFQCI/22/,IFQBI/23/,IFQS/24/
    DATA NFQW/25/,NFQWI/26/,IFQV/27/
    DATA IFQE/30/,IFQSC/31/,IFQSB/32/
    DATA NFIEL/28/
    DATA ISEQ/1/,IRAN/2/,IMEM/3/
  /GRIDQ/
    DATA NX/17/,NY/17/,NZ/17/
DATA XI /0./, XF /1./, YI /0./, YF /1./, ZI /0./, ZF /1./
C INCANG/
DATA NASW/1/, NANG(1)/1/
C /INCFLQ/
DATA AANG(1) /0./, AANG(2) /0./, AANG(3) /0./
DATA FREQ /3.E-8/, EMAG /(1.,0.)/
C /ITERQ/
DATA TOL /1.E-10/
DATA NITER /30/, ITPRT /0/, KITER /20/, MSGVL/3/
C /MATDAT/
DATA NMAT/2/
DATA ALPHA(1)/(0.,0.), ALPHA(2)/(0.,0.),
DATA BETA(1)/(0.,0.), BETA(2)/(0.,0.)
C /NUMRCN/
DATA CONE /(0.,1.)/
C /PHYSNC/
DATA VLIGHT /2.997925E8/
C END

*DECK INIT
SUBROUTINE INIT
C INITIALIZE DATA
C CALL FORMARK
C CALL COMMARK
C CALL LOCMARK
C CALL CODMARK
C CALL INICJM
C CALL INICJM
C RETURN
END
APPENDIX E - INITIALIZATION OF PARAMETERS IMBEDDED IN SOLVER VISOL SOURCE PROGRAM AA3DS

*DECK AA3DS
program aa3ds
C
C
C
*CALL FORMARK
C
*CALL COMMARK
C
*CALL PARAM
C
*CALL LOCMARK
C
*CALL CODMARK
C
*CALL /CBPARM/
*CALL /CHRDAT/
*CALL /DIAGN/
*CALL /FILDAT/
*CALL /GRIDQ/
*CALL /INCANG/
*CALL /INCFLQ/
*CALL /ITERQ/
*CALL /MADAT/
*CALL /NUMRCN/
*CALL /PHYSNCN/
*CALL /SCRTCH/
*CALL /STDOP/
*CALL /STENCL/
C
C
CALL CSTPRT('BEGIN')
C
C
initialize data
C
block data inicom
C
block data initialization
C
initialize fundamental common block variables
C
/CBPARM/
C
/CHRDAT/
data revdat/'7/31/87'/
data title /'DEFAULT DEFAULT DEFAULT DEFAULT'/
data name /'DEFAULT DEFAULT DEFAULT DEFAULT'/
data chpol/'ee'/,chpw/'pw'/
C
/迪AGN/
data eps/1.e-8/,limit/0/,ndout/6/
55
DATA DEBUG/.FALSE./
C /FILDAT/
DATA NIN/5/, NOUT/6/, NERR/6/, NPRSL/7/, NFCJ/9/
DATA NFRCS/10/
DATA NFOUT/4/
DATA NFGRN/11/, IFSRCH/12/, IFOP/13/
DATA IFNJ/14/, IFJ/15/, IFGRH/16/, IFGAM/17/
DATA IFW/18/, IFDJ/19/
DATA NFMRCS/29/
DATA IFQC/20/, IFQB/21/, IFQBI/22/, IFQS/24/
DATA NFQW/25/, NFQWI/26/, IFQV/27/
DATA NFGRN/31/, IFQSB/32/
DATA NTIEL/28/
DATA ISEQ/1/, IRAN/2/, TEM/3/
C /GRIDQ/
DATA NX/17/, NY/17/, NZ/17/
DATA XI/0./, XF/1./, YI/0./, YF/1./, ZI/0./, ZF/1./
C /INCANG/
DATA NASW/1/, NANG(1)/1/
C /INCFIQ/
DATA AANG(1)/0./, AANG(2)/0./, AANG(3)/0./
DATA FREQ/3.E-8/, EMAG/(1.,0.)
C /ITERQ/
DATA TOL/1.E-10/
DATA NITER/30/, ITPRT/0/, KITER/20/, MSGLVL/3/
C /MATDAT/
DATA NMAT/2/
DATA ALPHA(1)/(0.,0./, ALPHA(2)/(0.,0./)
DATA BETA(1)/(0.,0./, BETA(2)/(0.,0./)
C /NUMRCN/
DATA CONE/(0.,1.)
C /PHYSNC/
DATA VLIGHT/2.997925E8/
C ***END OF INITIALIZATION DATA***
C CALL DATE(ADATE)
CALL CLOCK(ATIME)
CALL VHEAD(NOUT, 'VISOL: ', REVDAT)
CALL CSTPRT('INIT')
C SET UP MEMORY MANAGER
CALL SETRA(MAXSCR, SCRCLO)
C WRITE(NOUT,2)
2 FORMAT(5X, '***JUST BEFORE CALL SUBROUTINE READIN***')
C CALL READIN(IREAD)
WRITE(NOUT,3) IREAD
3 FORMAT(5X,'**IREAD==',I5)
WRITE(NOUT,4)
4 FORMAT(5X,'**BACK FROM SUBROUTINE READIN**')
CALL CSTPRT('READIN')
IF (IREAD.NE.0) THEN
WRITE(NOUT,6)
6 FORMAT(5X,'**JUST BEFORE CALL SUBROUTINE SETUP**')
CALL SETUP
WRITE(NOUT,8)
8 FORMAT(5X,'**BACK FROM SUBROUTINE SETUP**')
CALL CSTPRT('SETUP')
WRITE(NOUT,10)
10 FORMAT(5X,'**JUST BEFORE CALL SUBROUTINE SOLVER**')
CALL SOLVER
WRITE(NOUT,12)
12 FORMAT(5X,'**BACK FROM SUBROUTINE SOLVER**')
CALL CSTPRT('SOLVER')
WRITE(NOUT,14)
14 FORMAT(5X,'**JUST BEFORE CALL SUBROUTINE OUTPUT**')
CALL OUTPUT
WRITE(NOUT,16)
16 FORMAT(5X,'**BACK FROM SUBROUTINE OUTPUT**')
CALL CSTPRT('OUTPUT')
ENDIF
STOP
END
APPENDIX F

IEQUAL DUMMY SUBROUTINE

This appendix has the listing for subroutines CLUSS, CLUSM and IEQUAL which are all contained in VISOL source program AA3DS. IEQUAL is called in both CLUSS and CLUSM to keep the memory space available for vector array CB and KCB intact.
APPENDIX F - SUBROUTINES CLUSS, CLUSM AND IEQUAL FROM SOLVER VI SOL USED TO PRESERVE VECTOR ARRAYS IN MEMORY

**DECK CLUSS**

```fortran
SUBROUTINE CLUSS(N,NCB,KCB,CB,NQC,KC,NQB,KB,R,X)

C FORIJARK
C DIMENSION R(2,N),X(2,N)
DIMENSION KCB(NCB),CB(NCB),KC(KC),KB(KB)
C
*C COMMAARK
C C CODMARK
C
COMPLEX Y
DIMENSION Z(2)
C CODMARK
C CALL CCOPY(N,R,1,X,1)
KCB(1)=0
CALL FCHRB(NQC,KCB,NCB,KC,1,LC,MC,1)
DO 50 J=1,N
NREC=KCB(1)
IREC=J-KCB(NCB)-1
IF(IREC.LE.NREC) THEN
MC=KCB(NCB-IREC)
ELSE
CALL FCHRB(NQC,CB,NCB,KC,J,LC,MC,1)
ENDIF
CALL IEQUAL(CB,KCB)
NC=KCB(MC)
JC=MC+NC+3
Y=CPLX(X(1,J),X(2,J))*CMPLX(CB(JC),CB(JC-1))
X(1,J)=REAL(Y)
X(2,J)=AIMAG(Y)
IF(NC.GT.1) THEN
Z(1)=X(1,J)
Z(2)=X(2,J)
ELSE
Z(1)=-Z(1)
Z(2)=-Z(2)
CALL CAXPYI(NC-1,Z,CB(JC+2),CB(MC+4),X)
ENDIF
IK=MC+4
M=JC+2
CDIR$ IVDEP
DO 50 L=1,NC-1
K=KCB(IK)
X(1,K)=X(1,K)-Z(1)*CB(M)+Z(2)*CB(M+1)
X(2,K)=X(2,K)-Z(1)*CB(M+1)-Z(2)*CB(M)
M=M+2
IK=IK+1
50
```

```fortran

DO 50 J=1,N
NREC=KCB(1)
IREC=J-KCB(NCB)-1
IF(IREC.LE.NREC) THEN
MC=KCB(NCB-IREC)
ELSE
CALL FCHRB(NQC,CB,NCB,KC,J,LC,MC,1)
ENDIF
CALL IEQUAL(CB,KCB)
NC=KCB(MC)
JC=MC+NC+3
Y=CPLX(X(1,J),X(2,J))*CMPLX(CB(JC),CB(JC-1))
X(1,J)=REAL(Y)
X(2,J)=AIMAG(Y)
IF(NC.GT.1) THEN
Z(1)=X(1,J)
Z(2)=X(2,J)
ELSE
Z(1)=-Z(1)
Z(2)=-Z(2)
CALL CAXPYI(NC-1,Z,CB(JC+2),CB(MC+4),X)
ENDIF
IK=MC+4
M=JC+2
CDIR$ IVDEP
DO 50 L=1,NC-1
K=KCB(IK)
X(1,K)=X(1,K)-Z(1)*CB(M)+Z(2)*CB(M+1)
X(2,K)=X(2,K)-Z(1)*CB(M+1)-Z(2)*CB(M)
M=M+2
IK=IK+1
50
```
25 CONTINUE
C
ENDIF
50 CONTINUE
KCB(1)=0
CALL FCHRB(NQB,KCB,NCB,KB,N,MB,1)
DO 90 I=1,N
J=N+1-I
IREC=J-KCB(NCB)+1
IF(IREC.GE.1) THEN
MB=KCB(NCB-IREC)
ELSE
CALL FCHRB(NQB,CB,NCB,KB,J,MB,1)
ENDIF
CALL IEQUAL(CB,KCB)
NB=KCB(MB)
JB=MB+NB-3
IB=MB-3+NB-1
Y=CMPLX(X(1,J),X(2,J)) * CMPLX(CB(IB),CB(IB-1))
X(1,J)=REAL(Y)
X(2,J)=AIMAG(Y)
IF(NB.GT.1) THEN
Z(1)=X(1,J)
Z(2)=X(2,J)
C
Z(1)=-Z(1)
Z(2)=-Z(2)
C
CALL CAXPYI(NB-,Z,CB(JB),CB(MB-3),X)
C
IK=MB+3
M=JB
CDIR$ IVDEP
DO 75 L=1,NB-1
K=KCB(IK)
X(1,K)=X(1,K)-Z(1)*CB(M)+Z(2)*CB(M-1)
X(2,K)=X(2,K)-Z(1)*CB(M+1)-Z(2)*CB(M)
M=M+2
IK=IK+1
75 CONTINUE
C
ENDIF
90 CONTINUE
RETURN
END

*DECK CLUSM
SUBROUTINE CLUSM(N,NCB,KCB,CB,NQC,KC,X,R)
C
*CA FORMARK
C
DIMENSION R(2,N),X(2,N)
DIMENSION KCB(NCB),CB(NCB),KC(*)
C
*CA COMMARK
C
C
*CA LOCMARK
DIMENSION Z(2)

CALL CZERO(N,R,1)
KCB(1)=0
CALL FCHRB(NQC,KCB,NCB,KC,1,LC,MC,1)
DO 50 J=1,N
NREC=KCB(1)
IREC=J-KCB(NCB)+1
IF(IREC.LE.NREC) THEN
MC=KCB(NCB-IREC)
ELSE
CALL FCHRB(NQC,CB,NCB,KC,J,LC,MC,1)
ENDIF
CALL IEQUAL(CB,KCB)
NC=KCB(MC)
JC=MC-NC-3
Z(1)=X(1,J)
Z(2)=X(2,J)
CALL CAXPYI(NC,Z(CB(JC)),CB(MC+3),R)

M=JC
IK=MC+3

DO 25 L=1,NC
K=KCB(IK)
R(1,K)=R(1,K)+Z(1)*CB(M)-Z(2)*CB(M+1)
R(2,K)=R(2,K)*Z(1)*CB(M+1)+Z(2)*CB(M)
M=M+2
IK=IK+1
25 CONTINUE

50 CONTINUE
RETURN
END