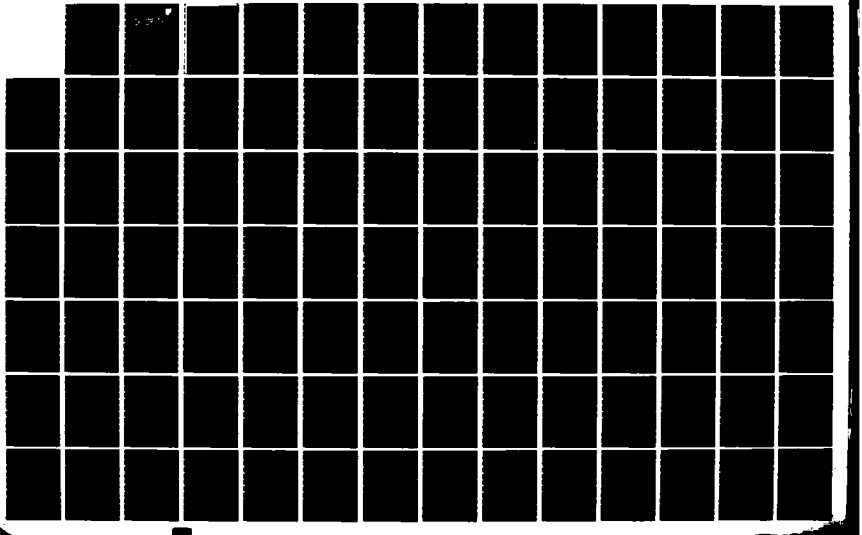
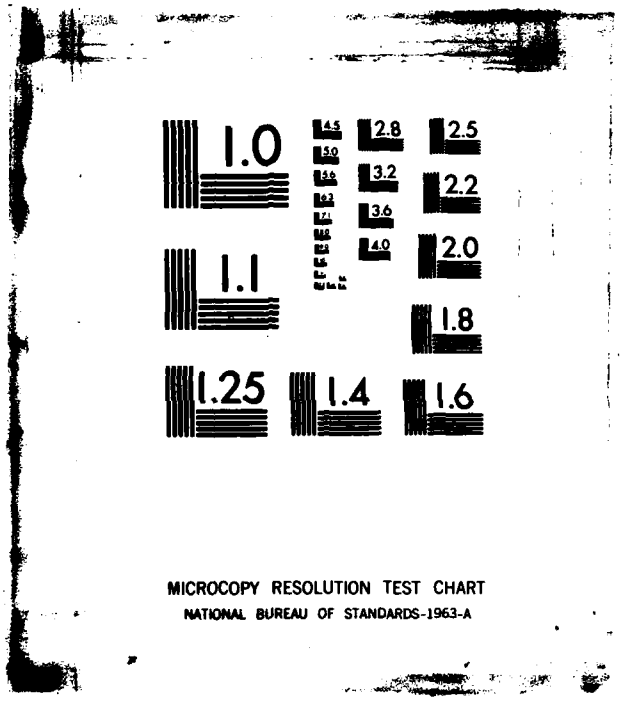
AD-A137 461 GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS..(U) BDM CORP ALBUQUERQUE NM D L KADLEC ET AL. SEP 83 BDM/A-83-170-TR 1/2

UNCLASSIFIED RADC-TR-83-217-VOL-1 F30602-81-C-0084 F/G 20/14 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RADC-TR-83-217, Vol I (of three)
Final Technical Report
September 1983

GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS) User Manual (Version 3)

The BDM Corporation

Dr. D. L. Kadlec and Dr. E. L. Coffey

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-83-217, Vol I (of three) has been reviewed and is approved for publication.

APPROVED: K R Siarkiewicz

KENNETH R. SIARKIEWICZ
Project Engineer

APPROVED: W S Tuthill

W.S. TUTHILL, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBCT) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-83-217, Vol I (of three)	2. GOVT ACCESSION NO. AD-A137 461	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS) USER MANUAL (VERSION 3)		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report February 81 - July 83 6. PERFORMING ORG. REPORT NUMBER BDM/A-83-170-TR
7. AUTHOR(s) Dr. Diana L. Kadlec Dr. Edgar L. Coffey		8. CONTRACT OR GRANT NUMBER(s) F30602-81-C-0084
9. PERFORMING ORGANIZATION NAME AND ADDRESS The BDM Corporation 1801 Randolph Road, S.E. Albuquerque NM 87106		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 23380333
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBCT) Griffiss AFB NY 13441		12. REPORT DATE September 1983 13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Kenneth R. Siarkiewicz (RBCT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Electromagnetic Compatibility
Method of Moments (MOM)
Geometrical Theory of Diffraction (GTD)
Antenna Analysis
Matrix Equation Solution
MOM/GTD Hybridization
EM Radiation and Scattering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

GEMACS solves electromagnetic radiation and scattering problems. The Method of Moments (MOM) and Geometrical Theory of Diffraction (GTD) are used. MOM is formalized with the Electric Field Integral Equation (EFIE) for wires and the Magnetic Field Integral Equation (MFIE) for patches. The code employs both full matrix decomposition and Banded Matrix Iteration (BMI) solution techniques. The MOM, GTD and hybrid MOM/GTD techniques in the code are used to solve electrically small object problems, electrically

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

large object problems and combination sized object problems.

Volume I of this report is the User Manual. The code execution requirements, input language and output are discussed.

Volume II is the Engineering Manual. The theory and engineering approximations implemented in the code are discussed. Modeling criterion are given.

Volume III is the Computer Code Documentation Manual. This manual contains extensive software information of the code.

FOREWORD

This document was prepared by The BDM Corporation, 1801 Randolph Road SE, Albuquerque, NM 87106, for Rome Air Development Center/RBCT, Griffiss AFB, NY 13441, under contract number F30602-81-C-0084. This is the User Manual for version 3 of the GEMACS computer code. The Engineering Manual and Code Documentation Manual are submitted under separate covers. These reports are submitted in accordance with Item Number A003 in The Contract Data Requirements List. The program manager on this contract was Diana L. Kadlec. The principal investigator was Dr. Edgar L. Coffey. Mariellen Kuna was the documentation editor.

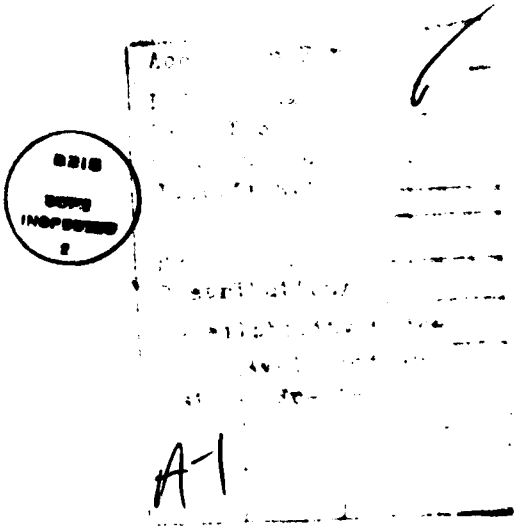


TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

A. INTRODUCTION

This manual contains instructions for using the General Electromagnetic Model for the Analysis of Complex Systems (GEMACS) computer program. The program is a highly user-oriented general purpose code designed for gradual incorporation of a variety of techniques for electromagnetic analysis of complex systems. The user is assumed to be an experienced electromagnetics analyst with a fair understanding of applied linear algebra. The current version (version 3) of the code supports all of the functions necessary for using one thin-wire and one surface patch (Method of Moments (MOM)) formalism with or without Geometrical Theory of Diffraction (GTD) interactions. GEMACS is implemented in four sequentially executable FORTRAN programs (called "modules"). The use of the modules in running GEMACS is described in section D. The GEMACS code uses a high level language and provides flexibility of control over the computational sequence by the user. Error messages, debug and trace options, and other features are included to aid the user in identifying sources of fatal errors.

1. Method of Moments Formulations

One MOM formalism used in the present code includes the thin-wire Pocklington integral equation, pulse plus sine plus cosine expansion functions, point matching, and a charge redistribution scheme at multiple wire junctions. This is the same formalism as used in the AMP (Antenna Modeling Program) code*. The GEMACS code includes most of the engineering features of the AMP code such as loading and ground plane effects. However, the range of applicability of the moments technique is extended to objects of larger electrical size in the GEMACS code by using a solution method for linear simultaneous equations called BMI (Banded Matrix Iteration). The user must have a limited understanding of the solution

*The Antenna Modeling Program is a general-purpose thin wire code developed and documented by MB Associates in *Antenna Modeling Program-Engineering Manual,* July 1972, AD-A025890.

method to ensure convergence and reasonable efficiency. The method is documented in Volume II (GEMACS Engineering Manual) of this report and in references noted therein. The thin-wire MOM can be used to solve general physical problems involving actual wires, wire grid models of conducting surfaces, or a combination of these. Wire grid modeling is not yet a highly defined process. Modeling guidelines developed in recent studies are discussed in Volume II. The user must reduce the physical problem to a thin-wire model. The GEMACS code includes a highly flexible geometry processor to aid in this task. The user specifies the frequency, additional features such as loading or the presence of a ground plane, and the excitation. Excitation options currently include plane or spherical waves, voltage sources for antennas, or arbitrary excitations on specified individual wire segments. Load options currently include fixed (as a function of frequency) lumped loads, series or parallel RLC networks, and finite segment conductivity.

A second MOM formalism is the use of the Magnetic Field Integral Equation (MFIE). Here the surface current is expanded in a set of pulse expansion functions, except in the region of a wire connection. Two orthogonal current directions are assumed for each surface patch. Point matching is used at the patch centers. In the region of a wire connection to a patch four subpatches are generated, and the continuity of current equation at the center of the patch takes into account the singular component due to the current flowing from the wire into the surface. The MFIE therefore provides a viable alternative to the wire grid modeling approach for a surface. (See section C.3.g of the GEMACS Engineering Manual.)

GEMACS can also use the physical symmetry of a MOM structure to decrease matrix fill time and matrix equation solution time. The symmetry may be either planar or rotational. Since only the physical symmetry is used, no restriction is placed on the excitation or loading of the structure regarding symmetry. (See section F.2 of the GEMACS Engineering Manual.)

2. Geometrical Theory of Diffraction Formulation

The GTD formalism used in the present code makes it possible to model electrically large objects in bulk instead of requiring the gridding of the object into a large number of wire segments or patches. Fields scattered by the object are determined by optic principles: ray tracing and reflection coefficients. The GTD extension* to basic geometrical optics principles adds diffracted ray paths, including creeping waves, and diffraction coefficients.

The present GTD capability in GEMACS contains the same computational procedures found in the BSC (Basic Scattering Code),** although much has been added to BSC to make it compatible with the GEMACS impedance matrix formalism. While GTD is a very powerful technique, the user must have some physical insight into the problem he is solving in order to interpret correctly the results obtained. This is particularly true when one sees field discontinuities at object shadow boundaries.

To use GTD techniques, the user must reduce his object geometry to a set of canonical objects: planar plates, a cylinder, and the cylinder's two endcaps. These shapes are described to GEMACS through the code's geometry processor. Next, the field sources which generate the field incident on the object, as well as the frequency of the source, are specified. Rather than calculate geometry element currents to obtain field pattern data (the MOM process), the scattered fields are obtained directly from the sources and geometry by tracing all geometrical optics paths from the sources to the field points and reflecting and diffracting the waves which follow these paths from the surfaces, edges, and corners of the geometry elements. With the GTD method one may find the geometry surface currents indirectly from the total fields at the surface, but not antenna impedances.

---

*J. B. Keller, "Geometrical Theory of Diffraction", *J. Opt. Soc. Am.*, 52:116, 1962.

**R. J. Marhefka and W. D. Burnside, "Numerical Electromagnetic Code - Basic Scattering Code Part I: User's Manual," The Ohio State University, September 1979.

GTD techniques are used when detailed structure elements are not important and when scattered fields are the only quantities desired. While theoretically it is possible to obtain any level of detail with enough GTD geometry elements, there is a practical limit to the detail that can be obtained because of the large number of second and third order interactions possible among the GTD elements. In practice, therefore, the user is forced to limit either the number of GTD elements or the number of physics interactions he wishes to consider. Guidance on these and other modeling considerations is provided in Volume II.

A limitation of the GTD formalism as implemented in GEMACS is that no advantage can be taken of symmetry. The full GTD geometry must be specified to GEMACS. Furthermore, all GTD physics assumes perfectly conducting geometry elements. Hence no dielectrics or surface impedance loadings are permitted. Perfectly conducting ground, however, can be accommodated by using a planar plate.

3. MOM/GTD Hybrid Formulation

The presence of both thin wire and large conducting elements on a geometry to be modeled presents a dilemma to the user if only MOM or GTD techniques are available. For this type of problem, the two solution methodologies have been combined into a hybrid formulation after the method of Thiele.* Thin wires are modeled with MOM wire segments and large conducting objects are modeled with GTD geometry elements. The impedance interactions normally computed in the MOM approach are augmented with fields generated from wire segments, scattered from the GTD objects, and received by the wire segments.

The formal mechanism of the hybrid formulation is transparent to the user; the only extra step is to divide his geometry into MOM and GTD elements. The same commands used to generate the interaction and excitation matrices for the MOM problem are used for the MOM/GTD solution.

---

*G. A. Thiele and T. H. Newhouse, "A Hybrid Technique for Combining Moment Methods with the Geometrical Theory of Diffraction," *IEEE Transactions on Antennas and Propagation*, January 1975.

The primary computational difference in the MOM/GTD technique is the small size of the matrices. For example, the interaction matrix size is determined from the number of MOM elements regardless of the number, size, or complexity of the GTD geometry. Hence, a blade antenna mounted on a B-52 aircraft would require the same size interaction matrix as the same antenna in free space. Each matrix element, however, consists of an interaction made up of a direct (MOM) interaction and (perhaps hundreds) of indirect (GTD) interactions. While each matrix element takes longer to generate, the small matrix size makes computation of the solution more efficient than attempting to grid the entire geometry with MOM objects and solving a huge set of simultaneous equations.

The restrictions on the hybrid formulation contain all the restrictions of both MOM and GTD formulations. The user is referred to Volume II for a discussion of these and the few additional restrictions applicable to the hybrid method.

4. Numerical Solution Methods

When executing a MOM or MOM/GTD problem, the code generates a set of linear simultaneous equations from the information provided. The user controls the process by which the equations are solved. If the total number of wire segments in the model is sufficiently small, standard solution methods are applicable. Solution by full matrix triangular decomposition is one of the least expensive general methods and is supported by GEMACS. For large problems, even this method is too expensive, and the BMI solution method should be specified by the user. This method is considerably less expensive provided the user carefully chooses the segment numbering and matrix bandwidth according to the guidelines discussed in Volume II.

The user may specify other quantities to be computed from the wire currents, such as impedances, coupling parameters, near fields or far fields. These are computed from currents regardless of the solution process specified. Regardless of the solution technique exercised, it is emphasized that the user must be familiar with general results from the

literature to ensure that the computed solution using the model for the system is of sufficient accuracy for the intended purpose. For example, the far fields can be computed from approximate currents obtained by specifying a weak convergence criterion when using the BMI solution method. This will allow the reduction of the required computer resources when large systems are being analyzed. Near-field computations will require more accurate wire currents, obtainable only with stricter, more costly convergence criteria.

The present code generates an interaction matrix from the MFIE and EFIE (Electric Field Integral Equation) discussed in the GEMACS Engineering Manual. The wire current is represented by a sine, cosine, and pulse expansion function with redistribution at junctions based on the fractional length of each segment with respect to the total length of all segments connected at the junction. The surface current is represented by a pulse function. The interaction matrix may be modified by loading the individual wire segments of the model using resistance, capacitance, and inductance in parallel or series configurations.

Associated with the geometric structure and interaction matrix is an excitation matrix which contains the total tangential electric field present at the midpoint of each segment or patch. The electric field may be caused by as many combinations of three types of sources as desired. These types are plane and spherical wave sources for scattering problems and voltage sources for antenna problems. In addition, the user may assign an arbitrary value to the excitation of any wire segment to force the desired boundary condition.

With the interaction matrix denoted by [Z] and the excitation matrix denoted by [E], the primary function of the code is to generate and solve the system of equations for the electric current [I].

$$[Z]\ [I] = [E]$$

This may be done using direct full matrix decomposition in which [Z] is decomposed into lower and upper triangular matrices [ZL] and [ZU]. Forward elimination and back substitution are then performed as indicated below.

$$[Z]\ [I] = [E]$$
$$[ZL]\ [ZU]\ [I] = [E]$$
$$[ZL]\ [I'] = [E]$$
$$[ZU]\ [I] = [I']$$

where, since [ZL] and [ZU] are triangular, the actual inverse is not required.

For very large problems, the direct solution method may be prohibitive due to the large amount of time required and the possible roundoff errors. In this case, the Banded Matrix Iteration (BMI) technique is available. In this method, [Z] is partitioned (<u>not</u> decomposed) into lower and upper triangular matrices ([LZ] and [UZ]) and a central or banded matrix [BZ]. This is illustrated graphically below.

$$[Z] = \begin{bmatrix} & & UZ' \\ & BZ' & \\ LZ' & & \end{bmatrix}$$

$$[Z] = [LZ] + [BZ] + [UZ]$$

where LZ', BZ', UZ' are the nonzero elements of LZ, BZ, and UZ. In this notation, the problem to be solved becomes:

$$[Z]\ [I] = [LZ]\ [I] + [BZ]\ [I] + [UZ]\ [I] = [E]$$

or:

$$[BZ]\ [I] = [E] - ([LZ] + [UZ])\ [I]$$

The banded portion of interaction matrix [BZ] should contain the dominant interactions while those contained in [LZ] and [UZ] may be viewed as perturbations in a properly posed problem. In this case, the BMI technique involves appending subscripts to the [I] vector and only solving the banded portion of the interaction matrix.

$$[BZ]\ [I]_n = [RHS]_n$$

where the right-hand side at the n^{th} iteration $[RHS]_n$ is given by:

$$[RHS]_n = [E] - ([LZ] + [UZ])\ [I]_{n-1}$$

The starting value for [I] or $[I]_0$ is zero unless preset by the user. There is very little if any advantage to presetting [I]. Since [BZ] is usually much smaller than [Z], the time to perform lower/upper decomposition is reduced considerably and the system to be solved becomes:

$$[BZL]\ [BZU]\ [I]_n = [RHS]_n$$

where [BZL] and [BZU] are the lower and upper triangular matrices obtained by decomposing [BZ]. The solution for $[I]_n$ is obtained exactly as for the full matrix decomposition. When using BMI, the user must provide the convergence measure and value to be used to stop the iterative procedure. Three criteria or measures are available, the BCRE (Boundary Condition Relative Error), the IRE (Iterative Relative Error), and the PRE (Predicted Relative Error). The BCRE is a measure of how well the solution matches the boundary condition. Mathematically:

$$BCRE = \frac{|[E] - [Z]\ [I]_n|}{|[E]|}.$$

While this form has great engineering appeal, mathematically it is not recommended since the system of equations may be ill-conditioned near resonances, and there is, by definition, a large variance in the elements of [I] which will result in a small BCRE.

The second criterion is the IRE. This is defined mathematically as:

$$IRE = \frac{|[I]_n - [I]_{n-1}|}{|[I]_n|}$$

and can be seen to be the relative change between successive approximations to the solution. For slowly converging problems, this criterion may cause premature termination of the iterative procedure.

Finally, the PRE may be used. This quantity is determined by using an exponential fit to the two previous values for the IRE and has been shown to be a good approximation to the ARE (Actual Relative Error) after four iterations. The ARE is defined as:

$$ARE = \frac{\|[I]_n - [I]\|}{\|[I]\|}$$

where [I] is the exact solution. Since the exact solution is not available, the PRE is the recommended criterion. (See the GEMACS Engineering Manual, section F.1.3 for a discussion of the PRE.)

The value of the convergence criterion depends largely on the output desired. If input impedance or near-field parameters are desired, a 1 percent value is not inappropriate; however, if normalized far-field patterns are desired, a 10 to 15 percent value may be sufficient.

Once the solution has been obtained, the input impedance of each voltage driven element (i.e., Antenna Feed Point) is output to the user. These are computed as $Z_a = \frac{V_a}{I_a}$ since a delta gap model is used for antenna sources. The currents may be used as inputs to the field computation routines to obtain the near- and/or far-electric field patterns, and the coupling between pairs of antennas. The user should note that the impedance value printed is equal to the antenna input impedance only when a single voltage driven element is specified. Otherwise Z_a is merely the ratio of element voltage to element current and cannot be related to an input impedance.

The above discussion is relevant only to MOM or MOM/GTD hybrid problems. A problem utilizing only GTD geometries does not require a numerical solution, since no simultaneous equations are generated. Rather, the GTD solution (radiated and scattered fields) is obtained directly from the ray tracing, reflection, diffraction, and shadowing

computations that take place where GTD fields are requested. In this case surface currents and input impedances are not calculated by the code.

5. GEMACS Capabilities Summary

The present version of GEMACS supports three electromagnetic solution methods and two numerical techniques. These are:

Method of Moments (wire gridding, patches)

Geometrical Theory of Diffraction (Ray tracing, optics, diffraction)

A MOM/GTD Hybrid

Matrix Triangular Decomposition

Banded Matrix Iteration

There are inherent limitations to the solution techniques available. The user who is not familiar with these techniques should consult the Engineering Manual and its references to avoid wasting valuable time and computer resources working an ill-posed problem.

B. COMPUTATIONAL APPROACH

The basic approach in the design of the GEMACS code is to permit the user to generate or define data sets and then to perform operations on the data contained in the data sets. The data sets are identified using symbolic names of six ANSI FORTRAN characters or less. All names must start with an alphabetic character. The result of this approach is a code which the user completely controls down to the functional or operational level. Associated with each symbol is a set of symbol characteristics referred to as attributes. These attributes are generated as the data associated with the symbol are generated or modified by an operation. The attributes are checked each time the data are used in an operation. This insures the integrity of the resulting data and the sanity of the operation. For example, if the user attempts to generate an interaction matrix using data which do not represent a geometrical structure, an error will occur since the operation is not defined. However, if an interaction matrix is generated for a properly defined geometry data set, it will have the attributes of a complex interaction matrix and will be identified as having been generated from the geometry data set specified. In this way, a symbol's lineage and the type of data (both physical and computational) are known. Likewise, the solution vector for a geometry data set is linked to the interaction matrix and thus to the geometry which was the parent data for all subsequent operations. Therefore, when field output is desired, the code will retrieve the correct electrical current data for use with the geometry data specified.

The primary function of the GEMACS overhead routines is to store and retrieve the data sets in order that user specified operations may be performed. The result of these operations will be a solution or analysis of the system described to the code.

As more operations are added to the GEMACS code, it is necessary to define the attributes of the data needed to assure correct program operation, modify the Input Module to recognize the command and add the

software to perform the operation. The data handling will be taken care of by the executive level programs in the GEMACS code.

As an example, a normal MOM scattering solution at a single frequency for which the far-field radiation pattern is desired could be done with the following input stream. The directives are discussed in the following section.

```
COMMAND
   1      GMDATA = GEOM1
   2      EINC = ESRC (GEOM1), FRQ = 180.0, SW = 1.,0.,THETA = 45.
   3      ZGEN GMDATA = GEOM1, ZMATRX = ZMAT1
   4      BNDZ1 = BAND (ZMAT1), BNDW = 50
   5      BNDZ1 = LUD (BNDZ1)
   6      BNDZ1 * CUR1 = EINC - ZMAT1 * CUR1, MAXITR = 10, CONVRG = PRE,
          VALUE = 20
   7      EFIELD (CUR1), LOGPLR, P1 = 0.  P2 = 180.  DP = 10.  T1 = 90.
   8      END
          {GEOMETRY DATA}
          END
```

Card 1 directs the geometry processor to generate the data to be associated with symbol GEOM1. Card 2 directs that data associated with symbol EINC be generated by a spherical wave at 180 MHz incident on the geometry specified by GEOM1 at a spherical angle theta of 45^{o}. Card 3 directs that data associated with interaction matrix ZMAT1 be generated using the sine + cosine + pulse expansion function on the geometry associated with GEOM1. Card 4 causes extraction of the elements from ZMAT1 which are located within 50 elements of the main diagonal and associates the data with symbol BNDZ1. Card 5 results in BNDZ1 being decomposed into upper and lower triangular matrices and the result stored in BNDZ1.

Card 6 invokes the BMI solution technique to obtain the data for symbol CUR1. The procedure is limited to 10 iterations and the PRE convergence criterion will be used. When the PRE $\leq$ 20 percent, the procedure will stop and return to execute the next directive. Card 7 directs the computation and output of the far field of structure GEOM1. The field will be computed at 10^{o} intervals from $\phi = 0$ to $\phi = 180$ for $\theta = 90^{o}$. In addition to a tabular print, the $\log_{10}$ of the field will be plotted on a polar graph. Card 8 indicates the end of the directives and is followed by the geometry data input which is also terminated by an *END* card. The input is completely free field and there are default values for most parameters.

All of the directives are processed before any execution begins. This precludes wasting considerable computer resources in the event of an error in the command directives.

C. GEMACS COMMANDS AND GEOMETRY LANGUAGE

The GEMACS inputs are in two categories. The command language directs the program execution while the geometry language describes the geometrical properties of the structure being analyzed.

The GEMACS command language is a free-field, keyword oriented input stream. The order of the data is generally not important, and the items on each card are delimited by a blank or a comma. An item is considered to be all of the input associated with a particular parameter such as THETA = 90. Note that an item may consist of several entries, and each entry is referred to as a field. Blanks may be imbedded between fields of an item but not within a field. Thus, THETA = 90. is acceptable while THETA = 9 0. is incorrect and will be interpreted as two items (i.e., THETA = 9 and 0). This extraneous item would be detected by the code and execution would be inhibited.

In order to prevent wasting computer resources, all of the GEMACS commands are read by the INPUT module and checked for errors prior to execution of the code. All errors in the user commands are identified; that is, one error does not preclude location of any other error during the same execution. This prevents the need for the user to make several submittals to debug the input.

The GEMACS geometry language is also a free-field language. However, the items must appear in the order specified or an error will occur which may not be detected. The reason for not using keyword-specified items on the geometry inputs is to decrease the effort required by the user since the geometry inputs are usually much larger than the command inputs.

For both input types there are several standard conventions. These relate to comment cards, comments on cards, and continuation cards. Comment cards are those cards which have a $ as the first nonblank character. Likewise, comments may be appended to a command or geometry input by preceding the comment with a $. If the last character encountered before a $ or the end of a card is a comma or arithmetic operator

(+,-,*,/), the next card must be a continuation card. If a card has a continuation character in column 1, it is treated as a continuation of the previous card. All continuation cards must have a continuation character in column 1. The continuation character may be installation dependent. It is an asterisk (*) in version 3. Other possible choices are the other arithmetic operators. The actual character is defined as variable NCONCH and is set in subroutine BLKDAT.

1. GEMACS Command Language

In describing the GEMACS command language, items enclosed in brackets [ ] have default values and need not be specified if the default value is acceptable to the user. Items enclosed in braces { } indicate a multiple choice. The only restrictions on symbolic names provided by the user are that they be six characters or less, the first character be a member of the alphabet (A-Z), and only characters contained in the alphabet or the digits (0-9) be present. That is, the characters =, +, -, *, /, $, and comma are not allowed. In addition, the following reserved keywords may not be used for symbolic names.

ONE LETTER KEYWORDS

A	C	D	N	O	R	V	X	Z

TWO LETTER KEYWORDS

CD	CR	CS	CW	CY	C1	C2	DC	DM	DP
DR	DT	DW	DX	DY	DZ	EC	ED	EI	ER
ES	EU	IS	LU	MM	NP	NR	ON	PC	PD
PL	PR	P1	P2	RC	RD	RR	R1	R2	SC
SW	T1	T2	VS	X1	X2	Y1	Y2	Z1	Z1

THREE LETTER KEYWORDS

ABS	CDP	ECC	END	FRQ	GTD	ILP	INV	IRE
LUD	OFF	PDR	PHI	PRE	RDP	SEQ	SET	

FOUR LETTER KEYWORDS

AXIS	BAND	BCRE	BNDW	COND	EPSR	ESRC
LOOP	PLOT	PRLC	READ	SCDP	SEGS	SIZE
SRDP	SRLC	TAGS	TIME	TYPE	VSRC	ZGEN
ZIMP						

FIVE LETTER KEYWORDS

CONJG	CPINC	CPNUM	DEBUG	INPUT	LABEL	PARTN
PIVOT	PRINT	PULSE	PURGE	SOLVE	STATS	
THETA	TRACE	VALUE	WRITE			

SIX LETTER KEYWORDS

BACSUB	CHKPNT	COLPSE	CONVRG	EFIELD	EXPAND
FILEID	GMDATA	LINLIN	LINLOG	LINPLR	LOGLIN
LOGLOG	LOGPLR	MAXITR	MODULE	NUMFIL	OUTPUT
PCESIN	REDUCE	REFLCT	REPLAC	RSTART	SETINT
SINCOS	SYMDEF	TRANSP	WIPOUT	ZCODES	ZLOADS
ZMATRX					

In addition to these names, three other cases are to be avoided. If an out-of-core matrix is to be decomposed and is identified symbolically, as XXXXXX where the Xs are legitimate characters, then the user must not define another data set with either XXXLWR, XXXUPR, or XXXPVT. These names will be internally generated to contain the lower triangular decomposed matrix, the upper triangular decomposed matrix, and the pivot vector as specified. These names may be referenced in output statements. However, in general, the user should simply ignore them. It is also correct to assume that a matrix which resides out of core is not destroyed when it is decomposed. However, unless a matrix is large, it will be difficult for the user to know a priori where it will be stored.

The second case is when an interaction matrix is generated with GTD interactions specified. A shadowing matrix is generated by the code, whose symbolic name is XXXSHD, where the Xs are the leftmost characters of the interaction data set name.

The third case involves generation of GTD scattered fields. Here, the code generates a Green's function scattering matrix XXXGFM, where the Xs are the leftmost characters of the field matrix data set name.

A list of commands is given in table 1 in which the following codes are used:

S - Previously undefined symbol

DS - Previously defined symbol

SDS - Either S or DS

N - Numeric value

DSN - Either a DS or numeric value

KW - Keyword

A - Alphameric word of at most six characters starting with a character A-Z and containing only characters A-Z and 0-9.

TABLE 1. LIST OF COMMANDS, FORMATS, AND MNEMONICS

COMMAND	FORMAT	MNEMONIC
FORWARD ELIMINATION/BACK SUBSTITUTION	BACSUB DS1*SDS = DS2	BACSUB
CONSTRUCT BANDED MATRIX	SDS = BAND (DS), BNDW = N	BAND
BMI SOLUTION PROCESS	DS1 * SDS1 = DS2 - DS3 * SDS2, MAXITR = N [,CONVRG = {BCRE / IRE / PRE}] [, VALUE = N]	BMI
CHECKPOINT COMMAND	CHKPNT [LU = N] [,CPINC = N] [,NR]	CHKPNT
DEBUG COMMAND	DEBUG {ON / OFF / TRACE / STATS} [,ILP]	DEBUG
ARITHMETIC OPERATION (SCALAR QUANTITIES ONLY)	SDS=DSN1 {+ / - / * / / / **} DSN2	DMP
ELECTRIC FIELD OUTPUT	SDS = EFIELD (DS) [, {LINLIN / LINLOG / LOGLIN / LOGLOG / LINPLR / LOGPLR}] [,U2=N] [,DU=N] [,V2=N] [,DV=N] [,W2=N] [,DW=N] [,U1=N] [,V1=N] [W1=N]	EFIELD
END OF COMMANDS	END	END

TABLE 1. LIST OF COMMANDS, FORMATS, AND MNEMONICS (Continued)

COMMAND	FORMAT	MNEMONIC
ELECTRIC FIELD EXCITATION	SDS = ESRC [(DS)] [,FRQ = DSN] SW = DSN, DSN [,R = DSN] [,THETA = DSN] [,PHI = DSN] [,ECC = DSN]	ESRC
GENERATE A STRUCTURE GEOMETRY	GMDATA [= S] [,LU = N]	GMDATA
COMMAND REPETITION	LOOP {A / N} N LABEL {A / N}	LOOP/LABEL
MATRIX DECOMPOSITION	SDS = LUD (DS)	LUD
PLOT GEOMETRY DATA SET	PLOT DS	PLOT
PRINT DATA COMMAND	PRINT SDS, SDS, . . . SDS	PRINT
PURGE DATA COMMAND	PURGE SDS, SDS, . . . SDS	PURGE
CHECKPOINT RESTART	RSTART [LU = N] [,MODULE = A] ,CPNUM = N	RSTART
DATA INITIALIZATION AND MODIFICATION	SET SDS = N [,N] [,R1 = N] [,R2 = N] [,C1 = N] [,C2 = N]	SET
SELECT THE PHYSICS INTERACTIONS FOR MATRIX GENERATION	SETINT [LIST OF INTERACTIONS]	SETINT

TABLE 1. LIST OF COMMANDS, FORMATS, AND MNEMONICS (Concluded)

COMMAND	FORMAT	MNEMONIC
SOLVE SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS	SOLVE DS1* SDS = DS2	SOLVE
VOLTAGE OR ANTENNA EXCITATION	SDS = VSRC [(DS)] [,FRQ = DSN] ,V = DSN, DSN, {TAGS / SEGS} = N, N, ..., N	VSRC
COMMAND STREAM MODIFICATION	WIPOUT N	WIPOUT
DATA OUTPUT COMMAND	WRITE DS [,LU = N] [,R1 = N] [,R2 = N] [,C1 = N] [,C2 = N] [,FILEID = A]	WRITE
INTERACTION MATRIX GENERATION	ZGEN [,GMDATA = DS] [,FRQ = DSN] ,ZMATRX = S [,ZLOAD = DS] [,COND = DSN] [,EPSR = DSN]	ZGEN
STRUCTURE LOADING	ZLOADS = SDS [,GMDATA=DS] , { COND = DSN / ZIMP = DSN, DSN / {PRLC / SRLC} = DSN, DSN, DSN } {TAGS / SEGS} = N, N, ...N	ZLOADS

BACSUB

Forward Elimination/Back Substitution Command

BACSUB DS1*SDS = DS2

This command causes the solution vector for a previously decomposed matrix DS1 to be found for the excitation vector DS2 and stored as symbol SDS. For example, assuming SRC1, SRC2, and SRC3 have been previously defined, then the solutions SOL1, SOL2, and SOL3 would be obtained by:

```
ZM = LUD(ZIJMAT)
BACSUB ZM*SOL1 = SRC1
BACSUB ZM*SOL2 = SRC2
BACSUB ZM*SOL3 = SRC3
```

This command permits the user to obtain several solutions with only one decomposition. Note that this is in contrast to multiple uses of the SOLVE command, each of which would initiate its own decomposition process.

BAND

Construct Banded Matrix

SDS = BAND (DS), BNDW = N

This operation causes the data associated with the matrix DS which are within N elements of a diagonal element to be transferred to the symbol identified as SDS. Note that SDS and DS may not be the same symbolic name. This operation is typically used to construct the banded matrix for use in the BMI solution process. This operation is illustrated in the figure.

Examples:

BNDZIJ = BAND (ZIJMAT), BNDW = 50

This operation will construct a banded matrix from the data associated with ZIJMAT.

DIAG = BAND (ZIJMAT), BNDW = 0

This operation will extract the diagonal elements from ZIJMAT and store them as DIAG.

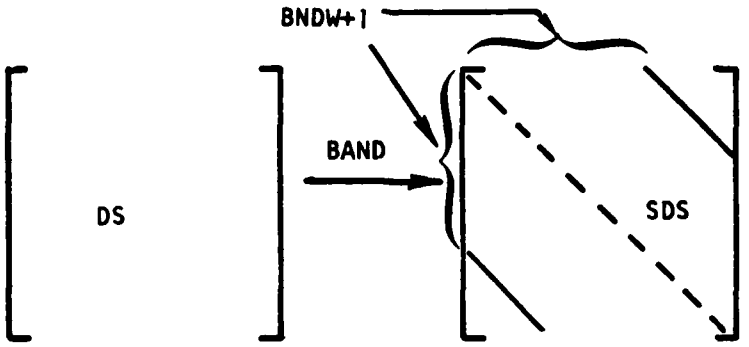


Illustration of BAND Operation

BMI

BMI Solution Process

$$\text{DS1} * \text{SDS1} = \text{DS2} - \text{DS3} * \text{SDS2},\ \text{MAXITR} = \text{N}\ \left[,\text{CONVRG} = \begin{Bmatrix}\text{BCRE}\\ \text{IRE}\\ \text{PRE}\end{Bmatrix}\right]$$

$$\left[,\text{VALUE} = \text{N}\right]$$

This command executes the BMI solution process. DS1 must be a banded decomposed matrix whose elements were originally contained in DS3. The solution will be stored in SDS1 upon completion. DS2 is the excitation or right-hand side of the original system of simultaneous equations and DS3 is the original interaction matrix of coefficients. DS3 will still contain the elements which were used to generate the banded matrix, but they will be ignored. The symbol SDS2 may be predefined and preloaded or it may be SDS1, in which case it will be initialized to zero. The convergence parameter to be used is contained in the CONVRG item. Their relative merits and definitions of the three convergence parameters are described in Volume II, GEMACS Engineering Manual. Default convergence item is CONVRG = PRE. The value in percent which the convergence parameter must reach is contained in the VALUE item. The default item is VALUE = 1. The MAXITR parameter defines an upper bound on the number of iterations. There is no default value for this item. It must appear before the CONVRG and VALUE items.

Example:

```
LUDZIJ * CUR = VDRV - ZIJMAT * CUR, MAXITR = 5 CONVRG = BCRE,
VALUE = 10
```

CHKPNT

Checkpoint Command

CHKPNT [LU = N] [,CPINC = N] [,NR]

This command designates the FORTRAN logical unit (LU) N in the LU = N item to be used to receive the checkpoint data. The default item is LU = 7, and, if the user specifies a different logical unit, he must assure the availability of the unit to the GEMACS code. The first word written on the checkpoint file will be the name of the module generating the checkpoint. Checkpoints will be taken at time increments of N CP minutes specified in the CPINC = N item. If the item is not specified, an immediate checkpoint is written when the command is encountered during execution. This type of command will not change the checkpoint increment specified on a previous command. Multiple CHKPNT commands may be used to vary the checkpoint increment and logical unit during execution.

Checkpoints are accomplished by writing all data contained in named commons and all data associated with symbolic names to the checkpoint file. Large problems can generate a very large amount of data and it is advisable to avoid using magnetic tapes for the checkpoint file since multiple reels may be required. Also, a large CP increment is recommended unless large data blocks are PURGED when no longer needed.

A historical record of checkpoint information is kept if the NR parameter is specified; otherwise, the checkpoint file is rewound after each checkpoint and overwritten with subsequent checkpoints. Due to the large amount of data, use of the NR parameter is not recommended. If the NR parameter is used, checkpointing should be controlled directly from the command language by omission of the CPINC item. This is the only mode in which restarting can be guaranteed with known data for multiple checkpoints on the same file in version 3 of GEMACS.

The default checkpoint file is rewound prior to writing the end-of-module checkpoint. If intermediate checkpoints are to be saved, an alternate logical unit must be specified in the LU = N item.

DEBUG

Debug Command

$$\text{DEBUG} \left\{ \begin{matrix} \text{ON} \\ \text{OFF} \\ \text{TRACE} \\ \text{STATS} \end{matrix} \right\} \quad [,\text{ILP}]$$

This command is used to obtain diagnostic information during program execution. Specifying the ON parameter causes all available information associated with the subsequent tasks to be printed. When the TRACE parameter is specified, the printout will include subroutine entry and exit information to allow the user to follow the program flow. When STATS is specified, the subroutine timing and entry statistics are accumulated and output upon completion of the execution. When OFF is specified, the program returns to the normal mode. The parameter ILP may be specified if the user needs to obtain diagnostic information during the execution of the Input Module.

Examples:

1.
```
DEBUG ON
BACSUB Z * I = V
DEBUG OFF
```

This command stream will cause a detailed printout to occur during execution of the BACSUB command.

2.
```
DEBUG ON,ILP
BACSUB Z * I = V
DEBUG OFF
```

This will cause a detailed printout to occur during input processing and during execution of the BACSUB command.

3.
```
DEBUG ON,ILP
END
```

This will cause all of the input language processes and execution tables to be printed on termination of input processing and before execution of the tasks specified by the user.

DMP

Arithmetic Operation (Scalar Quantities Only)

$$SDS = DSN1 \left\{ \begin{matrix} + \\ - \\ * \\ / \\ ** \end{matrix} \right\} DSN2$$

This command directs the arithmetic operation specified to be performed on the data associated with the symbolic name or the numeric value used. Note that the resultant symbol may be the same as an operand. Several global internal parameters may be defined by use of arithmetic operations. These are:

FRQ	(frequency in MHz)
TIME	(CP run time in minutes)
NUMFIL	(highest FORTRAN logical unit number available for use)
COND	(ground conductivity (in mhos/m))
EPSR	(relative dielectric constant for ground)

In version 3 of GEMACS there is no hierarchy of operation and the user must use parentheses to denote order of operations. Operations are performed from right to left and from innermost to outermost parenthesis. Thus,

```
A = 3**2 + 6 → 3^8
```

while

```
A = (3**2) + 6 → 3^2 + 6
```

Examples:

```
FRQ = FRQ + 1.                  $INCREMENT FREQUENCY
OMEGA = 6.28 * FRQ              $CONVERT TO RADIAN
VOLTS = ZIJMAT * CURENT         $MULTIPLY TWO SCALARS
TIME = 30                       $SET 30 MINUTE CP TIME LIMIT
```

EFIELD

Electric Field Output

SDS = EFIELD (DS) [, { LINLIN / LINLOG / LOGLIN / LOGLOG / LINPLR / LOGPLR }]

[,U2=N] [,DU=N] [,V2=N] [,DV=N] [,W2=N]
[,DW=N] [,U1=N] [,V1=N] [,W1=N]

This command will compute the incident, scattered, or total electric field due to the data set identified as DS. The resultant data are associated with the symbol in the SDS item.

The field quantity computed is determined by the data set type of DS (solution or source) and the interactions set by SETINT. Normally the scattered field is computed. However, if EI, ES, or EU has been specified in the last SETINT command, incident or total fields may be calculated. In this case, total fields (incident plus scattered fields) are calculated if DS is a solution data set; if DS is a source data set only incident fields are computed.

The location of all points at which the field is to be computed may be specified in spherical, cylindrical, or Cartesian coordinates. If spherical parameters are specified and R is omitted, the far electric field will be computed. If R is specified, the near field will be computed. The order in which the parameters are specified will determine the order of the output. U1, V1, and W1 may be R1, T1, P1, X1, Y1, Z1, with U2, V2, and W2 being R2, T2, P2, X2, Y2, Z2; and DU, DV, and DW being DR, DT, DP, DX, DY, DZ. In this way, the user specifies the first point, the increment, and the last point for each coordinate axis. If U,

W, V represent three coordinate specifications, then specifying U1 before W1 will cause the variation specified for W to be completed for each value of U. This is similar to nested FORTRAN DO loops. At the completion of each innermost variation, the electric field components will be printed and, if specified, plotted using the scales specified.

The dependent axis will be the magnitude of the electric field in volts/meter and the independent axis will be one of the geometric variables. Note that to specify a polar plot with either R, X, Y, or Z as the most rapidly varying coordinate is meaningless and will result in an error. However, requesting a linear or log independent axis for an angular coordinate is not meaningless and will be plotted. A combination R, T, P will imply a spherical coordinate system while R, T, Z implies a cylindrical system and X, Y, Z implies a Cartesian system. These are the only combinations allowed and the meanings of the primary coordinate identified (R, T, P, X, Y, Z) are given in the table.

KEYWORD DEFINITIONS FOR COORDINATE SYSTEMS

	SPHERICAL	CYLINDRICAL	CARTESIAN
R	Radius of Point from Origin	Radius in XY Plane	
T	Angle measured from Z axis toward X-Y plane	Angle measured from X axis counterclockwise	
P	Angle measured from X axis counterclockwise		
X			X Coordinate
Y			Y Coordinate
Z		Z Coordinate	Z Coordinate

The default values for U1, V1, W1, DU, DV, and DW are all zero. The coordinate specified will take all values from U1 to U2 in steps of DU. U2, V2, and W2 are defaulted to U1, V1, and W1. The only case in which a geometric parameter may be completely defaulted is the R item in the spherical coordinate system. Otherwise, the initial value or the final value must be given for each of the coordinate axes.

Example:

```
FLD = EFIELD (CUR)  LOGPLR R1 = 36.0   T1 = 0.0,   DT = 10.0,
                    T2 = 90.0,   P1 = 0.,   DP = 10.0,
                    P2 = 360.
```

This will cause the computation of the near field at a distance of 36 meters in the upper hemisphere. Log polar plots of E_r, E_θ, and E_ϕ will be made for theta angles of 0^o to 90^o in increments of 10^o with 37 points for each component per plot.

The plot output associated with the electric field output is unlabeled in version 3 of GEMACS. The plots are intended to be used in a qualitative manner, and the data are listed prior to the plot. The abscissa (X) axis is printed across the page and the ordinate (Y) is printed down the page. In the event of a polar plot, the origin is at the center of the display region while for nonpolar plots, the origin is as indicated by the axis data listed.

All plots use the most rapidly varying geometric parameter as the independent variable (angle for polar plots, abscissa for nonpolar plots). The dependent variable is the ratio of the magnitude of the electric field at the observation point to the maximum electric field computed for all observation points. The LOG specification results in the value being modified to 20 $\log_{10}$ of the ratio (Field Strength/Maximum Field Strength). The dynamic range of the plots is 100 dB.

When the independent variable is an angular coordinate for a polar plot, the location of the reference depends on the coordinate and the coordinate system in use. For cylindrical coordinates, the θ angle is

measured positive counterclockwise from the positive X-axis. The same convention holds for the ϕ angle in spherical coordinates. This is illustrated in figure a. For the spherical θ angle, the measure is positive counterclockwise from the plot X axis. In this case, the plane of the plot is the plane containing the vectors $\hat{r}$ and $\hat{z}$, where $\hat{r}$ is in the direction of the observation point and $\hat{z}$ is parallel to the Cartesian Z-axis. This is illustrated in figure b.

Angular coordinates for nonpolar plots are treated the same as any other independent variable and plotted as the abscissa.

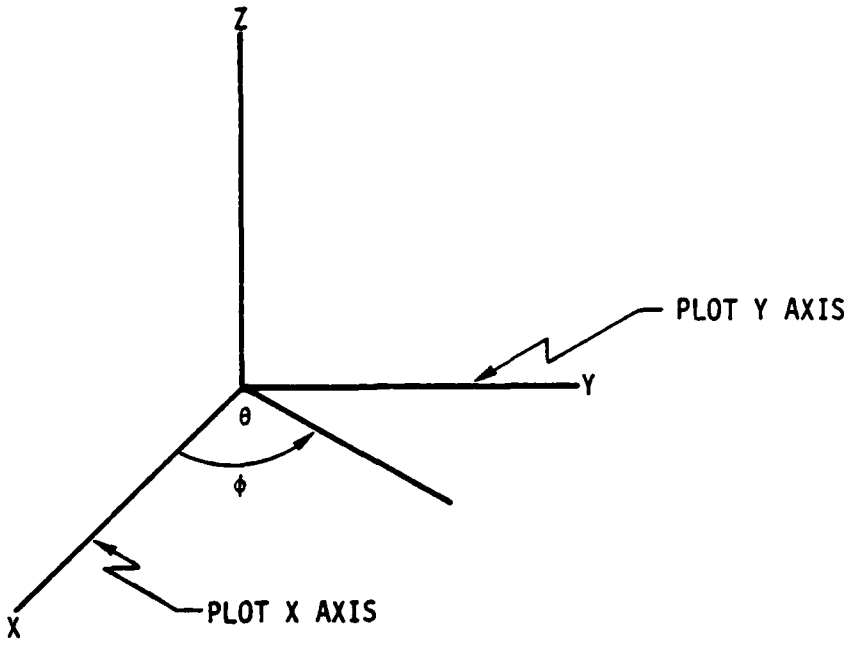


Figure a. Plot Axis for Spherical ϕ and Cylindrical θ Independent Coordinate

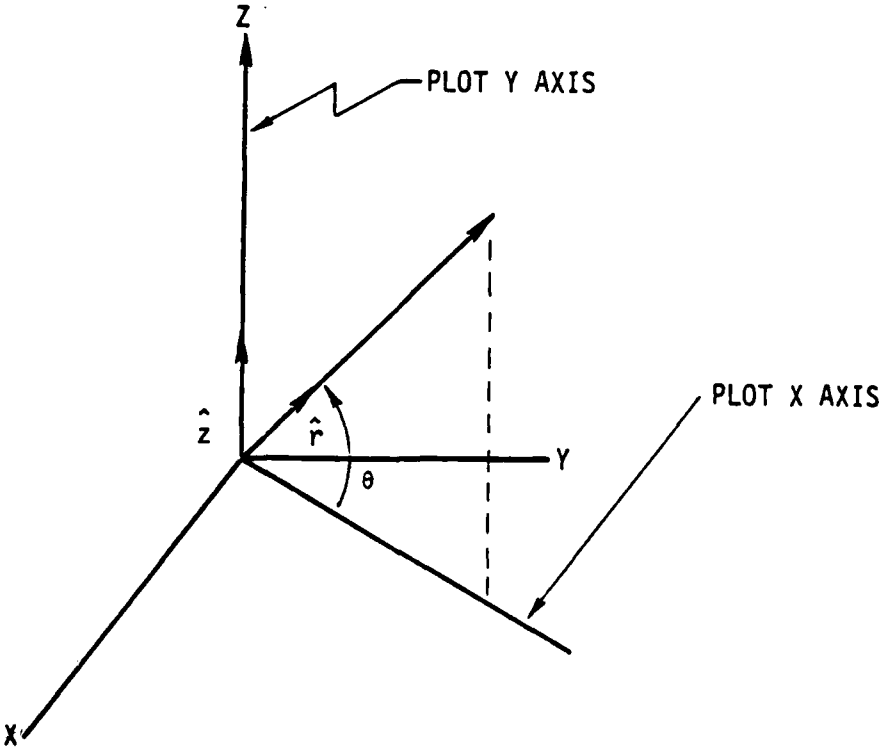


Figure b. Plot Axis for Spherical θ as Independent Coordinate

END

End of Commands

END

The END card designates the end of a command input deck. Any text may be on the same card as long as it is separated from the END by at least one blank or comma (i.e., no comment delimiter is needed). In addition, the END must be the first field encountered.

Examples:

```
END
END OF COMMANDS
END OF FCP747 ANALYSIS
```

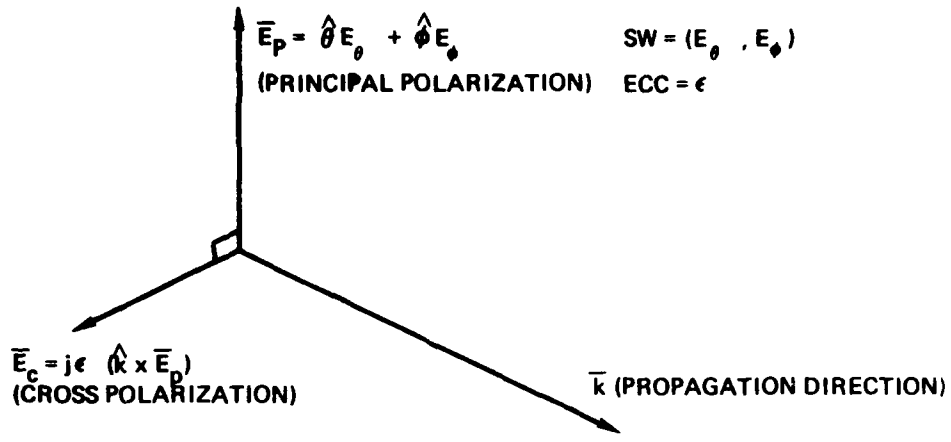
When an END card is encountered, execution is terminated, all files are closed, and an end-of-module checkpoint is written.

ESRC

Electric Field Excitation

SDS = ESRC [(DS)] [,FRQ = DSN] SW = DSN, DSN
[,R = DSN] [,THETA = DSN] [,PHI = DSN]
[,ECC = DSN]

This command generates or modifies the excitation specified in SDS by driving the structure identified by DS with an incident electric field. The default structure identifier GEODAT is supplied if omitted. The frequency in MHz is specified in the FRQ item; and, if omitted, the value used is the last value specified in a FRQ item of any command. If the frequency has changed since the last excitation (either ESRC or VSRC), the data are reinitialized to zero before the excitation is computed. Once computed, the excitation is superpositioned with previous excitation data. The angular coordinates of the source are illustrated in the figure. THETA is measured in degrees from the Cartesian Z axis and PHI is measured in degrees counterclockwise from the Cartesian X axis. The default values are THETA = 90., PHI = 0.. If the radial location specified in the R item is positive, a spherical incident wave from a source located at R, THETA, and PHI will be generated. If the radial location is omitted or negative, a plane wave incident from THETA and PHI will be generated. The default value is R = -1. The vector components of the source field are specified as the values in the SW item. The first value is the component of the field in the spherical $\hat{\theta}$ direction, and the second value is the component in the spherical $\hat{\phi}$ direction. If the item is SW = -1., 0. for example, it corresponds to a vertically polarized electric field with an intensity of 1 volt/m if $\theta = 90^{o}$. If an elliptically polarized incident field is to be generated, the ratio of the minor to major axis is specified in the ECC field. The default value is ECC = 0, indicating a linearly polarized wave. Left or right polarization is denoted by the sign of this parameter. Left polarization is positive while right polarization is negative.



$$\bar{E}_{TOTAL} = \bar{E}_P + \bar{E}_c$$

Field Excitation Geometry and Definition

When a ground plane has been specified by a previous ZGEN command, the total excitation is the vector sum of the source field and the reflected field. The reflected field is calculated using the reflection coefficient method discussed in the GEMACS Engineering Manual. Since this is applied only for a ground plane previously specified, the user must ensure that the ZGEN or DMP commands precede the ESRC commands in the execution stream.

Example:

VANT = ESRC SW = 0.,1., THETA = 45., ECC = 1

This will drive the default structure (GEODAT) with a left-hand circularly polarized plane wave incident from 45^{o} in the XZ plane of the structure.

ESRC (Concluded)

When ESRC is used to specify a source for a problem with only GTD geometry objects, a data set is not generated. Rather the SDS is used later by an EFIELD command to specify the source(s) which generated the incident field. As in the MOM and hybrid cases, more than one ESRC command can be used. However, when incident fields are requested through an EFIELD command, the only ESRC sources which will contribute to the incident field are those on cards prior to the EFIELD command but after a LOOP/LABEL pair, geometry change, or frequency change. In the example,

```
          FRQ = 500.
     1    SRC1 = ESRC (    )...
          FRQ = 1000.
     2    SRC1 = ESRC (    )...
     3    SRC1 = ESRC (    )...
          FLD = EFIELD (SRC1)...
```

only the last two ESRC commands generating SRC1 would contribute to FLD.

GMDATA

Generate a Structure Geometry

GMDATA [= S] [,LU = N]

This command calls the geometry processor to read the geometry data cards which follow the command stream END card. Note that geometry data do not get read until after all commands are read. The default symbol name of the geometry data set S is GEODAT. The processor will read the data from the FORTRAN logical unit specified in the LU = N item. Again, the user must assure that N is a valid logical unit and the file must be in card image format. The default is the user's computer system card input file and is set as variable LUTASK in BLOCK DATA.

Note that if the default name is used, it must not be explicitly referenced in a subsequent command (i.e., it must be defaulted on all commands for the remainder of the command stream). The only exception is when the PLOT command is used.

Examples:

```
GMDATA
GMDATA = FCP747
GMDATA = HUT, LU = 11
ZGEN FRQ=295.0 ZMATRX=MATZIJ
ZGEN GMDATA=FCP747 ZMATRX=FCPZIJ
ZGEN GMDATA=HUT ZMATRX=HUTZIJ
```

These examples result in three geometry data sets being generated and identified as GEODAT, FCP747, and HUT, respectively. These data sets are then used to generate interaction matrices. The first ZGEN command uses the default geometry data set and generates an interaction matrix with the name MATZIJ. The remaining two ZGEN commands use the geometry data specified and generate interaction matrices with the names identified in the respective ZMATRX parameter field.

LOOP/LABEL

Command Repetition

LOOP $\left\{\begin{matrix}A\\N\end{matrix}\right\}$ N

LABEL $\left\{\begin{matrix}A\\N\end{matrix}\right\}$

These commands cause the commands contained between them to be executed N times. Loops may be nested to any level as long as the total number of LOOP commands does not exceed 10. When using nested loops, the loops must be terminated from the innermost loop to the outermost loop. The $\left\{\begin{matrix}A\\N\end{matrix}\right\}$ field may be an up to six character alphanumeric entry or an integer.

Example:

```
LOOP   1, 5          $ EXECUTE SUBSEQUENT COMMANDS 5 TIMES
  .
  .
  .
LABEL 1

LOOP  LOOP1, 5       $ EXECUTE SUBSEQUENT COMMANDS 5 TIMES
LOOP  LOOP2, 10      $ EXECUTE SUBSEQUENT COMMANDS 10 TIMES
  .
  .                  $ COMMANDS EXECUTED 50 TIMES
  .
LABEL  LOOP2         $ INNER LOOP TERMINATOR
  .
  .                  $ COMMANDS EXECUTED 5 TIMES
  .
LABEL  LOOP1         $ OUTER LOOP TERMINATOR
```

LUD

Matrix Decomposition

SDS = LUD (DS)

This command results in the decomposition by rows of the matrix DS into lower and upper triangular matrices identified by SDS. In most cases the matrix DS will not reside in main memory, and subsequently, there will be two triangular matrices generated. They will be internally identified by symbols in which the rightmost three characters of the symbol specified for SDS are replaced with LWR and UPR for the lower and upper triangular matrices respectively. If SDS has three or less characters, then the matrices are simply LWR and UPR. This will be transparent to the user since any reference to SDS will result in the retrieval of the lower and upper triangular matrices as necessary for the operation. However, the user may reference the data symbolically by using the original symbol with LWR and UPR replacing the last three characters of the name.

No pivoting will take place during decomposition. This is due to the fact that, for EM problems, pivoting is usually of little value and can be quite time consuming for matrices not stored in main memory. Pivoting may be added in a subsequent version if a need is demonstrated.

Examples:

```
LUDZIJ   =   LUD (ZIJMAT)
ZIJMAT   =   LUD (ZIJMAT)
IM       =   LUD (ZIJMAT)
PRINT LUDLWR, LUDUPR
PRINT ZIJLWR, ZIJUPR
PRINT LWR, UPR
```

Note: When the matrix DS does not reside in main memory, it will still exist on the original file. The LWR and UPR matrices will reside on files other than this original. Therefore, purging DS will not affect the LWR and UPR matrices. Also, the user will retain DS if the LWR and/or UPR matrices are purged.

PLOT

Plot Geometry Command

PLOT DS

This command will generate a printer plot of the geometry in data set DS. Three projections will be made: X-Y, X-Z, and Y-Z. These provide a visual summary of the location of each wire segment and area patch.

Examples:

1. GMDATA
   PLOT GEODAT
   .
   .
   .
2. GMDATA=HUT
   PLOT HUT

In the first example the geometry was allowed to be given the default name GEODAT. In order to obtain a plot of the data, the default name for the geometry data set had to be explicitly referenced. In the second example the geometry was named HUT. The data in this geometry set will be plotted due to the presence of the second PLOT command.

PRINT

Print Data Command

PRINT SDS, SDS,... SDS

This command allows the user to obtain the entire contents of each data set specified. If the data are complex, the output will contain both the real and imaginary components as well as the magnitude and phase. For complex data there will be 2 elements per line; and for real data, there will be 10 elements per line. Printing out large complex arrays can consume a fair amount of paper, and the user is advised to use the WRITE command when the entire contents are not required.

Example:

PRINT VOLT, CURRNT

This command will cause the data associated with symbols VOLT and CURRNT to be printed sequentially.

PURGE

Purge Data Command

PURGE SDS, SDS,... SDS

This command will cause the internal main storage or FORTRAN logical unit associated with the specified symbols to be made available for other use. The data are not retrievable after a PURGE command. Due to the limited FORTRAN logical units, it is recommended that symbols be purged when no longer required. This will also make checkpoint files shorter. Purged symbols may be referenced if the actual data are not required. This could occur, for example, after a matrix has been decomposed and all that is required are the lower and upper triangular matrices.

Example:

```
GMDATA
ZGEN  FRQ = 123.0  ZMATRX = ZIJMAT
ZIJMAT = LUD (ZIJMAT)
PURGE  ZIJMAT
VOLT = VSRC,  V = 1.,0. SEGS = 51-60
BACSUB  ZIJMAT * CUR = VOLT
PRINT  CUR, VOLT
END OF COMMANDS
{GEOMETRY DATA}
END OF DATA
```

The PURGE command releases the file ZIJMAT on which the interaction matrix is stored. However, the lower and upper triangular matrices are unaffected since they are stored in files labeled ZIJUPR and ZIJLWR. Thus the BACSUB command can be executed, and a PRINT (or WRITE) command would output the data in the decomposed matrices.

RSTART

Checkpoint Restart

$$\text{RSTART} \quad [\text{LU} = \text{N}] \left[,\text{MODULE} = \begin{Bmatrix}\text{INPUT}\\ \text{GTD}\\ \text{MOM}\\ \text{OUTPUT}\end{Bmatrix}\right],\text{CPNUM} = \text{N}$$

This command is used to restart a job from checkpoint. The checkpoint file is to reside on the FORTRAN logical unit specified in the LU = N item. The default item is LU = 7, which is the same as the CHKPNT default logical unit. If an alternate logical unit is specified, the user is responsible for assuring that GEMACS can access the unit. The MODULE item is used to assure that execution is restarted with the correct module (INPUT, GTD, MOM, OUTPUT). The default item is MODULE = INPUT. The checkpoint number to be recovered is specified in the CPNUM item and is not defaulted. The value N specifies the integer number of the checkpoint to be recovered. This permits different operations to be performed on the data without requirements to regenerate all the data. For instance, if a checkpoint was written after the interaction matrix was generated, the user could restart at that point and use a different solution procedure than on a previous analysis. See the WIPOUT command for further discussion.

NOTE: RSTART should be the first command encountered for the restart run. If it is not, all previous commands in the restart input stream are ignored.

Example:

RSTART MODULE=GTD, CPNUM = 5

If a restart is executed from the same logical unit as the checkpoint was written to, the checkpoint file may be overwritten on subsequent checkpoints. If the user wishes to maintain the integrity of the original checkpoint file, the restart must take place from a different logical unit (LU), and thus cannot be defaulted.

SET

Data Initialization and Modification

SET SDS = N [,N] [,R1 = N] [,R2 = N] [,C1 = N]
[,C2 = N]

This command may be used to initialize or change the value of data associated with SDS. If the data are complex, then SDS = N, N (corresponding to the real and imaginary components) is used. If the data are real, then SDS = N is the correct form. The items R1, R2, C1, and C2 specify the row and column limits of the data to be loaded with the value specified. The default value of R1 and C1 is 1, while R2 and C2 default to R1 and C1 respectively.

This command would allow the user to alter an excitation data set if he wished to force a boundary condition. If a structure has interior wires and is excited by an external field, the field on the internal wires could be reset to zero. Also, the initial solution for the BMI process could be specified, as well as modifications made to the interaction matrix.

Example:

SET ZIJMAT = 0.,0. R1 = 10, C2 = 100

SET ZIJMAT = 0.,0. R2 = 100, C1 = 10

SET ZIJMAT = 1.,0. R1 = 10, C1 = 10

SET VDRV = 0.,0., R1 = 10

This sequence would load zeros into every element of the tenth row and tenth column of ZIJMAT and then reset the diagonal element to (1.,0). The tenth element of VDRV would be set to zero. This would have the effect of constraining the current in the tenth segment to be zero and not allowing any interaction between the tenth segment and the rest of the structure except for current continuity at junctions.

SETINT

Set the Physics Interactions for Matrix Generation

SETINT [KW, KW,... KW]

This command selects which MOM and/or GTD interactions are to be used to calculate interaction, excitation, and field matrices and whether or not incident fields will be included in the field matrix. Default is the MOM interaction. If a SETINT command is not present in the input stream, only MOM interactions will be computed. Below is the list of interaction keywords and their descriptions.

- MM Method of Moments interactions
- GTD All GTD interactions (equivalent to PL, CY, PC)
  - PL All plate interactions (PR, PD, RD, PDR, RR)
    - PD Diffraction from plate edges
    - PDR Diffraction from a plate edge; reflection from a second plate
    - PR Reflection from plates
    - RD Reflection from a plate; diffraction from a second plate
    - RR Reflection from one plate, then reflection from another plate
  - CY All cylinder interactions (CS, ER, ED)
    - CS Scattering from a cylinder
    - ED Diffraction from end cap rim
    - ER Reflection from cylinder end cap
  - PC All plate-cylinder interactions (RC, CR, CD, DC)
    - CD Scattering from a cylinder, then diffraction from a plate edge
    - CR Scattering from a cylinder, then reflection from a plate

DC Diffraction from a plate edge, then scattering from a cylinder

RC Reflection from plate, then scattering from cylinder

EI Include all excitation fields in incident field

ES Include only fields scattered by GTD geometry in incident field

EU Include only fields not shadowed by GTD geometry in incident field

A complete MOM/GTD hybrid problem can be worked with

```
SETINT GTD MM
```

A MOM problem does not require the SETINT command, though either of the two example commands could be used.

```
SETINT MM
SETINT
```

A GTD problem requires the presence of the SETINT command. All keywords but MM are valid in this case. The example

```
SETINT PL CY
```

would select plate and cylinder interactions, but not plate-cylinder multiple interactions.

The other interaction keywords provide the user with a way to isolate scatterers and determine which types of interaction contribute most to the final results.

SOLVE

Solve System of Simultaneous Linear Equations

SOLVE DS1* SDS = DS2

This command will solve for the solution vector SDS using lower/upper triangular decomposition on DS1 and back substitution using DS2. If DS1 is already decomposed, only the back substitution will be performed. SDS and DS2 may be the same symbol.

Example:

```
SOLVE  ZIJMAT * CUR = EINC
```

This will cause execution of the following equivalent input:

```
ZIJMAT = LUD (ZIJMAT)
BACSUB   ZIJMAT * CUR = EINC
```

VSRC

Voltage or Antenna Excitation

SDS = VSRC [(DS)] [,FRQ = DSN] ,V = DSN, DSN,

TAGS
SEGS = N, N, ..., N

This command will set up or add to the excitation specified by SDS on the structure DS. The default structure identified is GEODAT. The voltage source is applied as a delta-gap electric field at the midpoints of the specified wires in the geometry data. That is, the tangential electric field at the midpoints of the wires specified is $-V/\ell$ where ℓ is the segment length in meters. The frequency in MHz is specified in the FRQ item; and, if the item is omitted, the last frequency specified in a FRQ item on any command is used. If the value of the frequency has changed since the last excitation (either VSRC or ESRC), the symbol will be reinitialized to zero before the new source data are computed. If the frequency is unchanged, then the source data will be added to the existing data associated with SDS. This permits superpositioning of excitations. The real and imaginary components of the voltage source are specified by the V item. The values for this item may have been previously defined symbolically. The segment identification may have one of two forms. If TAGS is specified, then all wires which have the tag numbers specified in the parameter list will be excited. If SEGS is specified, only those wire segments listed will be excited. This list of segment and tag numbers must be the last entry of the command and may contain a minus sign between successive entries. That is N_1, N_2 - N_3, N_4 is valid and will cause tags or segments N_1, N_2 through and including N_3, and N_4 to be excited.

Example:

VANT = VSRC , FRQ = FRQMHZ, V = 0.707, - 0.707, TAGS = 1 - 4

The VSRC command is ignored for GTD-only calculations.

WIPOUT

Command Stream Modification

WIPOUT N

This command will cancel all commands beginning with the command identified by the sequence number N. The commands as input by the user are sequenced in ascending order starting with 1. Note that the command sequence number is not necessarily the same as the card number containing the command since comment cards and continuation cards may be present and are each counted. The entire command sequence from N up to and including the WIPOUT command are eliminated from the execution.

The WIPOUT command would normally be used after a RSTART command to change the command sequence. Additional commands may follow the WIPOUT command.

Example:

```
RSTART     CPNUM = 7
WIPOUT     5
.
.
.
```

This would eliminate all commands after the fourth command of the stream which generated the checkpoint being restarted. Subsequent commands input after the WIPOUT command would be executed.

(Note: LOOP/LABEL commands may be wiped out only as paired sets.)

WRITE

Data Output Command

```
WRITE DS  [,LU = N]  [,R1 = N]  [,R2 = N]  [,C1 = N]
          [,C2 = N]  [,FILEID = A]
```

Using this command, partial data associated with the symbol DS may be written to the file specified. If the logical unit item LU = N is not specified, the system printer is used and the FILEID item is ignored. If LU = N is specified, the field A in the FILEID item is written as the first word on the file, followed by the data specified. In this case, the output is in FORTRAN binary format and may be used as input to other analyses. R1, R2, C1, and C2 define the row and column limits of the data. Default values for R1 and C1 are 1, while default values for R2 and C2 are dependent on whether R1 and/or C1 are specified. If R1 is specified, R2 defaults to R1. If not, then R2 defaults to the number of rows in the symbol DS. The same procedure applies to C2.

Example:

```
WRITE ZIJMAT R1 = 10, C1 = 1, C2 = 5
```

This prints the first 5 elements in row 10 of ZIJMAT

```
WRITE ZIJMAT
```

This writes the entire matrix ZIJMAT, and is therefore equivalent to the results obtained by use of the PRINT command.

ZGEN

Interaction Matrix Generation

ZGEN [,GMDATA = DS] [,FRQ = DSN] ,ZMATRX = S
[,ZLOADS = DS] [,COND = DSN] [,EPSR = DSN]

This command generates an interaction matrix using the sine + cosine + pulse expansion function for wires, the pulse expansion function for patches, and impulse matching on the structure specified in the GMDATA item. The default structure is GEODAT, the default name for the GMDATA command. The frequency in MHz is specified by the value DSN of the FRQ item. The default is the frequency given in the last FRQ item explicitly stated in any previous command. Note that the frequency may be specified symbolically or numerically. The resultant impedance matrix is identified by the parameter in the ZMATRX item. If the structure is to be loaded, the value in the ZLOADS item must be the symbol name of a data set generated by a preceding ZLOADS command. The default is no loading or a null ZLOADS item. If a ground plane is to be used, the conductivity in mhos/meter must be specified in the COND item. COND = -1 implies a perfect ground, COND = 0.0 implies no ground. Default is COND = 0.0. If a nonperfect ground is specified, the relative dielectric constant ϵ_r may be specified in the EPSR item. The default item is EPSR = 1. For a perfect or no ground case, the contents of the EPSR item are ignored. When a ground is specified, it is assumed to be perpendicular to the global Z axis.

Example:

```
ZGEN    FRQ = FRQMHZ, ZMATRX = ZIJ, COND = 80.0
```

NOTE: When the GTD interactions are specified on a SETINT command, a ground plane may not be specified. In this case, both the COND and EPSR items must be defaulted. The effect of perfectly conducting ground may be modeled

by locating one of the GTD plates at $z = 0$ and specifying the PR keyword in a SETINT command. This, computationally, is equivalent to specifying COND = -1.

ZLOADS

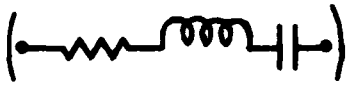
Structure Loading

$$\text{ZLOADS} = \text{SDS} \quad [,\text{GMDATA} = \text{DS}] \quad , \left\{ \begin{array}{l} \text{COND} = \text{DSN} \\ \text{ZIMP} = \text{DSN, DSN} \\ \\ \begin{matrix}\text{PRLC} \\ \text{SRLC}\end{matrix} = \text{DSN, DSN, DSN} \end{array} \right\}$$

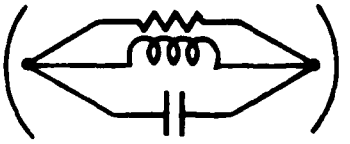
$$, \left\{ \begin{matrix}\text{TAGS} \\ \text{SEGS}\end{matrix} \right\} = \text{N, N,... N}$$

This command allows a user to place electrical loads on the wire structure identified in the GMDATA item. The default structure (GEODAT) will be used if the GMDATA item is omitted. The load information data will be associated with the symbol SDS.

The type of loading is specified by one of the parameters COND, ZIMP, PRLC, or SRLC. COND is used to specify the segment conductivity in mhos/meter. ZIMP is used to specify a frequency-independent lumped load resistance and reactance in ohms. The PRLC and SRLC parameters permit loading wire segments with parallel (PRLC) or series RLC (SRLC) circuits. The values of R, L, and C are contained in the parameter list in that order, the latter two being functions of frequency. The units of R, L, and C for the PRLC and SRLC items are ohms, millihenries, and microfarads respectively. The equivalent circuits are illustrated below:



SRLC



PRLC

ZLOADS (Concluded)

The TAGS/SEGS item is used to identify those wire segments to be loaded. Use of the TAGS option results in all segments which have one of the tags specified being loaded. Use of the SEGS option limits the loading to only those segments specified. The integer list in the TAGS/SEGS item may contain consecutive integers separated by a hyphen. In this case, all elements from the first integer to the last are effectively specified and the corresponding segments are loaded.

Multiple ZLOADS commands are permitted. In order to be effective, the ZLOADS command must occur before the ZGEN command.

Example:

```
ZLOADS = FCPLOD   GMDATA = FCP747, PRLC = 5.0, 13.0, 21.0,
                  SEGS = 1, 5,  7-23, 47
```

This command causes the structure FCP747 to have a parallel RLC load applied to wires 1, 5, 7, 8, 9,..., 21, 22, 23, and 47 with R = 5 ohms, L = 13 mH, and C = 21 μF.

2. Geometry Input Language Processor

The function of the GIP (Geometry Input Processor) is to translate the user's inputs related to structure geometry into a data set which may be operated on by an interaction matrix generator to provide the interaction matrix of the structure under analysis. The GIP is entered on encountering the GMDATA command in the task execution list. It is a re-entrant processor in that it may be called several times, either to extend a previously generated geometry data set or to create a new geometry data set. The attributes of the geometry data set are assigned by the GIP. On completion of the geometry processing, all data are written out to a peripheral storage device.

The user inputs to the GIP consist of card images in which the first nonblank item is an alphabetic code designating the type of data contained on the command. The data following the type code must be separated by blanks or commas. Blanks embedded within items are not allowed. An item is therefore defined as a contiguous group of characters which, when interpreted, correspond to a data input of the card being processed. Text following a $ is regarded as a comment and is ignored. There are a maximum of 256 items per command, not including comment fields or commas which are not processed. Continuation cards are indicated in the following ways. First, when a card ends in a comma, the next card is read as a continuation and must have a continuation character in column 1. Second, any card which has a continuation character in column 1 is assumed to be a continuation of the previous card. The only limit on the number of continuation cards is that dictated by the limit of 256 items per command. The continuation character and its definition are the same as described in section C.

The currently supported type codes and their meanings are:

Type Code	Definition
AT	Attach
CE	Combine Elements
CP	Connect Point
CS	Coordinate System

CY	Cylinder
DE	Define End
DF	Define
EC	Endcap
END	End
MP	Multiple Points
PA	Patch
PL	Plate
PSYM	Plane Symmetry
PT	Point
RA	Radii
RF	Reflect
RN	Renumber
RSYM	Rotational Symmetry
RX,RY,RZ	Rotate
SC	Scale
WR	Wire
XL	Translate

The command format and use of each of these types are presented in the following section.

General guidelines for modeling include the following:

a. Segments must be short compared to one wavelength. Lengths of 0.1 λ should be adequate for most purposes. For wire grids with square mesh, good results have been obtained with lengths up to 0.14 λ. (See Volume II, GEMACS Engineering Manual, section G.1.c.).

b. Actual wires should be modeled with the actual radius. Grid models should use a wire radius about one-fifth of the segment length in regions of square mesh. (See Volume II, GEMACS Engineering Manual, section G.1.c for more detailed comments.)

c. Grid mesh circumferences should not greatly exceed 0.5 λ. Larger circumferences lead to loop resonances and poor results.

d. Segments with lengths differing by more than a factor of two should not be joined. Small angles (less than about 20^{o}) between

joined segments should be avoided. Unjoined segments should be separated by a segment length or more. The maximum number of segments at a junction is limited to 50. Segments are considered to be joined when their end points are separated by no more than ZERO, a variable set in a data statement in BLOCK DATA BLKDAT. The value of ZERO should be set in the code by the user according to his needs and the limits of precision imposed by his computer. It is calculated using the following formula:

$$ZERO = 1/2 * 2^{-(m-1)}$$

where m is the number of bits in the mantissa of the computer system.

e. The maximum number of points, segments, defined elements, etc. that can presently be input for the geometry is discussed in section D.2 of this volume.

f. The renumbering command (RN) permits the user to specify the geometry in the most convenient manner available and to subsequently renumber the wire segments to locate the near-neighbor interactions close to the diagonal of the interaction matrix. The interaction terms between the i^{th} and j^{th} segments will be the (ij) and (ji) elements of the interaction matrix. If the BMI bandwidth chosen is b, then if i-j > b, the interaction of i^{th} and j^{th} segments will not be included in the band. Too many (an as yet unquantified number) large interactions omitted from the band will cause the BMI solution technique to fail to converge. Therefore, numbering the problem such that near neighbors have approximately equal segment numbers will cause large interactions to appear in the band. However, too many small coherent interactions outside the band may also cause the BMI solution technique to fail to converge. Note that patches, plates, cylinders, and cylinder end caps are to be input after all wire segments have been defined and renumbered. Therefore, this command and its operation do not apply to these objects.

g. Wire segments should be kept at least 0.25 λ from plate edges for maximum accuracy. Segments could be placed as close as 0.1 λ if only engineering accuracy ($\pm$ 25 percent) is required.

h. Wires, but not junctions of wires, may be connected to patches, ground, or plates, but not to cylinders or end caps. A wire is assumed connected if it is within ZERO meters of a patch center, ground plane, or within a plate boundary.

i. A plate is assumed attached to a cylinder if the attachment edge is within ZERO meters of the cylinder surface. Preferably the plate should be set slightly inside the cylinder volume. Attachment must be made with the plate edge parallel to the cylinder axis and edge binormal perpendicular to the cylinder surface normal at the attachment edge.

j. Plates may not be attached to end caps.

3. Geometry Input Language Commands

The general form of the commands available to the user is:

TYPE P1, P2, P3,...

where TYPE is one of the type codes listed in section C.2. P1, P2, P3 are the ordered items required or used in the processing of the command specified. The items may be separated by a comma or a blank. Two of the basic geometrical objects in the GIP are points and wire segments. In addition, points and wire segments may belong to larger groups. For points, the only larger group is referred to as a defined element generated by a DF command, and reference to such an element will automatically reference all of the points within the element. Line segments may also belong to a group identified by a tag number in addition to a line segment number. The former method (DF) is preferred due to programming considerations. Thus, the user may reference points, line segments, or a group of points and/or line segments. At present, there is no way to reference individual line segments unless they are part of a group; however, they may be the only object in the group. Whenever an object is operated on to form a new object, the known attributes of the source object are automatically given to the new object with the exception of group membership. This attribute will also be transferred if the group has not been closed by a DE (Define End) command. To the GIP, the attributes of points are:

(1) Point number

(2) Point location

(3) Group membership.

The attributes of a line segment are:

(1) Segment number
(2) Segment tag
(3) Location of end points
(4) Group membership
(5) Radius of wire segment
(6) Segment connection data.

A third basic geometrical object is the surface patch. Within GEMACS it is treated much like the wire segment. However, unlike the wire segments which can be grouped by tag number, each patch must have a tag number globally unique to that patch. Patches can be part of a defined element. The only way to reference an individual patch is by tag number, unless a defined element is made up of only one patch. In version 3 patches cannot be loaded or excited by a voltage source. They can only be treated as perfectly conducting scatterers. The attributes of the patch are:

(1) Tag Number
(2) Location of Center Point
(3) Group Membership
(4) Area
(5) Direction of Outward Normal
(6) Segment Connection Data.

The GTD objects make up the fourth type of geometry building blocks. GTD objects implemented in version 3 are:

Code	Geometry	Maximum Number
PL	Plates	14, each with no more than 6 corners
CY	Cylinders	1
EC	End caps	2

GTD elements cannot be excited like MOM elements, and currents cannot be found on them directly. They are used only to reflect, scatter, or diffract fields.

The attributes of the GTD elements are:

(1) Plates:
   (a) User-assigned number
   (b) Location of corners
   (c) Group membership

(2) Cylinder:
   (a) User-assigned number
   (b) Major and minor radii
   (c) Length
   (d) Coordinate system
   (e) Associated end cap numbers

(3) End cap:
   (a) User-assigned number
   (b) Associated cylinder number
   (c) Direction of outward normal.

In the discussion of the commands that follows, it must be remembered that the items are ordered. This is in contrast to the items of the execution commands which were keyword indexed to achieve order independence. Keywords are not used in the GIP in order to achieve a more succinct input.

It should also be noted that some of the commands have items that may be defaulted. An example of this is the connect points (CP) command. In general, for a command of the form

TYPE P1, P2, [,P3] [,P4] [,P5]

in which P3, P4, and P5 may be defaulted, it must be kept in mind that defaults can be achieved only from right to left. To specify P5, both P3 and P4 must be specified. To default P4, only P3 must be specified, while P5 must assume its default value. When one or more items are to be defaulted, then all items to the right must also be defaulted.

AT

Attach Operation

$$AT \left\{ \begin{matrix} PT \\ TG \\ DF \\ PL \end{matrix} \right\}, \left\{ \begin{matrix} n \\ n \\ name \\ n \end{matrix} \right\}, NCS$$

Parameter		Definition
AT	-	Attach operation code.
$\left\{ \begin{matrix} PT \\ TG \\ DF \\ PL \end{matrix} \right\}$	-	Point, tag, defined element, or plate to be operated on.
$\left\{ \begin{matrix} n \\ n \\ name \\ n \end{matrix} \right\}$	-	An integer for points, plates, or tag identifiers; the element name for defined elements.
NCS	-	Integer identifier of previously defined coordinate system to which the object or element is to be attached.

This command results in a group of points, segments and/or plates to be translated and rotated into the coordinate system specified. No new objects or elements will be generated. This allows the user to input parts of the geometry in local coordinate systems and then relocate them to their actual position.

Examples:

AT PT 10 3

This would result in the coordinates of point 10 being changed to those it would have after being relocated to the coordinate system 3. If point 10 originally had x, y, z coordinates (0,0,0) and the origin of coordinate system 3 were located at (10,0,0) with regard to the global system, then point 10 would have (10,0,0) as its coordinates after the operation.

AT (Concluded)

AT DF SPHERE 2

All parts of SPHERE would have their coordinates modified to those they would have if they had originally been defined in coordinate system 2.

AT PL 10 2

Plate number 10 would have its corner coordinates relocated to coordinate system 2.

CE

Combine Elements Operation

CE name1, name2, name3,...

Parameter		Definition
CE	-	Combine elements code.
name1	-	Element name by which resultant group will be known.
name2,name3...		Names of elements to be combined into element name. Note that these elements must have been generated by a Define Element (DF) operation and will not be available under their original names after this operation.

If the user has several elements that have been defined by a Define Element (DF) operation, he may combine them into one element with this operation. This is useful when a user has a collection of generic shapes and he needs to put them together to form another object. He may then combine all of the elements under one name for ease of future reference.

Example:

Assume all previous data are present.

```
AT  DF BOOM 3
AT  DF DISH 1
CE  SATLIT  CYLNDR  DISH  BOOM
AT  DF SATLIT 5
```

Coordinate systems 3 and 1 may have been defined such that the first two AT commands move elements BOOM and DISH to locations on element CYLNDR. Then CYLNDR, DISH, and BOOM are combined into element SATLIT and the resultant element positioned in coordinate system number 5. As can be seen, this has eased the input of geometry considerably when the relative locations of defined elements are known or easily calculated. If the user wanted to define the members of BOOM, DISH, and CYLNDR in coordinate system 5, a considerable amount of precomputation would have been required if the elements' dimensions were on separate drawings.

CP

Connect Points Operation

CP N1 N2 [NSEG] [NTAG] [NRAD]

Parameter		Definition
CP	-	Connect points code.
N1	-	Integer number of first point.
N2	-	Integer number of second point.
NSEG	-	Number of segments to be generated between N1 and N2. (Default = last NSEG parameter from any previous operation.)
NTAG	-	Tag number identifying all segments between N1 and N2. (Default = last NTAG parameter from any previous operation.)
NRAD	-	Integer index to radii table entry. (Default = last NRAD entry from any previous operation.)

After the user has defined the points N1 and N2 (PT command), this operation will connect NSEG segments between these points. Each segment generated will have a TAG number specified by NTAG and a radius retrieved from the NRAD entry of the radii table. The default values for NSEG, NTAG, and NRAD are those values left over from any previous operation for which they were defined. This implies that they must be defined on the first command which requires them.

Examples:

```
CP  1 2 3 0 5
```

or

```
CP  1,2,3,0,5
```

This would result in generating three wire segments between point 1 and point 2. They would have a TAG of 0 and a radius extracted from the fifth radii entry.

CP 1 4

Assuming the first example preceded this operation, three more segments would be generated between point 1 and point 4 with a TAG of 0 and radius extracted from the fifth radii entry.

CS

Coordinate System Specification

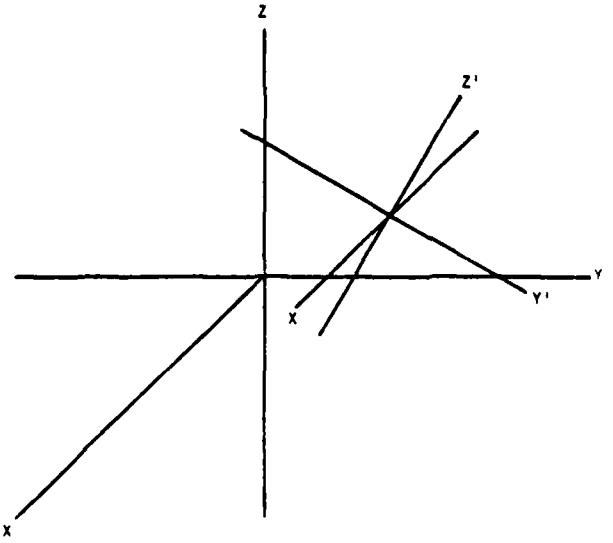
CS NCS XC , YC , ZC , RX , RY , RZ

Parameter		Definition
CS	-	Coordinate system code.
NCS	-	Unique integer identifier for this system.
XC,YC,ZC	-	(x,y,z) location of coordinate system origin with respect to the global coordinate system origin.
RX,RY,RZ	-	Rotation angles in degrees about the x, y, or z axis of the global coordinate system. Only one angle may be nonzero. Negative angle specification results in clockwise rotation when looking toward origin along the axis of rotation.

This command permits the user to specify additional coordinate systems. The NCS item on other commands references the coordinate system identified by this number. When the NCS parameter is specified on other commands, the transformation from or to this coordinate system will be made. This command must precede all commands referencing this NCS.

EXAMPLE:

CS 3 0.0 2.0 1.0 -30.0 0.0 0.0



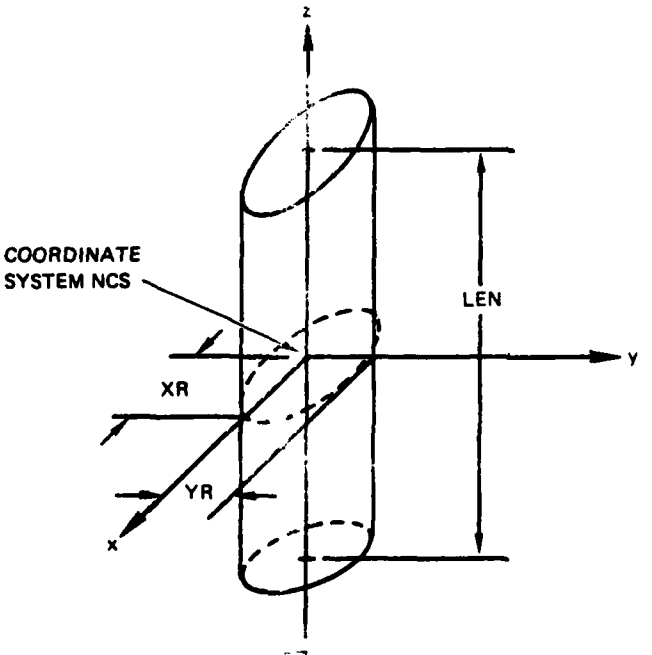
CY

Cylinder Specification

CY n XR YR LEN [NCS]

Parameter		Definition
CY	-	Cylinder specification code.
n	-	User-assigned cylinder number.
XR	-	X-axis radius in the x-y plane.
YR	-	Y-axis radius in the x-y plane.
LEN	-	Cylinder length (z-axis).
NCS	-	Integer identifier of previously defined coordinate system in which the cylinder is centered. (Default is the last previous NCS entry on any command.)

This command allows the user to generate a GTD elliptical cylinder at the origin of the NCS coordinate system. Cylinder length is from the center (X = Y = 0) of one end cap to the center of the other end cap, and it is centered on the x-y plane of the NCS coordinate system. End cap attributes are entered separately with the EC command. Version 3 permits only one cylinder per geometry data set.



DE

Define End Operation

DE

Parameter		Definition
DE	-	Define end code.

This operation ends or closes the group identified in the current Define Element (DF) operation. The group may not be extended except by a CE operation. All points, plates, patches and wire segments generated since the last DF operation belong to the group identified by the DF name. Points in this group may be referenced without regard to group membership, however, the segment and patch data may only be referenced by identifying the group. It is advisable not to generate points under a DF operation since the storage available for points is more restricted than that for segments and operations performed on the DF element will generate more points which are not usually required (see the example for the DF operation).

DF

Define Element Operation

DF name

Parameter		Definition
DF	-	Define element code.
name	-	A six character or less identifier must begin with alpha character, but can include integers.

All points, plates, patches and segments generated between this command and the next DE command will belong to the element "name." Nested DFs are allowed. Note even though points may belong to an element, they may still be referenced individually.

Examples:

```
DF BOX
PT 1  1.0  0.0  0.0
PT 2  0.0  1.0  0.0
PT 3 -1.0  0.0  0.0
PT 4  0.0 -1.0  0.0
CP 1, 2, 1, 0, 1
CP 2, 3
CP 3, 4
CP 4, 1
DE
AT DF BOX 1
```

Points 1, 2, 3, and 4, and segments generated in coordinate system 0 these points are identified as belonging to BOX. All of these points and segments are then moved to coordinate system 1. If an operation was performed on BOX which involved generation of additional elements, the points belonging to BOX would also be used as sources for additional points. For this reason, use of points in a defined element should be avoided whenever possible, since the storage available for points is more restricted than that for segments.

Cylinders and end caps are not allowable DF objects.

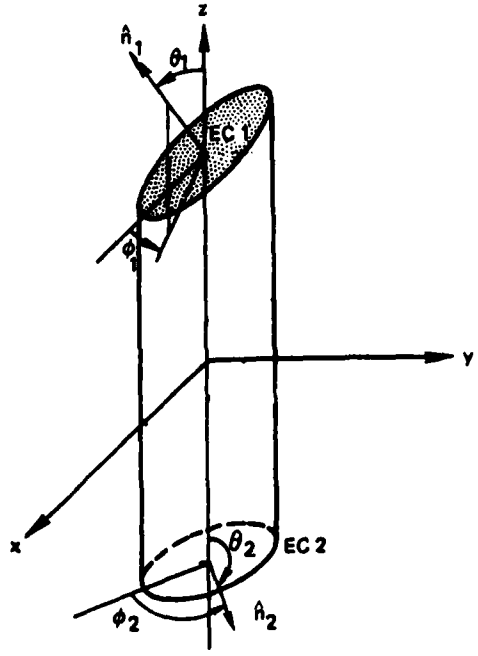
EC

End Cap Specification

EC n ±ncyl θ ϕ

Parameter		Definition
EC	-	End cap specification code.
n	-	User-assigned end cap number.
±ncyl	-	Cylinder to which end cap is attached: + indicates top end cap, - indicates bottom end cap.
θ	-	Polar angle of end cap outward normal.
ϕ	-	Azimuthal angle of end cap outward normal.

This command assigns an end cap to the cylinder ncyl in the geometry data set. Two EC commands are required for each CY command. Slanted end caps are permitted and slant angle is determined from the end cap normal. They need not be identical for both end caps. Version 3 of GEMACS limits the number of end caps to two and requires that the end cap normal be confined to the (x-z) plane ($\phi = 0$). The end cap is automatically located in the cylinder coordinate system.



END

End of Geometry

END

Parameter		Definition
END	-	End of current geometry designator.

This command causes the GIP to stop reading input. It will then look for wire junctions, identify all multiple segments, print out the geometry data set, and write the geometry data set to the user specified data set name.

Multiple segments are defined as those segments lying on an axis of rotation or in a plane of reflection. They are identical segments with the same end points as the generating segment. The generated segments do not enter into the interaction matrix calculation, and they are identified in the segment data output by a zero in the NSEG column.

MP

Multiple Point Connection Operation

MP NPTS, NP1, NP2, NP3,..., NPNPTS [,NSEG] [,NTAG] [,NRAD]

Parameter		Definition
MP	-	Multiple point connect code.
NPTS	-	Integer value of the number of points to be connected.
NP1,NP2,.. NPNPTS	-	Integer identification of points to be connected. There must be NPTS of these values.
NSEG	-	Number of segments between each pair of points. There will be (NPTS-1)*NSEG segments generated. (Default NSEG = last NSEG value from any previous command.)
NTAG	-	Integer tag number identifying all segments generated. (Default NTAG = last NTAG value from any previous command.)
NRAD	-	Integer value of location of wire radius in radii table. (Default NRAD = last NRAD value from any previous command.)

This command permits the user to generate a set of points to be connected with an equal number of segments between each pair of points as listed. There is no restriction on the location of the points and NPTS must be greater than 1.

Example:

MP 6 1 3 7 10 5 4 2 0 1

This command would connect points 1, 3, 7, 10, 5, and 4 with a wire whose radius is stored in the first entry of the radii table. There would be two segments between each pair of points, and all segments would have a tag of zero.

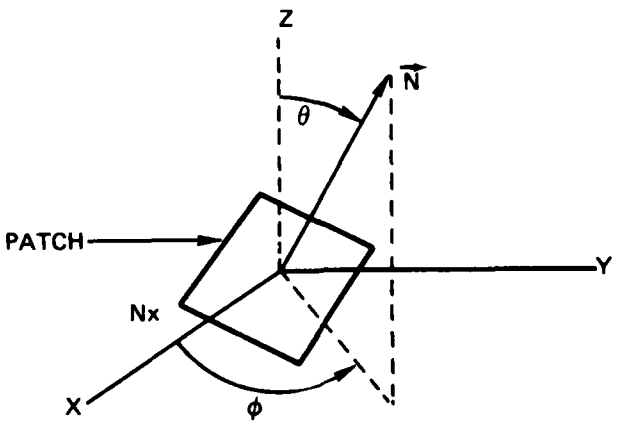
PA

Patch Specification

$$PA \left\{ \begin{matrix} x & y & z \\ & & \\ PT & & N \end{matrix} \right\} \left\{ \begin{matrix} \theta & & \phi & \\ & & & \\ PT & N1 & N2 & N3 \end{matrix} \right\} AREA\ NTAG\ \ [NCS]$$

Parameter		Definition
PA	-	Patch specification code.
PT	-	Two-letter code indicating that the patch is to be placed at a point, or that points will be used to define three corners of the patch.
N, N1, N2, N3		Integer point identifier.
x, y, z	-	X,Y,Z location of the center point of the patch.
θ	-	The polar angle for the normal vector of the patch.
ϕ	-	The azimuthal angle for the normal vector of the patch.
AREA	-	Surface area of patch in square meters or user scaled units.
NTAG	-	Integer tag identifier of each patch, unique for each patch.
NCS	-	Integer identifier of coordinate system. (Default to previous NCS entry of any operation.) See [illegible]

This command allows the user to specify the x,y,z location of the center point of the patch, or to place a patch at a previously defined point. The normal vector of the patch is defined by the polar angle θ and the azimuthal angle ϕ as shown in the figure, or the normal is identified by three previously specified points at its corners, given in a counterclockwise direction looking in the direction of the outward normal of the patch. The specification of the patch area, a unique tag number and an optional coordinate system complete the patch [illegible]



Patch Geometry

NOTES:

1) The NCS item is not applicable when the PT input format is used to define the patch center. All specifications are then given in the global coordinate system.

2) A wire can be connected between two patches. However, only one end of a wire segment may be connected to a patch. Hence the wire connecting the two patches must consist of at least two wire segments.

Examples:

1) PA 0.0 0.0 5.0 0.0 0.0 100.0 1

This command would create a surface patch parallel to and 5 meters above the x-y plane in the global coordinate system. The patch has a surface area of 100 m^2, and it is identified by the tag number 1.

2) PT 4 0.0 5.0 0.0
PA PT 4 90.0 90.0 100.0 2

These commands place a patch with an area of 100 m^2 and identified by tag 2 at point number 4, which has the (x, y, z) coordinates of (0.0, 5.0, 0.0). The normal vector of the patch is the y axis.

3) PT 1 0.0 0.0 0.0
PT 2 -1.0 0.0 0.0
PT 3 0.0 1.0 0.0
PT 4 1.0 0.0 0.0
PT 5 0.0 -1.0 0.0
PA PT 1 PT 2 3 4 2.0 1

These commands place a 2 m^2 patch in the x-y plane centered at the origin. The vertices of the patch are on the x, y, -x and -y axes. The normal vector of the patch is the z axis. The same patch would result if the last command of the input stream is

PA 0.0 0.0 0.0 PT 2 3 4 2.0 1

PL

Plate Specification

```
PL   n   N1   N2   N3  [N4]  [N5]  [N6]
```

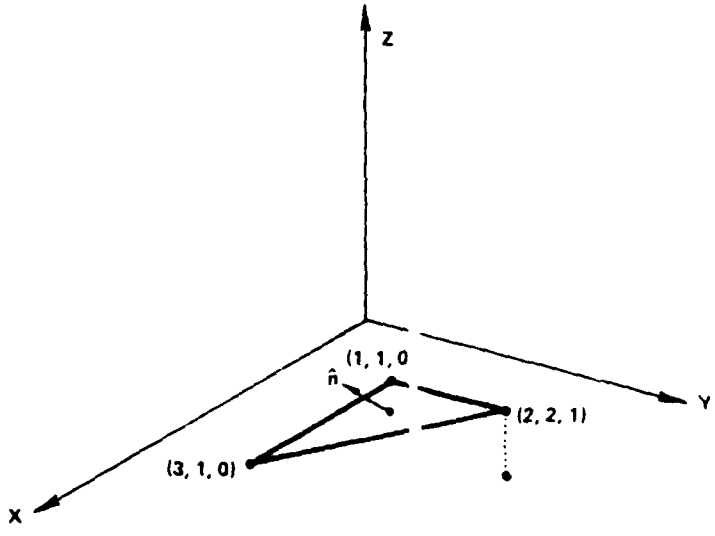
or

```
PL   n   X1  Y1  Z1   X2  Y2  Z2   X3  Y3  Z3
        [X4  Y4  Z4] [X5  Y5  Z5] [X6  Y6  Z6]
```

Parameter		Definition
PL	-	Plate specification code.
n	-	User-assigned plate number.
N1, N2,...,N6	-	Point number of plate corners (3-6).
(X1,Y1,Z1)... (X6,Y6,Z6)	-	Coordinates of plate corners (3-6) in global coordinate system.

This command allows the user to define a GTD plate by either referencing previously defined point numbers or specifying corner coordinates. The latter format is preferred as it does not require additional entries in the point table. A plate may have a minimum of three and a maximum of six corners. Fourteen plates are allowed in one geometry data set.

The plate in the figure can be defined by either of the following:



Triangle Plate Input Geometry

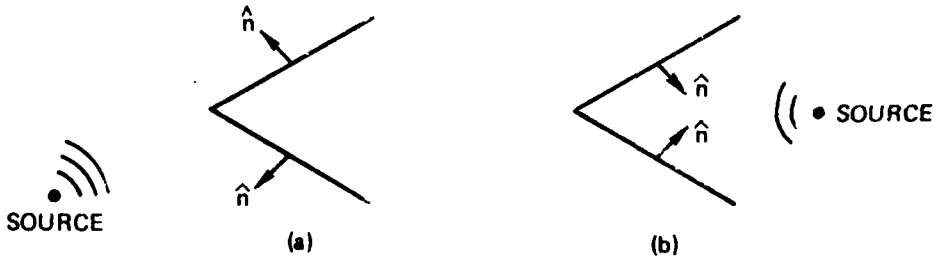
(a)

```
PT   1    1.0    1.0    0.0
PT   2    3.0    1.0    0.0
PT   3    2.0    2.0    1.0
PL  10    1    2    3
```

(b)

```
PL  10    1.0    1.0    0.0
*         3.0    1.0    0.0
*         2.0    2.0    0.0
```

The order of entry of the corners defines the plate normal by the right-hand rule. The plate normal must point outward from the geometry when diffraction interactions have been selected. This will define which angle is the exterior angle for edge and wedge diffraction coefficient calculations. For structures which are not strictly convex, there may be a dilemma if more than one source is used. In the figure below, the unit normals are correctly oriented for the source locations given.



However, had both sources been present in the same problem, incorrect answers would have resulted. The user may overcome this problem by working problems (a) and (b) separately and superposing the results.

A wire may be connected to a plate. The connection is determined by examining whether one end of the wire lies within the domain of the plate. Connecting a single wire segment to two plates is not allowed, nor is connecting a wire to a plate at less than a 20° angle from the plate. A wire connecting two plates must consist of at least two segments.

PSYM

Plane Symmetry Specification

PSYM N

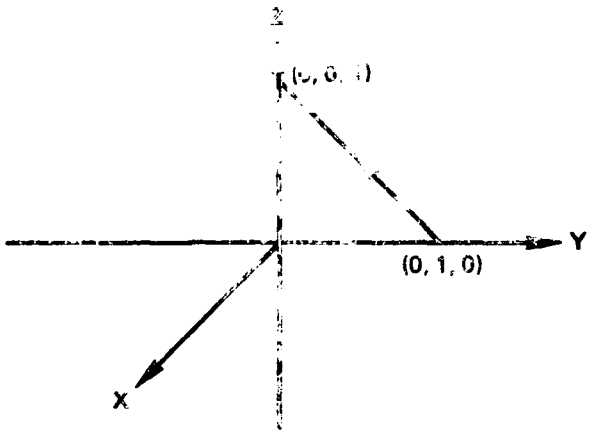
Parameter		Definition
PSYM	-	Plane symmetry specification code.
N	-	Number of planes of symmetry.

This command allows the user to specify that the geometry is made up of identical units that have been generated by use of the reflect command (RF). The presence of this command causes GEMACS to use the symmetry of the geometry to more quickly develop the interaction matrix and to solve the resulting matrix equation.

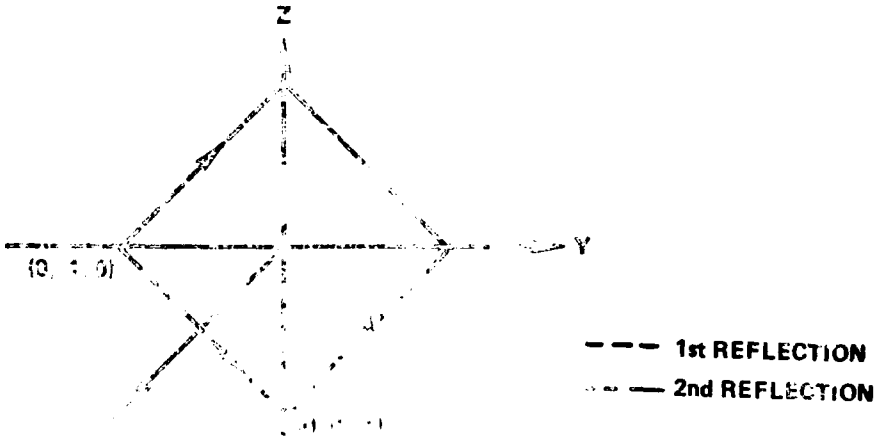
Example: (See figure)

```
RA 0.025
PT 1  0.0  1.0  0.0
PT 2  0.0  0.0  1.0
DF FIGURE
CP 1 2 30 1 1
RF DF FIGURE Y 1
RF DF FIGURE Z 2
DE
PSYM 2
END
```

The first three commands establish a radius for the segments and the locations of two points. The next five commands define 30 segments in the first quadrant of the Y-Z plane and then reflect them across the X-Z plane followed by a reflection of those 60 segments across the X-Y plane. The PSYM command specifies that there are two planes of symmetry in the geometry (X-Z and X-Y).



Original Wire



PT

Point Specification

PT NPT X Y Z [NCS]

Parameter		Definition
PT	-	Point specification code.
NPT	-	Integer identifier of this point. This must be a unique number.
X,Y,Z	-	(X,Y,Z) location of point with regards to NCS.
NCS	-	Integer identification of coordinate system for (x,y,z). (Default NCS = last defined NCS entry on any command.)

This command is used to specify points. The identifier NPT must be globally unique (not simply unique to coordinate system NCS). The point (x,y,z) will be transformed from coordinate system NCS into the global system before it is stored, if NCS is non-zero.

Examples:

```
CS 10  0.0  0.0  10.0  0.0  0.0  0.0
PT  3  1.   1.    1.   10
```

This defines point number 3 as being at (1.,1.,1.) in coordinate system 10. The point will be transformed to the global coordinate system before being stored, and its global coordinates in this case will be (1.0, 1.0, 11.0).

RA

Radii Specification

RA R1, R2, R3,...Rn

Parameter		Definition
RA	-	Radii table entry code.
R1,...Rn	-	Floating point values of the radii table will be loaded sequentially with these values.

Instead of appending the wire radius to each command that generates a wire segment, the user simply refers to an entry in the radii table. These entries are loaded sequentially from these RA items. Currently n $\leq$ 10.

Examples:

RA .001 .125 .0067

RA .025 1.00 .0003

Load radii entries 1 through 6 with the given values.

RF

Reflect Operation

$$RF \left\{ \begin{matrix} PT \\ TG \\ DF \\ PL \end{matrix} \right\} \left\{ \begin{matrix} n \\ n \\ name \\ n \end{matrix} \right\} \quad A1 \quad [A2] \quad [A3] \quad [INCTAG] \quad [NCS]$$

Parameter		Definition
RF	-	Reflection operation code.
$\left\{ \begin{matrix} PT \\ TG \\ DF \\ PL \end{matrix} \right\}$	-	Two-letter code to indicate point, tag, defined element, or plate is to be operated on.
$\left\{ \begin{matrix} n \\ n \\ name \\ n \end{matrix} \right\}$	-	For PT, PL and TG, the integer identifier; for DF, the alphanumeric name of the defined element.
A1	-	Alpha designation (x,y, or z) of axis along which first reflection will occur.
A2	-	Alpha designation of axis along which second reflection will occur, if one is designated.
A3	-	Alpha designation of axis along which third reflection will occur, if one is designated.
INCTAG	-	Tag or plate number increment parameter. (Default INCTAG = 0, wires only; must be specified for patches and plates.)
NCS	-	Integer identifier of coordinate system in which reflection is to take place. (Default NCS = last NCS value.)

This command causes the symmetry operation of reflecting through the plane normal to the axis specified. Wire segments in the planes of reflection are allowed. They are identified as such and will

not enter into the interaction matrix calculation. However, patches in the reflection plane are not allowed. Plates in the reflection plane are allowed but the duplicated plate serves no purpose. If the reflection operation takes place on an element currently being defined, all segments generated will be associated with the element being defined. When reflecting a patch or plate the increment must be specified. The user should use care to ensure that the new patch tags and plate numbers are globally unique. In addition, the user may not apply this command to a GTD cylinder and its end caps, or to a geometry in which a ground is present.

Examples:

1. RF DF WHEEL X 10

This will cause all points, segments, patches, and plates associated with WHEEL to be reflected through the y-z plane and their tags to be incremented by 10. A more complete geometry generating stream could be as follows:

```
DF WHEEL
PT 1 0.   0.   0.
PT 2 1.   0.   0.
RZ PT 2   2   45.0
RF DF WHEEL X Y 10
DE
```

This would generate the points in the positions shown in the figure. Note that points in the plane of symmetry are regenerated. This is strictly for the user's convenience in keeping up with the point identifiers. NPTS will result in 2^n*NPTS points incremented sequentially where n is the number of planes of reflection and NPTS is the number of points being reflected. The additional commands:

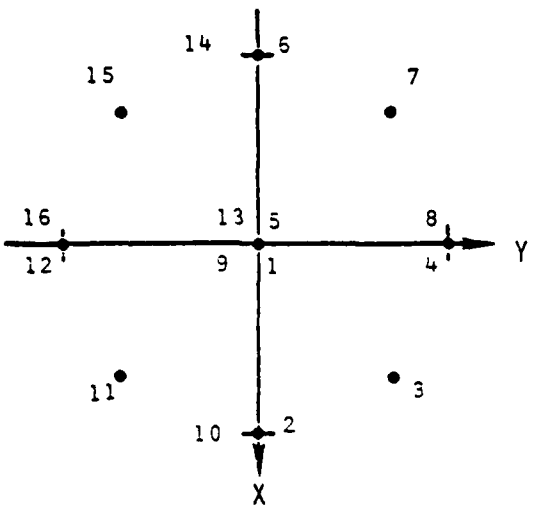
```
MP   13   1   2   3   1   4   7   1   6   15   1   16   11
*1   1   0   1
CP   11   10   1   0   1
CP   3   4   1   0   1
CP   7   6   1   0   1
CP   15   16   1   0   1
```

would generate the outline of a spoked wheel. The user could have substituted 5, 9, or 13 for point 1, 10 for 2, 8 for 4, 14 for 6, and 12 for 16 with the same result.

2.
```
DF WHEEL
PT  1  0.  0.  0.
PT  2  1.  0.  0.
RZ  PT  2  1  45.
PT  4  0.  1.  0.
MP  6  1   2   3   1   4   3   1   0   1
DE
RF  DF  WHEEL  X Y  5
```

These commands would generate not only identical points, but identical wire segments. These would eventually be detected during the search for multiple junctions and flagged as null segments. The results would be identical to those obtained in example 1.

RF (Concluded)



Point Positions

RN

Renumber Operation

RN I1, I2, I3,..., -In, In+2,..., I

Parameter		Definition
RN	-	Renumber operation code.
I1,...I	-	Integer numbers to control the resequencing of wire segment numbers.

When using the banded matrix iteration technique, it may be important to have the wire segments numbered correctly in order for the solution to converge. When a RN operation is encountered, all wire segments which have been generated since the last RN are renumbered according to the sequence specified by the integers on the RN command. The resequencing will start with the first segment generated since the last RN operation. The segment numbers are changed to correspond to the integers on the RN directive. When a negative integer is encountered (e.g., -In), the sequence number of the next wire segment will be the absolute value of the negative integer, and the sequence numbers will be incremented by 1 until the number of segments identified by the next item has been sequenced (e.g., In+2).

Example:

Suppose the user has generated segments 1 through 10 and wishes to renumber the segments:

RN 2 4 -5 , 5 , 1 , 3 , 10

Old Segment Numbers	1	2	3	4	5	6	7	8	9	10
New Segment Numbers	2	4	5	6	7	8	9	1	3	10

Neither GTD geometries nor patches are affected by the RN command.

RSYM

Rotational Symmetry Specification

RSYM N

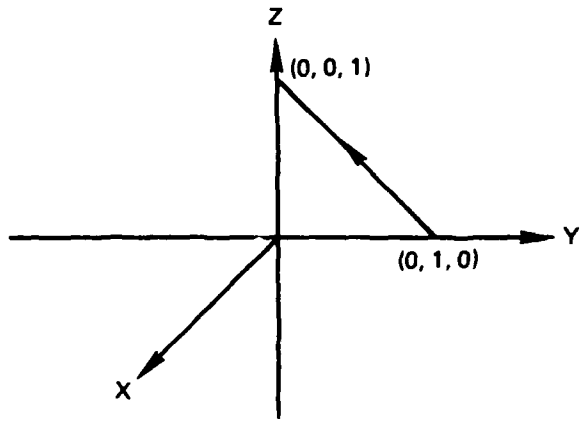
Parameter		Definition
RSYM	-	Rotational symmetry specification code.
N	-	*Number of symmetrical cells.*

This command allows the user to specify that the geometry is made up of identical cells that have been generated by use of the rotation command (RX, RY, RZ). The presence of this command causes GEMACS to use the symmetry of the geometry to more quickly develop the interaction matrix and to solve the resulting matrix equation.

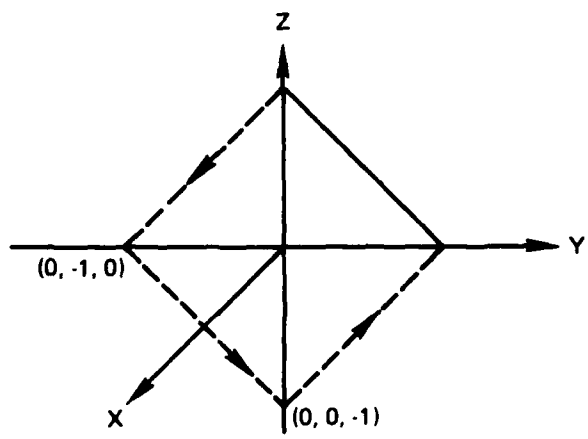
Example: (See figure)

```
RA  0.025
PT 1  0.0  1.0  0.0
PT 2  0.0  0.0  1.0
DF FIGURE
CP 1 2 30 1 1
RX DF FIGURE 3  90.0  1
DE
RSYM 4
END
```

The first three commands establish a radius for the wire segments and the locations of two points. The next four commands define 30 segments in the first quadrant of the Y-Z plane and then rotate it around the X-axis to generate 90 more segments in the Y-Z plane. The RSYM command specifies that there are four cells of symmetry in the geometry (one cell in each quadrant of the Y-Z plane).



Original Wire



Result of Rotation Command

RX, RY, or RZ

Rotation Operation

$$\left\{\begin{matrix}RX\\RY\\RZ\end{matrix}\right\}\left\{\begin{matrix}PT\\TG\\DF\\PL\end{matrix}\right\}\left\{\begin{matrix}n\\n\\name\\n\end{matrix}\right\} \quad \text{IADD ANGLE [INCTAG] [NCS]}$$

Parameter		Definition
$\left\{\begin{matrix}RX\\RY\\RZ\end{matrix}\right\}$	-	Rotation axis operation code.
$\left\{\begin{matrix}PT\\TG\\DF\\PL\end{matrix}\right\}$	-	Two-letter code designating point, tag, definition element or plate to be rotated.
$\left\{\begin{matrix}n\\n\\name\\n\end{matrix}\right\}$	-	For PT, PL and TG, n is the integer identifier for DF, "name" is the alphanumeric name of the defined element.
IADD	-	Integer indicating the number of additional objects or elements to be generated.
ANGLE	-	Rotation increment about specified rotation axis (positive is counterclockwise when looking toward origin along axis of rotation).
INCTAG	-	Tag or plate number increment for each additional element. (Default = 0, wires only; must be specified for patches and plates.)
NCS	-	Coordinate system identifier. (Default NCS = last NCS specified.)

The rotation operation can be used to generate structures which have axes of revolution. If the objects to be rotated are members of a defined element, the segment numbers of the original segments increment

by the number of segments in the original defined element for each additional element. If the DF operation is still in effect, all segments, patches, plates and points generated will be members of the element being defined. When rotating a patch or plate the increment must be specified. The user must ensure that the new patch tags or plate numbers are unique.

This command may not be applied to a GTD cylinder or to its end caps.

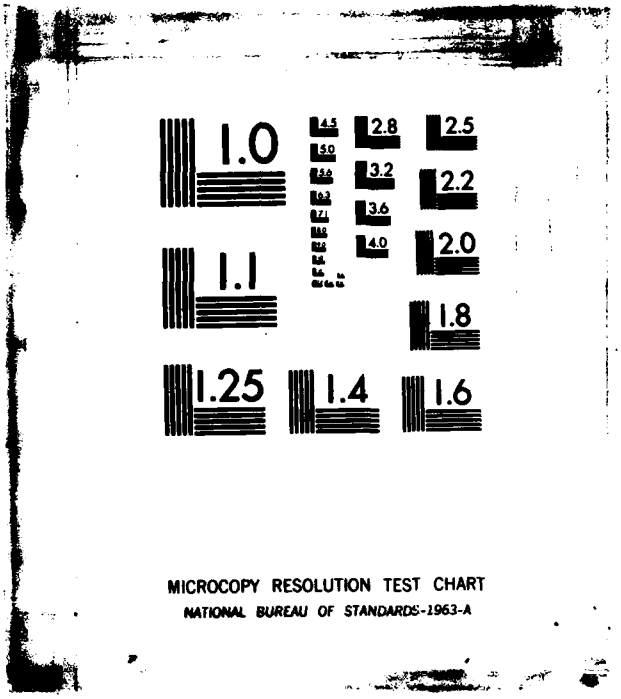
Example:

```
RA  .01
DF  CONE
PT  1  0.  0.  0.
PT  2  1.  0.  10.
PT  3  0.  0.  10.
RZ  PT  2  1  45.  $ ROTATE PT2  45° AROUND Z AXIS; GENERATE PT4
MP  3  3  2  4  1  0  1  $ CONNECT PTS 3, 2, 4
CP  2  1  10  $ CONNECT PTS 1 AND 2 WITH 10 SEG OF RADIUS .01
RZ  DF  CONE  7  45.  $ ROTATE BASIC ELEMENT TO COMPLETE CONE
DE
```

The RA command establishes the radius of all the segments generated. This is followed by the define element (DF) command. All points and segments generated from here to the next define end (DE) card will belong to the structure called CONE. The next three commands define the basic points, the result of which is shown in figure a. Then point 2 is rotated around the Z-axis to generate point 4. The next two commands connect these four points to define the basic pattern of the element CONE. This is shown in figure b. This basic pattern is then rotated around the Z-axis and regenerated seven more times to produce a wire-gridded cone. The DE command then closes the generation cards for CONE.

AD-A137 461 GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS. (U) BDM CORP ALBUQUERQUE NM D L KADLEC ET AL. SEP 83 BDM/A-83-170-TR 2/2

UNCLASSIFIED RADC-TR-83-217-VOL-1 F30602-81-C-0084 F/G 20/14 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

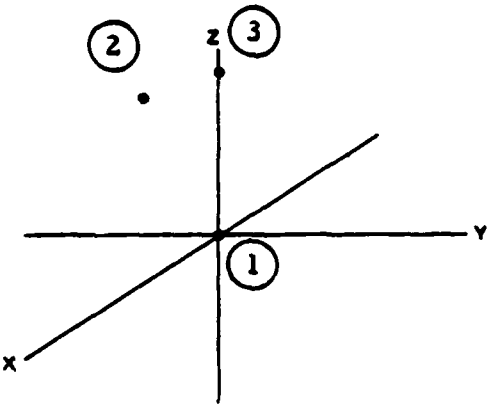


Figure a. Original Points 1, 2, 3

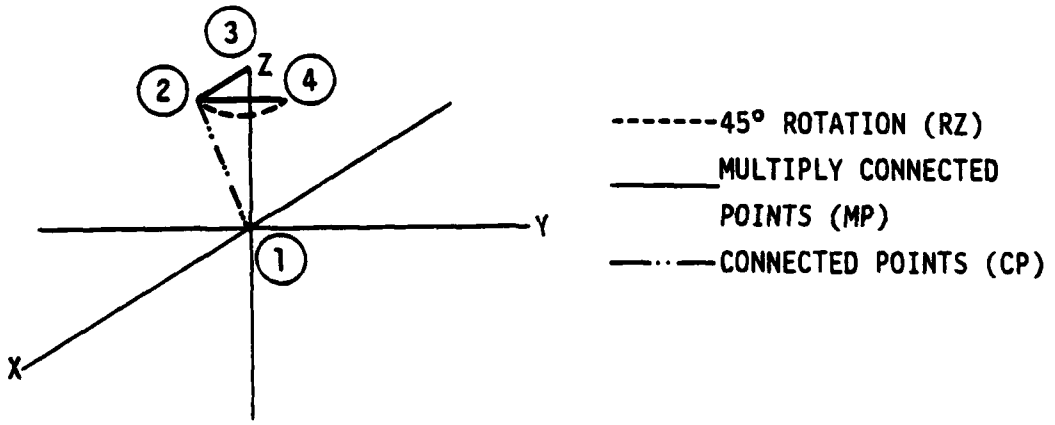


Figure b. RZ, MP, CP Operations Defining Basic Element of CONE

SC

Scale Parameter

$$SC \left[\left\{ \begin{matrix} value \\ FT \\ IN \\ CM \end{matrix} \right\} \right]$$

Parameter		Definition
SC	-	Scale specification code.
$\left\{ \begin{matrix} value \\ FT \\ IN \\ CM \end{matrix} \right\}$	-	Value is the numeric value of the scale factor in meters/unit. FT, IN, CM are two-letter codes that automatically scale the data from feet, inches, and centimeters to meters. (Default value = 1)

The GEMACS code uses the MKS system of units. Unless specified otherwise, all geometry input is assumed to be in meters. When an SC command is read, all subsequent dimensions until the next SC command are scaled by the item on the former SC command.

Example:

SC FT

All subsequent data would be converted from feet to meters before being stored.

WR

Wire Input

WR X1, Y1, Z1, X2, Y2, Z2, [,NSEG] [,NTAG] [,NRAD] [NCS]

Parameter		Definition
WR	-	Wire input designator.
X1,Y1,Z1	-	Coordinates of what will be considered the negative end of the wire segment.
X2,Y2,Z2	-	Coordinates of what will be considered the positive end of the wire segment.
NSEG	-	Number of segments into which the wire is to be divided. (Default to last NSEG entry of any previous operation.)
NTAG	-	Integer tag identifier of each segment. (Default to last NTAG entry of any previous operation.)
NRAD	-	The location of the radius of each segment in the radii table. (Default to last NRAD entry of any previous operation.)
NCS	-	Integer identifier of coordinate system. (Default to last NCS entry of any previous operation.)

This is the preferred method of generating wire segment data in GEMACS. It will generate NSEG segments between (X1, Y1, Z1) and (X2, Y2, Z2) with tag identifiers of NTAG, of radius specified by NRAD. If NCS is not equal to zero, (X1, Y1, Z1) and (X2, Y2, Z2) will be the end points in coordinate system NCS.

Example:

WR 1.63, 2.47, 3.67, 26.4, 16.3, 43., 10,2,1,1

This would generate 10 segments with tag number 2 and with a radius specified by the first radius entered previously in the radii table. The end points would be transformed from coordinate system 1 to the global system before storing segment end points.

XL

Translation Operator

$$XL \begin{Bmatrix} PT \\ TG \\ DF \\ PL \end{Bmatrix} \begin{Bmatrix} n \\ n \\ name \\ n \end{Bmatrix} \quad IADD \quad DX \quad DY \quad DZ \quad [INCTAG] \quad [NCS]$$

Parameter		Definition
XL	-	Translation operation code.
$\begin{Bmatrix} PT \\ TG \\ DF \\ PL \end{Bmatrix}$	-	Two letter code designating point, tag, defined element, or plate to be translated.
$\begin{Bmatrix} n \\ n \\ name \\ n \end{Bmatrix}$	-	For PT, PL and TG, n is the integer identifier; for DF, "name" is the alphanumeric name of the defined element.
IADD	-	Integer indicating the number of additional objects or elements to be generated.
DX,DY,DZ	-	Incremental translation vector. Each must be present, even if equal to zero.
INCTAG	-	Tag or plate number increment for each additional element. (Default = 0, wires only; must be present for patches and plates.)
NCS	-	Coordinate system in which translation is to take place. (Default = last NCS value used on any previous command.)

The translation operation will generate IADD additional segments, patches, points, plates and defined elements identical to those specified. Each will be displaced from the original by (DX,DY,DZ) in the coordinate system specified. If IADD is zero, the objects are simply

translated to a new location. This last operation can be performed by defining a coordinate system at the new location and performing an AT (attach) operation.

When translating a patch or plate the increment must be specified. The user must ensure that the new patch tags or plate numbers are globally unique. This command may not be applied to a GTD cylinder or its end caps.

Examples:

1.
```
RA   .01
PT   1  0.  0.  0.
PT   2  0.  1.  0.
PT   3  0.  0.  1.
DF   GRID
MP   3  3  1  2  1  1  1
XL   DF  GRID 9  0. 1. 0.
WR   0. 10. 0.  0.  10.   1. 1
XL   DF  GRID 9 0.  0.   1.  1
WR   0. 0. 10. 0.  10.  10. 10
DE
```

The MP command would generate segments 1 and 2 as illustrated in figure a. The first XL command would generate segments 3 through 20, and the WR command would result in segment 21 of figure b. The second XL command would generate segments 22 through 210, and the WR command would close the grid with segments 211 through 220. This is illustrated in figure c. Future references to GRID would encompass segments 1 through 220.

XL (Continued)

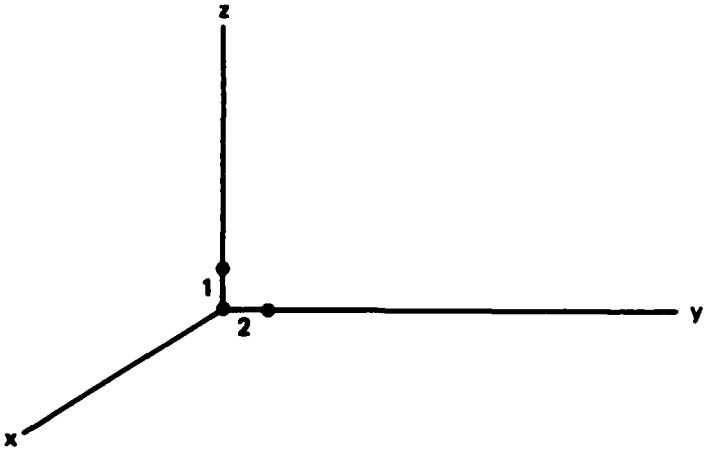


Figure a. Segment Locations

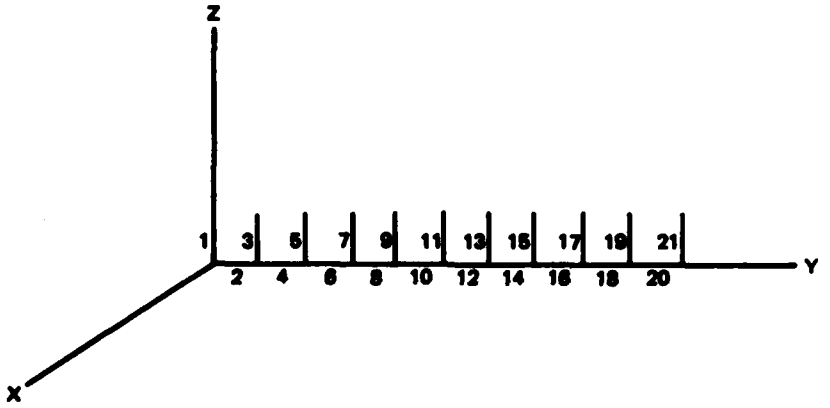


Figure b. First Row of Wire Grid

XL (Concluded)

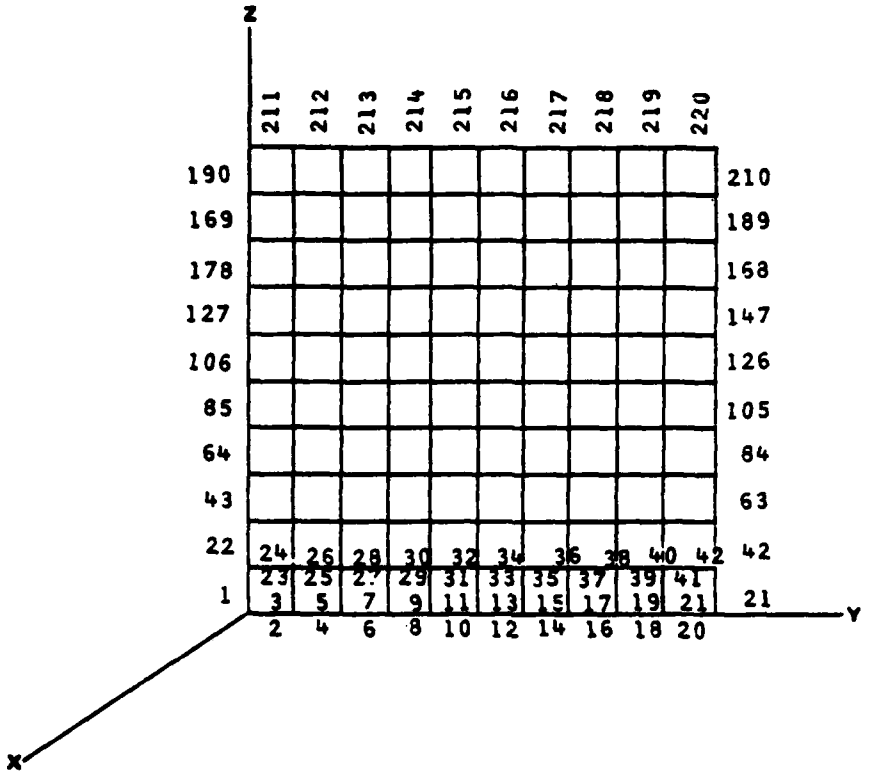


Figure c. Completed Wire Grid

2. PA 0.0 0.0 5.0 0.0 0.0 100.0 2
   XL TG 2 2 0.0 10.0 0.0 1

   This pair of commands would generate patches located at (0.0, 0.0, 5.0), (0.0, 10.0, 5.0), and (0.0, 20.0, 5.0) with tag numbers 2, 3, and 4 respectively. Each patch has a surface area of 100 m^2 and its normal vector is in the Z direction.

D. COMPUTER REQUIREMENTS

GEMACS is written in American Standard FORTRAN, X 3.9-1966. It is capable of executing with no library subroutines other than those required by the ANSI standard. The code consists of four sequentially executable modules, whose computer memory requirements are given in table 2. These values are approximate and depend on the specific machine and load method utilized.

TABLE 2. GEMACS COMPUTER MEMORY REQUIREMENTS

Module	Decimal Core Words	Module is required for MOM	GTD	Hybrid
INPUT	70K	X	X	X
GTD	120K		X	X
MOM	82K	X		X
OUTPUT	50K	X	X	X

The four modules execute sequentially, as shown in figure 1. However, should the functions of a module not be required in problem solution, that module may be bypassed. For instance, a MOM-only problem would not require the GTD module, so only the INPUT, MOM, and possibly OUTPUT modules would be needed.

The only communications medium between the modules is the checkpoint file. This file contains the contents of important commons and the contents of all data files present at the end of a module's execution. This allows two types of runs to be made. One may either execute all four modules in one batch job, or one may execute the modules in successive jobs, examining all intermediate results before proceeding to the next module. If the latter is the case, the checkpoint file should be a permanent file.

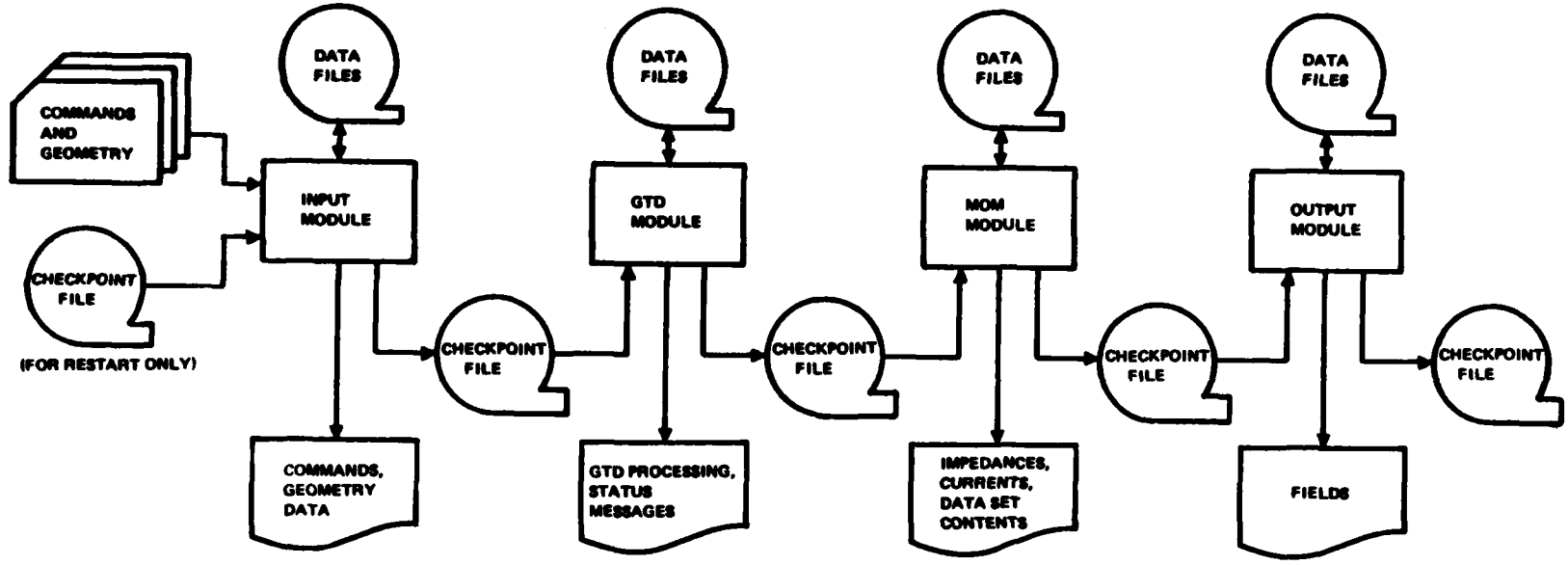


Figure 1. Communications Among the Four GEMACS Modules

Execution always begins with the INPUT module. The INPUT module examines the user's commands and checks for correctness and compatibility. The geometry inputs are processed to generate the geometry data sets required for problem execution.

The GTD module performs all calculations required for geometrical theory of diffraction modeling. Since it does not generate any results directly, the only output from the GTD module is a series of status messages which indicate the GTD tasks being accomplished.

The MOM module performs all calculations required for method of moments modeling and numerical solution of the resulting matrix problem. The MOM module generates solution currents and impedances and displays data set contents via PRINT or WRITE commands.

The OUTPUT module is the last module to execute. It produces field pattern printouts and plots.

1. I/O Requirements

GEMACS makes extensive use of peripheral file storage and must have several FORTRAN logical units available. These are listed in table 3 with the internal variable name, the logical unit number, and the file usage given. Data are stored starting on logical unit IOSYMB up to and including the logical unit number specified on the NUMFIL input. The user is responsible for assuring that GEMACS can access these files. If more files are required than made available, a fatal error will occur, and an attempt will be made to write a checkpoint. If a logical unit (LU) designator is specified on the CHKPNT, GMDATA, RSTART, or WRITE commands, it should be units 3 or 4, or greater than that specified on the NUMFIL command. This will ensure that the unit is not in use when required. Table 4 lists the types of data sets and the number of logical units required for storage for each type.

Execution of a PURGE command releases the logical unit used for the specified data set for other use.

TABLE 3. GEMACS LOGICAL UNITS

INTERNAL DESIGNATOR	LOGICAL UNIT NUMBER	USE
IOSCR1	1	Scratch File
IOSCR2	2	Scratch File
LUTASK	5	Card Image Input
LUPRNT	6	Formatted Output
IOCKPT	7	Checkpoint File (Binary)
IOSYMB	8	Data Storage (Binary)
LUNIT	9	Data Storage (Binary)
LUNIT	10	Data Storage (Binary)
	.	
	.	
	.	
	.	
LUNIT	NUMFIL	Data Storage (Binary)

TABLE 4. DATA SET LOGICAL UNIT REQUIREMENTS

DATA SET TYPE	NUMBER OF UNITS MOM	GTD	MOM/GTD
GEOMETRY	1	1	1
EXCITATION	1	0	1
SOLUTION	1	0	1
IMPEDANCE	1	0	2
BANDED*	1	0	1
DECOMPOSED	2	0	2
FIELD	1	1	2
MINIMUM VALUE FOR NUMFIL**	14-15	9	16-17

*Required for BMI solution; optional otherwise.
**Includes the five required files listed in table 3.

2. Internal Storage Requirements

There are several primary arrays used by GEMACS for problem execution. All arrays are stored in common blocks and have internal variables specifying their size.

SEGTBL is used to store the segment, patch, plate, GTD cylinder, and end cap data. It is dimensioned as a NPRSEG by MAXSEG array. The geometry data are stored on a logical unit (LU) and as needed are input to and output from main memory in blocks of at most MAXSEG columns. Although MAXSEG is presently set at 500, this does not limit the number of wire segments and patches that are used to model the MOM geometry. This limit is 20,000 segments and patches. As stated previously, version 3 of GEMACS allows for 1 cylinder, 2 end caps and 14 plates which may have at most 6 corners.

PTTBLE is used to store point data and is dimensioned as a MAXPTS by NPRPT array. MAXPTS is the maximum number of points to be stored, which is presently dimensioned for 100 points.

IDEFIN is used to store the data for defined elements and is dimensioned as a MAXDEF by NPRDEF array. MAXDEF is the maximum number of defined elements allowed, which is presently 100 elements.

CVAL is used to store coordinate system data and is dimensioned as a MAXCSY by 6 array. MAXCSY is the maximum number of coordinate systems allowed, now set to 10. CVAL is equivalent to CX in the CSYSTM common block.

The array TEMP is stored in common TEMP01 and must be of dimension NTEMPS, presently set to 5500. This array is used throughout the code for internal computation and intermediate storage. At present, TEMP must be capable of containing at least three columns of the interaction matrix. This implies that NTEMPS must be greater than six times the value of MAXSEG since the interaction matrix is complex.

All array parameters are stored in the same common block as the array and are preset in block data subroutine BLKDAT.

The maximum number of plates allowed for GTD modeling may be increased by changing (1) the dimensions of arrays associated with plate

modeling, (2) the variables used for plate geometry processing and (3) the variables directing execution of the GTD phsyics. The arrays that must be changed are:

ARRAY NAME	COMMON BLOCK
XI(14,14,3)	IMAINF
VXI(3,3,14)	IMAINF
LSURF(14)	SURFAC
LSHD(14)	LSHDT
LIHD(14,14)	LSHDT
LDC(14,6)	LDCBY
FNP(14,6)	FNANG
X(14,6,3)	GEOPLA
V(14,6,3)	GEOPLA
VP(14,6,3)	GEOPLA
VN(14,3)	GEOPLA
MEP(14)	GEOPLA
LRFI(14)	CLRFI
LRFS(14)	CLRFS
LDRC(14,6)	CLDRC
LRDC(14,6)	CLRDC
XX(14,6,3)	PLAIN

The plate geometry variables are located in COMMON /GTDDAT/ and are defined as (default in parentheses):

MAXPLT (14) maximum number of plates

The variables directing GTD physics are found in subroutine GTDDRV (GTD module):

MXPLT--maximum number of GTD plates

3. System Library Routines

No system library routines are required; however, some are desirable. The most important is a routine to return the elapsed processing time in minutes. This routine is called from subroutine

TICHEK and must be available when using the CHKPNT command with the CPINC item.

Auxiliary routines to return the date and time are called by subroutine SYSRTN. In the absence of these routines, zeros should be returned to the calling routine.

The file status function routine LUSTAT is called after each read to detect an end of file. If a library function is available to determine this information, it should be called from this routine. If none is available, simply return a zero value for the function.

4. Code Verification Example

The following example problem can be used to test that all four GEMACS modules have been properly loaded on the host computer. The INPUT module is run first with the commands and geometry data as input.

```
$
$ GEMACS GTD TEST PROBLEM -- THIELE'S MONOPOLE OVER FINITE PLATE
$       DISTANCE OF MONOPOLE FROM PLATE EDGE IS 0.75 METERS
$ TESTS STRUCTURE OF CODE
$       MODULE COMMUNICATIONS
$       ALL PLATE GTD INTERACTIONS
$       VALIDATION OF SIMPLE PHYSICS (NEAR-FIELD)
$
$-----------------------------------------------------------------
$
DEBUG STATS
NUMFIL=17
TIME=4.
FRQ=500.
SETINT PL MM EI
GMDATA=THIELE
ZGEN GMDATA = THIELE ZMATRX = ZGARY
SRC=VSRC(THIELE) V=1.,0. SEGS=1
SOLVE ZGARY*I=SRC
PRINT I
FLDDIP=EFIELD(I)  P1=0.  T1=0.  DT=5.  T2=180.
END

RA .00762
PT 1 0. 0. 0.
PT 2 0. 0. 0.25
CP 1 2 6 1 1
PT 3 -0.75 -0.75 0.
PT 4 0.75 -0.75 0.
PT 5 0.75 0.75 0.
PT 6 -0.75 0.75 0
PL 1 3 4 5 6
END
```

The INPUT module will generate a list of geometry elements.

```
                                    WIRE SEGMENT DATA

                ENDPOINTS                                       CENTERPOINTS                RADII      LENGTH     END1 END2 NSEG TAG
   XN          XP          YN          YP          ZN          ZP          XC          YC          ZC
0.          0.          0.          0.          0.          4.167E-02 0.          0.          2.083E-02 7.620E-03 4.167E-02  0   -2   1    1
0.          0.          0.          0.          4.167E-02 8.333E-02 0.          0.          6.250E-02 7.620E-03 4.167E-02  1   -3   2    1
0.          0.          0.          0.          8.333E-02 0.125     0.          0.          0.104     7.620E-03 4.167E-02  2   -4   3    1
0.          0.          0.          0.          0.125     0.167     0.          0.          0.146     7.620E-03 4.167E-02  3   -5   4    1
0.          0.          0.          0.          0.167     0.208     0.          0.          0.188     7.620E-03 4.167E-02  4   -6   5    1
0.          0.          0.          0.          0.208     0.250     0.          0.          0.229     7.620E-03 4.167E-02  5    0   6    1

TOTAL WIRE SURFACE AREA 1.197E-02
END 1 OF SEGMENT     1 IS CONNECTED TO PLATE 1 AT (X,Y,Z)= 0.          0.          0.

PLATE DATA
PLATE#          X          Y          Z          X          Y          Z          X          Y          Z
  1        -0.750     -0.750      0.       0.750     -0.750      0.        0.750      0.750      0.
           -0.750      0.750      0.
```

Upon termination, the INPUT module writes an end-of-module checkpoint to be used in starting the GTD module.

```
 CHECKPOINT NUMBER     1 STARTED AT TIME     0.124 ON LOGICAL UNIT     7
 COMMON BLOCK ADEBUG WRITTEN OUT TO   ICKFIL
 COMMON BLOCK AMPZIJ WRITTEN OUT TO   ICKFIL
 COMMON BLOCK ARGCOM WRITTEN OUT TO   ICKFIL
 COMMON BLOCK CSYSTM WRITTEN OUT TO   ICKFIL
 COMMON BLOCK DEFDAT WRITTEN OUT TO   ICKFIL
 COMMON BLOCK FLDCOM WRITTEN OUT TO   ICKFIL
 COMMON BLOCK GEODAT WRITTEN OUT TO   ICKFIL
 COMMON BLOCK IOFLES WRITTEN OUT TO   ICKFIL
 COMMON BLOCK JUNCOM WRITTEN OUT TO   ICKFIL
 COMMON BLOCK PARTAB WRITTEN OUT TO   ICKFIL
 COMMON BLOCK SCNPAR WRITTEN OUT TO   ICKFIL
 COMMON BLOCK SEGMNT WRITTEN OUT TO   ICKFIL
 COMMON BLOCK SYMSTR WRITTEN OUT TO   ICKFIL
 COMMON BLOCK SYSFIL WRITTEN OUT TO   ICKFIL
 COMMON BLOCK TEMPO1 WRITTEN OUT TO   ICKFIL
 COMMON BLOCK GTDDAT WRITTEN OUT TO   ICKFIL
 COMMON BLOCK MODULE WRITTEN OUT TO   ICKFIL
 COMMON BLOCK INTMAT WRITTEN OUT TO   ICKFIL
 PERIPHERAL FILE 8 SYMBOL THIELE NUMREC=     1
 CHECKPOINT COMPLETE AT      0.131
 ELAPSED TIME=      0.007
 CURRENT FILE LENGTH=      23421
STOP AT LINE      1......
```

The GTD module begins execution from the checkpoint. No input data are used.

```
GEMACS START-UP PROCESSOR CALLED ON 03/02/83 AT  10.02

     BEGIN START-UP PROCEDURE
     COMMON ADEBUG READ WITH   1107 WORDS
     COMMON AMPZIJ READ WITH     71 WORDS
     COMMON ARGCOM READ WITH    104 WORDS
     COMMON CSYSTM READ WITH     72 WORDS
     COMMON DEFDAT READ WITH    506 WORDS
     COMMON FLDCOM READ WITH      7 WORDS
     COMMON GEODAT READ WITH    120 WORDS
     COMMON IOFLES READ WITH    200 WORDS
     COMMON JUNCOM READ WITH    205 WORDS
     COMMON PARTAB READ WITH   3407 WORDS
     COMMON SCNPAR READ WITH    755 WORDS
     COMMON SEGMNT READ WITH   5531 WORDS
     COMMON SYMSTR READ WITH    102 WORDS
     COMMON SYSFIL READ WITH     36 WORDS
     COMMON TEMPO1 READ WITH   5502 WORDS
     COMMON GTDDAT READ WITH     14 WORDS
     COMMON MODULE READ WITH     13 WORDS
     COMMON INTMAT READ WITH    131 WORDS
PERIPHERAL FILE 8  SYMBOL THIELE NUMREC=     1
     START-UP COMPLETE
     MODULES PREVIOUSLY RUN: INPUT
```

The only output from the GTD module is a list of status messages, as shown below:

```
GEMACS-GTD    TASK EXECUTION STARTED ON 03/02/83 AT  10.02

NUMBER OF PERIPHERAL FILES AVAILABLE    17


RUN TIME SET TO    4.00    CPU MINUTES


FREQUENCY SET TO    500.    MEGAHERTZ
WAVLENGTH    0.600    METERS


FILL IMPEDANCE MATRIX ZGARY
USING BASIS FUNCTION SINCOS
ON GEOMETRY DATA SET THIELE
SHADOWING MATRIX      ZGSHD
FREQUENCY(MEGAHERTZ)   500.00

INTERACTIONS SET:    PR     RR     PD     RD     PDR    PL    MM    EU    ES
EXCITE GEOMETRY DATA THIELE
EXCITATION VOLTGE
EXCITATION DATA    SRC
INTERACTIONS SET:    PR     RR     PD     RD     PDR    PL    MM    EU    ES
VOLTAGE EXCITATION WILL BE PERFORMED IN MOM MODULE

GENERATING GREENS FUNCTION MATRIX FLDGFM
INTERACTIONS SET:    PR     RR     PD     RD     PDR    PL    MM    EU    ES

GENERATING INCIDENT FIELD MATRIX FLDDIP
INTERACTIONS SET:    PR     RR     PD     RD     PDR    PL    MM    EU    ES
```

Finally, an end-of-module checkpoint is taken at the conclusion of the GTD module.

The MOM module is the third module run. It starts up from the GTD module checkpoint, then begins execution as shown.

```
GEMACS-MOM    TASK EXECUTION STARTED ON 03/02/83 AT  10.03

NUMBER OF PERIPHERAL FILES AVAILIBLE    17

RUN TIME SET TO    4.00    CPU MINUTES

FREQUENCY SET TO    500.    MEGAHERTZ
WAVLENGTH    0.600    METERS

FILL IMPEDANCE MATRIX ZGARY
USING BASIS FUNCTION SINCOS
ON GEOMETRY DATA SET THIELE
SHADOWING MATRIX      ZGSHD
LOADS(IF SPECIFIED)IN
FREQUENCY(MEGAHERTZ)   500.00
GROUND COND (MHOS/M)  0.
RELATIVE PERMITIVITY   1.0000

INTERACTIONS SET:    PR    RR    PD    RD    PDR    PL    MM    EU    ES
EXCITE GEOMETRY DATA THIELE
EXCITATION VOLTGE
EXCITATION DATA    SRC
INTERACTIONS SET:    PR    RR    PD    RD    PDR    PL    MM    EU    ES

REAL COMP    1.00    IMAG COMP    0.
EXCITED SEGS
   1

DECOMPOSE ZGARY STORE RESULT IN ZGARY PIVOT=    N

MAX DIAG =  15564.    MIN DIAG =   1580.9
PIVOT RATIO =    9.85

SOLVE ZGARY*    I=   SRC

GEOMETRY DATA SET THIELE

*** NO LOAD FOR STRUCTURE ***

ANTENNA/LOAD PARAMETERS (SEE USER  MANUAL)

          INPUT      INPUT
  SEGMENT IMP(MAG)   IMP(PHZ) PWR INPUT  PWR LOAD
      1  428.952       32.389 0.984E-03 0.

******************

SYMBOL     I
LINEAGE-    SRC- ZGARY-THIELE-
COMPLX DATA

******************
COLUMN-     1
  REAL           IMAGINARY   MAGNITUDE   PHASE(DEG)   REAL          IMAGINARY   MAGNITUDE   PHASE(DEG)
  1   0.1969E-02 -0.1249E-02 0.2331E-02  -32.39    2  0.1024E-02 -0.3394E-02 0.3545E-02  -73.22
  3   0.2492E-03 -0.4312E-02 0.4319E-02  -86.69    4 -0.2539E-03 -0.4235E-02 0.4242E-02  -93.43
  5  -0.4487E-03 -0.3310E-02 0.3341E-02  -97.72    6 -0.3037E-03 -0.1588E-02 0.1617E-02  -100.8

GENERATING FLDDIP FROM: INCIDENT FIELD MATRIX FLDDIP
                        GREENS FUNCTION MATRIX FLDGFM
                        SOLUTION MATRIX      I

LOADING COLUMNS   1 TO 74 TO GENERATE COLUMN  1 OF FIELD MATRIX
```

The MOM module ends execution with a checkpoint, from which the OUTPUT module begins execution. The purpose of the output module is to print (and plot) the field patterns generated by the previous modules.

```
E-FIELD MATRIX FLDDIP
SPHERICAL COORDINATE SYSTEM

FAR FIELD FOR FIELD DATA=FLDDIP -CURRENT/SOURCE DATA= I -GEOMETRY DATA=THIELE
NORMALIZATION FACTOR  0.328  V/M
     CONSTANT PHI= 0
                E(THETA)                          E(PHI)
 TH=      MAGNITUDE       PHASE(DEG)      MAGNITUDE       PHASE(DEG)      NORMALIZED
  0.      0.              0.              0.              0.              -100.
  5.0     0.666E-01         120.          0.              0.              -13.8
 10.0     0.103             119.          0.              0.              -10.0
 15.0     0.937E-01         114.          0.              0.              -10.9
 20.0     0.502E-01          87.2         0.              0.              -16.3
 25.0     0.621E-01          2.86         0.              0.              -14.4
 30.0     0.132             -14.1         0.              0.              -7.89
 35.0     0.192             -12.1         0.              0.              -4.63
 40.0     0.234             -4.30         0.              0.              -2.93
 45.0     0.263              6.09         0.              0.              -1.91
 50.0     0.287             16.7          0.              0.              -1.16
 55.0     0.307             25.6          0.              0.              -0.549
 60.0     0.322             31.7          0.              0.              -0.141
 65.0     0.328             34.9          0.              0.              0.
 70.0     0.322             35.5          0.              0.              -0.152
 75.0     0.306             33.7          0.              0.              -0.587
 80.0     0.282             29.7          0.              0.              -1.29
 85.0     0.253             23.6          0.              0.              -2.24
 90.0     0.260             0.202         0.              0.              -2.00
 95.0     0.240             -10.2         0.              0.              -2.71
100.0     0.221             -21.4         0.              0.              -3.40
105.0     0.203             -32.9         0.              0.              -4.14
110.0     0.181             -44.6         0.              0.              -5.13
115.0     0.153             -57.0         0.              0.              -6.63
120.0     0.116             -72.7         0.              0.              -9.03
125.0     0.762E-01         -99.7         0.              0.              -12.7
130.0     0.597E-01         -154.         0.              0.              -14.8
135.0     0.848E-01          163.         0.              0.              -11.7
140.0     0.112              143.         0.              0.              -9.33
145.0     0.115              132.         0.              0.              -9.12
150.0     0.848E-01          121.         0.              0.              -11.7
155.0     0.316E-01           90.3        0.              0.              -20.3
160.0     0.467E-01          -27.7        0.              0.              -16.9
165.0     0.928E-01          -43.8        0.              0.              -11.0
170.0     0.101              -48.5        0.              0.              -10.2
175.0     0.654E-01          -50.5        0.              0.              -14.0
180.0     0.              0.              0.              0.              -100.
```

The OUTPUT module concludes with a checkpoint. Should the user wish further results from this problem (another pattern cut perhaps) he may begin execution from the OUTPUT module checkpoint and not have to rework 75 percent of the problem.

```
CHECKPOINT NUMBER    4 STARTED AT TIME    0.079 ON LOGICAL UNIT    7
COMMON BLOCK ADEBUG WRITTEN OUT TO    ICKFIL
COMMON BLOCK AMPZIJ WRITTEN OUT TO    ICKFIL
COMMON BLOCK ARGCOM WRITTEN OUT TO    ICKFIL
COMMON BLOCK CSYSTM WRITTEN OUT TO    ICKFIL
COMMON BLOCK DEFDAT WRITTEN OUT TO    ICKFIL
COMMON BLOCK FLDCOM WRITTEN OUT TO    ICKFIL
COMMON BLOCK GEODAT WRITTEN OUT TO    ICKFIL
COMMON BLOCK IOFLES WRITTEN OUT TO    ICKFIL
COMMON BLOCK JUNCOM WRITTEN OUT TO    ICKFIL
COMMON BLOCK PARTAB WRITTEN OUT TO    ICKFIL
COMMON BLOCK SCNPAR WRITTEN OUT TO    ICKFIL
COMMON BLOCK SEGMNT WRITTEN OUT TO    ICKFIL
COMMON BLOCK SYMSTR WRITTEN OUT TO    ICKFIL
COMMON BLOCK SYSFIL WRITTEN OUT TO    ICKFIL
COMMON BLOCK TEMPO1 WRITTEN OUT TO    ICKFIL
COMMON BLOCK GTDDAT WRITTEN OUT TO    ICKFIL
COMMON BLOCK MODULE WRITTEN OUT TO    ICKFIL
COMMON BLOCK INTMAT WRITTEN OUT TO    ICKFIL
PERIPHERAL FILE  8  SYMBOL THIELE NUMREC=    1
PERIPHERAL FILE  9  SYMBOL  ZGARY NUMREC=    6
PERIPHERAL FILE 11  SYMBOL    SRC NUMREC=    1
PERIPHERAL FILE 16  SYMBOL      I NUMREC=    1
PERIPHERAL FILE 13  SYMBOL FLDDIP NUMREC=    1
PERIPHERAL FILE 10  SYMBOL  ZGSHD NUMREC=    1
PERIPHERAL FILE 12  SYMBOL FLDGFM NUMREC=   74
PERIPHERAL FILE 14  SYMBOL  ZGUPR NUMREC=    6
PERIPHERAL FILE 15  SYMBOL  ZGLWR NUMREC=    6
CHECKPOINT COMPLETE AT       0.092
ELAPSED TIME=       0.012
CURRENT FILE LENGTH=     25149
STOP AT LINE    1......
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

END

FILMED

02 — 84

DTIC