April 1976

EISI/A2200

SPAR

REFERENCE MANUAL

by

W. D. Whetstone

(NASA-CR-149943) SPAR REFERENCE MANUAL (Engineering Information Systems, Inc.) 312 p HC $9.75 CSCL 09B

N76-28861

Unclas

G3/61 45853

ENGINEERING INFORMATION SYSTEMS, INC.
5120 CAMPBELL AVENUE, SUITE 240
SAN JOSE, CALIFORNIA 95130
(408) 379-0730

FOREWORD

The SPAR system is developed and maintained by Engineering Information Systems, Inc. (EISI) under NASA prime contracts NAS8-30536 and NAS1-13977 through subcontract agreements LL90A1760K and LL90A1800K with Lockheed Missiles & Space Company, Sunnyvale, California. SPAR is funded jointly by the George C. Marshall Space Flight Center (MSFC) and the Langley Research Center (LaRC) of the National Aeronautics & Space Administration. The Contracting Officer's technical representatives are L. A. Kiefling, MSFC, and J. C. Robinson, LaRC.

The purpose of the SPAR reference manual is to define the functions and rules of operation of the system. This document is not intended to stand alone as an introductory guide for the new user. It is expected that new users will either attend introductory courses or be assisted and advised by analysts who have substantial experience with SPAR.

It is assumed that users are familiar with finite element theory and execution procedures (run set-up, control cards, etc.) for either Univac 1100 or CDC 6000-series computer systems.

CONTENTS

Section

Section

Section

Section

Section 1

INTRODUCTION

SPAR is a system of computer programs used primarily to perform stress, buckling,and vibrational analyses of linear finite element systems. As illustrated on Fig. 1-1, individual processors within the SPAR system are able to communicate directly and automatically through a body of information known as the data complex, which resides in temporary and/or permanently cataloged files. The data complex contains one or more direct access libraries, within which may reside any number of data sets produced by processors in the SPAR system. Each data set has an identifying name. For example, a certain set of vibrational modes might be named VIBR MODE 400 2. Through these names the SPAR processors are able to locate and access automatically all of the information needed to perform a particular function.

This form of system structure — multiple independent processors communicating via a common data base — has many advantages. From the user point of view, the following are among the principal advantages:

- Interactive operation. Effective interactive operation, via teletype and/or graphics terminals, is made possible. Most large-scale applications involve both batch and interactive runs.

- Data salvage. All information generated in a run may be retained in the data complex, thereby remaining available for use in future runs. This retention is accomplished automatically, without complicated restart

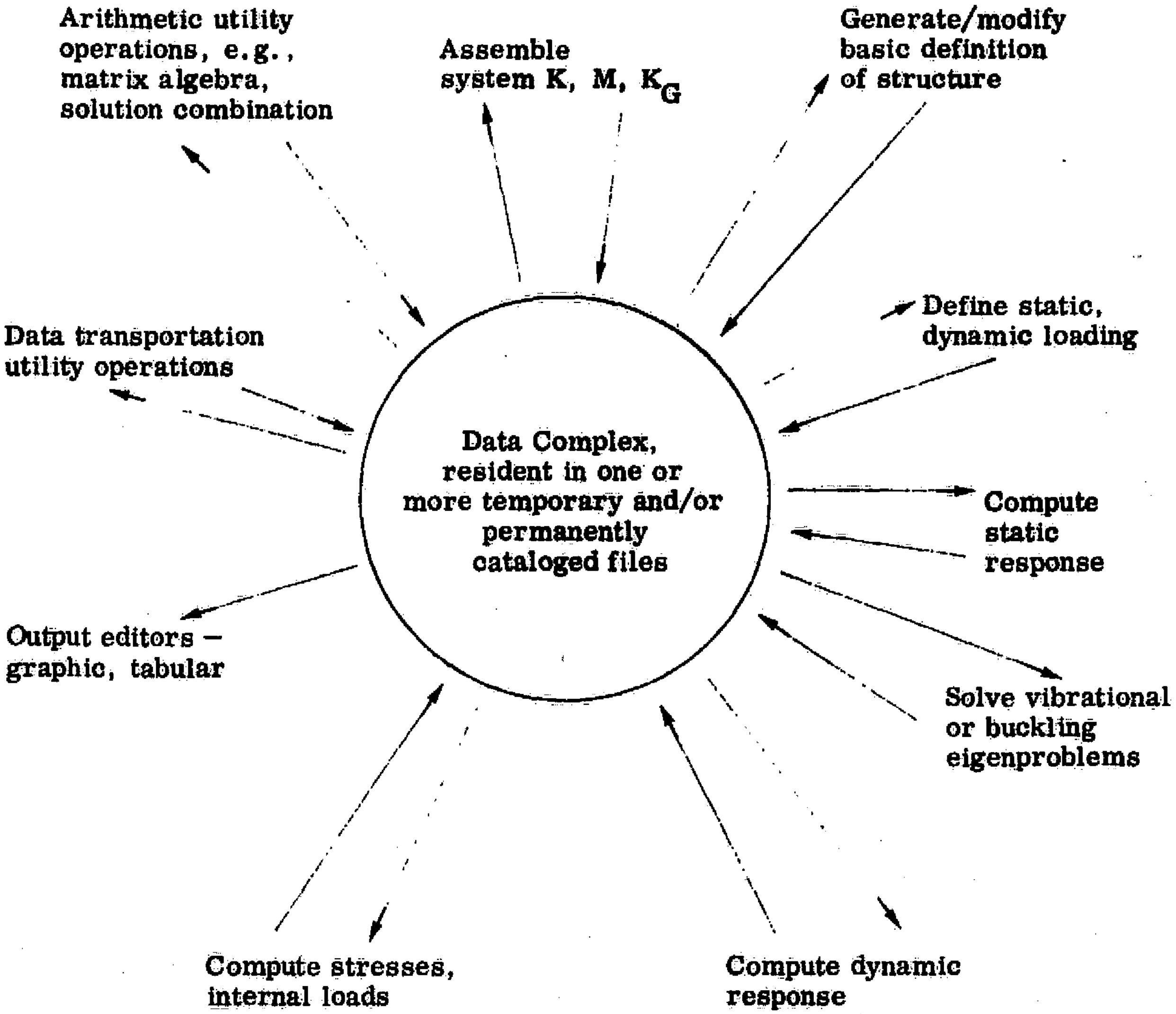

Arithmetic utility operations, e.g., matrix algebra, solution combination
Assemble system K, M, K_G
Generate/modify basic definition of structure
Data transportation utility operations
Define static, dynamic loading
Data Complex, resident in one or more temporary and/or permanently cataloged files
Compute static response
Output editors – graphic, tabular
Solve vibrational or buckling eigenproblems
Compute stresses, internal loads
Compute dynamic response

Figure 1-1 Typical Functions of Separate Programs, Communicating Through the Data Complex

procedures, and without requiring the user to be concerned with the internal structure of the data complex.

- Data sharing. Any number of users may obtain simultaneous access to the data complex, with file security (read/write authorization) provided by the cataloged file facilities of the host system.

Other characteristics of SPAR include the following:

- Efficiency. Execution time, central memory storage, and secondary data storage requirements are minimized by (1) the use of sparse matrix solution techniques, and (2) other computational and data management procedures developed to handle problems ranging in size from small one-shot studies to extensive analysis/design projects involving many millions of words of data.

  For most SPAR executions, $20,000_{10}$ to $30,000_{10}$ words of central memory are sufficient. On UNIVAC 1100 systems, for example, static displacement solutions to 6,000 degree-of-freedom problems may be computed using a total central memory field length of about $25,000_{10}$ words (all instructions plus data). During the course of a run, the central memory field length can be modified dynamically, if necessary, to permit additional memory resources to be temporarily acquired for the duration of a specific computational activity. The efficient use of system resources has the following principal effects:

  (1) Heavy interactive usage is possible without serious detrimental effects on the host operating system. This makes available the full range of benefits of interactive operation (e.g., major

reductions in labor cost and calendar time, with improved quality of results).

(2) Problems of very large size may be solved. One user has reported solving a 33,000 degree-of-freedom problem on a UNIVAC-1108. The size capacity of the eigensolver used to compute buckling and vibrational modes is approximately the same as that of the static displacement analyzer, for a given amount of central memory. Maximum degree-of-freedom capacity on typical UNIVAC 1108/1110 systems is in excess of 50000.

(3) Very low computer costs are achieved. The low cost of executing the eigensolver makes it possible, in many large-scale applications, to use the same finite element model for both static and dynamic analysis, resulting in further savings in labor costs and calendar time.

- Execution control and data input. All input is free-field. The command-oriented executive control language allows the user to design execution sequences optimally suited to the requirements of each individual application.

- Utility operations. The following operations are typical of the many available through SPAR utility processors:

  (1) Matrix operations through a high-level symbolic language, e.g., X = SUM(2.5 Y, 3.4 B), ZZ = PRODUCT(3.4 P,Q), etc.

(2) Communication, through ordinary files, with programs outside of the SPAR system.

(3) Display of selective summaries of the names and characteristics (size, type, etc.) of data sets resident in the data complex. These are called TOC's (Tables of Contents).

(4) Editing and display of information contained in the data complex.

- Input data generation, verification, and correction. Extensive facilities are provided for automated mesh generation, etc., and for checking the validity of input data. Where errors are detected, they can be patched locally, without having to re-run unaffected portions of the analysis.

Section 2

BASIC INFORMATION

On UNIVAC-1100 systems, SPAR consists of an array of separate absolute programs, usually resident in a read-only public cataloged file. On CDC systems, the entire system is contained in a single absolute program, configured to simulate UNIVAC operation; that is, CDC card input records appear as follows:

Card Image	Meaning
@XQT PROGX	Begin execution of program PROGX.
data cards	Input (usually optional) to PROGX.
@XQT PROGB	Begin execution of program PROGB.
data cards	Input to PROGB.
⋮	⋮
@XQT EXIT	Exit from SPAR (CDC only).
7/8/9	

The following are typical execution sequences:

Input	Function
@XQT TAB data cards	Create data sets containing tables of joint locations, section properties, material constants, etc.
@XQT ELD data cards	Define the basic mesh of finite elements.
@XQT E @XQT EKS	Form a complete detailed finite element model (element geometry, intrinsic stiffness and stress matrices, etc.) of the structure. The resulting data sets are referred to collectively as the E-State.
@XQT TOPO	Analyze element interconnectivity.
@XQT K	Form system K.

Input	Function
@XQT M	Form system consistent M, if required.
@XQT INV	Factor system K (or other designated system matrix, e.g., K + Kg or K - cM).
For static analysis:	
@XQT AUS data cards	Define applied loading, temperatures, etc.
@XQT SSOL	Compute static displacement solutions.
@XQT GSF	Compute stresses and internal loads.
@XQT PSF	Edit and display stresses.
For dynamic analysis:	
@XQT EIG control cards	Compute vibrational modes.
@XQT AUS data cards	Define applied loading, etc.
@XQT DR	Compute dynamic response.
For buckling analysis:	
@XQT KG	Form geometric stiffness matrix Kg, based on a designated internal load state.
@XQT EIG control cards	Compute buckling modes.

Detailed information concerning the functions and rules of operation of individual processors is provided in subsequent sections of this manual.

2.1 REFERENCE FRAME TERMINOLOGY

The term frame k will be used to refer to the global reference frame (k = 1), or to any alternate reference frame (k = 2, 3, 4, --) the analyst elects to define. Each joint in the structure has associated with it a unique joint reference frame, to which joint displacement and rotation components are relative. Individual joint reference frames may have any orientation designated by the analyst. Each element has an associated element reference frame, to which section properties and stress components are relative.

2.2 THE DATA COMPLEX

The data complex may consist of any number of files considered appropriate to a particular application. There are two kinds of files, namely:

- SPAR-format direct-access libraries, resident on random-access devices (disk, drum). Libraries are the media through which programs in the SPAR system are able to communicate. Users often elect to house the entire data complex in a single library file.
- Sequential files, resident on tape, drum, or disk. A large percentage of SPAR runs do not involve any sequential files. They are primarily used to store libraries on tape between runs. See Section 5.2, TWRITE and TREAD commands. These files are also used to communicate with programs outside the SPAR system. See Section 5.2, XCOPY and XLOAD.

Files are known by SPAR logical file numbers: 1 , 2 , . . . , 26. These are not Fortran logical unit numbers. The corresponding UNIVAC file names are SPAR-A, SPAR-B, . . . , SPAR-Z. The corresponding CDC file names are SPARLA, SPARLB, . . . , SPARLZ. If a SPAR program must use a file which does not already exist, it will generate internally the necessary requests to the host-operating system to assign (i.e., create) the file as a temporary file resident on random-access storage. The following examples illustrate the correspondence between UNIVAC external file names, internal file names, and SPAR logical file numbers.

- Example. A new study is being initiated, and it is desired to retain library 1 as a cataloged file named QUAL*NAME.

```
@ASG, UP          QUAL*NAME., F//POS/20
@USE    SPAR-A., QUAL*NAME.
@XQT    TAB (or any required program)
```

- Example. A library generated in preceding runs is resident in a cataloged file named GENGHIS*KAHN. To make the file known to SPAR programs as library 4, the following control cards would be used:

```
@ASG, A           GEHGHIS*KAHN.
@USE    SPAR-D., GENGHIS*KAHN.
```

SPAR logical files 1 through 20 are available for general use. Files 21 through 26 are usually reserved for temporary internal use. On CDC systems, not more than 10 files may be accessed concurrently from within a single program.

Individual data sets within a library are identified by four-word names. For each data set, there is an associated entry in a table of contents (TOC) which is automatically maintained within the library. Table 2.2-1 is a printout of a table of contents, produced by means of the TOC command in the Data Complex Utility program (see Section 5.2). TOC items are explained in Table 2.2-2.

In most cases, the program generating a data set will automatically assign the four-word name of the set. In certain instances, however, the user may assign part or all of the name. In such cases, the first two words should always be alphanumeric, and the last two should be integers.

In referencing a data set, one or more of the four words of the set name may be masked by replacing each word to be masked by the word, MASK. The primary uses of MASKing are (1) when less than four words of a data set name are sufficient to uniquely identify the set, or (2) when scanning TOC's in search of certain categories of data sets.

- Example. All of the following data sets correspond to VIBR MASK 2 MASK:

```
VIBR MODE   2   1
VIBR EVAL   2   1
VIBR MODE   2   2
VIBR EVAL   2   2
```

  Of these sets, only the first and third would correspond to VIBR MODE 2 MASK.

If a data set being inserted into a library has exactly the same name as one already contained in the library, the old data set will be automatically flagged as disabled. Although the disabled data set remains intact within the library, it cannot be read unless the user subsequently elects to re-enable it (see the ENABLE statement, Section 5.2), which will automatically result in the other data set being disabled.

When operating interactively, program execution should never be interrupted (e.g., on UNIVAC demand, a break followed by @@X T) if any data sets originated in the program. This is because certain TOC information is not finalized until the normal program exit procedure is executed.

Table 2.2-1: Example of TOC

```
TABLE OF CONTENTS, LIBRARY  2

BEAM KG COMPARISONS.

                                  E                              T   DATA SET NAME
SEQ       RR   DATE    TIME       R    WORDS      NJ  NI*NJ  Y    N1   N2   N3    N4
  1      -17  080775 172617     0      18        1     18   0 JDF1 BTAB    1     8
  2      -18  080775 172617     0       5        5      5   0 JREF BTAB    2     6
  3      -19  080775 172617     0      12        1     12   1 ALTR BTAB    2     4
  4       20  080775 172617     0      18        1     18   0 JDF1 BTAB    1     8
  5       21  080775 172617     0       5        5      5   0 JREF BTAB    2     6
  6       22  080775 172617     0      12        1     12   1 ALTR BTAB    2     4
  7       23  080775 172617     0      15        5     15   1 JLOC BTAB    2     5
  8       24  080775 172617     0      10        1     10   1 MATC BTAB    2     2
  9       25  080775 172617     0       5        1      5   1 MREF BTAB    2     7
 10       26  080775 172617     0      31        1     31   1 BA   BTAB    2     9
 11       28  080775 172617     0       5        5      5   0 CON          1     0
 12       29  080775 172617     0      45        5     45   1 QJJT BTAB    2    19
 13       31  080775 172617     0      64        4    896   0 DEF  E21     1     2
 14       34  080775 172617     0       2        1      2   0 GD   E21     1     2
 15       35  080775 172617     0      15        1     15   0 GTIT E21     1     2
 16       36  080775 172617     0      12       12     12   0 MELZ BTAB    1    11
 17       37  080775 172617     0       5        1      5   0 KE           0     0
 18       38  080775 172617     0       7        1      7   0 NS           0     0
 19       39  080775 172617     0       1        1      1   3 ELTS NAME    0     0
 20       40  080775 172617     0       1        1      1   0 ELTS LTYP    0     0
 21       41  080775 172617     0       1        1      1   0 ELTS NNOD    0     0
 22       42  080775 172617     0       1        1      1   0 ELTS ISCT    0     0
 23       43  080775 172617     0       1        1      1   0 ELTS NELS    0     0
 24       44  080775 172617     0       1        1      1   0 ELTS LE3     0     0
 25       45  080775 172617     0     560        4    140   4 E21  EFIL    1     2
 26       65  080775 172617     0      20       20     20   0 DIR  E21     1     2
 27       66  080775 172617     0      30        5     30  -1 DEM  DIAG    0     0
 28       68  080775 172617     0     896        5    896   0 KMAP         9     3
 29      100  080775 172617     0    1792        5   1792   0 AMAP @@@@    9     3
 30      164  080775 172617     0    2240        5   2240   1 K    SPAR   36     0
 31      244  080775 172617     0    3584        5   3584   1 INV  K       1     0
 32      372  080775 172617     0      30        5     30  =1 APPL FORC    1     1
 33      388  080775 172617     0      15        1     15   4 CASE TITL    1     1
 34      389  080775 172617     0      30        5     30  -1 STAT DISP    1     1
 35      391  080775 172617     0      30        5     30  -1 STAT REAC    1     1
 36      393  080775 172617     0     204        4   5600  -1 STRS E21     1     1
 37      401  080775 172617     0    2240        5   2240   1 KG   SPAR   36     0
 38     -481  080775 172617     0      18        1     18   4 NDAL         0     0
 39      492  080775 173455     0      18        1     18   4 NDAL         0     0
```

Table 2.2-2: TOC Information Summary

TOC Item	Meaning
SEQ	Sequence number, i.e., order of insertion into the library.
RR	Drum address pointer. A preceding minus sign means that the data set has been disabled.
DATE	Date of insertion.
TIME	Time of entry into the program which inserted the data set into the library.
ER	Error Code (0 = no error detected during generation of the data set; 1 = minor error; 2 = fatal error; -1 = incomplete data set).
WORDS	The total number of words in the data set. Data sets are generally comprised of a sequence of physical records, or "blocks." Each block is a two-dimensional matrix dimensioned (NI, NJ), i.e., NI rows, NJ columns. The block length is always NI*NJ.
NJ	See above.
NI*NJ	See above.
TY	Type code (e.g., 0 = integer, -1 = real, -2 = double precision, 4 = alphanumeric).

2.3 CARD INPUT RULES

The SPAR free-form input decoder recognizes three types of words: integer, floating-point, and alpha (typeless). Leading blanks are ignored. Each word is ended by a blank, comma, equal sign, slash, left parenthesis, or a right parenthesis, or by a record terminator (e.g., end-of-card). If used, commas, etc., should be carefully placed; for example 45 , is equivalent to 45, 0, . Floating-point numbers are identified by the presence of a decimal point. Alpha words must begin with a letter. Allowable forms for floating-point numbers are:

5000.= .5+4= .05+5= 50.00+02, etc.

Note that the Fortran form x.xExx is not permitted.

Each card begins an input record. A record is terminated by end-of-card, or by any of the three following symbols:

Record Terminator	Function
$	Characters to the right of a $ are ignored. This is used for two purposes: (1) to allow the input decoder to stop scanning, and (2) to allow the user to insert comments in the data deck. A card with a $ in Column 1 is interpreted as a comment card, and is ignored by the decoder.
5/8 punch*	The 5/8 punch (: on Univac) terminates one record and initiates a new record on the same card. For example,

*On CDC systems the 12-8-7 punch (;) serves this purpose.

3, 4 : A, 9.5 $

is the same as

3, 4 $
A, 9.5 $

4/8 Punch All characters to the right of a 4/8 punch (' on Univac, ≠on CDC) form a continuous alphanumeric label of up to 76 characters. Examples of label usage are shown below.

GROUP 42' RING FRAME, STA. 420.

CASE 240' 2.5 G GUST + 9.4 PSI PRESSURE.

Throughout this document, input card descriptions will use the Univac symbols (: for 5/8 punch, ' for 4/8 punch, @ for 7/8 punch).

Typeless words longer than 4 characters are truncated to 4 characters. Integer or floating-point words must not exceed 7 digits. Exponents may have one or two digits. Floating-point numbers must not exceed host system limits.

2.3.1 Equivalence of Word Terminators

The word terminators, (blank), comma, equal sign, left parenthesis, and right parenthesis, are equivalent. For example, the following statements have identical meaning:

```
Z= SUM( 3.5  R, 4.2  Q)
Z,  SUM  3.5  R  4.2  Q
```

2.3.2 Continuation Cards

If an input record will not fit on a single 80-character card, the 6/8 punch (a > on UNIVAC systems) may be used to indicate that the current record is continued on the next card. For example, the following record:

```
1.  2.  3.  4.  5.  6.  7.  $
```

could be written as

```
1.  2.>
3.  4.  5.  6.>
7.  $
```

The > symbol also acts as a word terminator. Continuation cards must not be used to extend an input record beyond 40 words.

2.3.3 Loop-Limit Format

There are many instances in which it is necessary to input an ordered list of integers; for example, a list of joint numbers, or a list of element index numbers, etc. For example, the input description given for a particular program may indicate that a list of integers, $n^1, n^2, n^3, n^4, \ldots,$ etc., should appear in the following form:

$$N= n^1: \; n^2: \; n^3: \; n^4:, \text{ etc.}$$

If it is specifically stated that "loop-limit format" is permitted, then any of the input records in the above sequence may also appear in the form $\underline{n^a, n^b, inc}$; implying the following sequence:

$$n^a, \quad n^a + inc, \quad n^a + 2(inc), \quad n^a + 3(inc) \text{ ---- } n^b .$$

If inc is omitted, it defaults to 1 .

For example:

J= 10: 11: 12: 13: 5: 6: 7: 2: 30: 40: 50: 60 $

may also be written as

J= 10, 13: 5, 7: 2: 30, 60, 10$

<u>Restrictions</u>:

- Unless specifically stated otherwise, inc should be positive.
- The sequence n^a, n^a + inc, etc. must terminate exactly on n^b; that is, $(1 + n^b - n^a)/inc$ must not be fractional.
- Unless specifically stated otherwise, a single loop-limit sequence must not consist of more than 100 input records.

2.4 RESET CONTROLS, CORE SIZE CONTROL, AND THE ONLINE COMMAND

RESET statements allow the analyst to alter certain parameters which control processor operation. Typical parameters available through RESET are listed below.

- Names of input and output data sets.
- Unit numbers of source and destination libraries (default is always SPAR logical 1).
- Flags indicating action to be taken if errors are encountered.
- Flags controlling display modes.
- Miscellaneous quantities, such as zero-test values, scale factors, output data set block lengths, formulation selections, constants, iteration controls, etc.

The form of a RESET statement is:

RESET $p_1=w_1$, $p_2=w_2$, $p_3=w_3$, - - -, where w_i is the value to be assigned to parameter p_i. One or more RESET statements may immediately follow a @XQT Processor command; for example:

```
@XQT INV
RESET  K=K+KG,  CON= 2 $

@XQT  EIG
RESET PROBLEM=  STABILITY$
RESET  CON=2, INIT=7, NDYN=8 $
```

In addition to specific RESET controls defined for individual processors, the following two parameters apply to all processors:

```
RESET  ABORT= 1$
```

which will cause the processor not to make an error abort if it encounters a serious error (e.g., if required input data sets do not exist), and

RESET CORE= n$(available on UNIVAC, only)

which will result in issuance of an executive request to change the total core size (both instructions and data) to n words.

On CDC systems, the user controls core size through RFL cards.

The statement, DATA SPACE = n, appearing at the beginning of execution of each program, indicates n = total field length less instruction storage.

Most SPAR programs generate little or no printed output. In some programs, the kind and quantity of output are controlled by a command (not a reset parameter) in the following form:

ONLINE= n$

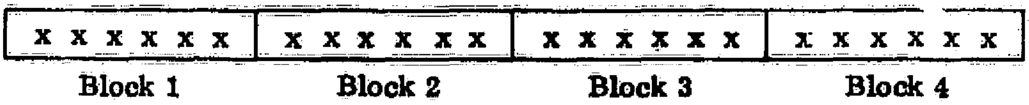
where n = 0 for minimum printout, 1 for normal printout, and 2 for maximum printout. If desired, the ONLINE statement may be used more than once within the same program execution.

2.5 DATA SET STRUCTURE

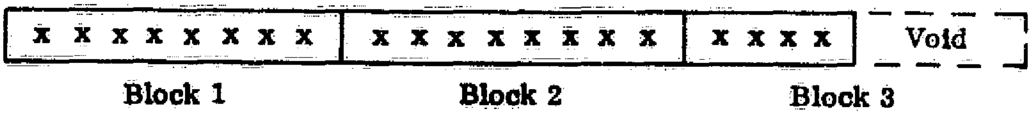
There are several standard forms of data set structure that are used, in various applications, by almost all SPAR programs. To avoid repetition in explaining these forms, standardized terminology for them is defined in this section. There are four such data-set forms, which are designated as TABLE, SYSVEC, ELDATA, and ALPHA.

In the following discussion, NWORDS, NI, and NJ have the same meaning as indicated in the explanation of library tables of contents (TOC's) appearing in Table 2.2-2; that is, any data set may be interpreted as a sequence of two-dimensional matrices, each dimensioned (NI, NJ). The physical record or "block" length is always NI times NJ. The total number of words in the data set is NWORDS. In some cases, NWORDS is an integral multiple of the block length; in other cases, it is not. The following cases are representative:

- Example. NI = 2, NJ = 3, NWORDS = 24



- Example. NI = 2, NJ = 4, NWORDS = 20



It is a uniform convention that the matrix elements contained in each block are ordered by column; that is, where x_{ij} is the element in row i, column j, the sequence within the block is $(x_{11}, x_{21}, x_{31}$ --- $x_{NI, 1}, x_{12}, x_{22}, x_{32},$ --- $x_{NI, 2},$ etc.). In the first of the preceding examples, the order of elements within each block would be $x_{11}, x_{21}, x_{12}, x_{22}, x_{13}, x_{23}$.

2.5.1 TABLE

The TABLE form of data set has the following attributes:

- All data are real (floating-point).
- NWORDS is an integral multiple of the block length, NI*NJ; that is, the number of blocks contained in the data set is exactly equal to NWORDS/NI*NJ.

The following are examples of data sets in TABLE form:

- Example. Node point position coordinate data produced by TAB subprocessor JLOC appear in a single-block data set for which NI equals 3 and NJ equals the total number of node points. Element x_{ij} in this (3, NJ) array is the i-direction position coordinate of node point j.
- Example. Temperatures at node points are defined by multiblock data sets for which NI equals 1 and NJ equals the number of node points. Element x_{1j} in the (1, NJ) array is the temperature of node j. Each separate block of the data set is a separate temperature case.

2.5.2 SYSVEC

SYSVEC is a special case of the TABLE form. SYSVEC is used primarily to represent either (1) displacements and rotations of all joints in the structure, or (2) forces and moments acting on all joints. The SYSVEC form is used to represent static deformation, reactions, vibrational and buckling eigenvectors, mechanical loading applied directly to joints, and various forms of equivalent joint loadings. This form is also used for diagonal mass matrices.

In the following discussion, it will initially be assumed that no joint motion components have been identically excluded for all joints via the START command (see Section 3.) In this case, NI is 6 and NJ is equal to the total number of joints in the structure. Each block of length NI*NJ corresponds to a separate case, e.g., a distinct load case, eigenvector, etc.

The six elements in the j-th column of a given block are either

$$u_{1j} \quad u_{2j} \quad u_{3j} \quad r_{1j} \quad r_{2j} \quad r_{3j} \text{ , or}$$

$$f_{1j} \quad f_{2j} \quad f_{3j} \quad m_{1j} \quad m_{2j} \quad m_{3j} \text{ ,}$$

depending on whether the data consists of (1) joint motions, or (2) joint forces and moments. In the above, u_{ij}, r_{ij}, f_{ij}, and m_{ij} are, in order, direction-i displacement, rotation*, force, and moment* components at joint j. The components are always relative to the joint reference frame uniquely associated with joint j, which may or may not be parallel to the global frame (see the JREF discussion in Section 3.1.6).

*Right-hand rule for rotations and moments.

If certain joint motion components are identically excluded via the START command, the number of elements in each column will be reduced accordingly. For example, if the direction-3 displacement and rotations about axes 1 and 2 have been excluded (plane frame analysis), NI would be 3 and column j would contain either:

$$u_{1j}, u_{2j}, r_{3j} \quad \text{or}$$

$$f_{1j}, f_{2j}, m_{3j} .$$

2.5.3 ELDATA

ELDATA is a data form used to represent certain categories of data bearing a one-to-one relationship with structural elements of a given kind, e.g. element temperature, or element pressures. Although a data set in ELDATA form often consists of many blocks, it will initially be assumed, for simplicity of explanation, that the data set consists of a single block. In this case, the number of columns, NJ, would be equal to the total number of structural elements of a particular kind.

For example, if the data set contains data associated with type E21 (general beam) elements, and the structure contains 70 E21 elements, the data set will contain 70 columns. NI, the number of words in each column, will depend upon the specific kind of data involved. If the 70 elements appear in three groups (say, 20 in group 1, 40 in group 2, and 10 in group 3), then the first 20 columns of an associated ELDATA-form data set would correspond to the 20 elements of group 1 (in sequential order, by structural element index number), the next 40 to the 40 in group 2, and the last 10 to the 10 in group 3.

The block length, which is always an integral multiple of NI (the number of data items per column), does not often exceed a few thousand words, and is sometimes less than one thousand. Small block lengths are used to minimize core storage requirements. If the selected block length is not sufficient to accommodate the data for all of a given kind of structural element, additional blocks are appended. In such cases, NJ is interpreted as the number of columns of structural element data per block, and NWORDS/NI is the number of associated structural elements.

2.5.4 ALPHA

The ALPHA form of data set is used to store lines of alphanumeric text. Each line (column) contains 60 characters. Each block contains one such character string. One use of the ALPHA form is to store static load case titles.

2.6 ERROR MESSAGES

Various error messages, usually self-explanatory, may originate in SPAR processors. Messages indicating insufficient core space refer to data space requirements, not to total (instruction + all data) space. At the beginning of execution of each processor, a message, DATA SPACE = XXXXX, indicates the amount of working core space currently available. This information enables the analyst to estimate the total field length needed in the rerun.

Another common message is ERROR IN CONTROL STATEMENT SYNTAX. This means that an illegal RESET name has been given, or that the wrong type of data (integer, floating-point, typeless) was given.

Another common error message is

```
LIB READ ERROR.  NU,L,KORE,IERR=  XX  XX  XX  XX
NAME=  N1  N2  n3  n4
```

Symbols in this message are defined as follows:

NU = SPAR logical file number.

L = Data set block length (if the data set exists).

KORE = Core space available for loading the data set.

IERR = Error code (see Section 2.2). If IERR = 2, core space is not sufficient to load a single block of the data set.

NAME = Data set name. The last two words of the data set name are always printed as integers. Accordingly, if they are MASKed, they will not be readable (e.g., on UNIVAC systems, they will be displayed as ******).

Many fatal error messages appear in the following form:

```
FATAL ERROR.  NERR, NIND= XXXX, xxxx
```

Other explanatory information usually precedes messages of this form. The following is a summary of error messages involving communication with the data complex.

NERR, NIND	Meaning
XLIB , n	Attempt is made to refer to non-existent SPAR logical file n.
ASG , n	Attempt is made to refer to non-existent SPAR logical file n.
NIND , ℓ	The user has supplied a file not in SPAR direct-access library format to be used as a library. The parameter ℓ is used in diagnosis.
RIND , n	The maximum number of data sets allowed for library n is reached. The normal limit for SPAR Level 9 is 2048 data sets per library.

Other error messages involving data complex communication appear in the following form:

```
** RIO ERR.  NU,IWR,IOP,KSHFT,L,ISTAT= ----
    INDEX= ----
```

These parameters have the following meanings:

NU	=	SPAR logical file number (1 to 26).
IWR	=	1, if writing on mass data storage; 2, if reading.
IOP	=	Addressing mode indicator.
KSHFT	=	Addressing parameter.
L	=	Number of words in the requested transmission.
ISTAT	=	I/O status code.
INDEX	=	Parameters used in diagnosis.

Section 3

STRUCTURE DEFINITION

Four programs – TAB, ELD, E, and EKS – are used to generate and store in the data complex data sets that define the finite element model of the structure.

TAB and ELD are used to generate the basic definition of the structure. Subprocessors within TAB translate card input data into tables of joint locations, material constants, section properties, etc. Subprocessors within ELD translate card input data into data tables that define individual finite elements of various types.

Using the data produced in TAB and ELD, programs E and EKS generate an array of data sets, collectively known as the E-state, that contain a complete description of every element in the structure, including details of element geometry, intrinsic stiffness matrices, etc.

Except where specifically noted to the contrary, all data sets produced by TAB, ELD, E, and EKS should be retained in library 1.

3.1 TAB — BASIC TABLE INPUT

Function. TAB contains an array of sub-processors which generate tables of material constants, section properties, joint locations, etc., and various other data sets comprising a substantial portion of the definition of the structure. Each of these sub-processors is identified either by a multi-word (long form) name, such as JOINT LOCATIONS, or by a short name, e.g., JLOC. Each sub-processor generates a data set having the same name as the sub-processor. Sub-processor names and functions are summarized in Table TAB-1.

TAB may be used to either (1) create new data sets, or (2) update existing data sets by replacing individual entries in them. The update mode is commonly used in Demand (teletype) interactive operation. Only one direct access library, SPAR logical 1, is usable in conjunction with TAB. When beginning a new problem, the first data card following @XQT TAB (and any RESET commands) must be

START j, m_1, m_2, - - -

In the above, j = the total number of joints in the structure. It is not harmful to have some unused joints (i.e. joints connected to no elements), for convenience in interpreting the output. This should not be carried to extremes, however, since it wastes core storage. It should be noted that it is not necessary to set aside large blocks of unused joint numbers in areas where you expect to later augment the model. The JSEQ sub-processor and an array of data modifier statements allow models to be extended without extensive repunching of existing data cards.

The parameters m_1, m_2, - - - on the START card list joint motion components which are identically zero for all joints. Those components are relative to the joint reference frames. For m=1, 2, or 3 a direction m displacement is indicated. For m=4, 5, or 6, a direction m-3 rotation is indicated.

For example, a 1000 joint space truss would begin with

```
START    1000,    4,5,6
```

The above indicates that all three rotation components are zero at all joints. As another example, a 500 joint plane frame would begin with

```
START  500   3, 4, 5
```

The above assumes that the the direction-3 displacements are normal to the plane of the frame, and the frame deforms in-plane only.

SPAR uses the m's given on the START card to determine submatrix storage block size (see Ref.1, Sec. 4). Therefore the START card (rather than the CON sub-processor) should be used to suppress joint motions in cases such as those illustrated above.

Following the START card, or directly following @XQT TAB if Library 1 already exists, the TAB card input stream is as illustrated below.

Proc_1

-

-

data cards read by Proc_1

-

-

Proc_2

-

-

data cards read by Proc_2

-

-

-

-

-

etc.

In the above, Proc_i represents a data card containing either the short or long-form name of a sub-processor to be executed. Sub-processors may be executed in any order, subject to the following restriction: if the data read by one sub-processor refers to another TAB-generated table, the other table must already exist. It is always safe to execute sub-processors in the order they appear in Table TAB-1.

TAB input card rules

In addition to the rules generally applying to SPAR free-field input, the following rules apply specifically to TAB.

(1) Trailing items omitted on data cards are assumed to be zero, except when indicated otherwise in discussions of specific sub-processors.

(2) Input should be in real (fixed on floating point) format, except for integer items such as entry numbers, joint numbers, table entry pointers, and control variables.

(3) Within the data card stream being read by any sub-processor, the following commands may be injected.

FORMAT = j

MOD= m

NREF = n

In the above,

- j identifies which of several admissible formats is to apply to subsequent cards. Details are given in discussions of individual sub-processors.

- m is added to the table entry indicator (usually called k), in all processors <u>other than</u> TEXT, JLOC, RMASS, JREF, CON and JSEQ. In JLOC, RMASS, JREF, and CON, m is added to joint numbers indicated on subsequent input cards.

- In processors JLOC, JREF, and MREF, reference frame n applies to data on subsequent cards.

These three commands may be used repeatedly in the card streamof any sub-processor. Upon beginning execution of a new sub-processor, these parameters are internally reset to their default values, FORMAT=1, MOD=0, and NREF=1.

Correcting existing data sets

To enter the "update" mode of operation, the following command is injected in the input stream.

```
UPDATE=1
```

To leave the update mode, the command is

```
UPDATE=0
```

UPDATE commands should immediately precede sub-processor execution commands. When operating in the update mode, the output data set produced in the current execution is identical to that produced in the preceding execution, except for entries defined by the card input of the current execution. The only sub-processors which cannot be operated in the update mode are TEXT and JSEQ.

As an example, suppose the location of joint 1742 is found to be in error. The JLOC data set could be repaired by the following card sequence.

```
UPDATE=1
JLOC
1742, 947.62, 1841.9 23.487$
```

Library Title

The following statement will cause an identifying title to be embedded in library 1:

TITLE' 60-character alphanumeric title

The title, which resides in a data set named NDAL---, is displayed at the beginning of printouts of library Tables of Contents (e.g., via DCU/TOC).

RESET Controls

No special RESET controls are available in TAB.

Core Requirements

Working core requirements will not significantly exceed the larger of (1) 13 times the number of joints, or (2) the longest table generated (see Table TAB-1).

Code Release Data

Level 9 (UNIVAC, CDC) July 1975, coded by W. D. Whetstone.

Table TAB-1: TAB Sub-Processors

Short Name	Table Length*	Long Form Name, Function
TEXT	15 n	TEXT. Creates a data set containing alphanumeric text documenting the analysis.
MATC	10 n	MATERIAL CONSTANTS. Entries define linear material constants E, ν, etc.
NSW	n	DISTRIBUTED WEIGHT. Entries define nonstructural distributed weight parameters (weight/length or weight/area).
ALTREF	12 n	ALTERNATE REFERENCE FRAMES. Entries describe reference frames selected by the analyst for convenience in defining joint locations, etc.
JLOC	3 j	JOINT LOCATIONS.
JREF	j	JOINT REFERENCE FRAMES. Entries define the orientation of reference frames associated with joints, used in defining constraint, applied loading, etc.
MREF	5 n	BEAM ORIENTATION. Entries define beam cross-section orientation.
BRL	8 n	BEAM RIGID LINKS. Entries define rigid links offsetting the end points of elastic 2-node elements from the joints they connect.
BA	31 n	E21 SECTION PROPERTIES.
BB	21 n	BEAM S6x6. Entries define elastic characteristics of type E22 and E25 elements.
BC	6 n	E23 SECTION PROPERTIES.
BD	6 n	E24 SECTION PROPERTIES.
SA	19 n	SHELL SECTION PROPERTIES.
SB	4 n	PANEL SECTION PROPERTIES.
CON	j, 6j	CONSTRAINT DEFINITION.
JSEQ	j	JOINT ELIMINATION SEQUENCE.
RMASS	6j	RIGID MASSES.

*n is the number of entries in the table, and j is the total number of joints in the structure.

TAB/
TEXT
MATC
NSW

3.1.1 TEXT

The function of the TEXT sub-processor is to allow the analyst to embed in the output library a data set containing alphanumeric text descriptive of the analysis being performed. Each card has a 4/8 punch in column one, followed by a 60 character string of text.

3.1.2 MATERIAL CONSTANTS (MATC)

MATC generates a table of linear material constants. Each input card defines one entry in the table. The data sequence on the input card defining the k-th entry in the table is k, E, Nu, Rho, Alpha_1, Alpha_2 ; where

E	=	modulus of elasticity,
Nu	=	Poisson's ratio,
Rho	=	Weight per unit volume, and
Alpha_1, Alpha_2	=	Thermal expansion coefficients effective in directions 1 and 2, relative to element reference frames, for 3- and 4-node elements. If Alpha_2 is omitted, the program sets $\text{Alpha}_2 = \text{Alpha}_1$. Zero or negative values of both Alpha's are permitted. For 2-node elements, the thermal expansion coefficient is Alpha_1.

Material constants associated with specific elements are defined in processor ELD by pointing to entries in the MATC table.

3.1.3 DISTRIBUTED WEIGHT (NSW)

A table of non-structural distributed weight parameters is defined. The data sequence of the input card defining the k-th entry in the table is

k, W.

For 2-node elements, W is weight/length.

For 3 and 4 node elements, W is weight/area.

Non-structural weight attached to specific elements is defined in processor ELD, by pointing to entries in the NSW table.

3.1.4 ALTERNATE REFERENCE FRAMES (ALTREF)

In addition to the global reference frame, the analyst may find it convenient to define additional reference frames. These frames have several uses, including the following:

(1) Joint locations may be defined in any frame the analyst finds most convenient (see JLOC).

(2) Joint reference frame orientations may be defined via the alternate frames (see JREF).

Each frame is uniquely identified by a positive integer. The global frame is always frame 1; accordingly the analyst is free to define only frames 2, 3, - - -. The order of data on the input card defining frame k is

$$k,\ i_1, a_1,\ \ i_2, a_2,\ \ i_3, a_3,\ \ x_1,\ x_2,\ x_3.$$

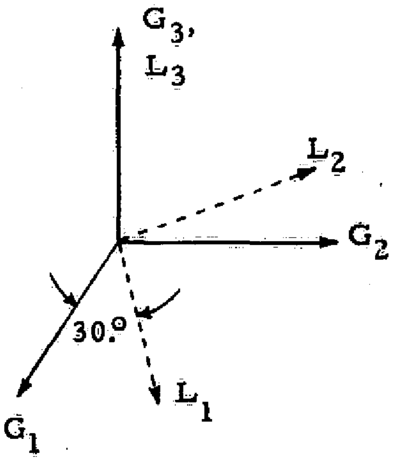
In the above x_1, x_2, and x_3 are position coordinates, relative to the global frame, of the origin of frame k. The x's need not be given if only the orientation of frame k is of significance, which often is the case.

The parameters i_1, a_1, - - - i_3, a_3 indicate ordered rotations defining axis orientations. Two formats are provided. In both cases, we begin with a local frame parallel to the global frame. After the ordered rotations are completed, the local frame is parallel to frame k. Each of the i's may be 1, 2, or 3, in any order. The a's are angles in degrees.

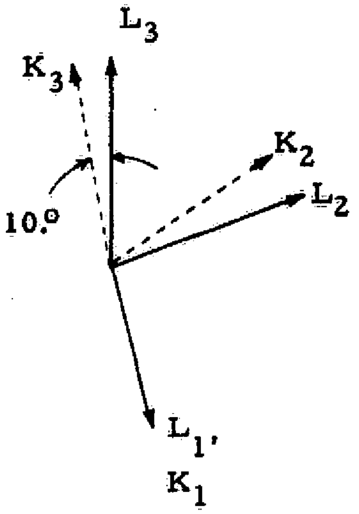
If FORMAT=1(default), the sequence is: (1) rotate the local frame a_1 degrees about local axis i_1, then (2) from the new position, rotate the local frame a_2 degrees about axis i_2, then (3) from the resulting position, rotate the local frame a_3 degrees about axis i_3. If FORMAT=2, the i's and a's indicate rotation of the global frame relative to frame k.

In the following example of an input card defining frame 27, it is assumed that FORMAT=1.

```
27        3,30.      1,10.$
```



First rotation,
G_i = Global axes



K_i = axes of Frame 27.

3.1.5 JOINT LOCATIONS (JLOC)

JLOC produces a table containing the position coordinates of the joints. The data sequence on input cards is as follows:

$k,\ xa_1,\ xa_2,\ xa_3,\ xb_1,\ xb_2,\ xb_3,\ ni,\ ijump,\ nj$

If nj is given, a second card must appear,

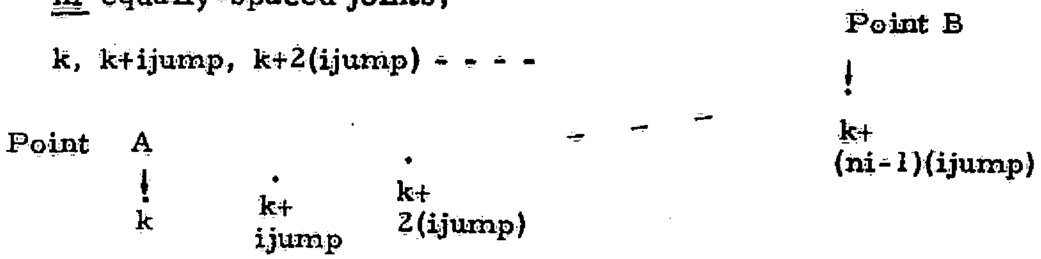
$jjump,\ xc_1,\ xc_2,\ xc_3,\ xd_1,\ xd_2,\ xd_3$

There are three possible interpretations of the above:

(1) If only k, xa_1, xa_2 and xa_3 are given, the xa's are coordinates of joint k.

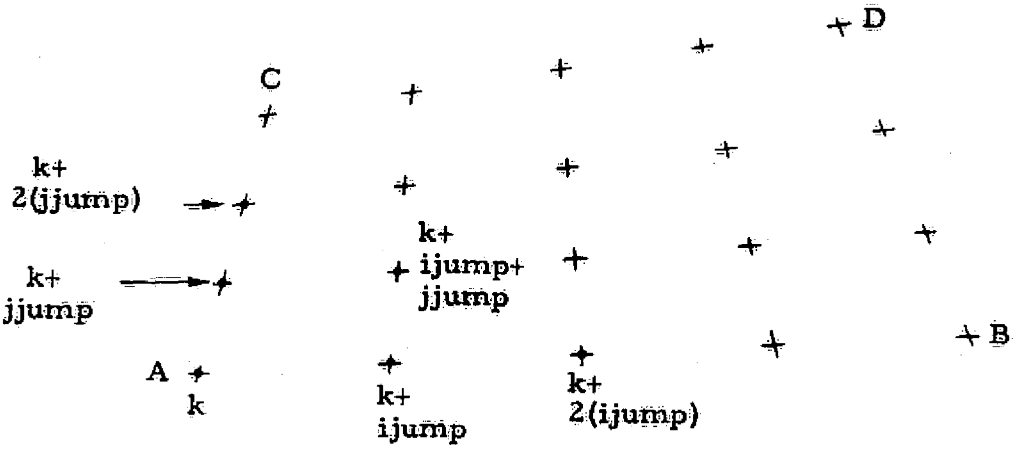
(2) If k, xa_1, - - - - xb_3, ni, and ijump are given, the xa's and xb's are coordinates of points A and B terminating a string of ni equally-spaced joints,

k, k+ijump, k+2(ijump) - - - -



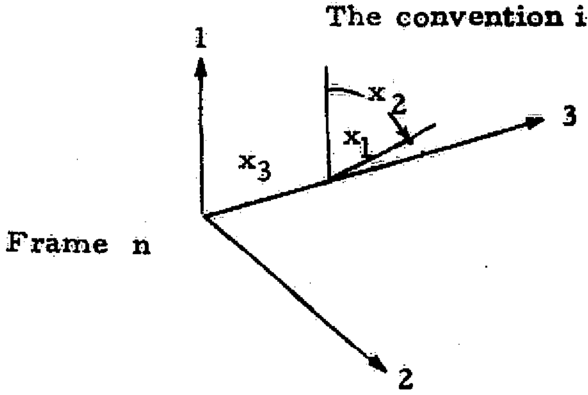
The default value of ijump is 1.

(3) If nj is given, a linearly interpolated two-dimensional mesh of (ni)(nj) joints is defined, as indicated below.



Although the output table generated by JLOC is in rectangular coordinates relative to the global frame, coordinate data appearing on the input cards may be either rectangular or cylindrical, and may be relative to any frame already defined in ALTREF. The associated control cards are summarized below.

Control	Meaning
NREF= n	Coordinate data on subsequent cards is relative to frame n (until another NREF command is encountered).
FORMAT=2	Subsequent data is in cylindrical coordinates, relative either to frame 1 (global) or to any other frame selected by an NREF command. The convention is shown below.
FORMAT=1	Switch back to rectangular coordinates.



x_1 = radial distance from 3-axis.

x_2 = angle, <u>in degrees</u>, from the <u>1-3 plane</u>.

x_3 = linear distance from the 1-2 plane.

Switching among frames, and between rectangular and cylindrical coordinates is unrestricted.

If cylindrical coordinates are used in connection with mesh generation, interpolation is performed before transformation to rectangular coordinates; so that regular meshes on circles, cylinders and cones are readily generated.

3.1.6 JOINT REFERENCE FRAMES (JREF)

A unique right-hand rectangular reference frame is associated with each joint. Through JREF the analyst may designate the orientation of any joint reference frames. All joint reference frames not defined in JREF are, by default, parallel to frame 1 (global). The orientation of these frames is of considerable significance, since joint motion components and mechanical loads applied at joints are defined relative to them (see discussions of TAB sub-processor CON, and SYSVEC format).

To identify a set of joints with reference frames parallel to frame n (see ALTREF), an input card containing NREF=n is followed by a sequence of records in 'loop limit' format naming the joints. 'Loop limit' format is: j_1, j_2, inc, meaning joints j_1, j_1+inc, j_1+2(inc), - - - j_2. Default for j_2 is j_1. Default for inc is 1.

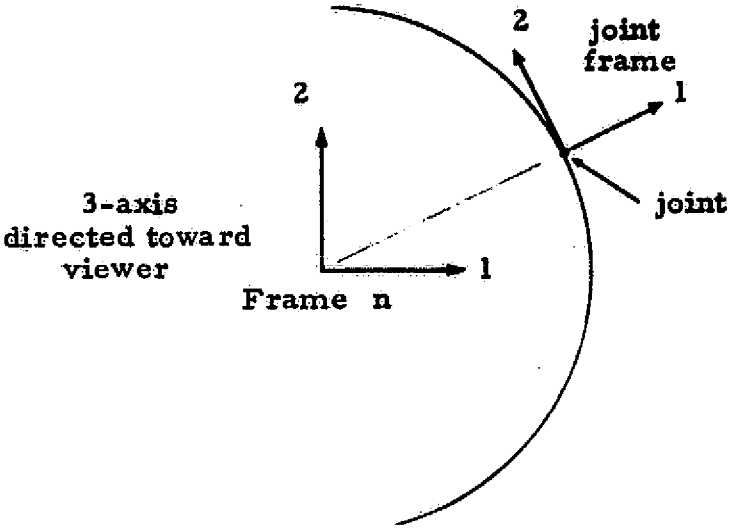
For example, the following records indicate that the reference frames of joints 5, 6, 7, 8, 19, 30, 32, 34, and 36 are parallel to frame 10.

NREF=10: 5, 8: 19: 30, 36, 2$

The language NREF=-n indicates that the joint reference frame is oriented as follows:

> The 3-axis of the joint frame is parallel to the 3-axis of frame n

- The 1 axis of the joint frame is perpendicular to the 3-axis of frame-n, as shown below:*



3.1.7 BEAM ORIENTATION (MREF)

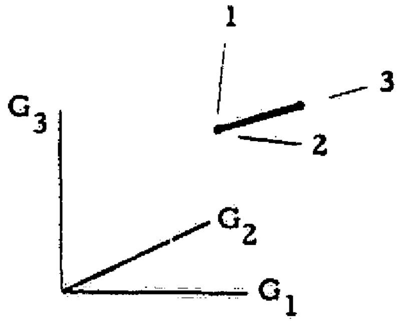
Each two-node element has an "element reference frame" associated with it (see processor ELD discussion). Beam section properties, stresses, etc., are defined relative to these frames. The 3-axis of the frame is directed from the beam's origin to its terminus. The origin and terminus are the ends connected to joints J1 and J2, respectively, as defined in processor ELD. The beam end points coincide with J1 and J2, unless offset by rigid links defined via the BRL sub-processor.

Entries in the table produced by MREF are used to define the orientation of beam axes 1 and 2. Usually a single entry in this table will apply to many beams. Either of two data sequences may be selected through FORMAT control, as indicated below.

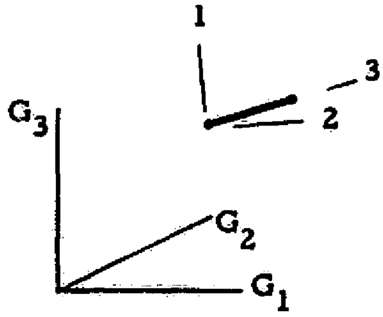
If FORMAT=1 (default), the data sequence defining table entry k is

k, nb, ng, isign, c

The above indicates that c is the cosine of the (smallest) angle between beam axis nb and global axis ng. Legal values of nb are 1 and 2, and ng may be 1,2,or3. The parameter isign resolves a possible ambiguity by indicating whether the cosine of the angle between beam axis (3-nb) and global axis ng is positive or negative, with values of +1 and -1 indicating positive and negative, respectively. Examples are shown below.



nb, ng, isign, and c are
1, 3, -1, .1.



nb, ng, isign, and c are
1, 3, +1, .1.

In both of the above cases the cosine of the angle between beam axis 1 and global axis 3 is .1, and isign defines the sign of the cosine of the angle between beam axis 2 and global axis 3.

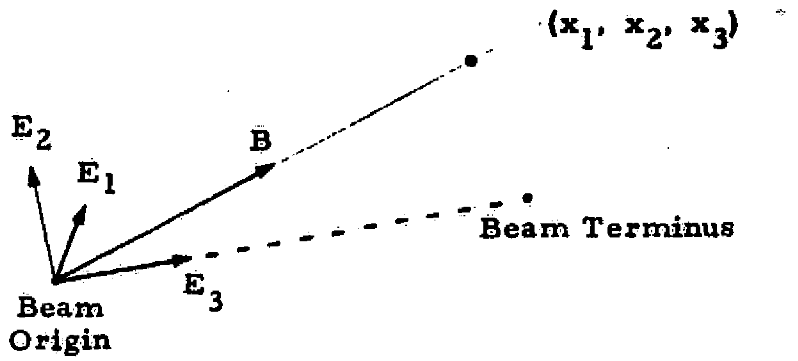
In about 95% of the cases c is either 1.0 or .0.

Care should be taken to supply meaningful information. For example, if beam axis 3 is parallel to global axis 2, it is useless to state that the cosine of the angle between global axis 2 and beam axis 1 (or 2) is zero.

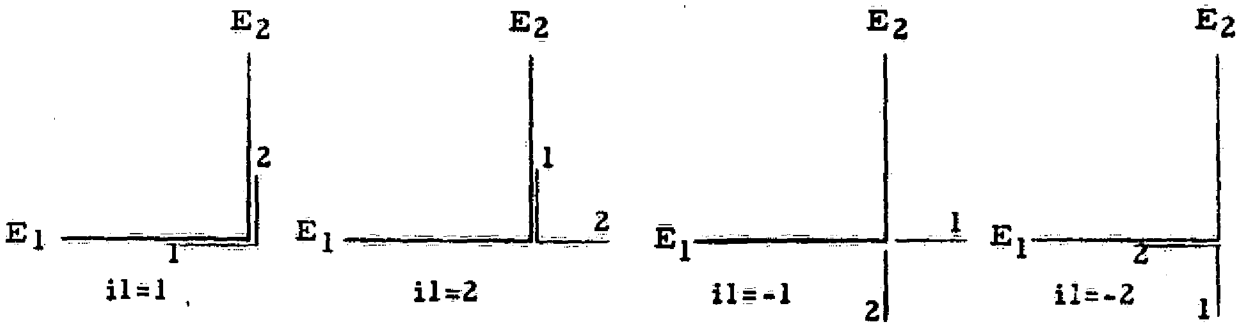
In FORMAT=2, the data sequence defining table entry k is

k, il, x_1, x_2, x_3.

The above indicates that beam axis orientation is determined by the location of a 'third point', located at (x_1, x_2, x_3), as illustrated below. The x's are in rectangular components, relative to any frame defined in ALTREF. The NREF=n command may be used to switch to frame n (default is frame 1), anywhere in the input stream.



In the above, E_1, E_3 and B are coplanar, and E_2 is parallel to $E_3 \times B$. The input parameter il selects one of four possible beam axis orientations, as shown below.



The analyst will find both formats 1 and 2 useful. For example, if rings and stringers reinforce a cylindrical or conical shell, it may be possible to define the orientation of all of the ring elements with a single format-1 MREF table entry, and to define the orientation of all of the stringer elements with one format-2 entry.

3.1.8 BEAM RIGID LINKS (BRL)

Each entry in the table generated by BRL defines rigid link offsets for both the origin and terminus of a 2-node element. The data sequence on the input card defining the k-th entry in the table is

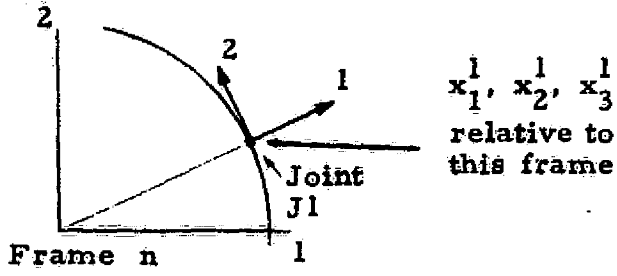
$$k,\ i^1,\ x_1^1,\ x_2^1,\ x_3^1, \qquad i^2,\ x_1^2,\ x_2^2,\ x_3^2$$

The above data is defined as follows.

- x_1^1, x_2^1, x_3^1 are the position coordinates of joint J1 (see ELD discussion) minus the position coordinates of the beam origin.

- x_1^2, x_2^2, x_3^2 are the position coordinates of joint J2 minus the position coordinates of the beam terminus.

The x's are always given in rectangular coordinates relative to reference frames defined by the parameters i^1 and i^2. If i^1=n (positive), it indicates that x_1^1, x_2^1, x_3^1 are relative to frame n (see ALTREF). If i^1=-n the reference frame of x_1^1, x_2^1, x_3^1 is shown below.



Interpretation of x_1^2, x_2^2, x_3^2 is controlled similarly by i^2.

3.1.9 E21 SECTION PROPERTIES (BA)

BA generates a table of section properties to which reference is made during definition of type E21 elements in processor ELD. Nine classes of cross section are allowed.

The first word in each input record is a typeless word (e.g. BOX, TEE, etc.) identifying the class. A single card defines a table entry, except for DSY which requires two cards. Data sequences are as indicated below.

BOX	k, b_1, t_1, b_2, t_2
TEE	k, b_1, t_1, b_2, t_2
ANG	k, b_1, t_1, b_2, t_2
WFL	k, b_1, t_1, b_2, t_2, b_3, t_3
CHN	k, b_1, t_1, b_2, t_2, b_3, t_3
ZEE	k, b_1, t_1, b_2, t_2, b_3, t_3
TUBE	k, inner radius, outer radius
GIVN	k, I_1, α_1, I_2, α_2, a, f, f_1, z_1, z_2, θ
DSY	k, I_1, α_1, I_2, α_2, a, f, f_1 (card 1)
	q_1, q_2, q_3, y_{11}, y_{12}, - - - y_{41}, y_{42} (card 2)

In the above, k identifies the table entry number. The b's and t's are cross-section dimensions defined on Figure BA-1.

In all cases the origin and terminus of the beam (see discussions of MREF, BRL, and ELD) coincide with the section centroid. For GIVN and DSY sections,

I_1, I_2 Principal moments of inertia. For DSY sections, principal axes must coincide with the element reference frame axes.

α_1, α_2 Transverse shear deflection constants associated with I_1 and I_2, respectively. For no shear deflection, set α_i equal to zero.

a = cross-sectional area.

f = Uniform torsion constant.* For uniform torsion, torque = Gf x (twist angle/unit length), where G is the shear modulus.

f_1= Nonuniform torsion constant,* accounting for flange-bending effects on torsional stiffness, etc.

z_1, z_2 Shear center - centroid offsets.

θ Inclination of principal axes relative to the element reference frame (see Figure BA-1). θ is in radians.

Use of the above quantities in calculating element stiffness matrices is discussed in Appendix B. Items given on the second card defining DSY sections have the following meaning:

q_1, q_2 Section shape factor such that maximum shear stress due to V_1, a shear in direction 1, is $= V_1\, q_1$. q_2 is similarly defined.

q_3 Section shape factor such that maximum stress due to twisting moment T is Tq_3.

y_{i1}, y_{i2} = Location, relative to the element reference frame, of the i-th point at which My/I combined bending stresses are to be computed. Up to 4 such points may be prescribed.

* In the notation of Appendix B,

$C = Gf$, and

$C_\Gamma = Ef_1$

Circled numbers identify locations where bending stresses are evaluated

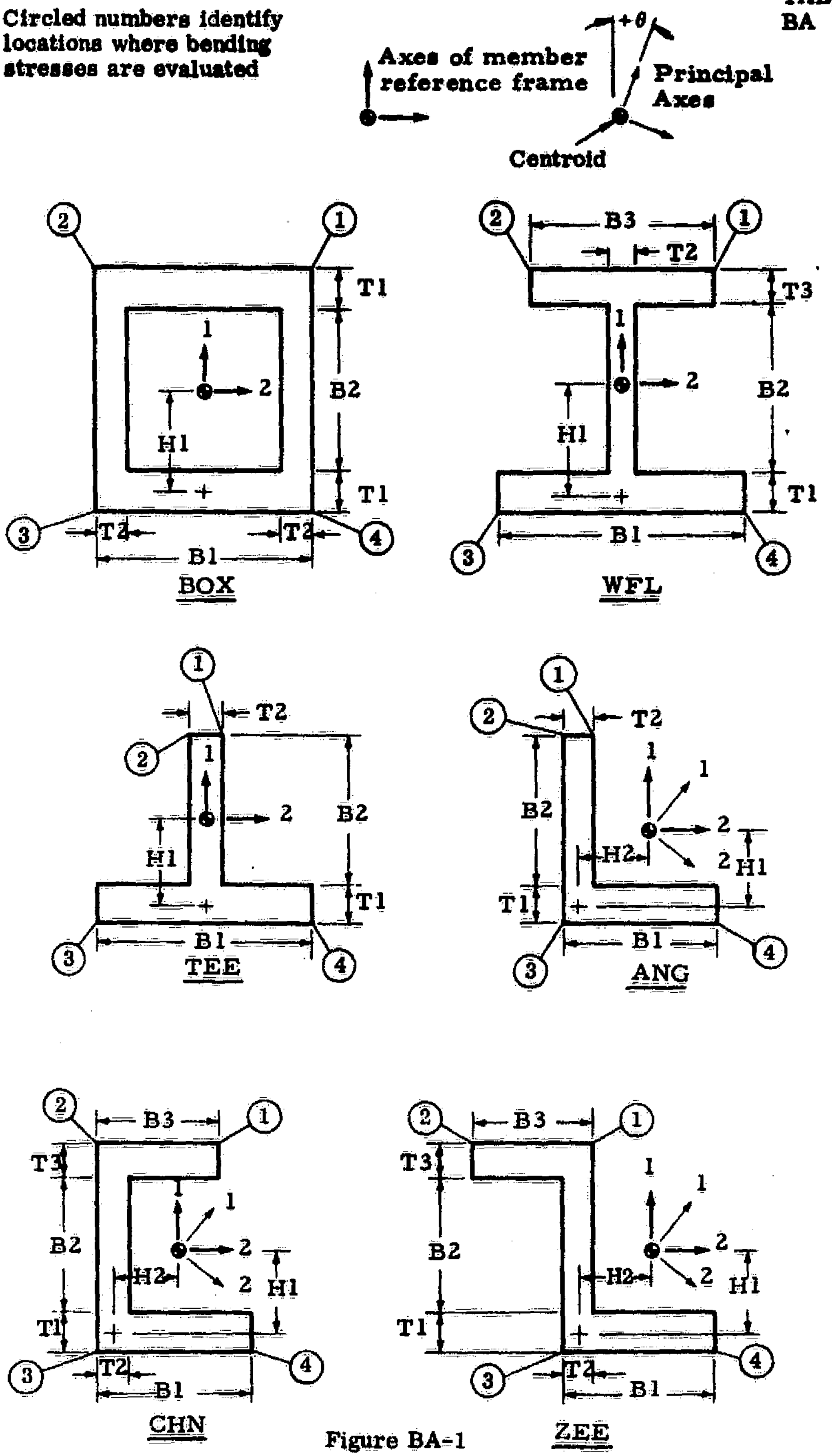


Figure BA-1

3.1.10 BEAM S 6 x 6 (BB)

BB generates a table of directly specified 6 by 6 intrinsic stiffness matrices to which reference is made during definition of type E22 and E25 elements in processor ELD. Six cards, as indicated below, are required to define one entry in the table.

k, s_{11}

s_{21}, s_{22}

s_{31}, s_{32}, s_{33}

s_{41}, s_{42} s_{43}, s_{44}

s_{51}, s_{52}, s_{53}, s_{54}, s_{55}

s_{61}, s_{62}, s_{63}, s_{64}, s_{65}, s_{66}

In the above, k identifies the table entry, and the s_{ij}'s are elements of a symmetric intrinsic stiffness matrix, S, which is defined as follows: if the terminus (see ELD, MREF, BRL discussions) of the element is completely fixed,

$$F = SU,$$

where $F = (f_1 \;\; f_2 \;\; f_3 \;\; m_1 \;\; m_2 \;\; m_3)^t$, and

$U = (u_1 \;\; u_2 \;\; u_3 \;\; r_1 \;\; r_2 \;\; r_3)^t$.

In the above, f_i, m_i, u_i, r_i = applied force, applied moment, displacement, and rotation components at the origin. All components are relative to the element reference frame.

3.1.11 E23 SECTION PROPERTIES (BC)

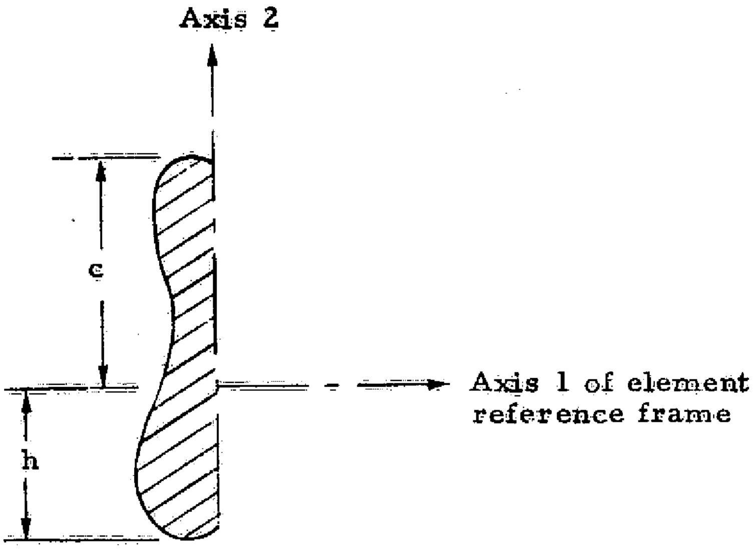
BC generates a table of axial element section constants to which reference is made during definition of type E23 elements in processor ELD. The data sequence on the input card defining the k-th entry in the table is k, a where a equals the cross-sectional area.

3.1.12 E24 SECTION PROPERTIES (BD)

BD generates a table of plane beam element section properties to which reference is made during definition of E24 elements in processor ELD. The data sequence on the input card defining the k-th entry in the table is

$$k, \; A, \; I_1, \; \alpha_1, \; h, \; c, \; q_1.$$

In the above, I_1, α_1 and q_1 are the same as for DSY sections, as defined in BA, the E21 section property sub-processor. Dimensions h and c are as shown below. A = cross sectional area.



3.1.13 SHELL SECTION PROPERTIES (SA)

SA generates a table of shell section properties to which reference is made during definition of element types E31, E32, E33, E41, E42, and E43, in processor ELD. Two formats are available to define table entries. Material constant table (MATC) entries associated with individual elements are defined in ELD via the NMAT parameter. Applicable entries in the material constant table must also be identified during execution of SA, using the same type of command, NMAT=entry number (default=1).

Two formats are available to define entries in the SA table. If FORMAT=1 (default), an isotropic section is indicated, and the data sequence on the single input card defining the k-th entry in the table is

k, thickness.

If FORMAT=2, an aeolotropic section is indicated. Three cards, containing the data sequences indicated below, are required to define the k-th entry.

$$k, T_{11}, T_{22}, T_{12}, F_1^a, F_2^a, F_1^b, F_2^b \$$$

$$c_{11}, c_{12}, c_{22}, c_{13}, c_{23}, c_{33} \$$$

$$d_{11}, d_{12}, d_{22}, d_{13}, d_{23}, d_{33} \$$$

The trailing items on the first record are stress coefficients, defined as follows:

Membrane stresses	Bending stresses, location a	Bending stresses, location b
$\sigma_x = N_{11}/T_{11}$	$\sigma_x = F_1^a M_{11}$	$\sigma_x = F_1^b M_{11}$
$\sigma_y = N_{22}/T_{22}$	$\sigma_y = F_2^a M_{22}$	$\sigma_y = F_2^b M_{22}$
$\tau_{xy} = N_{12}/T_{12}$		

The c's and d's are defined as follows. Where

N_{11}, N_{22}, and N_{12} are membrane stress resultants,

ϵ_{11}, ϵ_{22}, and γ_{12} are membrane strains,

M_{11}, M_{22}, and M_{12} are moment resultants, and

k_{11}, k_{22}, and k_{12} are

$$-\frac{\partial^2 w}{\partial x^2}, \quad -\frac{\partial^2 w}{\partial y^2} \quad \text{and} \quad 2\,\frac{\partial^2 w}{\partial x\,\partial y}, \quad \text{respectively,}$$

For Isotropic Sections

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \gamma_{12} \end{bmatrix}$$

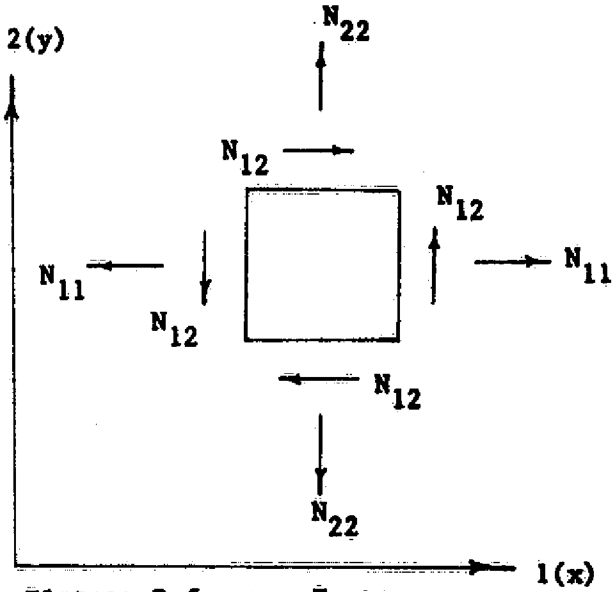
$c_{11}=c_{22}=Et/(1-\nu^2)$
$c_{33}=c_{11}(1-\nu)/2$
$c_{12}= c_{11}\nu$
$c_{13}=c_{23}=0$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} k_{11} \\ k_{22} \\ k_{12} \end{bmatrix}$$

$d_{11}=d_{22}=Et^3/12(1-\nu^2)$
$d_{33}=d_{11}(1-\nu)/2$
$d_{12}= d_{11}\nu$
$d_{13}=d_{23}=0$

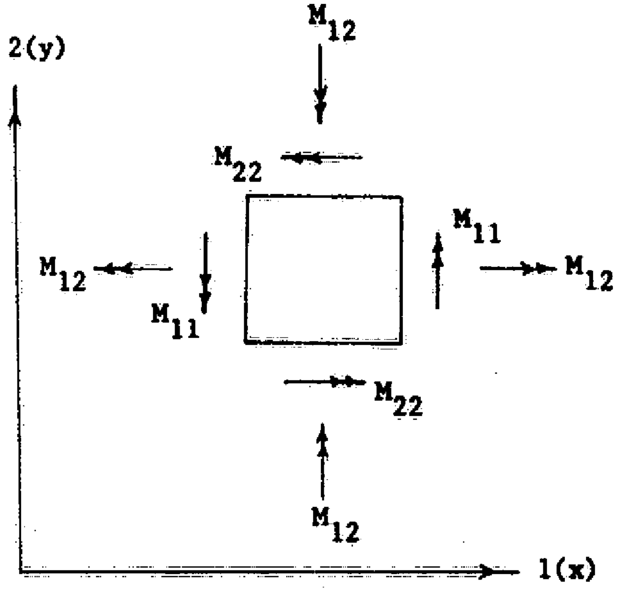
All of the above quantities are relative to the element reference frame (see ELD discussion). Stress resultant sign convention is shown on Fig. SA-1.

It should be noted that subprocessor SA computes c_{ij}'s and d_{ij}'s for isotropic sections, based on the material constant data currently resident in library 1. If material constants associated with isotropic sections are subsequently changed, SA should be re-executed.



Element Reference Frame

Membrane Stress Resultants Sign Convention



Element Reference Frame

Bending Stress Resultants Sign Convention

Figure SA-1

3.1.14 PANEL SECTION PROPERTIES (SB)

SB generates a table of shear panel section properties to which reference is made during definition of E44 elements in processor ELD. The data sequence on the input card defining the k-th entry is k,t .
where t = panel thickness.

3.1.15 CONSTRAINT DEFINITION (CON)

Any number of constraint cases may be defined, each uniquely identified by an integer assigned by the analyst. To define constraint case n, the CON sub-processor may be activated by any of the following commands.

```
CONSTRAINT DEFINITION  n
CONSTRAINT=  n
CON=n
```

Constraint is defined by specifying certain joint motion components to be either (1) unconditionally equal to zero, or (2) a specified value, possibly not zero. In the second category, specific values of such components may vary from one load case to another. (See Section 6.)

In the following, m=1, 2, or 3 indicates joint displacement in direction m , relative to a joint reference frame (see JREF). If m=4, 5, or 6, a direction m-3 rotation component is indicated.

To specify that joint motion components m_1, m_2, - - are unconditionally zero at certain joints, a command in the form

ZERO m_1, m_2, - - -:

is followed by a sequence of records in 'loop limit' format (see JREF) identifying the joints at which the indicated constraint is imposed. For example, to set the direction 3 displacement and direction 2 rotation equal to zero at joints 1, 2, 3, 17, 34, 36, and 38, the following language could be used.

```
ZERO 3, 5:  1, 3:  17:  34, 38, 2$
```

To declare joint motion components which will have prescribed nonzero values, a command in the form

NONZERO $m_1, m_2,$ - - - :

is followed by input cards in loop-limit format identifying the affected joints.

To release components erroneously constrained, the command

RELEASE $m_1, m_2,$ - - - :

is followed by cards identifying the affected joints, as above.

Additional commands, as summarized below, can be used to cause CON to automatically impose constraint to joints lying in global planes. Each joint in a symmetry or antisymmetry plane must have an axis of the joint reference frame (see JREF) perpendicular to the plane, although the other two axes need not be parallel to global axes.

Command		Meaning
SYMMETRY PLANE=	n	The plane normal to global axis n is a symmetry plane.
ANTISYMMETRY PLANE=	n	The plane normal to global axis n is an antisymmetry plane.
FIXED PLANE=	n	All joints in the plane normal to global axis n are completely fixed.
XZERO=	x	In processing the above commands, x is the linear tolerance used to determine if a joint is in a plane. Default x is .01.

Command		Meaning
RZERO=	r	In processing the above commands, joints found to be in a symmetry or antisymmetry plane must have an axis normal to the plane. The axis alignment tolerance is r (default value = .0001 radians).

3.1.16 JOINT ELIMINATION SEQUENCE (JSEQ)

As indicated in Reference 1 of Section 4, the sequence in which joints are eliminated during the factoring (reduction) sequence performed in processor INV significantly affects computer execution time and data storage requirements, both in factoring and subsequent execution of processors such as SSOL and EIG. We expect in the future to provide software which will substantially automate determination of joint elimination sequences. For the present, however, it is necessary for the user to perform one of the following.

(1) Carefully study Reference 1 of Section 4 to learn the factors governing choice of favorable joint numbering sequences. The rules are less restrictive and in many respects simpler than those associated with band matrix or wavefront procedures.

(2) Ask the advice of someone familiar with the sparse matrix solution procedure used in SPAR or in the SNAP program. For the infrequent user this is an excellent solution, since most experienced users can readily identify suitable sequences.

(3) Use a numbering sequence which would be appropriate for a band-matrix or wavefront procedure. This will forego the opportunity to use the full capability of SPAR's sparse matrix procedure, but it is the only alternative to the above. This will often give satisfactory costs for problems of small to moderate size.

Several methods of defining the joint elimination sequence are allowed. The simplest and least automatic is comprised of a sequence of input cards in string integer format, as illustrated below.

```
10/13, 8/5, 9$
4, 1/3, 14$
```

The above implies the following joint sequence:

10, 11, 12, 13, 8, 7, 6, 5, 9, 4, 1, 2, 3, 14.

For regular structures involving repeating patterns in the joint sequence, the command

```
REPEAT n, inc, jmod
```

may precede a packet of cards (terminated by another REPEAT card, or by end of data) defining a list of joints. The effect of the REPEAT card is to cause jmod to be added to each joint in the list. The list is repeated n times, with inc added to each joint number upon each repetition.

For example, suppose it is necessary to describe the following joint elimination sequence for a structure containing 140 joints:

111, 112, 113, - - - - - 120,
101, 102, 103, - - - - - 110,
131, 132, 133, - - - - - 140,
121, 122, 123, - - - - - 130,
100, 99, 98, - - - - - - - - 3, 2, 1.

The above is represented by the following JSEQ input.

```
REPEAT 2, 20, 100: 11/20, 1/10$

REPEAT 0, 0, 0: 100/1$
```

The default values of n, inc, and jmod are 1, 0, and 0, respectively.

3.1.17 RIGID MASSES (RMASS)

Two formats are available to define a rigid mass, as indicated below.

If FORMAT=1, the data sequence on each card is

k, M, I_1, I_2, I_3, where

k is the joint to which the mass is attached,

M = mass, and

I_i = mass moment of inertia about axis i of the joint reference frame. Note that it is required that the joint reference frame be coincident with the principal axes of the rigid mass. Rigid links and zero-length 2-node elements are often useful in modeling large rigid masses.

If FORMAT=2, the data sequence is

k, M_1, M_2, M_3, I_1, I_2, I_3, where the data has the same meaning as above, except that M_i is the mass effective in the direction of axis i of the joint reference frame.

If a data card in either of the above formats is preceded by the command

REPEAT n, j.

the effect of the REPEAT command is to cause the same mass to be defined at n joints, k, (k+j), (k+2j), - -. A REPEAT command applies only to the data card immediately following it.

To cause all subsequently defined masses (M, or M_1, M_2, and M_3) to be multiplied by p, and to cause all I's to be multiplied by q, a command is given in the following form.

CM p, q .

If successive CM commands are given, the new values of p and q replace the previous values.

The output data set generated by RMASS is, in effect, a diagonal mass matrix corresponding to the mass data read from the data cards defined above. Distributed mass associated with elements is not included in the RMASS output. However, if element distributed mass is negligible, RMASS output can be used as a system mass matrix by processors AUS and EIG.

3.2 ELD – ELEMENT DEFINITION PROCESSOR

Function - ELD translates element definition data from input cards into data sets which are readily usable by other SPAR processors. Elements may be defined singly or through a variety of network generators, or combinations thereof. An element is defined by specifying (1) the joints to which it is connected, and (2) pointers to applicable entries in tables of section properties, material constants, etc. The tables of section properties, etc., are usually generated by the TAB processor. As the ELD input is processed, checks are performed to detect errors such as references to nonexistent table entries, joint numbers, etc.; however, ELD does not extract any data from TAB generated tables. Accordingly, if TAB is subsequently re-executed to alter values of section properties, joint position coordinates, etc., it is not necessary to re-execute ELD, provided the connected joint numbers, table entry pointers, etc., of the elements are unchanged.

Alphanumeric names used to identify specific types of elements are summarized in Table ELD-1. Formulations on which these elements are based are discussed in Appendix B.

Table ELD-1: Summary of SPAR Elements

Name	Description
E21	General beam elements, such as angles, wide flanges, tees, zees, tubes, etc.
E22	Beams of finite length for which the 6 x 6 intrinsic stiffness matrix is directly specified.
E23	Bar elements having only axial stiffness.

Name	Description
E24	Plane beam.
E25	Zero-length element used to elastically connect two coincident joints.
E31	Triangular membrane, flat, aeolotropic, using TM constant-stress formulation.
E32	Triangular bending element, flat aeolotropic.
E33	Triangular membrane + bending element, flat, aeolotropic, using TM and TPB7 formulations.
E41	Quadrilateral membrane, flat, aeolotropic, using QMB5 hybrid formulation.
E42	Quadrilateral bending element, flat, aeolotropic, using QPB11 hybrid formulation.
E43	Quadrilateral membrane + bending element, flat, aeolotropic, using QMB5 and QPB11 formulations.
E44	Quadrilateral shear panel, flat, using QMB1 hybrid formulation.

For purposes of explaining card input, we will consider ELD to be comprised of an array of sub-processors, one for each type of element. The function of each sub-processor is to read input cards defining all of the elements of a particular type. A sub-processor is activated by an input card containing the name of the element type. Sub-processors may be called in any order. A typical input stream is shown below.

```
@XQT  ELD
  E43
  -
  -
  (Data cards defining all of the E43 elements in the structure)
  -
  -
  E21
  (Data cards defining all of the E21  elements in the structure)
  -
  -
  -
```

Execution of each sub-processor results in production of data sets having the following names (EXY represents the name of an element type, such as E21, E33, etc.):

DEF	EXY	=	Basic (integer form) element definitions.
GD	EXY	=	Group directory.
GTIT	EXY	=	Group titles.

When a sub-processor is executed, the resultant output data sets replace all data previously generated by the same sub-processor in any previous execution.*

Elements of each type may be assigned by the analyst to separate groups. Within each group, each element has an identifying index number. The grouping of elements serves various purposes, which new users will learn with experience. It is almost always best to use many groups. Definition of the elements in group n is preceded by an input card in the following form:

GROUP n' - - - title describing group n - - -. All elements defined by input cards following such a card belong to group n. Groups must appear sequentially (group 1, group 2, - -). If no GROUP card appears, all elements are assigned by default to group 1. The order in which individual element definitions appear following a GROUP card determines the index numbers they are assigned within the group.

* It is planned in a future release to provide an "update" mode of operation allowing deletions, additions, and corrections to be made.

Element definition cards have the forms shown below.

All 2-node elements:

J1, J2 $\overbrace{\text{NETOPT, NET(1), NET(2)}}^{\text{Optional}}$ - - -

All 3-node elements:

J1, J2, J3 NETOPT, NET(1), NET(2), - -

All 4-node elements:

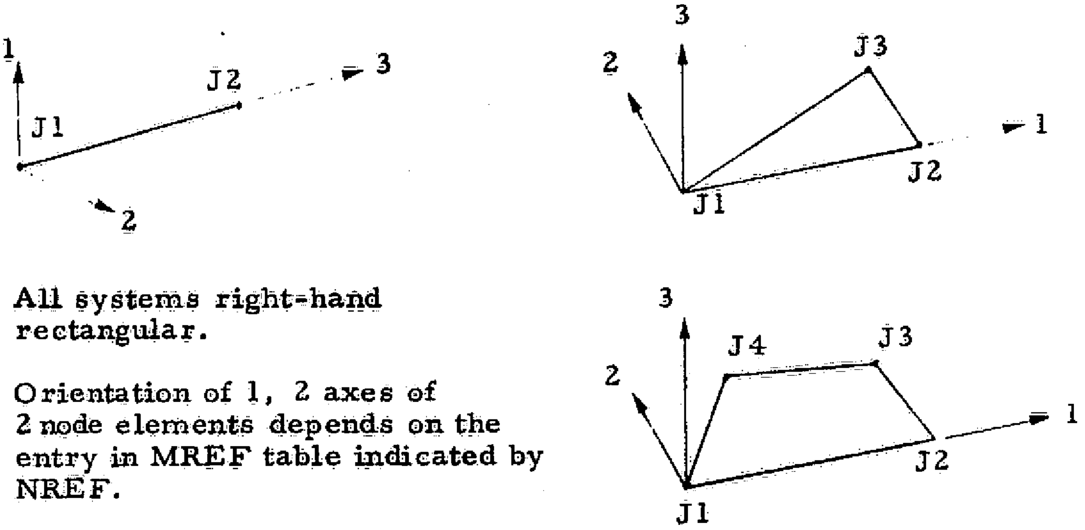
J1, J2, J3, J4 NETOPT, NET(1), NET(2), - - -

It should be noted that the order in which J1, J2 - - appear on the element definition cards is very important, since the orientation of element reference frames is defined on the basis of this information (see Fig. ELD-1). All element-related quantities, such as stiffness coefficients, stresses, etc., are relative to these reference frames.

In the above, J1, J2, - - indicate the joints interconnected by the first element defined by the card. If the optional network operation parameters, NETOPT, NET(1), NET(2), - -, are given, they result in definition of additional elements, in accordance with specific rules which will be defined subsequently.

The section properties, material properties, etc. associated with each element depend upon the values of various table reference pointers prevailing at the time the element definition card appears. These pointers are summarized below.

Table Pointer	Default Value	Associated Table
NMAT	1	Material constants, MATC.
NSECT	1	Section properties. E44 elements refer to Table SB. All other 3 and 4-node elements refer to Table SA. E21, E22, E23, E24, and E25 refer to BA, BB, BC, BD, and BB, respectively.
NOFF	0	Beam rigid link offsets, BRL (2-node elements, only).
NNSW	0	Nonstructural distributed weight, NSW.
NREF	1	Beam reference frame orientation, MREF (2-node elements other than E25, only). For E25 elements, NREF points to an entry in the ALTREF table. For 3 and 4-node elements NREF is not a table entry pointer. Instead, NREF is used to specify the direction of action of positive pressure exerted on the element. If NREF=0, pressure exerts no force on the element. If NREF=1, positive pressure acts in the direction of the 3-axis of the element reference frame (see Fig. ELD-1, below). If NREF=-1, pressure acts in the opposite direction.



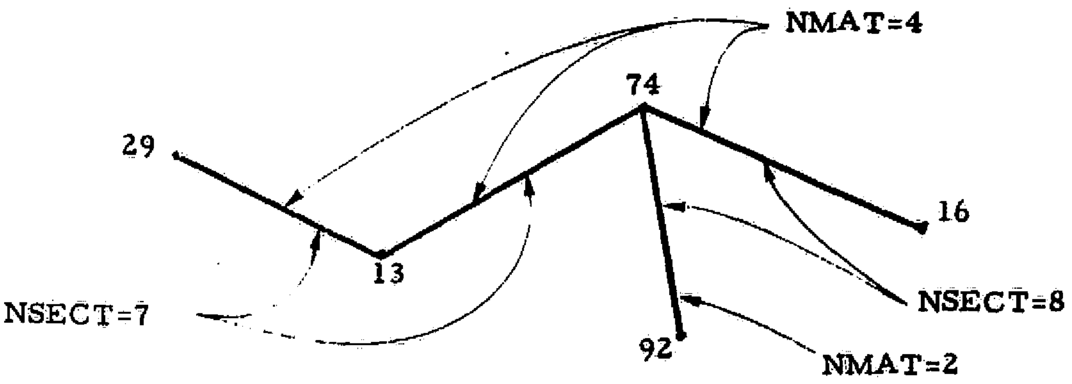
(1) All systems right-hand rectangular.

(2) Orientation of 1, 2 axes of 2 node elements depends on the entry in MREF table indicated by NREF.

(3) For 3, 4-node elements, J3 lies in quadrant 1 of plane 1-2.

Figure ELD-1. Element Reference Frames

An example of the use of the table reference parameters is shown below.



```
NMAT=4:  NSECT=7:  29,13:  13,74:  NSECT=8:  74,16: NMAT=2:  74,92$
```

ELD also accepts input commands which have the effect of modifying the data given on subsequent input cards. These commands are summarized below.

Command			Meaning of n (n default is zero in all cases)
MOD	JOINT=	n	Add n to all joint numbers.
MOD	GROUP=	n	Add n to all group numbers.
MOD	NSECT=	n	Add n to NSECT pointers.
MOD	NMAT=	n	" NMAT "
MOD	NNSW=	n	" NNSW "
MOD	NREF=	n	" NREF "
MOD	NOFF=	n	" NOFF "
INC	NSECT=	n	See below.
INC	NMAT=	n	" "
INC	NNSW=	n	" "
INC	NREF=	n	" "
INC	NOFF=	n	" "

A frequently encountered situation is the one in which a substantial portion of a finite element model is comprised of a regular mesh, which can be automatically defined by a mesh generation command, but the section properties are so nonuniform that it becomes necessary or highly desirable for simplicity of input preparation to associate with each element a unique entry in the section property table. A common example is an aircraft fuselage, where cutouts, doublers, stiffeners, etc., result in extensive variations in section properties over a mesh that is topologically and geometrically regular. A similar situation involving distributed nonstructural weight also sometimes occurs; e.g. thermal protection material of varying thickness. The INC command is intended primarily to address situations of this type.

The INC commands are used to cause the associated table reference pointers to be automatically incremented by n as successive elements are defined. This is especially useful in conjunction with network generation options, which are defined in detail on subsequent pages. For example, the element definition command 701, 702, 1, 50 causes fifty elements to be defined, connecting joint pairs (701, 702), (702, 703) - - - - (750, 751). If section property table entries 101, 102, - - -150 apply, successively, to these elements, the following input would suffice.

```
NSECT= 101: INC NSECT=1: 701,702, 1,50$
```

All MOD and INC parameters are internally reset to zero after conclusion of each sub-processor execution. The MOD and INC commands are not cumulative. That is, MOD NSECT 7: - - - - MOD NSECT 3: - - is not equivalent to MOD NSECT 10.

Reset Controls

Only one reset command is permitted:

OUTLIB = Destination library for ELD output. The TAB-generated data sets must already be resident in this library.

Core Requirements

ELD rarely requires more than 1500-3000 words of working core space, regardless of the size of the problem.

Code Release Data

Level 7 (UNIVAC, CDC), July 1974, coded by W. D. Whetstone.

Two-node element network generators.

If NETOPT=1,

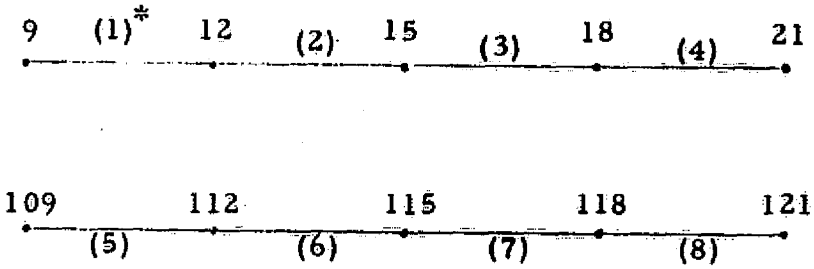
NI= NET(1) (default=1),
NJ= NET(2) (default=1),
JINC= NET(3).

Implied sequence:

```
        N1=J1
        IDIFF=J2-J1
        DO  200  J=1,NJ
        DO  100  I=1, NI
        N2=N1+IDIFF
        Define element connecting node N1 to N2
  100   N1=N2
  200   N1=J1 + J * JINC
```

Example:

J1	J2	Netopt	NI	NJ	JINC
9	12	1	4	2	100



*The order in which the elements are defined is indicated by the number enclosed in parentheses. The index number identifying elements within each group are determined by the order in which the elements are defined.

Two-node element network generators (continued).

If NETOPT=2,

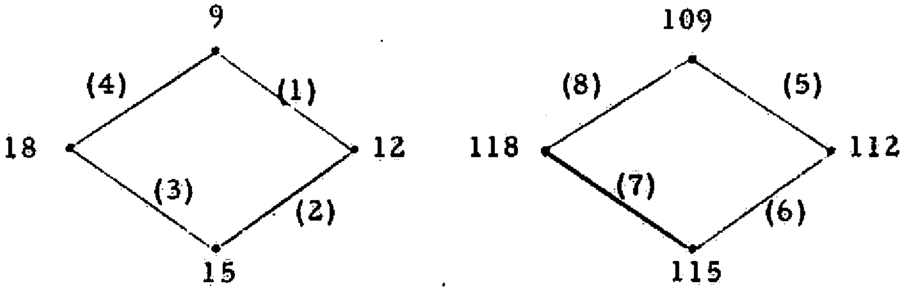
```
NI=     NET(1)  (default=1),
NJ=     NET(2)  (default=1),
JINC=   NET(3).
```

Implied sequence:

```
        N1=     J1
        IDIFF= J2-J1
        DO   200   J=1,NJ
        DO   100   I=1,NI
        N2 = N1 + IDIFF (except for closing element, when I=NI)
        Define element connecting N1,N2.

100     N1=N2
200     N1=J1 + J*JINC
```

Example:

J1	J2	Netopt	NI	NJ	JINC
9	12	2	4	2	100



Two-node element network generators(continued).

If NETOPT=3,

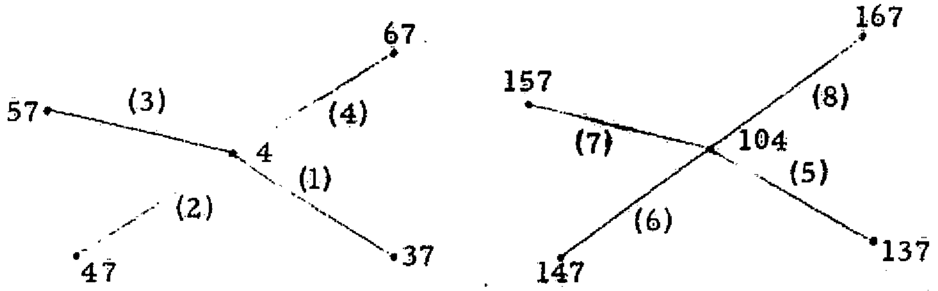
NI= NET(1),
IINC= NET(2),
NJ= NET(3) (default=1)
JINC= NET(4).

Implied sequence:

```
         N1=J1
         N2=J2
         DO  200  J=1, NJ
         DO  100  I=1, NI
         Define element connecting N1 to N2
  100    N2 = N2 + IINC
         N1 = J1 + JINC*J
  200    N2 = J2 + JINC*J
```

Example:

J1	J2	Netopt	NI	IINC	NJ	JINC
4	37	3	4	10	2	100



Three-node element network generators.

If Netopt=1,

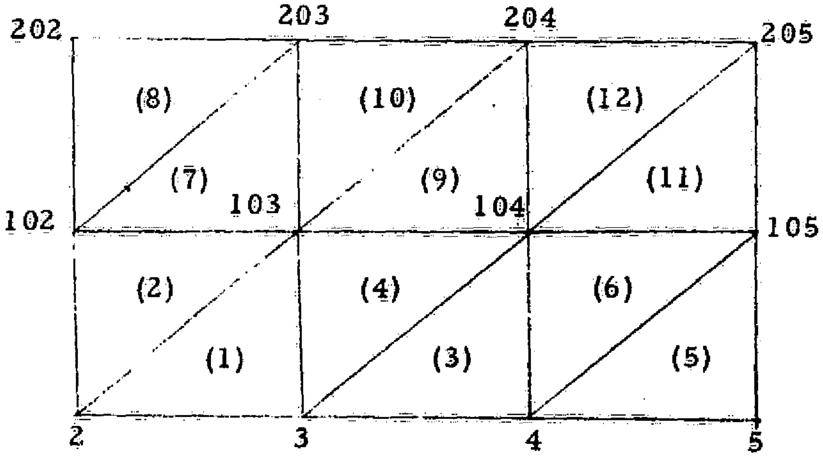
NI= NET(1),
NJ= NET(2) (default=1).

Implied sequence:

```
        IINC=  J2-J1
        JINC=  J3-J2
        N1=  J1
        N2=  J2
        N3=  J3
        N4=  J3-IINC
        DO   200   J=1, NJ
        DO   100   I=1, NI
        Define element connecting  N1, N2, N3
        Define element connecting  N3, N4, N1
        N1=  N1 + IINC
        N2=  N2 + IINC
        N3=  N3 + IINC
  100   N4=  N4 + IINC
        N1=  J1  + JINC
        N2=  J2  + JINC
        N3=  J3  + JINC
  200   N4=  N3 - IINC
```

Example:

J1	J2	J3	Netopt	NI	NJ
2	3	103	1	3	2



Three-node element network generators (continued).

If Netopt=2,

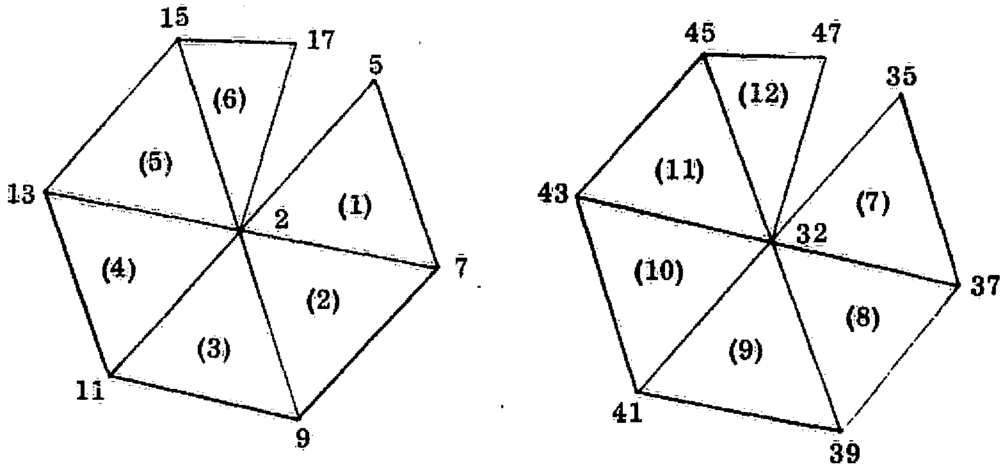
NI= NET(1),
NJ= NET(2) (default=1),
JINC= NET(3).

Implied sequence:

```
        IINC= J3-J2
        N1= J1
        N2= J2
        DO  200  J=1, NJ
        DO  100  I=1,  NI
        N3= N2 + IINC
        Define element connecting N1,N2,N3
    100 N2=N3
        N1= J1 + J*JINC
    200 N2= J2 + J*JINC
```

Example:

J1	J2	J3	Netopt	NI	NJ	JINC
2	5	7	2	6	2	30



Three-node element network generators (continued).

If <u>Netopt=3,</u>

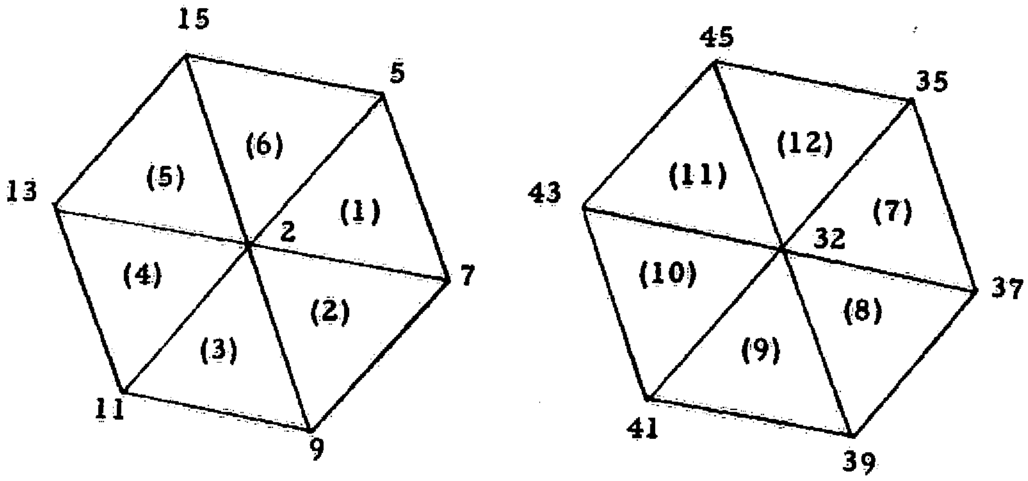
NI= NET(1),
NJ= NET(2) (default=1),
JINC= NET(3).

Implied sequence:

```
          IINC=  J3-J2
          N1= J1
          N2= J2
          DO   200   J=1,  NJ
          DO   100   I=1,  NI
          N3=  N2 + IINC (except closure when I=NI)
          Define element connecting  N1,N2,N3
    100   N2=N3
          N1 = J1 + J*JINC
    200   N2 = J2 + J*JINC
```

Example:

J1	J2	J3	Netopt	NI	NJ	JINC
2	5	7	3	6	2	30



Four-node element network generators.

If Netopt=1,

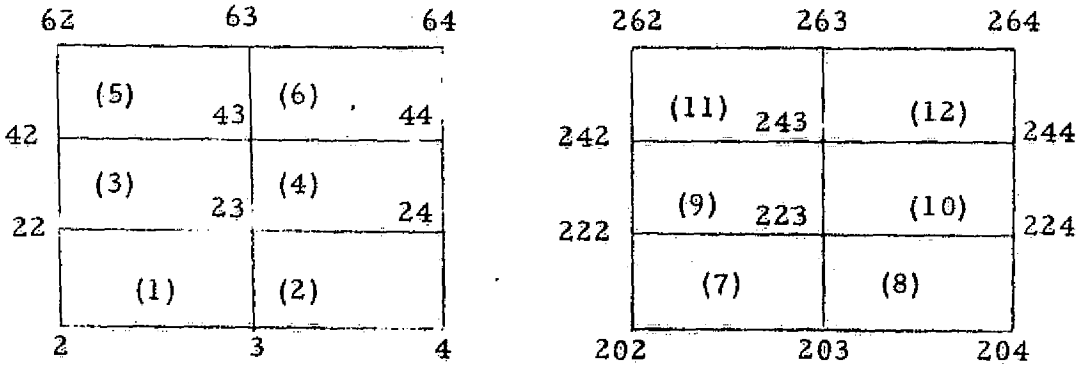
NI= NET(1),
NJ= NET(2) (default=1),
NK= NET(3) (default=1),
KINC= NET(4).

Implied sequence*:

```
          IINC=  J2-J1
          JINC=  J4-J1
          DO   300   K=1,  NK
          N1  = J1
          DO   200   J=1,  NJ
          DO   100   I=1,  NI
          N2=  N1+IINC
          N3=  N2+JINC
          N4=  N1+JINC
          Define element connecting N1, N2, N3, N4
    100   N1  = N1+IINC
    200   N1  = J1+J*JINC
    300   J1  = J1+KINC
```

Example*:

J1	J2	J3	J4	NETOPT	NI	NJ	NK	KINC
2	3	23	22	1	2	3	2	200



*Note: J3 must be present, although not used.

Four -node element network generators (continued).

If Netopt=2,

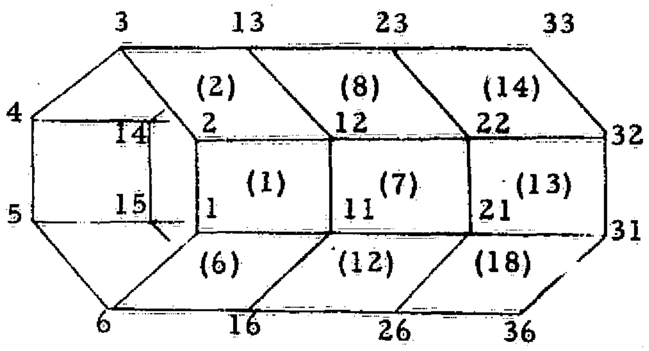
NI= NET(1),
NJ= NET(2) (default=1).

Implied sequence:

```
      IINC=    J2-J1
      JINC=    J4-J1
      N1=      J1
      DO   200   J=1,NJ
      DO   100   I=1, NI
      N2 = N1 + IINC
      N3 = N2 + JINC  (except closure when I=NI)
      N4 = N1 + JINC  (except closure when I=NI)
      Define element connecting N1,N2,N3,N4
  100 N1 = N1 + IINC
  200 N1 = J1 + J*JINC
```

Example:

J1	J2	J3	J4	Netopt	NI	NJ
1	11	12	2	2	6	3



3.3 E – E-STATE INITIATION

Function. For each element in the structure there is an "element information packet." Depending upon the particular type of element, each packet usually contains information in each of the following categories:

(1) General information such as the connected joint numbers, table reference integers, etc.

(2) Material constants.

(3) Geometrical data, e.g. dimensions, transformation matrices, etc.

(4) Section property data.

(5) Intrinsic stiffness matrix (or equivalent).

(6) Intrinsic stress matrix (or equivalent).

Processor E constructs these packets in skeletal form (omitting the intrinsic stiffness, and stress data). Input to E consists of:

- All TAB-produced data sets except for rigid mass data, constraint data, and the joint numbering sequence.
- All data sets produced by ELD.

Output from E consists of two data sets for each element type present. Where Exx represents the element type (e.g., E21, E22, - - - E31, - - E41, E42, etc.), the output data sets are

(1) Exx = all element information packets for element type Exx.

(2) DIR Exx = an associated directory.

In addition, a SYSVEC-format data set named DEM is produced. This data set is a system mass matrix in (lumped mass) diagonal form. DEM includes only the mass associated with elements and any distributed nonstructural mass attached to elements. Only displacement-dependent terms are included (i.e., rigid link offsets, if any, are ignored , and rotation-dependent terms are not included).

RESET Controls.

Name	Default Value	Meaning
BLIB	1	Library containing all input data sets.
ELIB	1	Destination library for all output data sets.
LZERO	.001	Used to detect identically zero lengths.
WARP	.001	Test for allowable warp of quadrilateral elements, such as E41, which are required to be flat. Warp is defined as Z/L, where Z is the distance from the fourth node to the plane of the first three nodes, and L is the distance between nodes 1 and 2.
RCH	.0001	Where R is any 3 x 3 orthogonal transformation, terms of $R^t R$ must equal the identity matrix, within a tolerance of RCH.
G	1.	Gravitational constant used in constructing the system diagonal mass matrix, DEM.

Core Requirements. It is rarely necessary to allocate more than about 20,000 words of working core space for execution of E. Requirements vary from problem to problem depending upon the size of the TAB-generated tables. All of the geometry, material constant and distributed weight tables are held continuously in core during execution of E. As packets are formed for each successive element type, the associated section property table is loaded into core. In addition to the above, about 1000 words should be allowed for input of the basic element definition data sets from ELD, plus 5000-10000 words to serve as an output buffer for the element information packets.

Code Release Data. Level 9 (Univac), July 1975, coded by W. D. Whetstone.

3.4 EKS - ELEMENT INTRINSIC STIFFNESS AND STRESS MATRIX GENERATOR

Function. Based on the dimensions, section properties, etc., currently embedded in the element information packets originated by processor E, EKS computes intrinsic stiffness and stress matrices and inserts them into the packets.

RESET Controls.

Name	Default Value	Meaning
ELIB	1	Library containing the element information packets.
TIME	0	Nonzero value causes printout of intermediate CP and wall clock times.
GAZERO	$10.^{-20}$	Zero-test parameter, (beam area) x (shear modulus).
C1ZERO	$10.^{-20}$	Zero-test parameter, beam non-uniform torsion constant.

Core Requirements. EKS requires only a buffer area through which element information packets are transmitted. About 5000-15000 locations are usually suitable. IO counts will vary in inverse proportion to core space.

Code release data. Level 7 (Univac, CDC), July 1974, routines QMB1, TM, TPB7, QMB5, QPB11 coded by C. L. Yen, all others coded by W. D. Whetstone.

Section 4

SPAR-FORMAT SYSTEM MATRIX PROCESSORS

Six processors are directly associated with the assembly, factoring, and display of SPAR-format system matrices. They are TOPO, K, M, KG, INV, and PS. Figure 4-1 illustrates the data exchange involved in execution of these processors. This figure should be carefully analyzed before the remainder of this section is read.

For a discussion of methods of combining system matrices (e.g., K + Kg, K - CM, adding rigid mass data to CEM or DEM, etc.), see the SUM subprocessor of AUS.

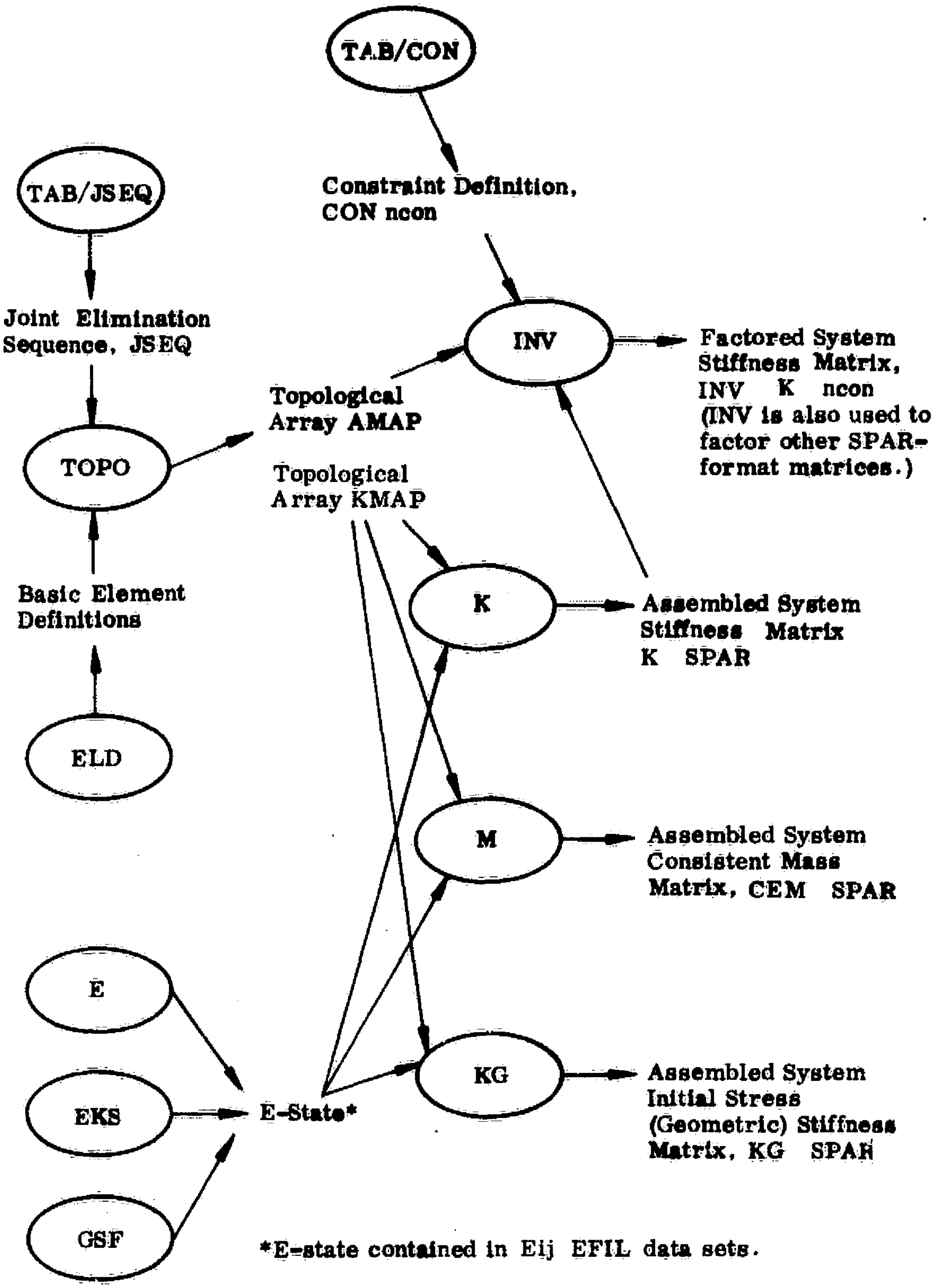
Reference 1 presents a procedure for solving high-order systems of linear equations of the kind which occur in displacement-method finite element analysis. The system stiffness matrix is regarded as an array of submatrices, each submatrix corresponding to the connection of one joint to another. Each submatrix is n by n, where n is the number of degrees of freedom at each joint. For general shells and frames, n = 6 (three displacements, three rotations); for a plane frame, n = 3 (two displacements, one rotation); etc. The only nonzero submatrices in a system stiffness matrix are those corresponding to pairs of joints connected by elements. Accordingly in all but the smallest finite element models only a tiny fraction of the submatrices are nonzero - usually less than 1%. The characterizing feature of the procedure described in Ref. 1 is that it operates exclusively with data contained in the nonzero submatrices, virtually eliminating the unessential arithmetic (multiplying, adding zeros) and wasted data storage (storing zeros) associated with conventional band matrix techniques.

Originated in the early 1960's, this method was first applied in a series of large capacity frame analyzers, and later in the SNAP series (e.g., Ref. 2) of finite element network analyzers. SPAR uses essentially the same procedure, except for certain differences in the order in which certain of its phases are performed. SPAR users should become familiar with Ref. 1, especially with regard to the effects of joint numbering sequence on execution costs and data storage requirements.

References

1. Whetstone, W. D., "Computer Analysis of Large Linear Frames," J. Struct. Div., ASCE, Nov. 1969.

2. Whetstone, W. D., "Structural Network Analysis Program Users Manual, Static Analysis Version V70E," LMSC-HREC D162812, Lockheed Missiles & Space Company, Inc., Dec. 1970.



*E-state contained in Eij EFIL data sets.

Fig. 4-1 Assembly and Factoring of SPAR-Format System Matrices

4.1 TOPO – ELEMENT TOPOLOGY ANALYZER

Function. TOPO analyzes element interconnection topology and creates two data sets, KMAP and AMAP. KMAP is used by processors K, M and KG to guide assembly of system stiffness and mass matrices in the SPAR standard sparse-matrix format. AMAP is used by processor INV in factoring system matrices. Input data sets are:

- All basic element definition data sets generated by processor ELD.
- JSEQ, the joint elimination sequence, generated by TAB subprocessor JSEQ. If JSEQ is not present, serial elimination sequence: 1,2,---, is assumed.

RESET Controls

Name	Default Value	Meaning
BLIB	1	Library containing all input data sets
HLIB	1	Destination library for KMAP
ILIB	1	Destination library for AMAP
LRKMAP	896	KMAP block length
LRAMAP	1792	AMAP block length
PRTKMAP	0	Nonzero value results in printout of topological data during generation of KMAP.
PRTAMAP	0	Same as above for AMAP.
MAXSUB	1000	Maximum allowable value of size index I_s (see Ref. 1, Sec. 4).
ILMAX	(none)	Maximum joint connectivity to be permitted in allocating space for topological tables.

Core Requirements. Core space requirements for TOPO are not easily defined because they depend on the number of nodes, element interconnection topology, and the joint numbering sequence prescribed. The working core required is not often greater than

$3000 + 2 * J + 4 * I^2$, where

J = the number of joints, and

I = maximum joint connectivity.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

4.2 K – THE SYSTEM STIFFNESS MATRIX ASSEMBLER

__Function.__ K assembles unconstrained system stiffness matrices in the standard SPAR sparse-matrix format. Input consists of the following:

- The element information packets produced by processors E and EKS.
- Topological information contained in data set KMAP produced by processor TOPO.
- Certain minor directory information generated by processors TAB and ELD.

The output is a data set named K SPAR, the unconstrained system stiffness matrix.

__RESET Controls__

Name	Default Value	Meaning
BLIB	1	Library containing TAB, ELD output.
ELIB	1	Library containing element information packets.
HLIB	1	Library containing data set KMAP, produced by TOPO.
OUTLIB	1	Destination library for output data.
LREC	2240	Block length of output data set.
SPDP	1	Value of 2 causes output to be generated in double precision.

Core Requirements. It is rarely necessary to allocate more than 10000 – 20000 locations. Space is required to concurrently accommodate the following:

- One block of KMAP
- One block of K SPAR
- A core pool area in which intermediate results are held during assembly of the system matrix. Specific requirements depend on element interconnection topology and the joint numbering sequence. Required space will be equal to KSIZE (as printed by TOPO) multiplied by the square of the number of degrees of freedom per joint.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

4.3 M — SYSTEM CONSISTENT MASS MATRIX ASSEMBLER

Function. M assemblies unconstrained system consistent mass matrices, considering only the structural and nonstructural distributed mass associated with the elements. Rigid link offsets are ignored by this processor. The name of the output data set is CEM SPAR.

The SUM command in AUS can be used to add rigid mass data (e.g., via TAB/RMASS) to CEM, if required.

M requires the same input data sets as K, and has identical working core requirements.

RESET Controls. Reset controls are the same as for processor K,* plus the following:

Name	Default Value	Meaning
G	1.	Constant used to convert element distributed "weight" to mass. For example; if distributed weights were given in lbs/in and lbs/in^2, and material specific weights were given in lbs/in^3, one would normally reset G = 386., depending upon the planned use of CEM SPAR.
IBEAM	0	Any nonzero value will cause all beam rotatory inertia terms, both lateral and torsional, to be ignored.
INERT	0	Any nonzero value results in assumption of linearly varying lateral displacement fields in calculating mass matrices for all elements. That is, beams are treated as axial (bar) elements, shell elements are treated as

(continued on next page)

*Except that double-precision output is not allowed.

RESET Controls, continued

Name	Default Value	Meaning
		membranes, etc. This often is an excellent approximation, and requires only a small fraction of the CP time used in the default mode.

Code Release Data. Level 7 (Univac, CDC), July 1974, routines M33, M34, M63, and M64 coded by C. L. Yen, all others coded by W. D. Whetstone.

4.4 KG – SYSTEM INITIAL STRESS (GEOMETRIC) STIFFNESS MATRIX ASSEMBLER

Function. KG assembles unconstrained system initial-stress (geometric) stiffness matrices, K_g, in the standard SPAR sparse matrix format. K_g will be based on the stress state currently embedded in the E-State (EIJ EFIL data sets), as computed by GSF. Rigid link offsets are ignored by KG. The output data set is named KG SPAR. Required input data sets are the same as for processor K. Core requirements are the same as for K.

RESET Controls. Reset controls are the same* as for processor K, plus the following:

Name	Default Value	Meaning
IKG2	0	Any nonzero value results in all beams being treated as though they were axial elements (bars) in computing Kg.
IKG34	0	Any nonzero value results in all shell elements being treated as though they were membranes in computing Kg.
IZERO	$10.^{-10}$	Zero-test parameter used to detect identically-zero beam moments of inertia.
AZERO	$10.^{-10}$	Zero-test parameter used to detect identically-zero cross-sectional areas of beams.

Code Release Data. Level 7 (Univac, CDC), July 1974, routines GTM, GQM, GTP, GQP coded by C. L. Yen, all others coded by W. D. Whetstone.

*Except that double-precision output is not allowed.

4.5 INV – SPAR FORMAT MATRIX DECOMPOSITION PROCESSOR

Function. INV factors assembled system matrices in the SPAR standard sparse matrix format, using a procedure essentially the same in principal as the one outlined in Ref. 1, Sec. 4. The following input data sets are required.

- An unconstrained assembled system matrix, such as K SPAR.
- A constraint definition, such as CON 1, produced by the CON sub-processor of TAB.
- Topological information contained in data set AMAP produced by processor TOPO.

Output consists of a factored system matrix having the name INV XXXX n, where

XXXX = the first word of the name of the input stiffness matrix, and

n = an integer identifying the constraint case.

For example, if K SPAR is factored using constraint case 1, the output is INV K 1.

RESET Controls

Name	Default Value	Meaning
KLIB	1	Library containing input system matrix.
KILIB	1	Destination library for INV output.
ILIB		Library containing topological data set AMAP. Default is the library containing the matrix to be factored.

Name	Default Value	Meaning
K	K	First word in name of input data set. Typical RESET actions are K= K+KG, K= K+KA, K= K+CM, etc.
CON	1	Integer identifying the constraint case to be used. This data set must be resident in library 1.
LRA	3584	Ouput data set block length.
DZERO	.00001	Zero-test parameter used to detect identically zero diagonal terms during factoring of the system matrix.

Single precision matrices are factored in single precision. INV uses double precision arithmetic in factoring double precision matrices, then truncates the output data set to single precision.

Use the ONLINE=2 statement to obtain printout of joints/components where negative diagonal terms occur during factoring.

Core Requirements.

Where J = the number of joints in the structure,

L_1 = block length of input matrix,

L_2 = block length of output matrix,

L_3 = block length of AMAP,

m = 1 if factoring in single precision, 2 if in double precision,

n = number of degrees of freedom per joint (usually 3, 4, 5 or 6), and

I_s = the size index (Ref. 1 Sec. 4) of the structure, the minimum core required is

$$J + L_3 + m\,(L_1 + L_2 + n^2\, I_s)\ .$$

Univac 1108 CPU Time Estimation.

TOPO prints the value of the cost index I_{c1} (Ref. 1, Section 4), associated with the selected joint numbering sequence. Where n is the number of degrees of freedom per joint, the number of inner product operations performed by INV is approximately $N = n^3 \times I_{c1}$. The Univac-1108 time required to execute INV is C x N. Approximate values of C, in U-1108 cpu microseconds, are tabulated below.

	Single Precision Execution	Double Precision Execution
n= 6	8.0	12.4
n= 3	10.4	15.3

Code Release Data. Level 9 (Univac, CDC), July 1975, coded by W. D. Whetstone.

4.6 PS — SPAR FORMAT MATRIX PRINTER

Function. SPAR-format matrices and factored SPAR-format matrices are printed by this processor. The user is cautioned that factored matrices usually contain a very large amount of data and should not normally be printed. PS does not have special RESET controls. Its activity is controlled by the following commands, which may appear any number of times within a single PS execution:

Command	Meaning
Lib= nlib	Source data is in library nlib (default = 1).
J= j_1, j_2	Print data for joints j_1 through j_2.
N1,N2,n3,n4	Print data from the named data set.

When printing factored matrices (produced by INV), the ONLINE statement may be used to further restrict the amount of printout. Set ONLINE= 0 to obtain only constraint and topological information. Set ONLINE= -1 to obtain only constraint information.

Core Requirements. Data space is required for one block of any data set printed.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

Section 5

UTILITY PROGRAMS

The following utility programs are described in this section:

- AUS – Arithmetic Utility System. This program performs an array of functions in the following areas:
  (1) Matrix arithmetic.
  (2) Construction, editing, and modification of data tables (e.g., applied loading).
  (3) Operations related to substructure analysis.
- DCU – Data Complex Utility. The following are typical DCU functions.
  (1) Display of library tables of contents.
  (2) Data transfer (e.g., library to library, sequential file to library, library to sequential file).
  (3) Enabling and disabling of data sets.
  (4) Changing data set names.
  (5) Printing data sets.
  (6) Storing/retrieving libraries on tape.
- VPRT – Vector Printer. This program is used to edit and display any data sets in SYSVEC format (e.g., eigenvectors, static displacements, reactions, nodal load vectors).

5.1 AUS – ARITHMETIC UTILITY SYSTEM

Function. AUS subprocessors are summarized in Table 5-1 according to their functional categories. Detailed information about individual subprocessors is contained in the remainder of this section.

It is mandatory that Library 1 contain the data set JDF1 produced as a result of the START command in TAB. Any SYSVEC data set or SPAR-format matrix operated on in AUS must correspond to the JDF1 data set present in Library 1. This is true even if the data set is present in a Library other than Library 1.

Reset Controls

No special RESET controls are provided.

Core Requirements

Specific core requirements are defined for individual subprocessors.

Code Release Data

Level 9 (UNIVAC), June 1975, coded by W. D. Whetstone.

Table 5-1 Summary of AUS Subprocessors

Miscellaneous	General Arithmetic	Data Set Constructors	Substructure
INLIB	SUM	TABLE	SSPREP
OUTLIB	PRODUCT	SYSVEC	SSM
DEFINE	UNION	ELDATA	SSK
ZERO	XTY	ALPHA	SSID
FIND	XTYSYM		
	XTYDIAG		
	NORM		
	RIGID		
	RECIP		
	SQUARE		
	SQRT		
	RPROD		
	RTRAN		
	RINV		

5.1.1 Miscellaneous AUS Commands

The following statements designate the primary data source and destination libraries, respectively:

INLIB= Source library (default = 1)

OUTLIB= Destination library (default = 1)

The above statements may be used repeatedly during a single AUS execution, if different libraries are to be used.

The DEFINE statement is provided to permit the use, in certain subprocessors, of single-word (alpha) names to identify specific data sets or portions of such sets. The most general statement form is as follows:

DEFINE X= Lib N1 N2 n3 n4 k_1 k_2

The above statement means that the short-form name X (any alpha word) will refer to a data set, named N1 N2 n3 n4, contained in library Lib. For multiblock data sets, the optional parameters k_1 and k_2 indicate that X consists only of blocks k_1 through k_2. If k_1 and k_2 are omitted, X is the entire data set.

The Lib parameter may be omitted; in which case the indicated data set is in the primary data source library established by the last INLIB statement. If the k_1 and k_2 parameters are not needed, it is permissible to omit trailing words of the data set name, provided they are not needed for unique identification of the data set (i.e., omitted trailing words are MASK-filled).

As an example, suppose the data set VIBR MODE 10 1 contains 5 blocks, $[U_1, U_2, U_3, U_4, U_5]$. To identify $[U_2\ U_3\ U_4]$ by the short-form name VA, the following statement would be used:

```
DEFINE VA= VIBR MODE 10 1, 2,4$
```

It should be noted that the DEFINE statement does not cause production of a data set. INLIB statements have no effect on definitions established in preceding DEFINE statements. For example:

```
INLIB= 4: ---: ----: ---
DEFINE V7= VIBR MODE 7
INLIB= 2
MV7= PRODUCT(M,V7)
```

In the last statement, V7 is understood as being in library 4 (not 2), although M would be in library 2, unless otherwise indicated by a DEFINE M= --- statement.

The following is permitted:

```
DEFINE X= A
=
-
=
DEFINE X= C D (supersedes preceding DEFINE X= A)
```

A maximum of 50 distinct short form names may be DEFINEd.

The command ZERO=e defines the current value of a zero-test parameter, e, that is used in connection with several functions. The default value of e is 1.0×10^{-20}.

The statement FIND Lib N1 N2 n3 n4 is used, primarily in interactive operation, to obtain a quick look at portions of library tables of contents. The resulting printout is similar to that produced by a DCU/TOC; however, all qualifying data sets in Lib are printed, if any parts of the data set name (N1 N2 n3 n4) are MASKed. Any omitted trailing words of a data set name are MASK-filled; i.e., FIND 1 X is equivalent to FIND 1 X MASK MASK MASK. For example, all SPAR-format matrices in library 3 would be located by the statement FIND 3 MASK SPAR.

5.1.2 General Arithmetic Operations

Table 5.1.2-1 summarizes commands in this category. All are in the following form, except where specifically noted otherwise.

$$Z= \quad \text{Oper}(\ c_1\ X_1,\ \ c_2\ X_2, ---)$$

where Oper is one of the operation names, such as SUM, PRODUCT, etc., and the X_i's are short-form names identifying source data. If a short-form name X has not appeared in a DEFINE X= ... statement, it is assumed that X is a data set named X MASK MASK MASK that is contained in the current primary data source library identified by the last INLIB statement. The c_i's are floating-point constants which may be omitted (default is 1.0).

The data set produced as a result of the command will be stored in the current destination library designated by the last OUTLIB command. The name of the output data set will depend on the form of Z, as summarized below:

Form of Z	Output Data Set Name
N1	N1 AUS 1 1
N1 N2	N1 N2 1 1
N1 N2 n3	N1 N2 n3 1
N1 N2 n3 n4	N1 N2 n3 n4

In the following discussions of individual operations, it will be convenient to use the abbreviations listed below to identify specific types of data sets:

Abbreviation	Kind of Data Set
SS	Single-precision SPAR format system matrix (e.g. K SPAR, formed by K in default mode).
DS	Double-precision SPAR format system matrix (e.g. K SPAR, formed in double-precision mode).
D	Diagonal system matrix, e.g. the diagonal mass matrix produced by processor E. Such data sets are stored as single vectors in SYSVEC format; i.e., D is a special case of V below.
V	System vectors in SYSVEC format, e.g. VIBR MODE 3 2.
R	Single block rectangular matrix, No. of rows = (NI*NJ)/NJ No. of cols. = NJ See Sections 2.2, 2.5
A	Any single or multiblock data set containing <u>all</u> floating point data.

Table 5.1.2-1 Summary of General Arithmetic Operations

Command Forms *	Meaning
Z= SUM(X,Y)	Z = X + Y (system matrices)
Z= PRODUCT(X,Y)	Z = X Y (system matrices)
Z= UNION(X_1,X_2,---)	Z = [X_1 | X_2 | X_3 ---]
Z= XTY(X,Y)	Z = X^tY
Z= XTYSYM(X,Y)	Z = X^tY, symmetric
Z= XTYDIAG(X,Y)	Z = X^tY, diagonal
Z= NORM(X,j,k,v)	System vector renormalization
Z= RIGID(j)	Rigid body motion vectors
Z= RECIP(X)	Each element z = 1./x
Z= SQRT(X)	Each element z = sign(x) $\sqrt{\|x\|}$
Z= SQUARE(X)	Each element z = x^2
Z= RPROD(X,Y)	Z = X Y (rectangular matrices)
Z= RTRAN(X)	Z = X^t (rectangular matrices)
Z= RINV(X)	Z = X^{-1} (square matrices)
Z= LTOG(X)	Converts system vector components from local joint reference frames to global
Z= GTOL(X)	Complement of LTOG

5.1.2.1 SUM. The general form of the command is as follows:

$$Z= \ SUM(c_x \ X, \ c_y \ Y) \quad .$$

This statement means that the output data set Z is c_x times X added to c_y times Y. If omitted, c_x and c_y default to 1.0. Only two data sets may be combined; i.e., in Z= SUM(P,Q,R), R would be ignored. The type of Z depends on the type of X and Y, as shown below (dashes indicates illegal operation).

		Y type				
		SS	DS	D	A	
X type	SS	SS	DS	SS	–	Z type
	DS	DS	DS	DS	–	
	D	SS	DS	D	–	
	A	–	–	–	A	

The following limitations must be observed:

(1) If X and Y are types SS or DS, they must have been formed on the basis of the same KMAP data set from processor TOPO.

(2) A type D matrix must be based on the same number of system joints as any D, SS, or DS data set with which it is combined.

(3) Two type A data sets must have the same block length (NI*NJ), and the same number of blocks. For example,

DEFINE A= STAT DISP 1 1 4,5 (2 blocks)

DEFINE B= STAT DISP 2 1 9,10 (2 blocks)

DEFINE C= STAT DISP 3 1 5,9 (5 blocks)

Z= SUM(A, B) is legal

Z= SUM(A, C) is not legal

Examples.

K+KG= SUM(K, 4.7 KG)	System stiffness matrix including effects of prestress.
M1= SUM(RMASS, DEM)	Diagonal system matrix composed of rigid mass data plus the lumped-mass equivalent of all distributed element mass.
M2= SUM(CEM, RMASS)	SPAR-format consistent mass matrix, plus rigid-mass data.
K24= SUM(K, -24000. M)	Shifted stiffness matrix to be used in EIG to compute eigenvalues near 24,000.

Core Requirement. One block of **X** plus one block of **Y**.

5.1.2.2 PRODUCT. The general form of the command is as follows:

$$Z= \text{PRODUCT}(\, c_x X, c_y Y)$$

This statement means that Z is c_x times c_y times X post-multiplied by Y. In standard applications, X is of type S, DS, or D, and Y is of type V. Z will have the same number of blocks (vectors) as Y.

Example. Construct inertia force vectors due to rigid-body acceleration. The command R= RIGID(j) would result in production of a 6-block data set containing system rigid-body motions in SYSVEC format. Where M is the system mass matrix,

MR= PRODUCT(M, R)

would produce a 6-block data set, in SYSVEC format, containing inertia force vectors due to unit rigid-body accelerations.

In addition to the above, PROD can also be used to perform element-by-element multiplication of data sets, provided that X and Y have the same block length (NI*NJ), and both contain only real data. Where:

x_i = the ith element in the first block of X,

y_{ij} = the ith element in the jth block of Y, and

z_{ij} = the ith element in the jth block of Z,

$z_{ij} = x_i y_{ij}$.

Core Requirement. One block each of X, Y, and Z.

5.1.2.3 UNION. The general form of the command is as follows:

$$Z= \text{UNION}(c_1\ X_1,\ c_2\ X_2,\ c_3\ X_3,\ \text{---})$$

This statement means that Z is formed by concatenating $c_1\ X_1$, $c_2\ X_2$,---. If omitted, the c_i's default to 1.0. The only conformability requirements are that (1) the block lengths of all X_i's be identical, and (2) the c_i's must not be used if the X_i's are not real. It is permissible for the X_i's to be of integer or alpha type. Z= UNION($c_1\ X_1$) is permitted.

Example. Collect eigenvectors from several data sets into a single set.

```
DEFINE V1= VIBR MODE 1 1 1,7
DEFINE V2= VIBR MODE 2 1 3,5
DEFINE V3= VIBR MODE 3 1 2,6
V= UNION( V1, V2, V3)
```

It should be noted that V contains 7 + 3 + 5 = 15 eigenvectors (blocks).

Example. Form a new array of case titles from parts of several existing arrays.

```
                DEFINE T1= CASE TITLES 1
                DEFINE T2= CASE TITLES 2 MASK 1,4
                DEFINE T3= CASE TITLES 3 MASK
CASE TITLES 123 1= UNION( T1, T2, T3)
```

Core Requirement. One block of Z.

5.1.2.4 XTY, XTYSYM, XTYDIAG. The general form of the XTY command is as follows:

$$Z= \mathrm{XTY}(\, c_x X,\ c_y Y)$$

This statement means that the output data set Z is c_x times c_y times the transpose of X post-multiplied by Y. X and Y may contain any number of blocks. but only one column per block (e.g., SYSVEC format). The block lengths of X and Y must be identical. Z will be a single block of length NI*NJ where NI and NJ are the number of columns in X and Y, respectively.

The operation XTYSYM may be used instead of XTY if it is known in advance that X^tY is symmetric. The XTYSYM operation requires about half the execution time and I/O activity of XTY. If XTYSYM is used, X and Y must have the same number of blocks. Z is in a single block containing z_{11}, z_{12}, z_{22}, z_{13}, z_{23}, z_{33}, ---.

The XTYDIAG command is in the same form as XTY. The output produced is a single block of length NI, where NI is the number of blocks in both X and Y. The ith element of Z is c_x times c_y times the vector inner product $(x_i^t\, y_i)$, where x_i = the ith column (block) of X, and y_i = the ith column (block) of Y.

In the following examples, it has been assumed that the following short-form names have been established:

K = system stiffness matrix

M = system mass matrix

U = an array of vectors in SYSVEC format

Example. Form reduced system mass and stiffness matrices, U^tMU and U^tKU, respectively.

MU= PRODUCT(M, U): UTMU= XTYSYM(U, MU)

KU= PRODUCT(K, U): UTKU= XTYSYM(U, KU)

Example. Where U contains vibrational eigenvectors not equally normalized, compute a vector of generalized mass terms, GM.

MU= PRODUCT(M, U): GM= XTYDIAG(U, MU)

Core Requirement. Minimum of one block of X plus one block of Y.

5.1.2.5 NORM. The general form of the command is as follows:

Z= NORM(X, j, k, v)

X must be in SYSVEC format; j and k must be integers; and v must be real. If the optional parameters j,k, and v are given, the ith vector (block) of Z will be $c^i x^i$, where x^i = the ith vector (block) of X, and $c^i = v$ divided by the term of x^i corresponding to joint j, component k.

If j,k, and v are omitted, c^i is the reciprocal of the largest (magnitude) term in x^i.

Example. Re-normalize a set of eigenvectors U so that the value of the direction-2 displacement at joint 420 is 1000.

NEWU= NORM(U, 420, 2, 1000.)

Example. Renormalize U to unit maximum values.

UNIT= NORM(U)

Core Requirement. One block of X.

5.1.2.6 RIGID. The general form of the command is as follows:

Z= RIGID(j)

Z will be in SYSVEC form, containing six vectors (blocks) that define rigid-body motion of the system. The first three blocks correspond to unit translations in global directions 1, 2, and 3. The second three blocks correspond to unit rotations about axes parallel to the global frame, passing through joint j. If the integer j is omitted, a default value of 1 is assumed.

Core Requirement. 18 times the number of joints in the structure.

5.1.2.7 RECIP, SQRT, SQUARE. These commands apply to single or multiblock data sets comprised entirely of real data. The output, Z, will be in the same form (block length, number of words, etc.) as the input, X. In the following definitions, z_i and x_i are the ith elements of Z and c_xX, respectively.

Z= RECIP(c_xX) indicates $z_i = 1.0/x_i$.

Z= SQRT(c_xX) indicates $z_i = (\text{sign of } x_i) \sqrt{|x_i|}$

Z= SQUARE(c_xX) indicates $z_i = x_i^2$.

The zero-test parameter established by the last ZERO= e statement (see Section 5.1.1) is used to avoid error stops in RECIP and SQRT. In these operations, $z_i = x_i$ if the magnitude of x_i is less than e.

Core Requirement. One block of X.

5.1.2.8 **RPROD, RTRAN, RINV.** Each of these commands applies only to single-block real data sets, interpreted as ordinary rectangular matrices. The output, Z, will be a single-block real data set. In the following definitions, X and Y may be replaced by $c_x X$ and $c_y Y$, if desired.

Z= RPROD (X,Y) indicates Z= XY.

Z= RTRAN(X) indicates Z= X^t.

Z= RINV(X) indicates Z= X^{-1}.

The usual rules of conformability apply to RPROD, and arguments of RINV must be square. Following is a tabulation of error codes produced by the subprocessor which handles these operations.

Code	Error
1	X not rectangular
2	X not real
3	Y not real
4	Y not rectangular
5	X, Y not conformable for multiplication (RPROD)
6	X not square (RINV)

Core Requirements. Data space requirements are as follows:

- For RPROD, the sum of the number of words in X, Y, Z.
- For RTRAN, twice the number of words in X.
- For RINV, $n^2 + n$, where X is n × n.

AUS/
LTOG
GTOL

5.1.2.9 LTOG and GTOL. The form of the LTOG command is:

Z= LTOG(X)

The source data set, X, must be in SYSVEC format. The output, Z, will contain joint motion (or force-moment) components relative to the global frame. The complementary operation is performed by GTOL; that is Z= LTOG(X) followed by Y=GTOL(Z) would produce a data set Y identical to X.

These commands may not be used if any joint motion components were excluded via the START command in TAB; that is, the SYSVEC vectors must contain 6 degrees of freedom per joint.

5.1.3 Data-Set Constructors

Several kinds of data sets may be generated or modified by the TABLE, SYSVEC, ELDATA, and ALPHA subprocessors. Examples are summarized in the following tabulation:

Subprocessor	Kind of Data Set Produced or Operated On
TABLE	TABLE.* Examples: Nodal pressures, temperatures, and TAB-produced tables.
SYSVEC	SYSVEC.* Examples: System vectors of applied forces, motions, and eigenvectors.
ELDATA	ELDATA.* Examples: Element-applied load data, e.g. pressures, temperatures, and dislocations.
ALPHA	ALPHA.* Examples: Arrays of alpha character strings used as load case titles, or to describe eigenvectors.

In the following, XXXX is any of the subprocessor names: TABLE, SYSVEC, ELDATA, or ALPHA. To initiate production of a new data set named N1 N2 n3 n4, the following language sequence is used:

```
XXXX( optional parameters): N1  N2  n3  n4 :  -- data cards ---
```

To initiate modification (updating) of an existing data set, the following language is used:

```
XXXX,  U( optional parameters): N1  N2  n3  n4: -- data cards ---
```

*See Section 2.5.

The output data set, N1 N2 n3 n4, is resident in the current data destination library defined by the last OUTLIB Statement. If omitted, N2, n3, and n4 assume default values of AUS, 1, and 1.

The following characteristics of the Oper, U (update) mode of operation are noted:

- The data set is actually overwritten in mass data storage, rather than modified and stored in a new area.
- Existing data sets must not be extended (i.e., additional blocks cannot be appended in this mode of operation).

The following example input indicates a typical application of a data set constructor. In this case SYSVEC is used to create a data set, named APPL FORC 88, comprised of two blocks. The first block defines load case 1 of static load set 88; and the second block defines case 2 of set 88.

```
@XQT AUS
  SYSVEC$                          Enter SYSVEC
  APPLIED   FORCES   88$           Name of output data set.
  CASE 1$
        I=         2,      3$      Direction 2 and 3 forces.
        J= 7,9$                    List of joint numbers.
                   7.2     7.3$    f2, f3  at joint 7.
                   8.2     8.3$       "    "      8.
                   9.2     9.3$       "    "      9.
  CASE 2$
        I= 1:  J=10:  14.6$        f1  at joint 10.
        I= 3:  J= 7:  19.2$        f3  at joint  7.
```

Many examples of SYSVEC, etc., input are given in Section 6. It is suggested that these examples be scanned briefly before reading Sections 5.1.3.1-4.

Because of the central role played by the data set constructors, it is strongly recommended that new users perform a series of test executions of them, using DCU/PRINT and/or VPRT to verify that the intended results are achieved.

The TRAN sub-activity of TABLE provides a very general method of transmitting information from one data set to another. New users should not attempt to use TRAN without advice from an experienced analyst.

5.1.3.1 TABLE. Execution begins as indicated below. If appropriate, any of the underlined words may be omitted.

TABLE, U(NI= ni, NJ= nj): N1 N2 n3 n4: -- data ---

The optional parameters NI and NJ have the same meaning as defined in Section 2.5; i.e., each block of the output data set is a rectangular matrix of dimension (ni, nj). The default value of ni is 1, and the default value of nj is the number of joints in the structure. The input defining the nth block of the output data set has the following structure:

BLOCK n

OPERATION = SUM $ or XSUM, or MULTIPLY, or DIVIDE

J= j_1: j_2: j_3: - - - $, a list of columns.*

$e^1_{j_1}$ $e^2_{j_1}$ $e^3_{j_1}$ - - - $e^{ni}_{j_1}$ $ Data record 1, applies to column j_1

$e^1_{j_2}$ $e^2_{j_2}$ $e^3_{j_2}$ - - - $e^{ni}_{j_2}$ $ Data record 2, applies to column j_2

=

=

etc.

*Loop-limit format is also permitted (see Section 2.3).

If there are fewer data records of the latter type than there are j's in the list of columns, the last record will be associated with each of the remaining j's. The meaning of the e_j^i's depends upon current arithmetic mode, as established by the OPERATION= XXXX statement. In the following, x_j^i is the element in row i and column j of block n of the output data set. The range of i is 1, 2, - - - ni, and the range of j is j_1, j_2, j_3 - - - .

Operation	Arithmetic mode
SUM	$x_j^i \leftarrow x_j^i + e_j^i$
XSUM	$x_j^i \leftarrow e_j^i$
MULTIPLY	$x_j^i \leftarrow x_j^i e_j^i$
DIVIDE	$x_j^i \leftarrow x_j^i/e_j^i$

The sequence (OPERATION=XXX ..... List of j's ..... data records for each j in the list) may be repeated as many times as necessary to produce the desired data in block n.

The OPERATION statement continues in effect, not only for the current block but for all subsequent blocks as well, until another OPERATION statement supersedes it. The default OPERATION is SUM.

The following control statement is frequently useful:

$$I= i^1, \; i^2, -- i^m \$ \text{ (each } i^k \text{ between 1 and ni)}$$

This statement indicates that the data records will have the following content:

$$e_j^{i^1}, e_j^{i^2} \; -- \; e_j^{i^m} \; \$.$$

For example:

BLOCK n: etc.: - - - $

I= 3,5: J= 7,9$

e_7^3, e_7^5: e_8^3, e_8^5: e_9^3, e_9^5\$

I= 1: J= 10: e_{10}^1\$

The I= i^1, i^2 --- statement continues in effect, not only for the current block but for all subsequent blocks as well, until another I= i^1, i^2 --- statement is encountered. Default is, in effect,

I= 1, 2, --- ni.

The following statement may immediately precede a list of j's:

$$\text{DDATA= } d_1, d_2, \text{ --- } d_m \ \$$$

When data records are being duplicated, the DDATA command has the effect of incrementing each item in the data record by the corresponding d_i. The DDATA statement will apply only to the j-list immediately following the statement.

Example:

```
I= 1,  3: DDATA= .2,  .1:  J= 2,  5: 7., 8.$
```

The foregoing statement produces the same result as:

```
I= 1,3 : J= 2,5$
```

$$7.0, 8.0 \qquad \$ \ = \ e_2^1, e_2^3$$

$$7.2, 8.1 \qquad \$ \ = \ e_3^1, e_3^3$$

$$7.4, 8.2 \qquad \$ \ = \ e_4^1, e_4^3$$

$$7.6, 8.3 \qquad \$ \ = \ e_5^1, e_5^3$$

To cause an integer constant, jshift, to be added to each j in subsequent lists of j's, the following command is used: JSHIFT= jshift. The jshift parameter will remain in effect until superseded by another JSHIFT command.

Rules for ordering control and input statements:

- No statement may precede the first BLOCK= n statement (Exception: if operation is confined to block 1, the BLOCK= 1 statement may be omitted).
- When the BLOCK= n statement appears, block n - 1 must already exist.
- No control card of any kind (OPERATION= --, I= ---, JSHIFT= ---, or DDATA= ---) may appear anywhere between the beginning of a j-list and the end of the subsequent data records $\left(\text{the } e_j^i\text{'s}\right)$.

Synonyms. If desired, the control word CASE may be used instead of BLOCK, and JOINT instead of J.

Multiple data records on a single card. . In the preceding, it has been indicated that each data record should contain exactly ni words, or m words if the statement $I = i^1, i^2 --- i^m$ is used. It is also permitted to place several such complete data records on the same card; i.e., both of the following are legal cards and have the same effect:

```
1.2, 2.9: 4.5, 2.7: 3.2, 9.$
1.2, 2.9, 4.5, 2.7, 3.2, 9.$
```

An error stop will occur if a card does not contain complete records.

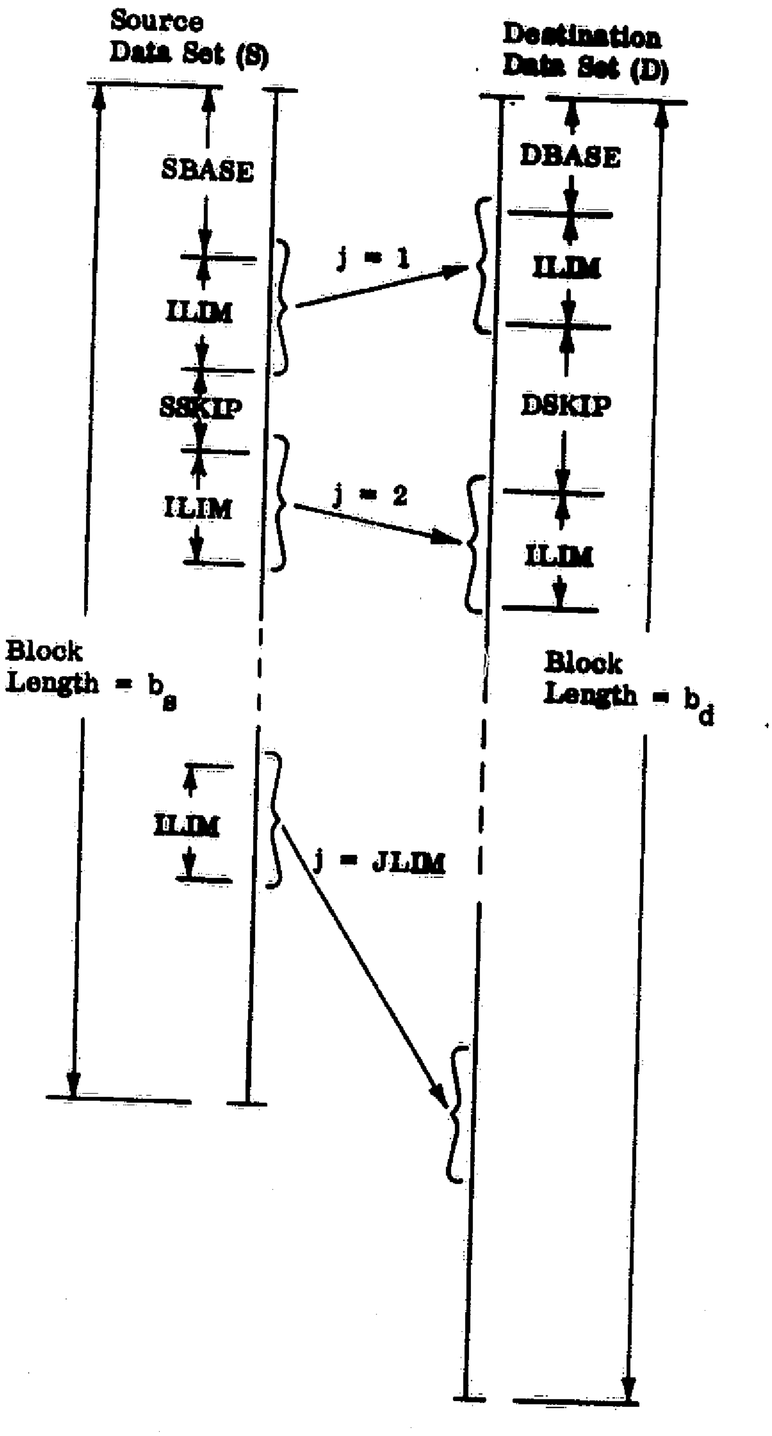
The TRANSFER Statement. This command provides a general method of extracting real data from designated areas of one data set, and adding them to data in designated areas of another data set. TRANSFER statements may appear as the first or last statement in a TABLE execution, or immediately prior to any BLOCK statement. The command form is as follows:

TRANSFER(List of arguments $P_1 = v_1, P_2 = v_2$, ---)

Any or all of the arguments summarized in the following table may be used.

Argument Name	Default Value	Meaning
SOURCE	None	Name (a single alpha word) of the source data set. May be established by a DEFINE statement.
L1	1	Successive blocks of the source data set are added to blocks L1, L1 + 1, --- L2 of the destination data set unless L1=L2.
L2	1	
ILIM	Lesser of b_s, b_d	See Fig. 5.1.3.1-1
JLIM	1	
SSKIP	0	
DSKIP	0	
SBASE	0	
DBASE	0	

The TABLE/operation statement has no meaning in TRANSFER.



Fortran equivalent, repeated for blocks L1, L1 + 1, . . . L2:

```
    IS = SBASE
    ID = DBASE

    DO 20  J = 1, JLIM
    DO 10  I = 1, ILIM
    IS = IS + 1
    ID = ID + 1
 10 D(ID) = D(ID) + S(IS)
    IS = IS + SSKIP
 20 ID = ID + DSKIP
```

Fig. 5.1.3.1-1: TRANSFER Parameters

Figure 5.1.3.1-1 illustrates the function of the TRANSFER statement for the case in which the number of blocks in the source data set is equal to the number of blocks in the destination data set (i.e., L2-L1+1).

If L1= L2, the function is similar, except that successive blocks of the source data set are transmitted to successive areas within block L1 of the destination data set.

The following are a few examples of the TRANSFER statement.

```
@XQT  AUS

DEFINE  OLDV= 2   VIBR MODE  200 1    1, 10  $     Salvage vibrational modes
                                                   from a previous analysis for
TABLE(NI=6,  NJ=1700):   VIBR  MODE  1  1    $     use as initial approximations
                                                   of modes of a modified struc-
TRANSFER( SOURCE= OLDV,   L1=1,   L2=10)     $     ture having additional joints.


DEFINE   U7=   STAT   DISP  1  1  7          $     Store node point locations of
                                                   the deformed structure in
TABLE(NI=3,  NJ=400):  DEFORMED  POSITION    $     DEFO POSI.

TRANSFER( SOURCE=  JLOC)                     $     JLOC= original position.

TRAN(SOURCE=U7, ILIM=3,  JLIM=400, SSKIP=3)  $     U7= Joint motion due to static
                                                   load case 7 of set 1.

DEFINE   X= VIBR   MODE   1  2  1  8         $     Form a single-block matrix having
                                                   8 columns, each column corres-
TABLE( NI=30,  NJ=8):  T   MODE   1  2       $     ponding to a mode.  Each column
                                                   contains all 6 motion components
TRANSFER( SOURCE=X,  SBASE=600,  ILIM=30)    $     of joints 101-105.
```

5.1.3.2 SYSVEC. SYSVEC is used specifically for creating and modifying data sets in SYSVEC format. (See Section 2.5.) SYSVEC is entered as follows:

SYSVEC: N1, N2, n3, n4 $,

or

SYSVEC, U: N1, N2, n3, n4 $

Input rules are identical to those of TABLE, except for the meaning of the command. I= i^1, i^2, i^3 --- i^m.

In SYSVEC, i^k = 1, 2, or 3 always indicates a direction-i^k displacement or force component; i^k = 4, 5, 6 indicates a rotation or moment about axis i^k-3. This is true, regardless of whether or not any joint degrees of freedom have been excluded via the START card in TAB.

If the number of degrees of freedom per joint is less than 6, then the I= i^1, i^2 --- i^m command must precede the first j-list.

5.1.3.3 ELDATA. This subprocessor is used to create and/or modify data in ELDATA form, as defined in Section 2.5. The ELDATA form is used to represent structural element temperatures, pressures, etc. To create a new data set named N1 N2 n3 n4, ELDATA is entered as follows:

```
ELDATA: N1, N2, n3 $
CASE  n4 $

     OPERATION= SUM $,   or   XSUM, etc., as in TABLE.

     Data cards establishing a list of structural elements.

     Data record applying to 1st element in list.
     Data record applying to 2nd element in list.
     Etc.
```

The above input sequence is identical to that of TABLE, except that:

- Instead of a list of columns (j-list), a list of structural elements is given in a form which will be defined subsequently.
- The fourth word of the output data set name, n4, is given in the CASE statement. In ELDATA, each separate CASE statement refers to a distinct data set whereas in TABLE the CASE statement refers to different blocks within the same data set.

The language used to establish lists of structural elements is as follows:

$$G= g_1: \; g_2: \; g_3: \; ---- \; g_{ng} \$ \quad = \text{ a list* of groups.}$$

$$E= e_1: \; e_2: \; e_3: \; ---- \; e_{ne} \$ \quad = \text{ a list* of element indexes.}$$

Where $(g_i \; e_j)$ indicates group i, element index j, the above indicates a list of structural elements in the following sequence:

$$(g_1e_1), \; (g_1e_2), \; --- \; (g_1e_{ne}), \qquad (g_2e_1), \; (g_2e_2), \; ---- \; (g_2e_{ne}), \text{ etc.}$$

If the E= --- statement is omitted, the index list defaults to all elements, in order, in each of the indicated groups. The G= --- statement is mandatory and, if omitted, will cause an error termination.

The I= ---- and DDATA= --- statements may be used in ELDATA, subject to the same rules of order, etc., as in TABLE.

N2, the second name of the data set, must be a legal element name (e.g., E21, E22, etc.). The first name must be one of several specific names (e.g., TEMP, PRESS, etc.) described elsewhere in this reference manual, in connection with the end use of the data set. ELDATA will automatically establish the correct value of ni, the number of data words per column (i.e., per structural element).

*Loop-limit format also permitted (see Section 2.3).

Any number of CASE statements are permitted within one ELDATA execution.

If existing data sets are to be modified, entry begins with the statement ELDATA, U: N1 N2 n3.

To cause an integer, gshift, to be added to each group number in subsequent lists of groups, the command GSHIFT= gshift is used. Element indexes are similarly shifted by eshift as a result of the statement ESHIFT= eshift.

5.1.3.4 ALPHA. To define a new data set or to modify an existing one (via the optional, U), ALPHA is entered as follows:

ALPHA, U: N1, N2, n3, n4 $

Each subsequent input record is in the following form:

m'--- 60-character alphanumeric string ---,

which defines the mth block of the data set. When producing a new data set, the blocks must be defined in serial order.

5.1.4 Substructure Operations

The following AUS commands are associated with substructure analysis.

```
SSID= Ssid
SSPREP( Kname, ncon)
SSM( Mname)
SSK( Kname)
```

The effect of the SSID command is to establish an alphanumeric identifier, Ssid (default is SS01), which will appear as the second word of the name of all data sets produced by the substructure subprocessors, SSPREP, SSM, and SSK, as summarized on Figure 5.1.4-1 and Table 5.1.4-1. The data sets produced by these subprocessors may be used in many ways. In the present discussion no attempt will be made to address details of possible end uses (e.g., vibrational, buckling analysis) of the output.

The terms and symbols used in Figure 5.1.4-1 and Table 5.1.4-1 are defined as follows.

Boundary nodes are any subset of joints, $(j_1, j_2, - - j_n)$, for which all motion components are declared to be NONZERO in constraint case ncon. Boundary node 1 is joint j_1, boundary node 2 is joint j_2, . . . , where $j_1 < j_2 < j_3 \cdots < j_n$.

Boundary node unit motion functions are in SYSVEC format. In the following,

m= the number of degrees of freedom per node,

n= the number of boundary nodes, and

u_k^i= motion component i of boundary node k.

There are mn such functions, each of which is a static displacement solution due to a unit value of one u_k^i, with all other boundary node motions identically equal to zero.

Fixed-boundary functions are any sequence of vectors of joint motions for which all motion components of all boundary nodes are identically equal to zero. Such functions are usually eigenvectors computed by EIG and/or static displacements computed by SSOL. In the latter category, uniform acceleration functions (i.e., static displacement fields due to uniform acceleration of the structure) are often useful in dynamic analysis. Fixed boundary functions (if any) must reside in a data set named FEF Ssid, created by the analyst via AUS/UNION.

M and K are mass and stiffness matrices of the (sub)structure, identified by the Mname and Kname arguments of SSM and SSK. Mname and Kname are the first words of the names of the data sets containing M and K. K must be in SPAR format. M may be diagonal, or in SPAR format.

All input and output data sets associated with AUS substructure operations should reside in library 1. The DEFINE, INLIB, and OUTLIB statements should not be used in any AUS execution in which these operations are performed.

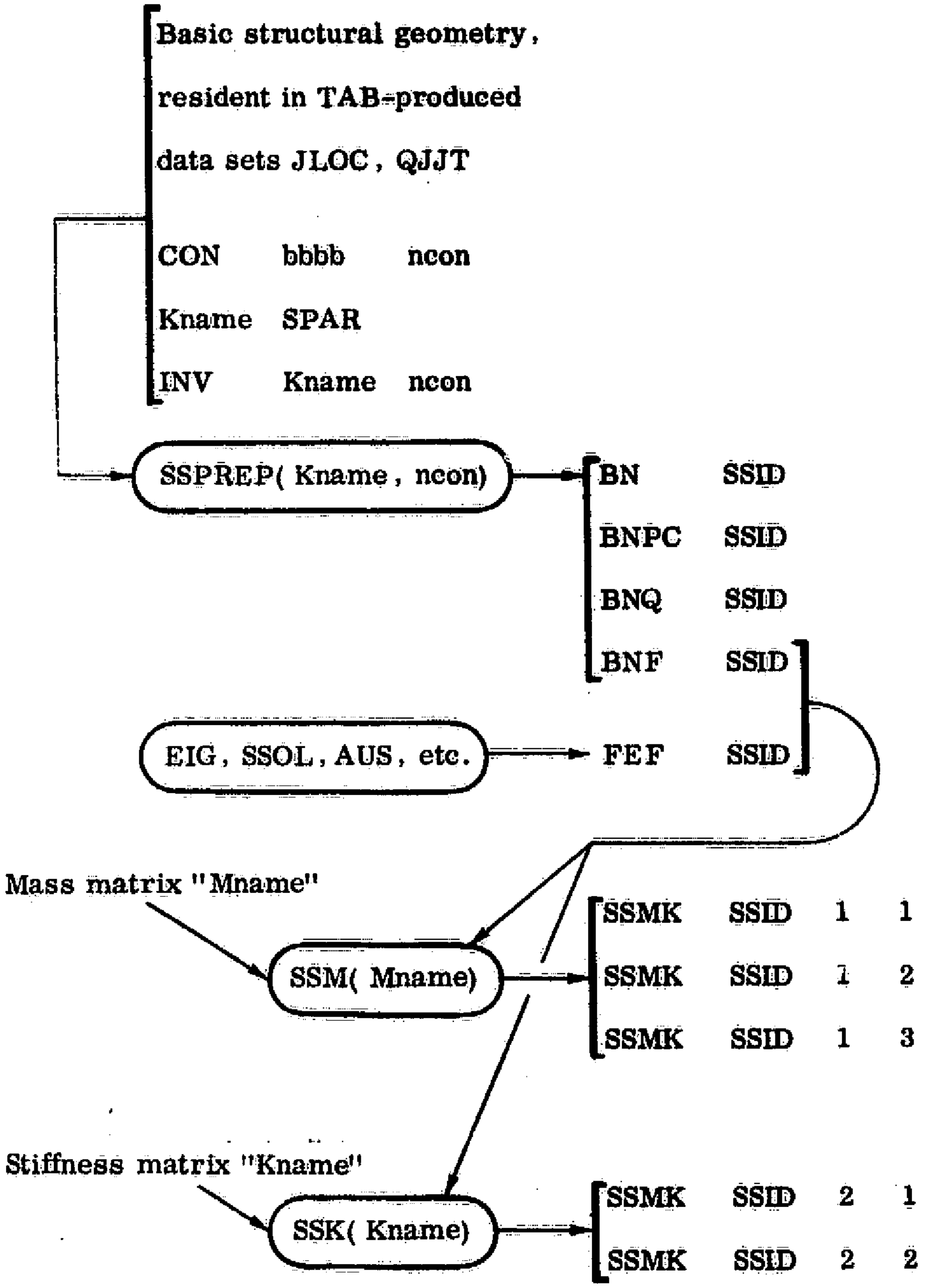


Figure 5.1.4-1 Information Flow, SSPREP, SSM, and SSK.

Table 5.1.4-1: Data Sets Associated With Substructure-Related Functions

Data Set Name				NWORDS	NJ	NI*NJ	Contents
BN	Ssid[1]	–	–	n	1	n[2]	List of boundary nodes.
BNPC	Ssid	–	–	3n	n	3n	Boundary node position coordinates.
BNQ	Ssid	–	–	9n	n	9n	Boundary node reference frame orientations.
BNF	Ssid	–	–	$\ell \cdot nx$[5]	j[3]	ℓ[4]	X, a matrix of boundary node unit motion functions.
FEF	Ssid	–	–	$\ell \cdot ny$[6]	j	ℓ	Y, a matrix of fixed-boundary functions.
SSMK	Ssid	1	1	$\overline{\overline{nx}}$[7]	1	$\overline{\overline{nx}}$	X^tMX
SSMK	Ssid	1	2	$\overline{\overline{ny}}$[8]	1	$\overline{\overline{ny}}$	Y^tMY
SSMK	Ssid	1	3	$nx \cdot ny$	ny	$nx \cdot ny$	X^tMY
SSMK	Ssid	2	1	$\overline{\overline{nx}}$	1	$\overline{\overline{nx}}$	X^tKX
SSMK	Ssid	2	2	$\overline{\overline{ny}}$	1	$\overline{\overline{ny}}$	Y^tKY

(1) Ssid= substructure identifier attached via the SSID= Ssid command.

(2) n= the number of boundary nodes.

(3) j= the number of nodes (joints) in the (sub)structure.

(4) ℓ= $j \cdot m$, where

m= the number of degrees of freedom per node, usually 6.

(5) nx= $n \cdot m$= the number of boundary node motion functions.

(6) ny= the number of fixed-boundary functions.

(7) $\overline{\overline{nx}}$= $nx \cdot (nx + 1)/2$. Symmetric array, triangular form.

(8) $\overline{\overline{ny}}$= $ny \cdot (ny + 1)/2$. Symmetric array, triangular form.

5.2 DCU – DATA COMPLEX UTILITY PROGRAM

Function – DCU performs an array of utility functions, as subsequently defined in this section. New users should review Sections 2.2 and 2.5 before reading this section.

In defining DCU command syntax, the symbol Lib will be used repeatedly to represent a SPAR library internal designation (1, 2, etc.), and Id will identify one or more data sets. Id may be in any of the three forms indicated below, unless specifically stated otherwise:

(1) The four-word data set name, MASK-filled.
Example: K SPAR is the same as
K SPAR MASK MASK.

(2) An integer, n, indicating the data set associated with sequence number n in the table of contents.

(3) Integers n, m, indicating (m-n + 1) data sets associated with sequence numbers n, n+1, - - - m.

(4) Omitted - meaning all data sets in a library.

Consider as an example, the most commonly executed command,

TOC Lib Id.

Examples:

- TOC 1 K SPAR$ Id form (1) causes a single line of the Table of Contents of library 1 to be printed, i.e. the line corresponding to the first data set named K SPAR MASK. MASK to be located.

- TOC 1 27$ Id form (2) causes printout of line 27 of the Table of Contents of library 1.

- TOC 1 32, 50$ Id form (3) causes printout of lines 32 through 50 of the Table of Contents of library 1.

- TOC 1 $ Id form (4) causes the entire Table of Contents of library 1 to be printed.

Other commands currently available in DCU are summarized below:

- DISABLE Lib Id$. Data set(s) are marked as disabled. The data set(s) are still present in the library, but cannot be accessed until they are re-enabled via the following command:

- ENABLE Lib, Id$. Only forms (2) or (3) of Id are allowed for this command.

- PRINT Lib Id\$, or

  PRINT Lib N1, N2, n3, n4, j_1, j_2, i_1, i_2, b_1, b_2. The one or more identified data sets are printed in tabular form*. Only matrix-form data sets (e.g., TABLE, SYSVEC, ELDATA, and ALPHA forms, as defined in Section 2.5) should be PRINTed. If the second form is used, the printout will be restricted to columns j_1 through j_2, rows (items) i_1 through i_2, for successive blocks b_1, $b_1 + 1$ --- b_2. Using the terminology of Section 2.5, default parameters are $j_1 = i$, $j_2 = NJ$, $i_1 = 1$, $i_2 = NI$, $b_1 = 1$, b_2 = total number of blocks in the data set.

  > NOTE
  >
  > PRINT displays data set items sequentially, so that matrices appear in transposed form.

  Data produced by processors E, EKS, TOPO, K, M, KG, and INV should not normally be named in a PRINT command.

- COPY Lib_1, Lib_2, Id\$. Copy the indicated data sets from Lib_1 to Lib_2. Disabled data sets are not copied. This is the recommended method of packing libraries.

- XCOPY Lib_1, n, Id\$. The indicated data set (in Lib) is written on ordinary sequential file n in a sequence of physical records identical to individual blocks of the data set as it resides in Lib. As an example,

* An auxiliary command, NCPL= n, controls the number of columns per printed line. Default NCPT.= 10. For teletype display, select NCPL= 5.

XCOPY 1, 5, VIBR MODE$ causes eigenvectors to be written onto a file known externally as SPAR-E (UNIVAC) or SPARLE (CDC), corresponding to n = 5. The output file will contain one physical record for each eigenvector.

- XLOAD n, Lib, nwords, nj, ninj, type, N1, N2, n3, n4 $. This command causes data from sequential file n to be loaded as a data set named N1, N2, n3, n4 in Lib. The other parameters have the same meaning as defined in Section 2.2. As an example, suppose a sequential file, SPAR-D, contains five blocks (physical records produced by direct binary writes, not unformatted Fortran writes) of real data and that each block is a matrix with 6 rows and 100 columns. To load these data into library 1 as a data set named XX YY 1 2, the following command would be used:

```
XLOAD  4,1,   3000,  100,  600,  -1,  XX  YY  1  2$
```

- REWIND n $. Used in conjunction with XCOPY, XLOAD, this command causes sequential file n to be rewound (i.e., set to starting point). It should be noted that neither XCOPY nor XLOAD rewinds sequential files either before or after the data transmission, so that one sequential file may contain many multiblock data sets.

- SCALE C_1, $C_2$$. This command may precede a COPY or XCOPY command. If it does, the output data sets will = C_1 (source) + C_2.

- CHANGE Lib Id_{old}, Id_{new}\$. This command causes the name of data set to be changed from Id_{old} to Id_{new}. Only the full 4-word name form is permitted for both Id's.

- DUPLICATE Lib_1, Lib_2\$. Lib_2 is created identical to existing Lib_1, including disabled data sets, if any.

- TWRITE Lib$. Lib is written onto tape nt (see NTAPE command). The complete library is written in physical records as large as the available core will allow.

- TREAD Lib$. Lib is read from tape nt (see NTAPE command). Available working core space must be as large as it was when the TWRITE was executed.

- NTAPE= nt$. The internal unit number of the tape to be used in the next TWRITE or TREAD command is nt (default=20). Note that logical 20 is Univac file SPAR-T, CDC file SPARLT.

- STORE Lib, ID$. Lib is stored as a data set named Id, in library nℓ (see LIBLIB command). Id may only be a full 4-word name, the first two words of which are typeless.

- RETRIEVE Lib, Id$. The data set Id is recovered from the library nℓ and constituted as library Lib.

- LIBLIB= nℓ$. The internal unit number of the library library is nℓ (default=12). Note that logical 12 is Univac file SPAR-L, CDC file SPARLL.

- TITLE Lib' - - - Alphanumeric title for Lib - - - -. The label-field title is embedded in Lib, and will be displayed at the beginning of each table of contents printout produced by a TOC command.

- STATUS Lib$. The number of library entries and the current I/O counts for Lib are printed.

- ABORT n$. To cause an error-abort if an abnormal event occurs in DCU, set ABORT=1$.

Core Requirements. Working core must be sufficient to accommodate one block of each data set transmitted through core (e.g. via COPY, XCOPY, etc.). See also the discussion of TWRITE and TREAD.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

5.3 VPRT – VECTOR PRINTER

Function. VPRT is used to edit and display information in SYSVEC format (see Section 2.5), such as static displacements, buckling and vibrational eigenvectors, reactions, equivalent nodal loading, etc.

The name convention associated with data sets in SYSVEC format is usually as follows (name = N1, N2, n3, n4):

N1, N2 = descriptive names such as VIBR MODE, STAT DISP, APPL FORC, etc.

n3 = identifies the data set as associated with static solution set n3, or eigenvector set n3, etc.

n4 = associated constraint case (if any).

The following command will cause data set N1 N2 n3 n4 to be printed (N2, n3, n4 MASK-filled, if omitted).

```
PRINT  N1 N2 n3 n4'  optional heading (60 characters)
```

If the optional heading is omitted, VPRT will supply appropriate headings for data sets bearing standard names.

If the data source library also contains an ALPHA-format data set named CASE TITLES n3 (the same n3 as the SYSVEC data set), case titles will be displayed beneath the heading line if the command TPRINT is used instead of PRINT. The usual applications of case titles are (1) to describe individual cases within a set of static loads, and (2) to describe individual eigenvectors.

If there is a constraint case* associated with the SYSVEC data set, each constrained component will be flagged with an asterisk.

If eigenvalues are associated with the data set, they will be displayed.

The following control statements may appear any number of times during a VPRT execution. Each continues in effect until superseded by another command of the same kind.

Control Statement	Meaning
LIB= m	Library m contains the data to be printed. Default Lib = 1.
VECTORS v_1, v_2	Print only vectors (blocks) v_1 through v_2. Default is all vectors.
JOINTS $j_1 : j_2 : j_3 :$ ---**	Print data only for joints j_1, j_2, j_3, ---. Default is all joints.
COMPONENTS= $i_1, i_2,$ --	Display only components i_1, i_2, ---, where each i is between 1 and 6. Default is all components.

*Contained in CON bbbb n4, residing in library 1.

**Loop-limit format, i.e., j^a, j^b, inc, also permitted (see Section 2.3). Up to 300 input records are allowed.

Control Statement (Cont.)	Meaning (Cont.)
FILTER= e_1, e_2 --- e_6.	Data for a given joint will be printed only if the absolute value of some component, v_i, exceeds e_i. All six e_i's must be given for this command to be valid. Default $e_i = 1. \times 10^{-20}$ for all six components.
FORMAT= k	Values of k from 1 to 4 select, in order, one of the following Fortran print formats: E9.3, F9.5, F9.1, E15.7. Only the first three are suitable for 72-character teletype output. Default FORMAT = 1.
LINES= n	Print n lines per page. Default LINES = 50.
HEADING= m	Any non-zero value causes headings to be repeated at the top of each page. Default HEADING = 0.

Synonyms. In the preceding commands, J may be used instead of JOINTS, and I instead of COMPONENTS.

Core Requirements. Data space must be sufficient to contain one block of any data set to be printed.

Restrictions. Data sets printed by VPRT must correspond to the TAB/START information (no. of joints, etc.) in data set JDF1, contained in Library 1. Data sets should not contain more than 300 vectors.

Examples. Several examples of VPRT output are appended to this section.

Code Release Date. Level 9, July 1975, coded by W. D. Whetstone.

VPRT Example

```
STATIC DISPLACEMENTS,
1200 KIP SHEAR LOAD, DIR. 2                                              ID=   5/ 1/  1

JOINT       1             2             3             4             5             6
   1    -.148-04       .123+00       .634-05       .153-02       .790-05       .425-02
   2     .571-01       .105+00       .737-02       .880-04      -.147-02      -.367-02
   3     .163+00       .452-01       .183-02      -.699-03      -.119-02      -.579-02
   4     .163+00      -.452-01       .173-02       .664-03      -.118-02       .581-02
   5     .571-01      -.105+00       .711-02      -.735-04      -.143-02       .367-02
   6     .147-04      -.123+00      -.633-05      -.153-02      -.790-05      -.425-02
   7    -.571-01      -.105+00      -.737-02      -.880-04       .147-02       .367-02
   8    -.163+00      -.452-01      -.183-02       .699-03       .119-02       .579-02
   9    -.163+00       .452-01      -.173-02      -.664-03       .118-02      -.581-02
  10    -.571-01       .105+00      -.711-02       .735-04       .143-02      -.367-02
  11     .147-05       .649-01       .295-04       .122-02      -.270-05       .760-03
  12     .347-01       .543-01       .102-01       .122-02      -.152-02      -.285-03
  13     .667-01       .216-01       .190-01       .616-03      -.304-02      -.671-03
  14     .667-01      -.216-01       .190-01      -.614-03      -.304-02       .663-03
  15     .348-01      -.544-01       .102-01      -.123-02      -.153-02       .286-03
  16    -.155-05      -.649-01      -.295-04      -.122-02       .270-05      -.760-03
  17    -.347-01      -.543-01      -.102-01      -.122-02       .152-02       .285-03
  18    -.667-01      -.216-01      -.190-01      -.616-03       .304-02       .671-03
  19    -.667-01       .216-01      -.190-01       .614-03       .304-02      -.663-03
  20    -.348-01       .544-01      -.102-01       .123-02       .153-02      -.286-03
  21     .413-04       .238-01      -.878-05       .548-03       .250-06      -.494-03
  22     .192-01       .190-01       .580-02       .563-03      -.906-03      -.448-03
  23     .316-01       .722-02       .110-01       .291-03      -.170-02      -.201-03
  24     .316-01      -.722-02       .110-01      -.288-03      -.170-02       .205-03
  25     .191-01      -.191-01       .583-02      -.564-03      -.909-03       .452-03
  26    -.413-04      -.238-01       .878-05      -.546-03      -.249-06       .494-03
  27    -.192-01      -.190-01      -.580-02      -.563-03       .906-03       .448-03
  28    -.316-01      -.722-02      -.110-01      -.291-03       .170-02       .201-03
  29    -.316-01       .722-02      -.110-01       .288-03       .170-02      -.205-03
  30    -.191-01       .191-01      -.583-02       .564-03       .909-03      -.452-03
  31     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  32     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  33     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  34     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  35     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  36     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  37     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  38     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  39     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
  40     .000    *     .000    *     .000    *     .000    *     .000    *     .000    *
```

```
STATIC REACTIONS, FORCE ERRORS.
1200 KIP SHEAR LOAD, DIR. 2                                                      ID=   5/ 1/  1

JOINT       1            2             3             4             5             6
   1    .198-02     -.238+00       .747-02      -.284-02       .897-03       .973-02
   2    .302-01     -.910-01       .989-02       .289-02       .628-03       .632-02
   3   -.879-01      .433-01      -.234-01       .386-02      -.766-03       .220-01
   4   -.254-01     -.231-01      -.151-01      -.192-02      -.218-02       .438-01
   5   -.481-02      .366+00      -.436-02       .410-04       .196-03      -.662-01
   6   -.578-01      .131+00      -.181-01       .216-02      -.167-02      -.511-01
   7   -.470-01     -.185+00      -.219-01      -.374-02      -.914-03      -.251-01
   8    .625-01      .915-01      -.596-02       .924-03       .642-03       .161-01
   9   -.176-01     -.372-01      -.327-02       .380-03       .223-02       .868-01
  10    .552-01     -.372+00       .211-01      -.740-02      -.466-03       .111+00
  11    .744-02     -.109+00       .181-01       .129-02      -.510-04       .393-02
  12    .570-02     -.137+00       .158-01      -.540-02       .665-04       .122-01
  13   -.696-03      .317-01      -.449-02      -.106-02      -.167-03       .200-01
  14   -.479-01      .447-02      -.788-01       .392-02      -.486-02       .264-01
  15   -.541-01      .141+00      -.642-01       .689-02       .340-03       .182-01
  16   -.760-01      .235+00      -.662-01       .529-02       .190-02      -.134-01
  17   -.136-01      .160+00       .183-01       .354-02      -.834-03      -.282-01
  18    .219-01      .330-01       .875-01      -.217-03      -.283-02      -.287-01
  19    .481-01     -.339-01       .115+00      -.618-02       .665-03      -.768-02
  20    .647-01     -.261+00       .507-01      -.416-02      -.437-02      -.261-03
  21   -.426-02     -.970-02      -.960-02       .208-02       .588-04      -.214-03
  22   -.701-02      .659-02      -.179-01      -.161-03      -.607-03      -.488-03
  23   -.180-01      .322-02      -.330-01      -.773-03      -.235-03      -.403-02
  24   -.181-01     -.915-02      -.370-01      -.486-03      -.147-02      -.385-02
  25    .195-02     -.155-02      -.187-01      -.461-03      -.163-02      -.458-02
  26   -.681-02     -.668-02      -.212-01       .340-03      -.751-03       .702-03
  27    .298-02      .246-01       .156-01       .172-02      -.227-03       .453-02
  28    .151-01     -.161-01       .787-01       .258-03       .134-02      -.201-02
  29    .270-01      .615-02       .483-01      -.160-02       .965-03       .385-02
  30    .126-01     -.694-01       .294-01      -.204-02      -.334-03      -.171-02
  31    .136+03*    -.960+05*     -.256+02*     -.270+03*      .101+02*      .581+03*
  32   -.806+05*    -.758+05*     -.201+06*     -.212+03*      .279+04*      .428+03*
  33   -.131+06*    -.270+05*     -.333+06*     -.776+02*      .441+04*      .139+03*
  34   -.131+06*     .274+05*     -.333+06*      .782+02*      .440+04*     -.141+03*
  35   -.808+05*     .758+05*     -.201+06*      .213+03*      .277+04*     -.429+03*
  36   -.136+03*     .960+05*      .258+02*      .270+03*     -.101+02*     -.581+03*
  37    .806+05*     .758+05*      .201+06*      .212+03*     -.279+04*     -.428+03*
  38    .131+06*     .270+05*      .333+06*      .776+02*     -.441+04*     -.139+03*
  39    .131+06*    -.274+05*      .333+06*     -.782+02*     -.440+04*      .141+03*
  40    .808+05*    -.758+05*      .201+06*     -.213+03*     -.277+04*      .429+03*
```

VPRT Example

```
VIBRATIONAL MODE.
EIGENVALUE=     .3313871+08,  FREQ=  916.1950 HZ                    IL=    6/ 1/  1
RING STIFFENED CONE. N=2 DEFORMATION

JOINT       1            2            3            4            5            6
  1     .568+00     -.119+01     -.103+00      .862-01      .140-01      .344+00
  2    -.224+01     -.619+00      .413+00      .447-01     -.563-01      .179+00
  3    -.196+01      .806+00      .358+00     -.585-01     -.489-01     -.235+00
  4     .103+01      .112+01     -.191+00     -.809-01      .261-01     -.323+00
  5     .260+01     -.114+00     -.476+00      .850-02      .650-01      .330-01
  6     .571+00     -.119+01     -.103+00      .862-01      .141-01      .344+00
  7    -.224+01     -.621+00      .413+00      .448-01     -.564-01      .180+00
  8    -.196+01      .804+00      .358+00     -.586-01     -.489-01     -.235+00
  9     .103+01      .112+01     -.192+00     -.809-01      .262-01     -.323+00
 10     .259+01     -.114+00     -.476+00      .862-02      .650-01      .331-01
 11     .494+00     -.952+00     -.755-01      .474-01      .497-02      .241+00
 12    -.195+01     -.496+00      .298+00      .247-01     -.196-01      .126+00
 13    -.170+01      .646+00      .260+00     -.320-01     -.172-01     -.163+00
 14     .900+00      .897+00     -.137+00     -.446-01      .898-02     -.226+00
 15     .226+01     -.911-01     -.345+00      .444-02      .228-01      .229-01
 16     .499+00     -.953+00     -.764-01      .475-01      .510-02      .241+00
 17    -.196+01     -.498+00      .298+00      .249-01     -.197-01      .126+00
 18    -.171+01      .646+00      .260+00     -.322-01     -.172-01     -.163+00
 19     .902+00      .896+00     -.138+00     -.447-01      .913-02     -.227+00
 20     .226+01     -.917-01     -.345+00      .461-02      .228-01      .234-01
 21     .259+00     -.443+00     -.210-01      .111-01     -.559-02      .106+00
 22    -.103+01     -.231+00      .829-01      .580-02      .221-01      .552-01
 23    -.894+00      .301+00      .724-01     -.752-02      .193-01     -.717-01
 24     .474+00      .417+00     -.383-01     -.104-01     -.102-01     -.996-01
 25     .119+01     -.419-01     -.959-01      .106-02     -.256-01      .102-01
 26     .261+00     -.443+00     -.212-01      .111-01     -.563-02      .106+00
 27    -.103+01     -.232+00      .831-01      .579-02      .221-01      .553-01
 28    -.895+00      .300+00      .722-01     -.754-02      .193-01     -.719-01
 29     .474+00      .417+00     -.384-01     -.104-01     -.102-01     -.995-01
 30     .119+01     -.422-01     -.958-01      .107-02     -.256-01      .103-01
 31     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 32     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 33     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 34     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 35     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 36     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 37     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 38     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 39     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
 40     .000    *    .000    *    .000    *    .000    *    .000    *    .000    *
```

```
   BUCKLING MODE,
 EIGENVALUE=     .1390661+02                                                      ID=    7/ 1/ 10
 1ST BUCK MODE CORRESPONDING TO 1000 KIP COMPRESSIVE LOAD

 JOINT        1               2               3               4               5               6
    1    -.530-06        .863-06         .382-06        -.857-06        -.129-03        -.311-05
    2    -.463-06        .801-06         .371-06        -.850-06        -.129-03        -.307-05
    3    -.666-06        .777-06         .407-06        -.849-06        -.129-03        -.306-05
    4    -.719-06        .830-06         .415-06        -.856-06        -.129-03        -.310-05
    5    -.603-06        .862-06         .393-06        -.856-06        -.129-03        -.310-05
    6    -.562-06        .846-06         .392-06        -.856-06        -.129-03        -.310-05
    7    -.516-06        .806-06         .382-06        -.857-06        -.129-03        -.310-05
    8    -.554-06        .767-06         .387-06        -.853-06        -.129-03        -.308-05
    9    -.697-06        .786-06         .406-06        -.853-06        -.129-03        -.307-05
   10    -.706-06        .850-06         .407-06        -.856-06        -.129-03        -.310-05
   11     .668-06        .139-05        -.835-07         .146-05         .418-03         .575-05
   12     .685-06        .138-05        -.812-07         .146-05         .418-03         .575-05
   13     .681-06        .138-05        -.108-06         .146-05         .418-03         .576-05
   14     .660-06        .138-05        -.115-06         .146-05         .418-03         .576-05
   15     .681-06        .140-05        -.111-06         .146-05         .418-03         .575-05
   16     .739-06        .138-05        -.119-06         .146-05         .418-03         .575-05
   17     .727-06        .135-05        -.107-06         .146-05         .418-03         .576-05
   18     .648-06        .134-05        -.893-07         .146-05         .418-03         .576-05
   19     .627-06        .136-05        -.970-07         .146-05         .418-03         .576-05
   20     .648-06        .138-05        -.101-06         .145-05         .418-03         .575-05
   21     .275-05       -.167-06        -.100-05         .146-05         .418-03         .575-05
   22     .273-05       -.164-06        -.994-06        -.146-05        -.382-03        -.427-05
   23     .273-05       -.161-06        -.100-05        -.146-05        -.382-03        -.427-05
   24     .274-05       -.162-06        -.101-05        -.146-05        -.382-03        -.427-05
   25     .274-05       -.163-06        -.101-05        -.146-05        -.382-03        -.427-05
   26     .274-05       -.167-06        -.100-05        -.146-05        -.382-03        -.426-05
   27     .274-05       -.171-06        -.100-05        -.146-05        -.382-03        -.426-05
   28     .275-05       -.175-06        -.101-05        -.147-05        -.382-03        -.427-05
   29     .273-05       -.173-06        -.101-05        -.147-05        -.382-03        -.427-05
   30     .274-05       -.167-06        -.101-05        -.146-05        -.382-03        -.427-05
   31     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   32     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   33     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   34     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   35     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   36     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   37     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   38     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   39     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
   40     .000     *     .000     *     .000     *     .000     *     .000     *     .000     *
```

```
INERTIA FORCES ASSOCIATED WITH VIBR MODE 1                              ID=   1/ 1/  1

JOINT        1            2            3
    1     .462-02     -.964-02     -.836-03
    2    -.182-01     -.503-02      .335-02
    3    -.159-01      .655-02      .291-02
    4     .841-02      .908-02     -.155-02
    5     .211-01     -.923-03     -.387-02
    6     .464-02     -.965-02     -.839-03
    7    -.182-01     -.504-02      .335-02
    8    -.159-01      .653-02      .291-02
    9     .840-02      .908-02     -.156-02
   10     .211-01     -.923-03     -.387-02
   11     .781-02     -.150-01     -.119-02
   12    -.309-01     -.783-02      .470-02
   13    -.269-01      .102-01      .411-02
   14     .142-01      .142-01     -.216-02
   15     .358-01     -.144-02     -.545-02
   16     .788-02     -.151-01     -.121-02
   17    -.309-01     - 787-02      .472-02
   18    -.270-01      .102-01      .411-02
   19     .143-01      .142-01     -.218-02
   20     .357-01     -.145-02     -.545-02
   21     .512-02     -.874-02     -.415-03
   22    -.203-01     -.456-02      .164-02
   23    -.177-01      .594-02      .143-02
   24     .936-02      .825-02     -.757-03
   25     .235-01     -.829-03     -.190-02
   26     .517-02     -.876-02     -.418-03
   27    -.203-01     -.458-02      .164-02
   28    -.177-01      .593-02      .143-02
   29     .937-02      .824-02     -.758-03
   30     .234-01     -.833-03     -.189-02
   31     .000    *    .000    *    .000    *
   32     .000    *    .000    *    .000    *
   33     .000    *    .000    *    .000    *
   34     .000    *    .000    *    .000    *
   35     .000    *    .000    *    .000    *
   36     .000    *    .000    *    .000    *
   37     .000    *    .000    *    .000    *
   38     .000    *    .000    *    .000    *
   39     .000    *    .000    *    .000    *
   40     .000    *    .000    *    .000    *
```

Section 6

STATIC SOLUTIONS

As illustrated in Fig. 6-1, the normal execution sequence for static solutions is as follows:

- Subprocessors of AUS are used to form tables of applied loading data in any or all of the following categories:

  (1) Point forces and moments acting on joints.

  (2) Directly specified joint motion components.

  (3) Inertial loading.

  (4) Nodal (joint) temperatures.

  (5) Nodal (joint) pressures.

  (6) Thermal loading defined for individual structural elements. Transverse temperature gradients are permitted in beam elements.

  (7) Dislocational (initial mismatch) loading, defined for individual elements.

  (8) Pressure loading defined for individual elements.

- If any of the last five of the above types of loading are present, EQNF is executed to compute the corresponding equivalent nodal (joint) loading, and to determine initial strains subsequently needed for stress computation.

- SSOL is executed to compute static displacements and reactions.
- GSF is executed to compute stresses.
- PSF is executed to print stresses.
- VPRT is executed to print displacements and reactions.

The following terminology will be used in explaining the data sets involved in the static solution process. Static loadings are arranged in _sets_. There may be any number of sets, and each set may contain any number of _cases_. Users may select any set/case arrangements they find convenient. Table 6-1 lists the names of the data sets involved in the static solution process. In Table 6-1, and throughout the remainder of this section, the following definitions apply:

iset = a positive integer identifying the _load set_

icase = a positive integer between 1 and _ncases_, identifying a load case within set _iset_

ncon = the constraint case associated with the factored stiffness matrix used in SSOL to obtain displacement solutions. (See INV, Section 4.)

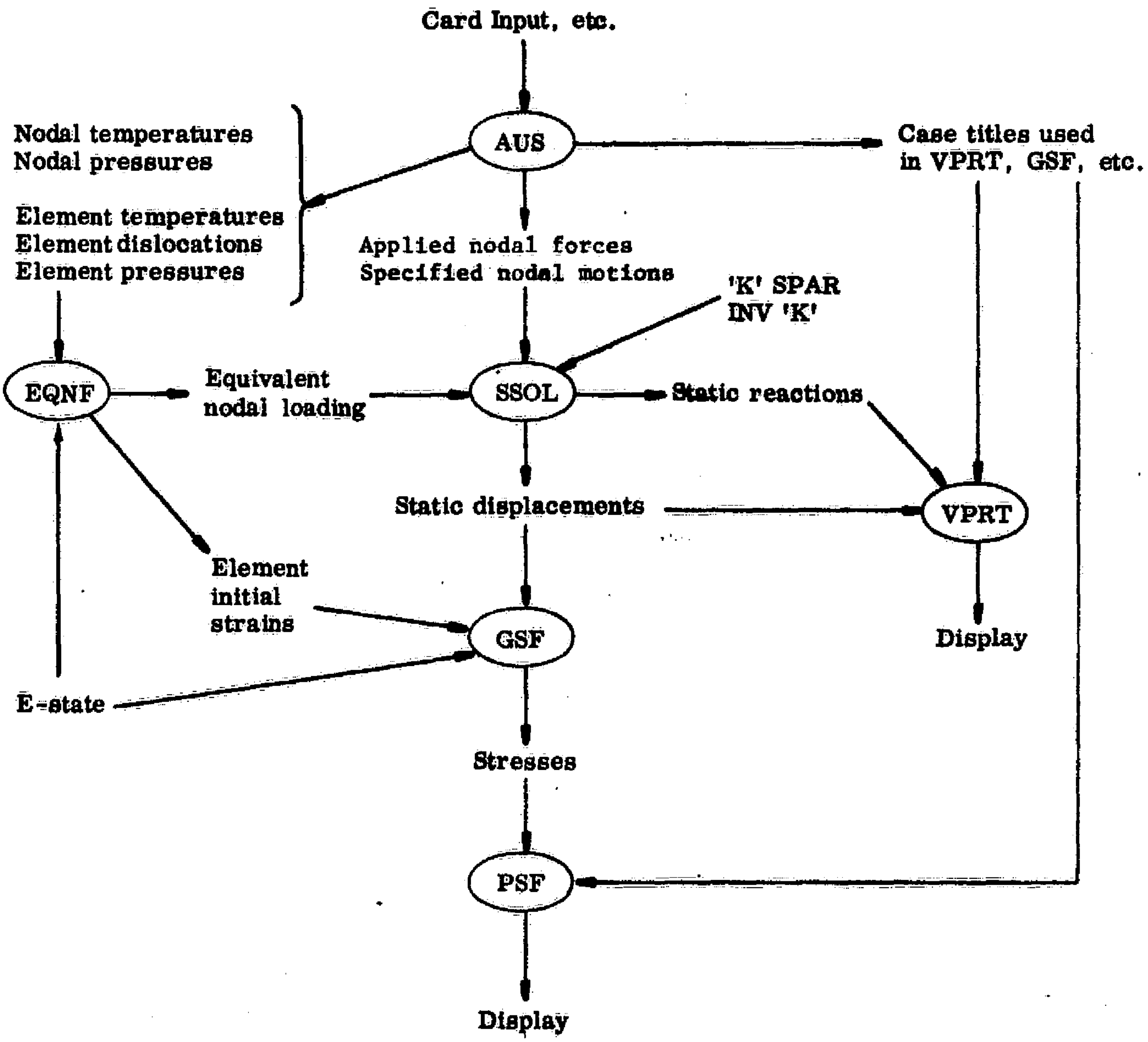


Fig. 6-1 The Static Solution Process

Table 6-1 Summary of Data Sets Involved in Static Solutions

Data Set Name				Main Source-Destination Programs	Data Form	Contents
CASE	TITL	iset		AUS-Misc.	ALPHA[1]	Case title for set iset.
APPL	FORC	iset		AUS-SSOL	SYSVEC[1]	Applied forces and moments (at joints).
APPL	MOTI	iset		AUS-SSOL	SYSVEC[1]	Applied motions (at joints).
NODA	TEMP	iset		AUS-EQNF	TABLE[1]	Nodal temperatures.
NODA	PRES	iset		AUS-EQNF	TABLE[1]	Nodal pressures.
TEMP	Eij[3]	iset	icase	AUS-EQNF	ELDATA[2]	Element temperatures.
DISL	Eij[3]	iset	icase	AUS-EQNF	ELDATA[2]	Element dislocations.
PRES	Eij[3]	iset	icase	AUS-EQNF	ELDATA[2]	Element pressures.
EQNF	FORC	iset	icase	EQNF-SSOL	SYSVEC[2]	Equivalent nodal forces.
IS	Eij[3]	iset	icase	EQNF-GSF	ELDATA[2]	Initial strains.
STAT	DISP	iset	ncon	SSOL-GSF	SYSVEC	Static displacements (of joints)
STAT	REAC	iset	ncon	SSOL-Misc.	SYSVEC	Static reactions and error forces.
STRS	Eij[3]	iset	icase	GSF-PSF	Special	Element internal loads, stresses, etc.

(1) Block 1 = Case 1, Block 2 = Case 2, etc.
(2) One case per data set.
(3) Element type identifier (e.g., E21, E22, etc.)

6.1 APPLIED LOAD INPUT

Input requirements for each category of applied loading are detailed below. The data forms SYSVEC, TABLE, and ELDATA are explained in Section 2.5.

6.1.1 Point Forces and Moments Acting on Joints

These data must reside in a SYSVEC format data set named APPL FORC iset. Case 1 resides in block 1, case 2 in block 2, etc.

In the following example, set 20 contains two cases:

```
@XQT AUS
      SYSVEC:  APPLIED FORCES 20
      CASE 1
            I= 2 : J= 4, 10 : 420.$      Direction 2, joints 4 through 10,
                                         420 lb.
            I= 1 : J= 7 : 3500.$         Direction 1, joint 7, 3500 lb.
      CASE 2
            I= 1, 2, 3 : J= 7, 9$
            17., 27., 37.$               Direction 1,2,3 forces at joint 7.
            18., 28., 38.$               Direction 1,2,3 forces at joint 8.
            19., 29., 39.$               Direction 1,2,3 forces at joint 9.
```

6.1.2 Specified Joint Motions

These data must reside in a SYSVEC format data set named APPL MOTI iset. Case 1 resides in block 1, case 2 in block 2, etc.

In the following example, set 14 contains two cases.

```
@XQT AUS
     SYSVEC: APPLIED MOTIONS 14

     CASE 1
          I= 2 : J= 110 : -4.2$        Direction-2 displacement of
                                       joint 110 is -4.2.
     CASE 2
          I= 1 : J= 2 , 8 : 3.2$       Direction-1 displacement of
                                       joints 2 through 8 = 3.2.
```

Specified motion is permitted only for components declared NONZERO in the constraint case associated with the current solution (see the CON subprocessor of TAB). Motions specified for components not declared NONZERO are ignored, and no error message is produced.

6.1.3 Inertial Loading

A very general capability is provided in AUS for computing equivalent nodal forces corresponding to inertial loading (elastic, rigid body, or both), utilizing any appropriate system mass matrix. Such loading may be included in the APPL FORC iset data set via the AUS/SUM command.

For example, assume that the following data sets are already known in AUS:

F = vectors of directly applied nodal forces.

X2 = vectors of nodal accelerations (e.g., rigid body motion defined via the RIGID command, or combined rigid body and elastic motion).

M = system mass matrix of any kind.

The following procedure would store the inertial loading in MX2, and the combined direct and inertial loads in APPLIED FORCES 27$.

```
@XQT AUS
         -
         -
         MX2= PRODUCT(M, X2)
         APPLIED FORCES 27= SUM( F, MX2)
```

6.1.4 Nodal Temperatures

Nodal temperatures are in a TABLE format data set named NODA TEMP iset. Case 1 resides in block 1, case 2, in block 2, etc. The block length is equal to the total number of joints in the structure. The structure is stress-free at a uniform temperature of zero. (See Section 6.1.6.1 for further information concerning interpretation of the temperature data.)

The following example contains three cases:

```
@XQT AUS
      TABLE:  NODAL TEMPERATURES iset

      CASE 1
            J=  1 , 4 : 10.$          Joints 1 through 4, temperature = 10 .
      CASE 2
            J=  1 : 14.$              Joint 1, temperature = 14 .
            J=  10 : 27.$             Joint 10, temperature = 27 .
      CASE 3
            J=  4 : 44.$              Joint 4, temperature = 44 .
```

6.1.5 Nodal Pressures

Nodal pressures are in a TABLE format data set named NODA PRES iset. Case 1 resides in block 1, case 2 in block 2, etc. The block length is equal to the total number of joints in the structure. The direction of action of nodal pressure on individual 3- and 4- node elements is established by the NREF statement in ELD. If NREF= 1, positive pressure acts in the +3 direction of the element reference frame. If NREF= -1, positive pressure acts in the opposite direction. If NREF= 0, the pressure loading does not act on the element. (See Section 6.1.6.3 for further information.)

The following example defines load set 9, containing two cases:

```
@XQT AUS
      TABLE: NODAL PRESSURES 9

      CASE 1
            J= 1, 6: 1.3$          Pressure at joints 1 through 6 is 1.3.
      CASE 2
            J= 2, 10: 4.2$         Pressure at joints 2 through 10 is 4.2.
            J= 20, 30: 5.0$        Pressure at joints 20 through 30 is 5.0.
```

6.1.6 Loading Defined For Individual Elements

Applied load data defined for individual structural elements resides in ELDATA format data sets (see Section 2.5) with the following names:

Name				Type of Loading
TEMP	Eij	iset	icase	Thermal
DISL	Eij	iset	icase	Dislocational (initial mismatch)
PRES	Eij	iset	icase	Pressure

In the above list, Eij is any valid structural element type (e.g., E21, E33, E41, etc.). Note that each of these data sets corresponds to a single load case, icase, within set iset. For example, suppose that, in load set 4, case 7, thermal loads are applied to type E21 and E43 elements, and pressure loading is applied to type E43 and E33 elements. The names of the corresponding data sets would be as follows:

TEMP E21 4 7

TEMP E43 4 7

PRES E43 4 7

PRES E33 4 7

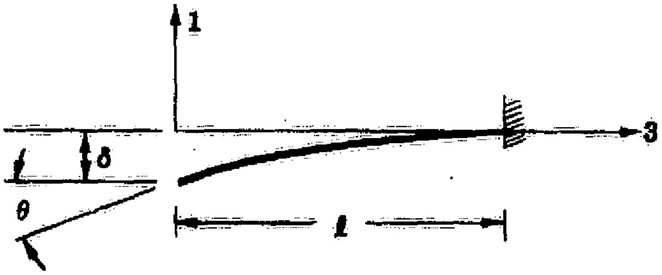
Each of the data sets contains an entry (column of data) for each structural element of the indicated type. Individual entry details for each class of loading are defined in Sections 6.1.6.1 through 6.1.6.3.

6.1.6.1 Temperatures. The content of each entry (data column) within a TEMP Eij <u>iset</u> <u>icase</u> data set is described in this section.

- For E21, E22, E23, and E24 elements, each entry contains the following three words:

$$T_a,\ T_1',\ T_2'$$

T_a is the average temperature of the element, and T_1' and T_2' are transverse gradients in directions 1 and 2, respectively, of the element reference frame. The following sketch shows how a beam would deform due to T_1', if the beam origin were free and the terminus fixed. The thermal expansion coefficient is α.



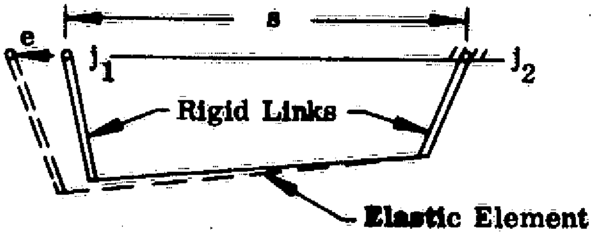
$$\theta = \alpha T_1' \ell$$

$$\delta = \alpha T_1' \frac{\ell^2}{2}$$

T_1' and T_2' produce no deformation in rigid links.

The following sketch shows how the beam, together with the rigid links (if any) by which it is attached to joints j_1 and j_2, would deform due to

T_a, if it were fixed at j_2 and disconnected from j_1. T_1 and T_2 are the nodal temperatures (if any, from block icase of NODA TEMP iset) at joints j_1 and j_2, respectively.



$$e = \alpha s \, [\, T_a + \tfrac{1}{2}(T_1 + T_2)\,]$$

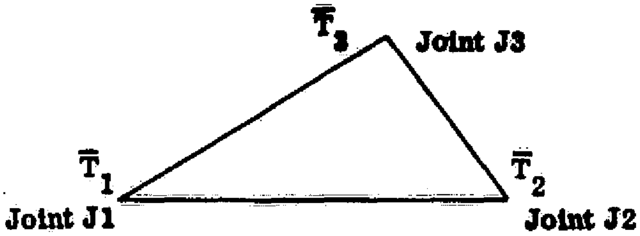
Example. Cases 5 and 7 of set 48, type E21 elements.

@XQT AUS	
–	
–	
ELDATA: TEMP E21 48	
CASE 5	
G= 4: E= 10, 20$	Elements 10 through 20 of group 4.
100., 10., 20.$	$T_a = 100$, $T_1' = 10.$, $T_2' = 20.$
CASE 7	
G= 4: E= 50$	Element 50 of group 4.
22., 1.5, 2.7$	$T_a = 22$, $T_1' = 1.5$, $T_2' = 2.7.$
I= 1$	Following data records contain T_a only.
G= 4: E= 1, 5$	
1.2, 1.3, 1.4, 1.5, 1.6$	T_a's for elements 1 – 5 of group 4.

- For E31 and E33 elements, each entry contains three elements:

$$\bar{\bar{T}}_1, \ \bar{\bar{T}}_2, \text{ and } \bar{\bar{T}}_3$$

The $\bar{\bar{T}}_i$'s are temperatures at the element corners, as shown below.



In the above, J1, J2, and J3 have the same meaning as defined in ELD. Where T_{J1}, T_{J2}, and T_{J3} are the nodal temperatures (if any, from block icase of NODA TEMP iset), the total effective corner temperatures are $\bar{\bar{T}}_1 + T_{J1}$, $\bar{\bar{T}}_2 + T_{J2}$, and $\bar{\bar{T}}_3 + T_{J3}$. The temperature distribution within the element is assumed to be linear .

Example. Case 19 of load set 6, type E33 elements.

```
@XQT  AUS
          ELDATA:  TEMP  E33  6
          -
          -
          CASE 19
                G= 2:  E= 10:  4.5,  6.2,  9.4$
                G= 9:  E= 92:  3.7,  6.8,  9.9$
```

- For E41, E43, and E44 elements, each entry contains four elements:

$$\bar{T}_1, \ \bar{T}_2, \ \bar{T}_3, \ \text{and} \ \bar{T}_4$$

The $\bar{T}_i$'s have the same meaning as for the triangular elements. The temperature distribution in 4-node elements is assumed to be linear, resulting in stress-free deformation of the decoupled element. An averaging procedure is used to determine a linear distribution approximating the given $\bar{T}_i$'s. Element meshes should be made fine enough to support this assumption.

- TEMP loading is not defined for E25, E32, or E42 elements.

6.1.6.2 Dislocations (initial strains). Dislocational loading, which is similar to thermal loading, is used to describe situations in which element strains are not zero in the null structure. The term "null structure" indicates the state in which all joints are held motionless. Dislocations are the deformations an element would undergo if disconnected from the null structure, allowing it to assume a strain-free state.

The content of each entry (data column) within a DISL Eij iset icase data set is described below.

- For E21, E22, E23, E24, and E25 elements, each entry contains six words:

  $$d_1, d_2, d_3, r_1, r_2, r_3$$

  These quantities have the following interpretation: the d's and r's are displacements and rotations of the origin, relative to a reference frame, parallel to the member reference frame, embedded in the terminus.

  For example, the thermal (TEMP) loading defined for the E21, etc., elements could also be modeled as dislocational loading if the following values of the d's and r's were used:

  $$d_1 = -\alpha T_1' \frac{\ell^2}{2} \quad , \quad d_2 = -\alpha T_2' \frac{\ell^2}{2} \quad , \quad d_3 = -\alpha T_a \ell$$

  $$r_1 = -\alpha T_2' \ell \quad , \quad r_2 = \alpha T_1' \ell \quad , \quad r_3 = 0 .$$

Example Input. **Case 7 of set 4, E21 elements:**

```
@XQT AUS

     ELDATA: DISL  E21  4


     CASE 7

          G= 3 : E=  7$                             Element 7 of group 3.

          1.2, 1.5, 1.1, .01, .017, .18$            d1, d2, d3, r1, r2, r3

          I= 1, 3, 6$                               Identify  d1, d3, r3

          G= 4:  E= 2, 3$

          1.1, 4.2, .00817$                         Element 2, group 4.

          3.2, 6.7, .00903$                         Element 3, group 4.
```

- For E31 and E33 elements, each entry contains three words, as follows:

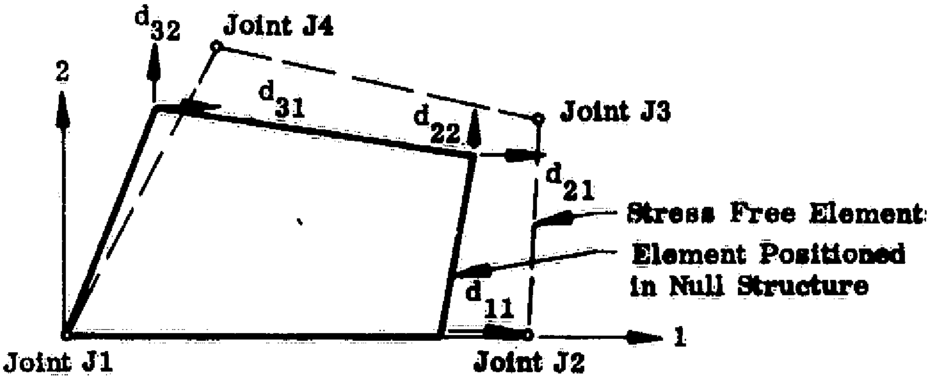
$$d_{11}, d_{21}, d_{22}$$

The d_{ij}'s have the same meaning as defined subsequently for the E41, etc., elements.

- For E41, E43, and E44 elements, each entry contains five words:

$$d_{11},\ d_{21},\ d_{22},\ d_{31},\ d_{32}.$$

As shown below, d_{ij} is the direction j displacement of node i. The Ji's are the connected joints, in the order given in the element definition (see ELD).



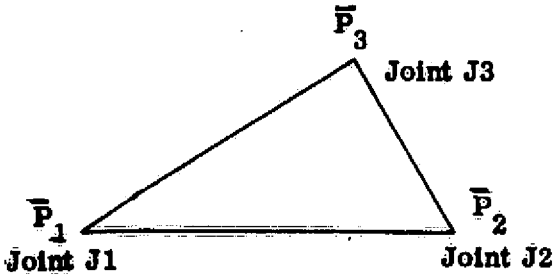
- Dislocational loading is not defined for E32 and E42 elements.

6.1.6.3 Pressure. The content of each entry (data column) within a PRES Eij <u>iset</u> <u>icase</u> data set is described below.

- For E31, E32, and E33 elements, each entry contains three words, as follows:

$$\bar{P}_1, \; \bar{P}_2, \; \bar{P}_3$$

The $\bar{P}_i$'s are pressures at the corners of the element, as shown below. The Ji's have the same meaning as defined in ELD.



It is assumed that pressure varies linearly over the surface of the element. The 3-axis of the element reference frame is the direction of action of positive $\bar{P}_i$'s.

The NREF statement in ELD has no effect on the $\bar{P}_i$'s. The reason for this convention is to provide a means of introducing pressure loading when NREF has been set equal to zero to inhibit nodal pressure loading as defined in Section 6.1.5.

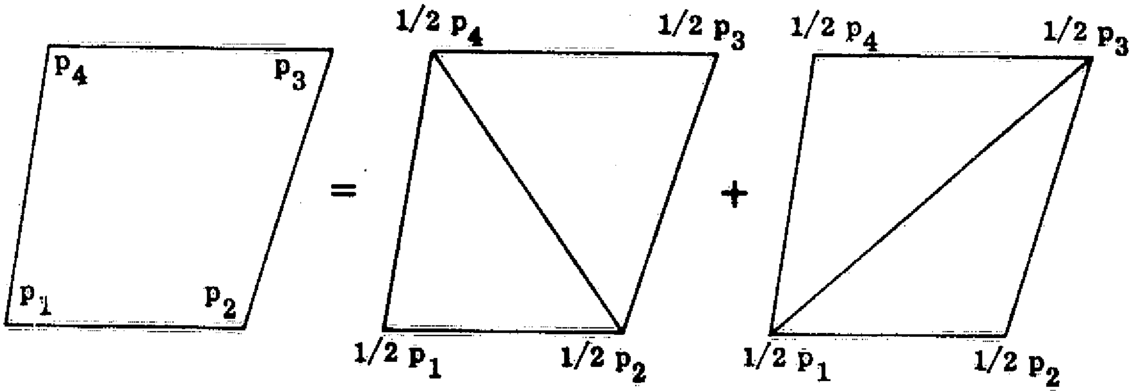
- For E41, E42, E43, and E44 elements, each entry contains four words:

$$\overline{P}_1, \overline{P}_2, \overline{P}_3, \overline{P}_4$$

The $\overline{P}_i$'s have the same meaning as defined previously for triangular elements. For purposes of determining equivalent loading, the following assumptions are made:

(1) The element is comprised of four independent triangular elements shown below.

(2) The pressure distribution in each of the four triangles is linear. The p_i's are total corner pressures.



6.2 EQNF – EQUIVALENT NODAL FORCE GENERATOR

Function. As indicated in the preceding section, EQNF computes equivalent joint loading associated with thermal, dislocational, and pressure loading. For thermal and dislocational loading, EQNF also computes the corresponding initial strain data necessaiy for subsequent stress computations.

Figure 6-1 shows the relationship of EQNF to other processors, and Table 6-1 indicates the names of all input and output data sets.

RESET Controls

Name	Default Value	Meaning
SET	1	Load set identifier (iset).
L1	1	First load case.
L2	1	Last load case (cases L1 through L2 will be processed.)
INLIB	1	Source library, all applied load data.
FEFLIB	1	Destination library, equivalent nodal force data.
ISLIB	1	Destination library for initial strain data (if any).
ISBL	896	Block length, initial strain data output.

Core Requirements.

Where JT is the number of joints in the structure, the data space requirements will not substantially exceed the sum of the following:

6 * JT

+ JT (if nodal temperatures are involved)

+ JT (if nodal pressures are involved)

+ block length of initial strain output (if any)

+ L_{max} , where

L_{max} is the largest value of L, defined below, occurring for any kind of element, Eij:

L = block length of TEMP EIJ *iset* *icase* (if present)

+ block length of DISL EIJ *iset* *icase* (if present)

+ block length of PRES EIJ *iset* *icase* (if present)

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

6.3 SSOL – STATIC SOLUTION GENERATOR

Function. SSOL computes displacements and reactions due to point loading applied at joints. Where iset is the load set identifier, and ncon is the constraint case (see SSOL Reset Controls and Figure 6-1), the SYSVEC-format output data sets are

STAT DISP iset ncon, and

STAT REAC iset ncon.

Each of the output data sets consists of n blocks, where n is the number cf cases in the designated load set. Block 1 is the solution corresponding to case 1, block 2 corresponds to case 2, etc. The number of cases, n, is the largest of the following:

(1) n_f, the number of blocks in APPL FORC iset,

(2) n_m, the number of blocks in APPL MOTI iset, or

(3) n_e, the largest value of icase in any EQNF FORC iset icase resident in QLIB.

If n_f or n_m is less than n, the omitted input load vectors (e.g., applied forces for cases $n_f + 1$, $n_f + 2$, etc.) are assumed to be identically zero. Similarly, any omitted EQNF FORC iset icase is assumed to be identically zero.

Components of the STAT REAC iset ncon set corresponding to constrained or specified joint motion components are reactions. All other items are residual error forces; i.e., F - KU, where F = total applied forces, K = stiffness matrix, and U = computed displacements. The error forces should always be scanned for evidence of round-off error.

If the error print (EP) option is in effect, three items, entitled F*U, U*KU, and ERR, will be printed for each load case. Where F and U are applied force and computed joint motion vectors, respectively, these three items are in order, F^tU, U^tKU, and the absolute value of $F^tU - U^tKU$ divided by the greater of F^tU or U^tKU. If joint motions are specified via APPL MOTI iset, the ERR quantity should not be interpreted as an error measure.

RESET Controls.

Name	Default Value	Meaning
K	K	Name of stiffness matrix.
CON	1	Constraint case (see INV discussion).
KLIB	1	Library containing stiffness matrix.
KILIB	1	Library containing factored stiffness matrix.
QLIB	1	Data source library, all load input.
SET	1	Load set (iset).
REAC	1	Nonzero value causes STAT REAC iset ncon to be produced.
EP	1	Nonzero value causes error analysis to be performed.

Core Requirements.

In the following L is the greater of (1) the block length of K, or (2) the block length of INV K. JT is the total number of joints in the structure. Data space requirements are approximately as follows:

$$L + n * JT$$

where n = 12 if K is single-precision, or 18 if K is double-precision.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

Section 7

STRESSES

GSF is used to generate data sets containing element stresses and internal load details. PSF is used to produce various forms of stress printout. Figure 6-1 illustrates the relationship of GSF and PSF to the other processors involved in static stress analysis, and Table 6-1 lists the names of the associated input and output data sets.

7.1 GSF – STRESS DATA GENERATOR

Function. GSF generates data sets containing element stress and internal load information. Input data sets are usually (1) structural deformations resident in STAT DISP iset ncon, and (2) initial strain data resident in IS Eij iset icase. Production of these data sets is described in Section 6. Output data sets are named STRS Eij iset icase, where Eij, iset, and icase have the same meaning as used throughout Section 6.

RESET Controls

Name	Default Value	Meaning
QLIB	1	Source library for STAT DISP iset ncon, and IS Eij iset icase data sets, if applicable. Stresses will be stored in QLIB.
SET	1	Load set (iset).
L1	1	$icase_1$.
L2		$icase_2$.
CON	MASK	Fourth word of STAT DISP iset ncon.
EMBED	0	If nonzero, all stresses computed in the current GSF execution will be embedded in the E-state, for use in computing geometric stiffness matrices, Kg. (See Section 4.)
ACCUM	0	If nonzero, all stresses computed in the current GSF execution will be added to those already resident in the E-state.
LREC	5600	Block length of output data sets.

Execution Control Parameters

The following statement may appear immediately after the last RESET card, or after @XQT GSF if no RESET cards are given:

SOURCE= N1 N2 n3 n4 (trailing words MASK-filled, if omitted)

The SOURCE statement names a data set, resident in QLIB, which replaces STAT DISP iset ncon as the source of joint motion data.

Stress data sets, STRS Eij iset icase, will be produced for icase = icase_1, $\text{icase}_1 + 1$, --- icase_2. If L2 is not RESET, icase_2 assumes, as a default value, the number of cases (blocks) resident in the input data set containing joint motion data.

Subsequent control cards designate specific groups of elements for which stress data is to be computed, as illustrated below:

E21: 4, 7: 10$	E21 groups 4 through 7, and group 10.
E33$	All E33 elements.
E43: 1: 7: 10, 12$	E43, groups 1, 7, 10, 11, 12.
E44: 1, 5: 7, 17: etc.	

If control cards of this kind are not given, stresses will be computed for all elements.

Core Requirements

The data space requirements are approximately the sum of the following:

Six times the number of joints in the structure

+ the longest IS Eij block length (if any)

+ the STRS Eij block length.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

7.2 PSF – STRESS TABLE PRINTER

Function. PSF prints element stresses and internal load information contained in the STRS Eij iset icase data sets produced by GSF. If present, load case titles contained in CASE TITL iset are displayed.

Reset Controls

Name	Default Value	Meaning
QLIB	1	Data source library, STRS Eij iset, icase, and CASE TITL iset.
SET	1	iset (load set identified).
L1	1	icase$_1$
L2	1	icase$_2$ – Default icase$_2$ is the number of cases (blocks) in CASE TITL iset.
DISPLAY	1	DISPLAY = 1 produces standard stress print. DISPLAY = 2 designates output of end force data for beams, bars, etc., and stress resultants for two-dimensional elements.(See Page 7.2-5a) DISPLAY = 3 produces detailed display of beam bending stress data.
NODES	1	For 3- and 4-node elements, set NODES= 0 to eliminate printout of stresses, etc., at element corners. (See Fig. 7.3-1.)
CROSS	1	For 3- and 4-node elements, set CROSS= 0 to restrict the printout to mid-surface stresses. (See Fig. 7.3-1.)
LINES	56	Lines per page.
IEA	1	Set IEA= 0 to cause the run to be aborted if an error occurs (e.g., designated source data sets do not exist).

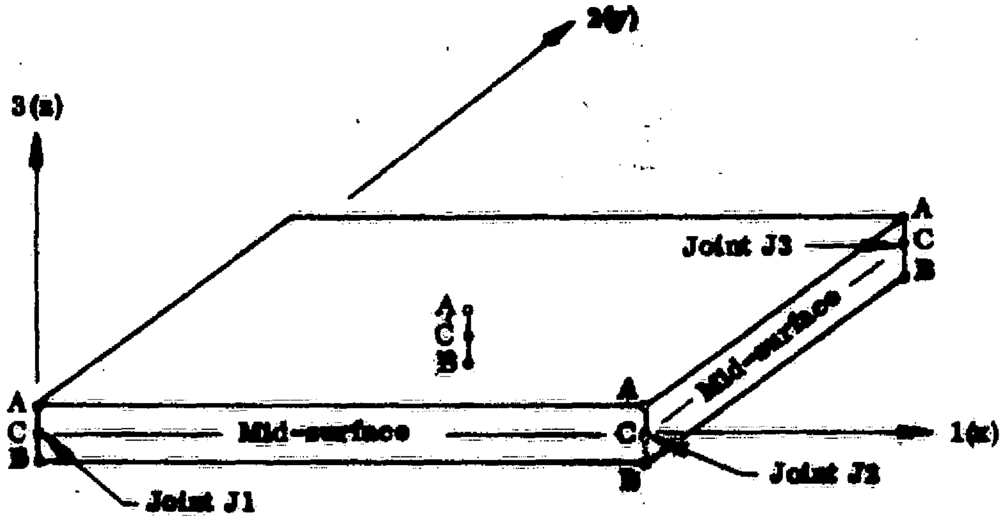


Fig. 7.3-1 Stress Display Convention – 4-Node Elements

Execution Control Parameters

The following control card may appear after the last RESET card (if any):

$$DIV= f_1 , f_2 , f_3 , f_4$$

The f_i's are divisors for the following categories of data, in the order indicated:

(1) Stresses.

(2) Membrane stress resultants.

(3) Bending stress resultants.

(4) End forces for beam, bar, etc. elements.

The default value of each f_i is 1.0.

Data will be displayed for cases $icase_1$, $icase_1 + 1$, --- $icase_2$ of set iset. Subsequent control cards (if any) will indicate specific element types (Eij) for which data are to be displayed, as indicated below. If no such cards are given, the data will be displayed for all element types.

```
E21:  E23$          |   Data will be displayed only for these
                    |   element types.
E31:  E41:  E43$    |
```

Several examples of PSF printout are shown on the following pages.

Core Requirements

Data space required is approximately equal to one block of STRS Eij.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

```
1200 KIP SHEAR LOAD, DIR. 2                                                     CASE  S-  1

 RING STIFFENER, 2-20                                                            *GROUP   3

-21 STRESSES, DIVIDED BY   1000.0000

                      MAX COMBINED                                 TRANSVERSE
          CONNECTED   P/A +BENDING                                 SHEAR STRESS          TWIST
INDEX     JOINTS      TENSION          COMP          P/A           S1           S2          SHEAR

     1     21    22       7.69          .00          4.26          .00          .00          .00
     2     22    23      15.91          .00         12.99          .00          .00          .00
     3     23    24      19.06          .00         16.86          .00          .00          .00
     4     24    25      15.80          .00         12.89          .00          .00          .00
     5     25    26       7.54          .00          4.11          .00          .00          .00
     6     26    27        .00        -7.69         -4.26          .00          .00          .00
     7     27    28        .00       -15.91        -12.99          .00          .00          .00
     8     28    29        .00       -19.06        -16.86          .00          .00          .00
     9     29    30        .00       -15.80        -12.89          .00          .00          .00
    10     30    21        .00        -7.54         -4.11          .00          .00          .00
```

Note: This printout was produced with DISPLAY=1 and the control parameter DIV=1000.

```
1200 KIP SHEAR LOAD, DIR. 2                                          CASE  5=  1
 RING STIFFENER, Z=20                                                #GROUP    3
F21 FORCES,    DIVIDED BY       1.0000
INDEX  JOINT      P1        P2        P3        P4        P5        P6

   1      21     86.73    570.97  -9200.52  -1173.97    171.36   2393.08
          22    -86.73   -570.97   9200.52  -4707.38    721.98  -2393.08

   2      22    163.25    140.43-28064.40   2554.90   -642.98   2269.01
          23   -163.25   -140.43  28064.40  -4001.37   2324.54  -2269.01

   3      23     10.03     -1.01-36423.49   3010.12  -2379.70      3.58
          24    -10.03      1.01  36423.49  -2999.74   2483.03     -3.58

   4      24   -183.99   -140.31-27839.44   3989.37  -2429.59  -2268.92
          25    183.99    140.31  27839.44  -2544.10    534.36   2268.92

   5      25    -70.59   -567.75  -8882.02   4699.58   -613.79  -2400.72
          26     70.59    567.75   8882.02   1148.55   -113.31   2400.72

   6      26    -86.72   -570.97   9200.53   1173.97   -171.32  -2393.08
          27     86.72    570.97  -9200.53   4707.38   -721.99   2393.08

   7      27   -163.25   -140.43  28064.41  -2554.91    642.99  -2269.00
          28    163.25    140.43-28064.41   4001.37  -2324.54   2269.00

   8      28    -10.03      1.01  36423.48  -3010.13   2379.70     -3.58
          29     10.03     -1.01-36423.48   2999.73  -2482.99      3.58

   9      29    183.99    140.31  27839.43  -3989.36   2429.55   2268.92
          30   -183.99   -140.31-27839.43   2544.11   -534.36  -2268.92

  10      30     70.58    567.75   8882.02  -4699.58    613.77   2400.71
          21    -70.58   -567.75  -8882.02  -1148.55    113.27  -2400.71
```

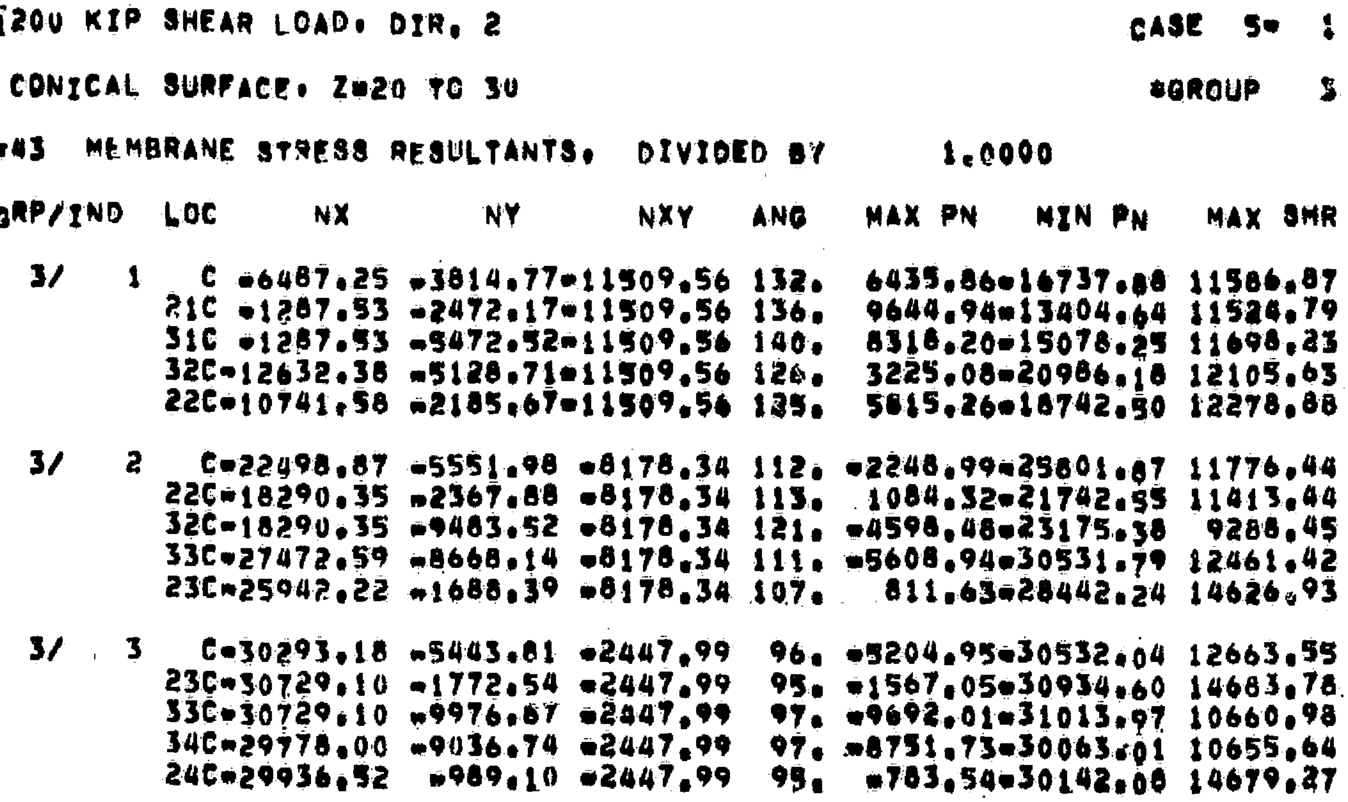
Note: PI's are point force and moment components (relative to element reference frames) acting on element end points. This was printout produced with DISPLAY=2.

```
CASE A. WIND ON TOWER=57.6 PSF                                    CASE  I=  1

TOWER BODY CORNER ANGLES                                          *GROUP    9

E21 STRESSES, DIVIDED BY      1.0000

                                           MY2/I1                 MY/I'S
INDEX      LOC        MY2/I1     MY1/I2   +MY1/I2       P/A        +P/A

  5
         41-1        1635.70    3029.33   4665.03    5137.33     9802.37
         41-2        1744.75    2662.50   4407.25    5137.33     9544.58
         41-3           -.00   -3206.83  -3206.83    5137.33     1930.50
         41-4       -1744.75    2662.50    917.75    5137.33     6055.09

         51-1       -2511.48   -4546.61  -7058.09    5137.33    -1920.75
         51-2       -2678.91   -3996.04  -6674.95    5137.33    -1537.62
         51-3            .00    4813.01   4813.01    5137.33     9950.34
         51-4        2678.91   -3996.04  -1317.13    5137.33     3820.20

  6
         40-1       -6038.17  -11607.92 -17646.09  -12866.94   -30513.03
         40-2       -6440.71  -10202.27 -16642.99  -12866.94   -29509.93
         40-3            .00   12288.07  12288.07  -12866.94     -578.87
         40-4        6440.71  -10202.28  -3761.57  -12866.94   -16628.51

         50-1        3861.71   11106.91  14968.62  -12866.94     2101.68
         50-2        4119.16    9761.93  13881.09  -12866.94     1014.15
         50-3           -.00  -11757.71 -11757.71  -12806.94   -24624.65
         50-4       -4119.16    9761.94   5642.78  -12866.94    -7224.16

  7
         42-1       -1653.93    3023.05   1369.11    5137.04     6506.16
         42-2       -1764.19    2656.97    892.78    5137.04     6029.82
         42-3            .00   -3200.18  -3200.18    5137.04     1936.86
         42-4        1764.19    2656.98   4421.17    5137.04     9558.21

         52-1        2495.67   -4599.35  -2103.68    5137.04     3033.36
         52-2        2662.05   -4042.40  -1380.35    5137.04     3756.69
         52-3           -.00    4868.85   4868.85    5137.04    10005.69
         52-4       -2662.05   -4042.40  -6704.45    5137.04    -1567.41
```

Note: This printout was produced with DISPLAY=3.

```
CASE A, WIND ON TOWER=57.6 PSF                                         CASE  1-  1

SECTION F-F                                                            -GROUP  17

E21 STRESSES, DIVIDED BY      1.0000

                                                  MY2/I1                      MY/I'S
INDEX       LOC          MY2/I1       MY1/I2     +MY1/I2        P/A          +P/A

   13
            97-1         -35.33     -2086.51    -2121.84     -7703.13     -9824.96
            97-2          35.33     -2086.51    -2051.18     -7703.13     -9754.31
            97-3         376.82       829.97     1206.79     -7703.13     -6496.34
            97-4        -376.82       829.97      453.15     -7703.13     -7249.98

            99-1          75.56      3374.26     3449.82     -7703.13     -4253.31
            99-2         -75.56      3374.26     3298.71     -7703.13     -4404.42
            99-3        -805.92     -1342.22    -2148.14     -7703.13     -9851.27
            99-4         805.92     -1342.22     -536.29     -7703.13     -8239.42

   14
            99-1          76.22      3403.57     3479.80     -7157.43     -3677.63
            99-2         -76.22      3403.57     3327.35     -7157.43     -3830.08
            99-3.       -813.06     -1353.87    -2166.94     -7157.43     -9324.37
            99-4         813.06     -1353.87     -540.81     -7157.43     -7698.25

           101-1         -42.77       415.57      372.80     -7157.43     -6784.63
           101-2          42.77       415.57      458.34     -7157.43     -6699.09
           101-3         456.22      -165.31      290.92     -7157.43     -6866.52
           101-4        -456.22      -165.31     -621.53     -7157.43     -7778.96

   15
           101-1         -37.96       623.39      585.43     -6628.46     -6043.03
           101-2          37.96       623.39      661.34     -6628.46     -5967.11
           101-3         404.89      -247.97      156.92     -6628.46     -6471.54
           101-4        -404.89      -247.97     -652.86     -6628.46     -7281.31

           103-1           -.33      1917.79     1917.46     -6628.46     -4711.00
           103-2            .33      1917.79     1918.12     -6628.46     -4710.34
           103-3           3.52      -762.86     -759.34     -6628.46     -7387.80
           103-4          -3.52      -762.86     -766.37     -6628.46     -7394.83
```

Note: This printout was produced with DISPLAY=3.



```
1200 KIP SHEAR LOAD, DIR. 2                                        CASE  5-  1
 CONICAL SURFACE, Z=20 TO 30                                       *GROUP   3
#3  MEMBRANE STRESS RESULTANTS,  DIVIDED BY       1.0000
GRP/IND  LOC       NX          NY          NXY   ANG    MAX PN    MIN PN    MAX SHR

  3/    1    C -6487.25 -3814.77-11509.56 132.   6435.86-16737.88 11586.87
           21C -1287.53 -2472.17-11509.56 136.   9644.94-13404.64 11524.79
           31C -1287.53 -5472.52-11509.56 140.   8318.20-15078.25 11698.23
           32C-12632.38 -5128.71-11509.56 126.   3225.08-20986.18 12105.63
           22C-10741.58 -2185.67-11509.56 135.   5815.26-18742.50 12278.88

  3/    2    C-22498.87 -5551.98 -8178.34 112. -2248.99-25801.87 11776.44
           22C-18290.35 -2367.88 -8178.34 113.  1084.32-21742.55 11413.44
           32C-18290.35 -9483.52 -8178.34 121. -4598.48-23175.38  9288.45
           33C-27472.59 -8668.14 -8178.34 111. -5608.94-30531.79 12461.42
           23C-25942.22 -1688.39 -8178.34 107.   811.63-28442.24 14626.93

  3/    3    C-30293.18 -5443.81 -2447.99  96. -5204.95-30532.04 12663.55
           23C-30729.10 -1772.54 -2447.99  95. -1567.05-30934.60 14683.78
           33C-30729.10 -9976.67 -2447.99  97. -9692.01-31013.97 10660.98
           34C-29778.00 -9036.74 -2447.99  97. -8751.73-30063.01 10655.64
           24C-29936.52  -989.10 -2447.99  95.  -783.54-30142.08 14679.27
```

Note: This printout was produced with DISPLAY=2. For each element, the first line printed refers to the center of the element. Subsequent lines refer to nodes 1,2,3, and 4, with the connected joint number listed on the left. The C refers to mid-surface. On the following pages, all of which were produced with DISPLAY=1, A and B have the meaning indicated on Figure 7.3-1.

```
1200 KIP SHEAR LOAD, DIR. 2                                          CASE  5-  1

 CONICAL SURFACE, Z=20 TO 30                                          *GROUP   3

F43  STRESSES,  DIVIDED BY   1000.0000

GRP/IND  LOC     SX        SY        TXY   ANG     MAX PS    MIN PS     MAX SHR

  3/    1   C   -12.97     -7.63    -23.02 132.     12.87    -33.48      23.17
            A   -12.72     -7.60    -23.02 132.     13.00    -33.32      23.16
            B   -13.23     -7.66    -23.02 132.     12.74    -33.63      23.19
          21C    -2.58     -4.94    -23.02 136.     19.29    -26.81      23.05
            A    -5.07     -6.14    -23.02 136.     17.42    -28.63      23.03
            B     -.08     -3.75    -23.02 137.     21.18    -25.01      23.09
          31C    -2.58    -10.95    -23.02 140.     16.64    -30.16      23.40
            A      .34    -10.58    -23.02 142.     18.54    -28.78      23.66
            B    -5.49    -11.31    -23.02 139.     14.80    -31.60      23.20
          32C   -25.26    -10.26    -23.02 126.      6.45    -41.97      24.21
            A   -21.61     -9.29    -23.02 128.      8.38    -39.28      23.83
            B   -28.92    -11.22    -23.02 124.      4.59    -44.73      24.66
          22C   -21.48     -4.37    -23.02 125.     11.63    -37.48      24.56
            A   -23.74     -5.49    -23.02 124.     10.15    -39.38      24.76
            B   -19.22     -3.25    -23.02 125.     13.13    -35.60      24.36

  3/    2   C   -45.00    -11.10    -16.36 112.     -4.50    -51.60      23.55
            A   -44.27    -10.94    -16.36 112.     -4.25    -50.96      23.35
            B   -45.73    -11.27    -16.36 112.     -4.74    -52.25      23.76
          22C   -36.58     -4.74    -16.36 113.      2.17    -43.49      22.83
            A   -42.72     -7.71    -16.36 112.     -1.25    -49.17      23.96
            B   -30.44     -1.77    -16.36 114.      5.65    -37.85      21.75
          32C   -36.58    -18.97    -16.36 121.     -9.20    -46.35      18.58
            A   -28.32    -17.50    -16.36 126.     -5.69    -40.14      17.23
            B   -44.84    -20.43    -16.36 117.    -12.23    -53.04      20.41
          33C   -54.95    -17.34    -16.36 111.    -11.22    -61.06      24.92
            A   -46.59    -15.43    -16.36 113.     -8.42    -53.60      22.59
            B   -63.30    -19.25    -16.36 108.    -13.84    -68.71      27.43
          23C   -51.88     -3.38    -16.36 107.      1.62    -56.88      29.25
            A   -57.33     -5.97    -16.36 106.     -1.20    -62.10      30.45
            B   -46.44      -.79    -16.36 108.      4.47    -51.69      28.08

  3/    3   C   -60.59    -10.89     -4.90  96.    -10.41    -61.06      25.33
            A   -59.61    -10.65     -4.90  96.    -10.16    -60.09      24.97
            B   -61.56    -11.13     -4.90  95.    -10.66    -62.04      25.69
          23C   -61.46     -3.55     -4.90  95.     -3.13    -61.87      29.37
            A   -68.31     -6.81     -4.90  95.     -6.42    -68.70      31.14
            B   -54.61      -.28     -4.90  95.       .16    -55.04      27.60
          33C   -61.46    -19.95     -4.90  97.    -19.38    -62.03      21.32
            A   -51.32    -17.97     -4.90  98.    -17.27    -52.02      17.38
            B   -71.60    -21.94     -4.90  96.    -21.46    -72.07      25.31
          34C   -59.56    -18.07     -4.90  97.    -17.50    -60.13      21.31
            A   -49.58    -15.94     -4.90  98.    -15.24    -50.28      17.52
            B   -69.53    -20.21     -4.90  96.    -19.73    -70.01      25.14
          24C   -59.87     -1.98     -4.90  95.     -1.57    -60.28      29.36
            A   -66.74     -5.23     -4.90  95.     -4.85    -67.13      31.14
            B   -53.00      1.28     -4.90  95.      1.71    -53.44      27.58
```

```
1200 KIP SHEAR LOAD, DIR. 2                                                CASE  5-  1

 CONICAL SURFACE, Z=20 TO 30                                               *GROUP   3

F43  STRESSES,  DIVIDED BY   1000.0000

GRP/IND  LOC       SX          SY         TXY   ANG     MAX PS     MIN PS     MAX SHR

  3/   1   C     -12.97      -7.63     -23.02 132.      12.87     -33.48       23.17
           A     -12.72      -7.60     -23.02 132.      13.00     -33.32       23.16
           B     -13.23      -7.66     -23.02 132.      12.74     -33.63       23.19

  3/   2   C     -45.00     -11.10     -16.36 112.      -4.50     -51.60       23.55
           A     -44.27     -10.94     -16.36 112.      -4.25     -50.96       23.35
           B     -45.73     -11.27     -16.36 112.      -4.74     -52.25       23.76

  3/   3   C     -60.59     -10.89      -4.90  96.     -10.41     -61.06       25.33
           A     -59.61     -10.65      -4.90  96.     -10.16     -60.09       24.97
           B     -61.56     -11.13      -4.90  95.     -10.66     -62.04       25.69

  3/   4   C     -50.06      -6.25       8.70  79.      -4.58     -51.72       23.57
           A     -49.37      -6.04       8.70  79.      -4.36     -51.05       23.34
           B     -50.76      -6.45       8.70  79.      -4.80     -52.40       23.80

  3/   5   C     -21.63        .82      20.27  59.      12.77     -33.58       23.17
           A     -21.44        .92      20.27  59.      12.89     -33.41       23.15
           B     -21.82        .72      20.27  60.      12.65     -33.74       23.19

  3/   6   C      12.97       7.63      23.02  42.      33.48     -12.87       23.17
           A      12.72       7.60      23.02  42.      33.32     -13.00       23.16
           B      13.23       7.66      23.02  42.      33.63     -12.74       23.19

  3/   7   C      45.00      11.10      16.36  22.      51.60       4.50       23.55
           A      44.27      10.94      16.36  22.      50.96       4.25       23.35
           B      45.73      11.27      16.36  22.      52.25       4.74       23.76

  3/   8   C      60.59      10.89       4.90   6.      61.06      10.41       25.33
           A      59.61      10.65       4.90   6.      60.09      10.16       24.97
           B      61.56      11.13       4.90   5.      62.04      10.66       25.69

  3/   9   C      50.06       6.25      -8.70 169.      51.72       4.58       23.57
           A      49.37       6.04      -8.70 169.      51.05       4.36       23.34
           B      50.75       6.45      -8.70 169.      52.40       4.80       23.80

  3/  10   C      21.63       -.82     -20.27 149.      33.58     -12.77       23.17
           A      21.44       -.92     -20.27 149.      33.41     -12.89       23.15
           B      21.82       -.72     -20.27 150.      33.74     -12.65       23.19
```

```
1200 KIP SHEAR LOAD, DIR. 2                                              CASE  5-  1

CONICAL SURFACE, Z=20 TO 30                                              *GROUP   3

F43  STRESSES,  DIVIDED BY   1000.0000

GRP/INC  LOC      SX         SY         TXY   ANG.   MAX PS     MIN PS     MAX SHR

   3/    1   C   -12.97     -7.63     -23.02 132.     12.87     -33.48      23.17
           21C    -2.58     -4.94     -23.02 136.     19.29     -26.81      23.05
           31C    -2.58    -10.95     -23.02 140.     16.64     -30.16      23.40
           32C   -25.26    -10.26     -23.02 126.      6.45     -41.97      24.21
           22C   -21.48     -4.37     -23.02 125.     11.63     -37.48      24.56

   3/    2   C   -45.00    -11.10     -16.36 112.     -4.50     -51.60      23.55
           22C   -36.58     -4.74     -16.36 113.      2.17     -43.49      22.83
           32C   -36.58    -18.97     -16.36 121.     -9.20     -46.35      18.58
           33C   -54.95    -17.34     -16.36 111.    -11.22     -61.06      24.92
           23C   -51.88     -3.38     -16.36 107.      1.62     -56.88      29.25

   3/    3   C   -60.59    -10.89      -4.90  96.    -10.41     -61.06      25.33
           23C   -61.46     -3.55      -4.90  95.     -3.13     -61.87      29.37
           33C   -61.46    -19.95      -4.90  97.    -19.38     -62.03      21.32
           34C   -59.56    -18.07      -4.90  97.    -17.50     -60.13      21.31
           24C   -59.87     -1.98      -4.90  95.     -1.57     -60.28      29.36

   3/    4   C   -50.06     -6.25       8.70  79.     -4.58     -51.72      23.57
           24C   -59.88     -1.35       8.70  82.      -.08     -61.14      30.53
           34C   -59.88    -12.29       8.70  80.    -10.75     -61.42      25.33
           35C   -38.46    -11.04       8.70  74.     -8.51     -40.99      16.24
           25C   -42.03      -.30       8.70  79.      1.44     -43.77      22.60

   3/    5   C   -21.63       .82      20.27  59.     12.77     -33.58      23.17
           25C   -32.49      1.78      20.27  65.     11.19     -41.90      26.54
           35C   -32.49      -.36      20.27  64.      9.44     -42.29      25.86
           36C    -8.80      -.11      20.27  51.     16.27     -25.19      20.73
           26C   -12.75      1.99      20.27  55.     16.19     -26.95      21.57

   3/    6   C    12.97      7.63      23.02  42.     33.48     -12.87      23.17
           26C     2.58      4.94      23.02  46.     26.81     -19.29      23.05
           36C     2.58     10.95      23.02  50.     30.16     -16.64      23.40
           37C    25.26     10.26      23.02  36.     41.97      -6.45      24.21
           27C    21.48      4.37      23.02  35.     37.49     -11.63      24.56

   3/    7   C    45.00     11.10      16.36  22.     51.60       4.50      23.55
           27C    36.58      4.74      16.36  23.     43.49      -2.17      22.83
           37C    36.58     18.97      16.36  31.     46.35       9.20      18.58
           38C    54.95     17.34      16.36  21.     61.06      11.22      24.92
           28C    51.88      3.38      16.36  17.     56.88      -1.62      29.25

   3/    8   C    60.59     10.89       4.90   6.     61.06      10.41      25.33
           28C    61.46      3.55       4.90   5.     61.87       3.13      29.37
           38C    61.46     19.95       4.90   7.     62.03      19.38      21.32
           39C    59.56     18.07       4.90   7.     60.13      17.50      21.31
           29C    59.87      1.98       4.90   5.     60.28       1.57      29.36
```

```
1200 KIP SHEAR LOAD, DIR. 2                                              CASE   5-   1

 CONICAL  SURFACE, Z=20 TO 30                                             *GROUP    3

F43   STRESSES,   DIVIDED BY    1000.0000

GRP/IND   LOC       SX          SY          TXY    ANG     MAX PS     MIN PS     MAX SHR

  3/   1   C     -12.97       -7.63      -23.02  132.      12.87     -33.48       23.17

  3/   2   C     -45.00      -11.10      -16.36  112.      -4.50     -51.60       23.55

  3/   3   C     -60.59      -10.89       -4.90   96.     -10.41     -61.06       25.33

  3/   4   C     -50.06       -6.25        8.70   79.      -4.58     -51.72       23.57

  3/   5   C     -21.63         .82       20.27   59.      12.77     -33.58       23.17

  3/   6   C      12.97        7.63       23.02   42.      33.48     -12.87       23.17

  3/   7   C      45.00       11.10       16.36   22.      51.60       4.50       23.55

  3/   8   C      60.59       10.89        4.90    6.      61.06      10.41       25.33

  3/   9   C      50.06        6.25       -8.70  169.      51.72       4.58       23.57

  3/  10   C      21.63        -.82      -20.27  149.      33.58     -12.77       23.17
```

Section 8

EIG – SPARSE MATRIX EIGENSOLVER

Function. EIG solves linear vibration and bifurcation buckling eigenproblems of the types indicated by Eqs. (1) and (2).

$$rMX - KX = 0 \tag{1}$$

$$rKgX + KX = 0 \tag{2}$$

K and Kg must be in the SPAR standard sparse matrix format; M may be in either SPAR sparse matrix or diagonal format; and K must be non-singular. K need not be positive-definite in Eq. (1), but must be positive-definite in Eq. (2).

EIG implements an iterative process consisting of a Stodola (matrix iteration) procedure followed by a Rayleigh-Ritz procedure, followed by a second Stodola procedure, etc., resulting in successively refined approximations of m eigenvectors associated with the m eigenvalues of Eqs. (1) or (2) closest to zero. Closely spaced roots do not adversely affect the process.

In the following discussion of application of the process to Eq. (1), it is assumed that M and K are n x n, and that m linearly independent system vectors, Y^1, Y^2, - - Y^m, are known. Methods of initiating these vectors will be discussed later. Usually m is chosen to be much less than n; that is, m is usually 4 to 30, while n may be extremely large (e.g. 10000 +).

In the following discussion, Z is a general linear combination of the Y's:

$$\begin{aligned} Z &= q_1 Y_1 + q_2 Y_2 + \dots + q_m Y_m \\ &= HQ, \text{ where} \end{aligned}$$

$$H = (Y_1\ Y_2 \text{ ---- } Y_m),\ \text{and}$$
$$Q = (q_1\ q_2 \text{ ---- } q_m)^*. \qquad (3)$$

The Rayleigh-Ritz procedure consists of replacing X with Z in Eq. (1); that is, substituting (3) into (1) and pre-multiplying by H^*,

$$r\ (H^*MH)Q + (H^*KH)Q = 0 \qquad (4)$$

Using the Cholesky-Householder method, this low-order eigenproblem is solved for all m eigenvectors Q_1, Q_2, ---- Q_m. The Rayleigh-Ritz procedure is concluded by using Eq. (3) to calculate m improved approximations of the n - order system eigenvectors,

$$\begin{array}{lcl} Z_1 & = & HQ_1 \\ Z_2 & = & HQ_2 \\ - & & - \\ - & & - \\ Z_m & = & HQ_m \end{array} \qquad (5)$$

The Stodola (matrix iteration) step is as follows. From each Z of Eq. (5) a new Y is computed, subject to the requirement that

$$MZ = KY. \qquad (6)$$

In performing MZ calculations and in solving for Y, SPAR's sparse-matrix algorithms are used. The iterative process continues by using the new Y's in another Rayleigh-Ritz procedure, etc. Vectors are regularly renormalized to avoid scaling problems.

The convergence resulting from Eq. (6) is readily observed by considering vectors Z and Y as linear combinations of the n system eigenvectors X_1, X_2 -- X_n.

$$Z = \sum_{i=1}^{n} a_i X_i , \quad Y = \sum_{i=1}^{n} b_i X_i \tag{7}$$

Substitution of (7) into (6) and pre-multiplication by X_j^* gives

$$\sum_{i=1}^{n} X_j^* M X_i a_i = \sum_{i=1}^{n} X_j^* K X_i b_i .$$

Since, for i not equal to j,

$$X_j^* M X_i = X_j^* K X_i = 0,$$

$$b_i = \frac{X_i^* M X_i}{X_i^* K X_i} a_i = \frac{a_i}{r_i} , \tag{8}$$

where r_i is the eigenvalue associated with X_i. Eq. (8) indicates that in the transition from Z to Y the magnitudes of corresponding eigenvector coefficients are diminished in inverse proportion to the associated eigenvalue magnitudes. Suppose, for example, that some of the system eigenvalues are $r_1 = .5$, $r_7 = 10$. $r_{25} = 1000$. Then

$$b_1 = 2.0 a_1,$$
$$b_7 = .1 a_7, \text{ and}$$
$$b_{25} = .001 a_{25}.$$

Since K need not be positive definite, EIG can be used to solve "shifted" vibrational eigenproblems. For example, if cMX is added and subtracted from the left side of Eq. (1),

$$(r - c) MX - (K - cM) X = 0. \tag{9}$$

Eq. (9) is of the same form as eq. (1), except that (r - c) has replaced r, and (K - cM) has replaced K. Shifting has several useful applications.

For example, suppose it is desired to compute only the vibrational modes associated with eigenvalues near c. The procedure is as follows:

(1) Use AUS/SUM to compute K - cM.

(2) Use INV to factor K - cM.

(3) Use EIG to compute the eigenvalues of Eq. (9), (r - c), near zero.

It is noted that INV can be used to determine the number of eigenvalues present in any range of interest. INV prints the number of negative terms encountered in factoring K - cM, which is equal to the number of roots below c in Eq. (1).

EIG measures convergence of the overall process by tracking changes in the eigenvalues of Eq. (4), as computed in successive steps.

Through RESET controls, the analyst may choose eigenvector initial approximations comprised of:

(1) random vectors generated internally by EIG, and/or

(2) data sets, from any source, resident in a library.

The following output data sets are produced by EIG:

Data Set	Name	Contents
VIBR MODE	nset ncon	The last set of vibrational eigenvector approximations $(Y_1\ Y_2 \text{ -- } Y_m)$, normalized to $Y_i^t\ M\ Y_i = 1.0$.
VIBR EVAL	nset ncon	The eigenvalue approximations associated with the last set of vibrational eigenvector approximations (not in Hz).
BUCK MODE	nset ncon	Last set of buckling mode approximations, normalized to $Y_i^t\ K_g\ Y_i = 1.0$.
BUCK EVAL	nset ncon	Eigenvalue approximations associated with the last set of buckling mode approximations.

The VIBR MODE and BUCK MODE data sets are in SYSVEC format, containing m blocks – one for each eigenvector approximation. The VIBR EVAL and BUCK EVAL data sets contain a single block of m words.

The fourth word of the names of the output data sets, ncon, is the constraint case identifier, designated by the CON Reset Control. The third word, nset, is a user-supplied identifier designated by the NEWSET Reset Control. A typical use of nset is to identify results associated with different shift points.

RESET Controls

Name	Default Value	Meaning
PROB	VIBR	VIBR indicates solution of Eq. (1). BUCK indicates solution of Eq. (2).
NDYN	8	Maximum number of passes through the EIG iterative process.

RESET Controls, Continued.

Name	Default Value	Meaning
CONV	.0001	Iteration controls. For iteration j, the measure of convergence of eigenvalue i is: $$e_i = \|(r_i^j - r_i^{j-1})/r_i^j\| .$$ The ith eigenvalue is said to be converged if e_i is less than the value indicated by CONV. Execution will terminate when both of the following conditions are met: (1) At least NREQ eigenvalues have converged. (2) There are no unconverged eigenvalue approximations within the range V1 < r_i < V2
NREQ	.5 m	
V1	.0	
V2	.0	
K	K	First word of the name of the data set to be used as K in Eqs. (1) or (2). Common RESETs are K= K+KG, and K= K+CM.
KG	KG	First word of the name of the data set to be used as Kg in Eq. (2).
M	DEM	First word of the name of the data set to be used as M in Eq. (1). A common RESET is M= M+RM, where M+RM is the sum of RMASS and DEM (or CEM), formed through AUS/SUM.
CON	1	Constraint case, <u>ncon</u>, applied in factoring of K, or the K surrogate, by INV.
KLIB	1	Library containing K, or the K surrogate.
KILIB	1	Library containing the factored K (or K surrogate) produced by INV.
KGLIB	1	Library containing KG, or the KG surrogate.
MLIB	1	Library containing M, or the M surrogate.
OUTLIB	1	Library into which EIG output is to be delivered.
NEWSET	1	Value of <u>nset</u>, the third word of the output data set names.

Name	Default Value	Meaning
INIT	0	These parameters designate initial approximations from either or both o' the following sources: (1) INIT = the number of vectors of random numbers. (2) Vectors L1 through L2 of an existing data set within library INLIB. The name of this data set is VIBR (or BUCK) MODE oldset ncon. If L1 and L2 are not given, and INIT is not given, all vectors in the data set VIBR (or BUCK) MODE oldset ncon will be used as initial approximations. The maximum total number of functions, m, is 100.
L1		
L2		
INLIB	1	
OLDSET	1	
XRAND	0	Advance the random number generator by XRAND numbers (to avoid re-creating the same vector as in a previous execution).
SHIFT	.0	SHIFT is added to the computed eigenvalues [i.e., shift corresponds to c in Eq. (9)] , before they are stored in VIBR EVAL or BUCK EVAL.
HIST	4	Nonzero value, 4 or 8, results in printout of eigenvalue convergence history, displayed as HIST columns per page.
TIME	0	Nonzero value results in printout of intermediate CP and wall clock times.
RRCH	0	Nonzero value results in execution of a check of the accuracy of each Rayleigh-Ritz analysis.

Following any RESET cards, two additional types of commands may appear:

(1) RSCALE $s_1, s_2, s_3, s_4, s_5, s_6$.

The above causes all direction − i joint motion components of all random vectors produced by the INIT Reset Control to be multiplied by s_i. Default values are $s_i = 1.$ for $i = 1, 2, 3,$ and $s_i = .01$ for $i = 4, 5, 6$ (rotations).

(2) PRINT K1 K2 K3 K4 K5

Nonzero values of the K's result in production of the following categories of printout:

K1: Vectors Y, KY, and MY are printed during each iteration. Bring a boxcar to haul your printout home.

K2: Terms in the m x m coefficient matrices of Eq. (4) are printed during each iteration.

K3: All Rayleigh-Ritz eigenvectors, Q_i in Eq. (4), are printed during each iteration.

K4: Vectors Z_i of Eq. (5) are printed during each iteration.

K5: The final set of eigenvector approximations are printed.

K1 through K4 should be used only to diagnose abnormal results.

Suggested Technique: The RESET controls allow EIG to be used in many ways. The best approach for a particular case will depend on many factors, such as problem size, the number of eigensolutions required, the mode of execution (batch or interactive), etc. Accordingly, no fixed rules of operation can be given; however, a few general guidelines can be stated:

(1) If eigenvectors for a similar structure are known, use them as initial approximations. This is very useful in parametric studies of the effects of changes in section properties, mass distribution, etc.

(2) To compute modes of structures for which rigid body motion is possible, use the spectral shift method. The shift constant, c in eq. (9), should be selected within the expected range of eigenvalues associated with the elastic modes. Too small a value of c results in excessive roundoff errors due to K - cM being nearly singular.

(3) In the first execution of EIG, choose a small value for NDYN, perhaps as small as 2 or 3 for large structures. This will give the analyst useful information about the eigenvalue distribution in many cases, providing a basis for determining whether vectors should be added or deleted in the ensuing execution. For example, suppose you want to compute vibrational eigenvalues from .0 to 100.; and that you RESET INIT= 12, NDYN= 3. If it is found that only the lowest three eigenvalue approximations are less than 100., it would likely be reasonable in the next run to continue iterating on only the lowest five or six modes; e.g. RESET L1= 1, L2= 6.

Generally, if p modes are to be computed, at least 1.2 p vectors should be used in the EIG iterative process, since the higher eigenvalues may converge much more slowly than the interior (close to zero) eigenvalues, depending upon the spread of neighboring eigenvalues.

If good turnaround is available (e.g. via an interactive terminal), it is almost always best to proceed in a sequence of short executions to avoid wasting computer resources by iterating on either too many or too few vectors. Note that the default operation (@XQT EIG, followed by no input cards) is to resume iteration on all vectors computed and stored in a library in the last execution.

(4) If a very large number of modes is required, it may be cost effective to make several shifts and compute clusters of modes spanning the desired eigenvalue range. The resulting several data sets can subsequently be merged, if necessary.

Core Requirements. In the following discussion, m is the number of vectors being iterated, L is the block length of the factored K (or K surrogate), and J is the number of joints. There are two parallel requirements, the second of which will dominate in almost all cases:

(1) $3m^2 + 16m$, and

(2) $2m + L + B$

where B is a pool area usable only in integral multiples of 12J. The I/O activity of EIG will diminish linearly as B is increased in multiples of 12J.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

Section 9

DYNAMIC RESPONSE

9.1 PRELIMINARY INFORMATION

The following subsections contain general information concerning terminology and solution procedures used in the dynamic response processors.

9.1.1 Terminology

The purpose of this subsection is to define terms and symbols that will be used in subsequent explanations of the functions, input data, and control requirements of dynamic response subprocessors.

9.1.1.1 **Linear Systems. It is assumed that the reader is familiar with Rayleigh-Ritz methods of dynamic response analysis.**

The basic equation of system motion is as follows:

$$M\ddot{U} + D\dot{U} + KU = P, \tag{1}$$

where

U = system displacement state vector (e.g., a vector of joint motion components)

M = system mass matrix,

K = system stiffness matrix,

D = system damping matrix, and

P = applied loading,

Generalized coordinates are defined as follows:

$$U = Xq_x + Rq_r, \tag{2}$$

where

$$X = [X^1\ X^2 - - - X^{nx}] \quad , \quad q_x = \begin{bmatrix} q_x^1 \\ \vdots \\ q_x^{nx} \end{bmatrix},$$

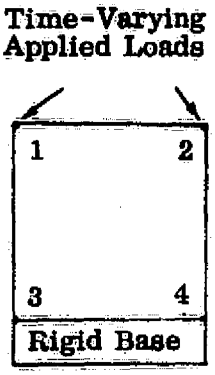
$$R = [R^1\ R^2 - - - R^{nr}] \quad , \text{ and } \; q_r = \begin{bmatrix} q_r^1 \\ \vdots \\ q_r^{nr} \end{bmatrix} \quad . \tag{3}$$

The X^i's are generalized functions (usually vibrational modes); and the q_x^i's are corresponding generalized coordinates.

Prescribed base motion, if any, is defined by Rq_r. The R^i's are rigid-body-motion vectors, corresponding to translation and rotation of a base point. The base point may be any joint in the structure.

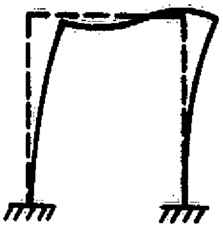
The base point and all other joints embedded in the rigid base must be constrained in the X^i's. If base motion is not prescribed, the X^i's are unrestricted.

Example. Suppose the motion of the base of the plane frame shown below is known, as are the time-varying forces acting on joints 1 and 2.



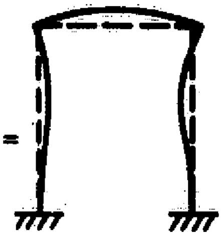
A typical selection of X's and R's would be as shown on the next page.

$X^1 =$



1st Elastic Mode

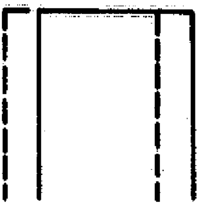
$X^2 =$



2nd Elastic Mode

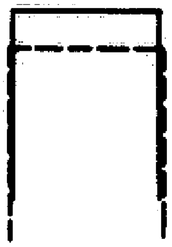
Etc.

$R^1 =$



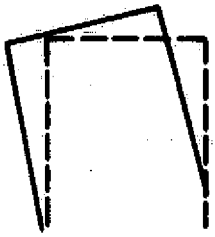
Translation

$R^2 =$



Translation

$R^3 =$



Rotation About Joint 3

Substituting Eq. (2) into Eq. (1), then premultiplying by X^t and R^t, respectively, gives Eqs. (4) and (5):

$$X^tM(X\ddot{q}_x + R\ddot{q}_r) + X^tD(X\dot{q}_x + R\dot{q}_r) + X^tK(Xq_x + Rq_r) = X^tP, \quad \text{or}$$

$$X^tMX\ddot{q}_x + X^tDX\dot{q}_x + X^tKXq_x = X^tP - X^tMR\ddot{q}_r \,. \tag{4}$$

$$R^tM(X\ddot{q}_x + R\ddot{q}_r) + R^tD(X\dot{q}_x + R\dot{q}_r) + R^tK(Xq_x + Rq_r) = R^tP, \quad \text{or}$$

$$R^tP = R^tMX\ddot{q}_x + R^tMR\ddot{q}_r \,. \tag{5}$$

In Eqs. (4) and (5), terms involving KR and DR are dropped because R contains only rigid-body-motion vectors.

The nodal force vector, P, includes both the known active applied loading and the unknown (reactive) forces associated with all joints attached to the base point. It will be convenient to define P_x, the nonzero elements of which are the known applied forces, and P_r, the nonzero terms of which are the unknown external force required to produce the specified base motion; that is:

$$P = P_x + P_r \tag{6}$$

The right side of Eq. (4) may be rewritten as follows:

$$X^t P = X^t P_x + X^t P_r = X^t P_x \tag{7}$$

It is a required characteristic of X that $X^t P_r = 0$, since all points fixed to the rigid base must be constrained in each X^i.

Using Eq. (6), Eq. (5) may be rewritten as follows:

$$R^t P_r = R^t M X \ddot{q}_x + R^t M R \ddot{q}_r - R^t P_x \tag{8}$$

If the R^i's involve unit base motions, $R^t P_r$ is the vector of base reactions.

P_x is generally defined in the following form:

$$P_x = [F^1 \; F^2 \; - - - \; F^{nf}] \begin{bmatrix} a^1(t) \\ a^2(t) \\ - \\ - \\ a^{nf}(t) \end{bmatrix} = Fa \tag{9}$$

The F^i's are applied force influence functions and the a^i's are corresponding amplitudes. Using Eqs. (7) and (9), Eq. (4) may be rewritten as follows:

$$X^t M X \ddot{q}_x + X^t D X \dot{q}_x + X^t K X q_x = X^t F a - X^t M R \ddot{q}_r \tag{10}$$

9.1.1.2 **Piecewise Linear Functions of Time in PLF Format.** This discussion is concerned with piecewise linear representation of functions of the following form, such as a and $\ddot{q}_r$ in Eq (10):

$$\begin{bmatrix} a^1(t) \\ a^2(t) \\ - \\ - \\ - \\ a^{nf}(t) \end{bmatrix}$$

PLF-format is the terminology used to refer to the following method of representing such functions as data sets within the data complex. Two data sets are involved, namely:

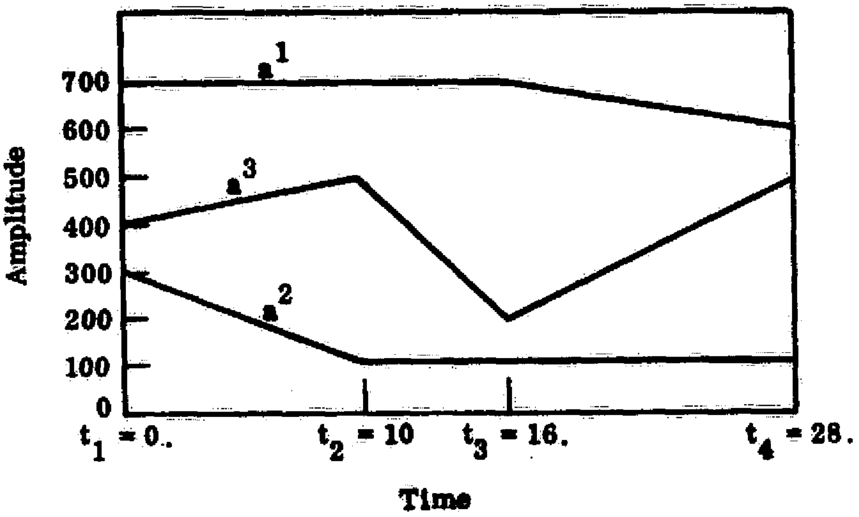
(1) TIME N2 ncase n4, and

(2) N1 N2 ncase n4

The meaning of N2, ncase, and n4 will vary, depending on the application. N1 will identify a particular function array, e.g., CA, to represent a in Eq. (10), or CQR2 to represent $\ddot{q}_r$, etc. The information contained in these data sets is summarized on the following page for the case of array a in Eq (10). The number of functions (a^i's) is nf, and the number of time points is nt.

Data Set	Value of NI, NJ in AUS/TABLE	Data Content
TIME N2 ncase n4	1, nt	$\begin{array}{cccc} j = 1 & j = 2 & & j = nt \\ \lbrack\ t_1 , & t_2 , & --- & t_{nt}\ \rbrack \end{array}$
CA N2 ncase n4	nf, nt	$\left[\begin{array}{cccc} a^1(t_1) , & a^1(t_2) , & --- & a^1(t_{nt}) \\ a^2(t_1) , & a^2(t_2) , & --- & a^2(t_{nt}) \\ - & & & \\ - & & & \\ a^{nf}(t_1) , & a^{nf}(t_2) , & --- & a^{nf}(t_{nt}) \end{array}\right] \begin{array}{c} i = 1 \\ i = 2 \\ - \\ - \\ i = nf \end{array}$

In the following example, nf = 3, nt = 4, N2 = X44, ncase = 70, and n4 = 1.



The following input to AUS/TABLE would represent the above information:

```
TABLE(NI= 1, NJ= 4): TIME X44 70 1$         nt = 4
        J= 1,4: .0, 10., 16., 28.$
TABLE(NI= 3, NJ= 4): CA   X44 70 1$         nf = 3, nt = 4

        I= 1: J= 1,3: 700.: J= 4: 600.$     a^1 defined
        I= 2: J= 1: 300.: J= 2,4: 100.$     a^2 defined
        I= 3: J= 1,4: 400., 500., 200., 500.$ a^3 defined
```

9.1.2 Matrix Series Expansion Method of Transient Response Computation

Given initial conditions for q and $\dot{q}$, Eq. (1) may be numerically integrated by repeated application of Eq. (2), to determine q and $\dot{q}$ at time $t = 0$, Δ, 2Δ, 3Δ, ---.

$$M\ddot{q} + D\dot{q} + Kq = Q \qquad (1)$$

$$\begin{bmatrix} q(t+\Delta) \\ \dot{q}(t+\Delta) \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} + \begin{bmatrix} N_{10} & N_{11} & N_{12} & -- & N_{1r} \\ N_{20} & N_{21} & N_{22} & -- & N_{2r} \end{bmatrix} \begin{bmatrix} Q(t) \\ \dot{Q}(t) \\ \ddot{Q}(t) \\ - \\ \frac{d^rQ}{dt^r} \end{bmatrix} \qquad (2)$$

Equation (2) involves the assumption that Q derivatives above order r vanish identically. For example, if all elements of Q are piecewise linear functions of time, $\ddot{Q}$ and higher derivatives are identically zero.

The W's and N's in Eq. (2) are defined in the following discussion. For simplicity, Eq. (1) is rewritten as

$$\ddot{q} = A\dot{q} + Bq + \eta$$

where

$$A = -M^{-1}D\ , \quad B = -M^{-1}K\ , \quad \text{and} \quad \eta = M^{-1}Q\ . \qquad (3)$$

The Taylor series expansions of q and $\dot{q}$ are

$$q(t+\Delta) = q(t) + \Delta\dot{q}(t) + \frac{1}{2}\Delta^2\ddot{q}(t) + ---$$

$$\dot{q}(t+\Delta) = \dot{q}(t) + \Delta\ddot{q}(t) + \frac{1}{2}\Delta^2\dddot{q}(t) + --- \qquad (4)$$

Using Eq. (3), higher derivatives of q may be expressed in terms of q and $\dot{q}$, as follows:

$$\begin{aligned}\dddot{q} &= A\ddot{q} + B\dot{q} + \dot{\eta} \\ &= A(A\dot{q} + Bq + \eta) + B\dot{q} + \dot{\eta} \\ &= (A^2 + B)\dot{q} + ABq + A\eta + \dot{\eta}\end{aligned}$$

Similarly,

$$\begin{aligned}\ddddot{q} &= (A^2 + B)\ddot{q} + AB\dot{q} + A\dot{\eta} + \ddot{\eta} \\ &= [(A^2 + B)A + AB]\dot{q} + (A^2 + B)Bq + (A^2 + B)\eta + A\dot{\eta} + \ddot{\eta}\end{aligned}$$

In general,

$$\frac{d^n q}{dt^n} = \overset{(n)}{q} = R_n q + P_n \dot{q} + P_{n-1}\eta + P_{n-2}\dot{\eta} + --- + P_1 \overset{(n-2)}{\eta} \qquad (5)$$

since

$$\overset{(n+1)}{q} = R_n \dot{q} + P_n(A\dot{q} + Bq + \eta) + P_{\eta-1}\dot{\eta} --- + P_1 \overset{(n-1)}{\eta} .$$

The recursion formulae for P_n and R_n are

$$\begin{aligned} P_{n+1} &= P_n A + R_n, \text{ and} \\ R_{n+1} &= P_n B, \text{ beginning with} \\ P_1 &= \text{Identity matrix, and} \\ R_1 &= \text{Zero matrix} \end{aligned} \qquad (6)$$

Substitution of Eq. (5) into Eq. (4) yields Eq. (7):

$$
\begin{aligned}
q(t + \Delta) \;=\;& q(t) + \Delta\dot{q}(t) \\
&+ \frac{\Delta^2}{2}\,[R_2 q(t) + P_2\dot{q}(t) + P_1\;(t)] \\
&+ \frac{\Delta^3}{3!}\,[R_3 q(t) + P_3\dot{q}(t) + P_2\eta(t) + P_1\dot{\eta}(t)] \\
&+ \frac{\Delta^4}{4!}\,[R_4 q(t) + P_4\dot{q}(t) + P_3\eta(t) + P_2\dot{\eta}(t) + P_1\ddot{\eta}(t)] \\
&+ \ldots.
\end{aligned}
$$

$$
\begin{aligned}
\dot{q}(t + \Delta) \;=\;& \dot{q}(t) \\
&+ \Delta\;[R_2 q(t) + P_2\dot{q}(t) + P_1\eta(t)] \\
&+ \frac{\Delta^2}{2}\,[R_3 q(t) + P_3\dot{q}(t) + P_2\eta(t) + P_1\dot{\eta}(t)] \\
&+ \frac{\Delta^3}{3!}\,[R_4 q(t) + P_4\dot{q}(t) + P_3\eta(t) + P_2\dot{\eta}(t) + P_1\ddot{\eta}(t)] \qquad (7) \\
&+ \ldots.
\end{aligned}
$$

If ℓ terms are taken in the series, the W's and N's of Eq. (2) are identified in Eq. (7) as follows:

$$W_{11} = I + \sum_{k=2}^{\ell} \frac{\Delta^k}{k!} R_k$$

$$W_{12} = \sum_{k=1}^{\ell} \frac{\Delta^k}{k!} P_k$$

$$W_{21} = \sum_{k=1}^{\ell} \frac{\Delta^k}{k!} R_{k+1}$$

$$W_{22} = I + \sum_{k=1}^{\ell} \frac{\Delta^k}{k!} P_{k+1}$$

and, for $j = 0, 1, 2, --- \ell - 2$,

$$N_{1j} = \left[\sum_{k=j+2}^{\ell} \frac{\Delta^k}{k!} P_{k-j-1} \right] M^{-1}$$

$$N_{2j} = \left[\sum_{k=j+1}^{\ell} \frac{\Delta^k}{k!} P_{k-j} \right] M^{-1} \tag{8}$$

Since 1962, the method described above has been used by the writer in a number of applications, both linear and nonlinear. Nonlinearities, if present, are included in Q .

Although a general analysis of convergence characteristics is beyond the scope of the present discussion, the following guidelines are suggested for application to linear systems. It is usually satisfactory to truncate the series at about ten terms [i.e., $\ell =$ 10 in Eq. (8)]. Where p_{min} is the shortest damped natural period of the system, the series indicated by Eq. (8) will diverge as Δ approaches 1/4 p_{min}. Accordingly, it is usually best to select Δ near 1/8 p_{min}. In the special case of an undamped, uncoupled system, the accuracy of the approximation associated with the suggested parameters (ℓ=10, Δ= 1/8 p_{min}) would be as follows:

$$\sin\theta \approx \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \frac{\theta^9}{9!}\ , \text{ and}$$

$$\cos\theta \approx 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} - \frac{\theta^{10}}{10!}$$

where $\theta = \pi/4$. The above approximations are accurate to about 7 decimal places.

9.2 DR – LINEAR DYNAMIC RESPONSE ANALYZER

Function. DR has no special RESET controls. All activities are initiated by commands of the following form:

$$\text{PROC}(p_1 = v_1, p_2 = v_2, \text{----}) ,$$

where PROC is a DR subprocessor name, such as DTEX, and the argument list is in the usual SPAR form, e.g., INLIB=3, CASE=14, etc. Details of subprocessor function, input requirements, output, core requirements, etc., are discussed in the following subsections.

Code Release Data. Level 9, July 1975, coded by W. D. Whetstone.

9.2.1 Transient Response of Uncoupled Systems

Subprocessors DTEX and TR1 implement the numerical integration procedure outlined in Section 9.1.2, for the special case in which:

- X^tMX, X^tDX, and X^tKX are diagonal matrices. As a result, the W's and N's in Eq. (2) of Section 9.1.2 also are diagonal.

- The generalized forces are piecewise linear functions of time. The generalized forces, Q (from Section 9.1.2), are generally identified as either or both of the terms on the right side of Eq. (10) in Section 9.1.1.1.

The correspondence between data set names recognized by these subprocessors and the terminology of Section 9.1 is summarized in Table 9.2.1-1. The normal flow of information among processors is shown on Fig. 9.2.1-1.

As indicated in Fig. 9.2.1-1, AUS is normally the source of all data input for DR subprocessors DTEX and TR1. The analyst, using the various tools available in AUS, creates the input data sets required for each particular analysis.

Typical DR control card input is as follows:

```
@XQT DR
     DTEX (optional parameters)
     TR1(optional parameters, e.g., CASE=26, --)
     TR1(---)
      -
      -
      -
```

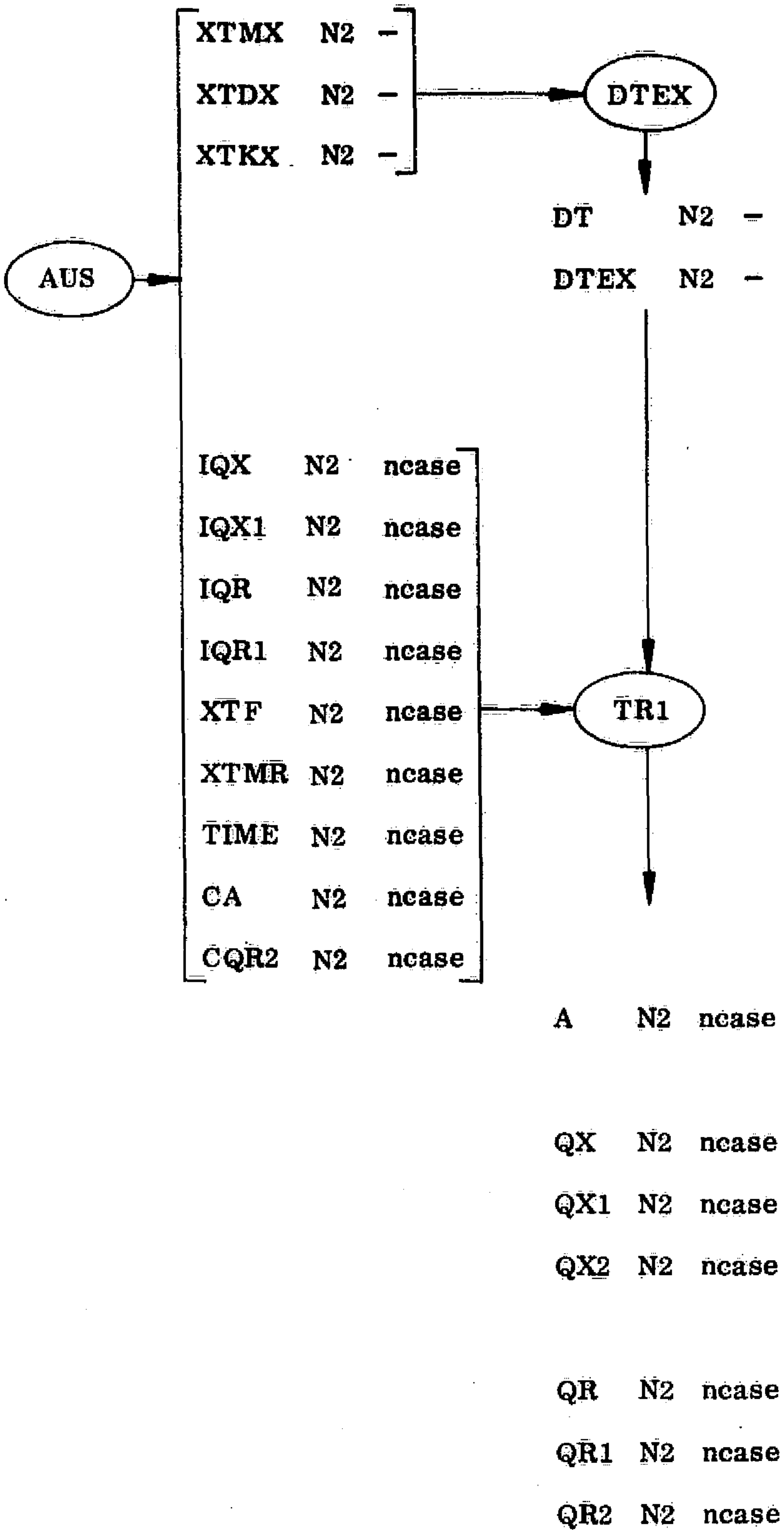


Fig. 9.2.1-1 Normal Flow of Information in Uncoupled Linear Transient Response

Table 9.2.1-1: Data Sets Involved in Transient Response of Uncoupled Systems

Data Set Name(*)			NWORDS	NJ	NI*NJ	Contents – See Eq. (10), Section 9.1.1.1	
XTMX	N2	–	nx	nx	nx	Diagonal matrices X^tMX, X^tDX, and X^tKX. The number of vectors in X is nx.	
XTDX	N2	–	nx	nx	nx		
XTKX	N2	–	nx	nx	nx		
DT	N2	–	1	1	1	Integration-step size, from DTEX.	
DTEX	N2	–	nx·11	1	nx	Diagonal matrices W_{11}, W_{12}, N_{10}, N_{11}, W_{21}, W_{22}, N_{20}, N_{21}, A, B, and M^{-1}, as defined in Section 9.1.2.	
IQX	N2	ncase	nx	1	nx	Initial q_x	
IQX1	N2	ncase	nx	1	nx	Initial $\dot{q}_x$	
IQR	N2	ncase	nr	1	nr	Initial q_r	
IQR1	N2	ncase	nr	1	nr	Initial $\dot{q}_r$	
XTF	N2	ncase	nx·nf	nf	nx·nf	X^tF. The number of vectors in F is nf.	
XTMR	N2	ncase	nx·nr	nr	nx·nr	X^tMR. The number of vectors in $\bar{R}$ is nr.	
TIME	N2	ncase	nt	nt	nt	In PLF format (see Section 9.1.1.2), TIME and CA define a TIME and CQR2 define $\ddot{q}_r$	
CA	N2	ncase	nf·nt	nt	nf·nt		
CQR2	N2	ncase	nr·nt	nt	nr·nt		
A	N2	ncase	n·nf	m	nf·m	a	TR1 output data sets. Each column corresponds to an integration time point. Block lengths are readjusted internally to synchronize the output.
QX	N2	ncase	n·nx	m	nx·m	q_x	
QX1	N2	ncase	n·nx	m	nx·m	$\dot{q}_x$	
QX2	N2	ncase	n·nx	m	nx·m	$\ddot{q}_x$	
QR	N2	ncase	n·nr	m	nr·m	q_r	
QR1	N2	ncase	n·nr	m	nr·m	$\dot{q}_r$	
QR2	N2	ncase	n·nr	m	nr·m	$\ddot{q}_r$	

*The user selects both the N2 (alphanumeric) and ncase (integer) identifiers. The fourth word of the name of all these data sets is masked by the DTEX and TR1 subprocessors.

9.2.1.1 DTEX Command. The function of the DTEX subprocessor is as indicated on Fig. 9.2.1-1 and Table 9.2.1-1. DTEX is entered via the following statement:

$$\mathrm{DTEX}(p_1 = \; v_1 , \; p_2 = \; v_2 \; ----)$$

DTEX control parameters are summarized below.

Parameter Name	Default Value	Meaning
INLIB	1	Data source library. If data set XTDX N2 is not present, an undamped system is assumed.
N2	MASK	N2 of input data set name XTMX N2 --. Output data set N2 is the same as in XTMX N2 --.
DT		Integration step size. Default is one eighth of the smallest undamped period.
NTERMS	10	Number of terms in matrix series expansion.
OUTLIB	1	Destination library for output data sets.

Core Requirements. Data space requirements are approximately 22 nx.

9.2.1.2 The TR1 Command. Figure 9.2.1-1 and Table 9.2.1-1 indicate the function of TR1. TR1 is entered through the following command.

$$TR1(p_1= v_1, p_2= v_2,---)$$

The control parameters are summarized below:

Parameter Name	Default Value	Meaning
INLIB	1	Data source library.
N2	MASK	N2 of input data sets. Default N2 for output data set is the same as in DTEX N2.
CASE	1	Value of ncase.
T1	–	Numerical integration start/stop times. Defaults are beginning and ending times in TIME N2 ncase.
T2	–	
ALIB	1	Destination libraries for output data sets A, QX, QX1,---. No output is produced if the associated library is 0, or if the quantity is not defined (i.e., if no CQR2 exists, QR2 will not be produced).
QXLIB	0	
QX1LIB	0	
QX2LIB	1	
QRLIB	0	
QR1LIB	0	
QR2LIB	1	
LB	896	Nominal block length of output data sets.

Figure 9.2.1-1 lists all of the input data sets recognized by TR1. If any of these are not present in the data source library, the associated information is considered to be null (all zero's). That is,

- If IQX is not present, generalized displacements, q_x, are zero at time = T1.

- If IQX1 is not present, $\dot{q}_x = 0$ at time = T1.

- If IQR is not present, $q_r = 0$ at time = T1.
- If IQR1 is not present, $\dot{q}_r = 0$ at time = T1.
- If XTF is not present, $X^tF = 0$.
- If XTMR is not present, there is no prescribed base motion.
- If CA is not present, XTF is not present.
- If CQR2 is not present, XTMR is not present.

In performing the numerical integration procedure indicated by Eq. (2), Section 9.1.2, TR1 assumes that the generalized forces are piecewise linear; i.e., $\ddot{Q}$ and higher derivatives are identically zero, within each time step, Δ. The value of $\dot{Q}$ during the time interval $j\Delta \leq t \leq \Delta + j\Delta$, is assumed to be

$$\dot{Q}_i = \frac{1}{\Delta}\left[Q_i(\Delta + j\Delta) - Q_i(j\Delta)\right]$$

where Q_i and $\dot{Q}_i$ are the ith elements of Q and $\dot{Q}$. This approximation should be taken into account when selecting time points for PLF input, and in choosing DT in DTEX, although it will rarely introduce significant error.

<u>Core Requirements.</u> The required data space is approximately the sum of the following:

11 nx (to accommodate DTEX N2)

+ one block of each input and output data set

9.2.2 Back Transformation Via the BACK Command

This subprocessor is entered through the following command:

$$\text{BACK}(p_1 = v_1 , p_2 = v_2 , \text{----})$$

The function of BACK is to perform a transformation of the following form:

$$Z = Z_c + T_1Y_1 + T_2Y_2 + T_3Y_3 + \text{---} + T_fY_f \qquad (1)$$

The output array Z is a function of time. Z contains nz rows. Z_c is a constant vector. The T_k's are transformation matrices, and the Y_k's are functions of time, represented by data sets such as QX, QX1, QX2, QR, etc., produced by DR/TR1. Each Y_k must contain data for nt time points. T_k contains nz rows (one for each output quantity), and ny_k columns, where ny_k is the number of rows in Y_k . The Y's may reside in multi-block data sets, but the T's must be single block data sets. The T's are usually arrays of stress or displacement eigenvector components constructed via AUS/TABLE/TRAN.

Output consists of either or both of the following:

(1) The time history of Z , in a format similar to that of Q, QX, etc.

(2) Maximum and minimum values of each element of Z, and the times of occurrence.

After the BACK($p_1 = v_1$,---) command, additional commands of the following types may appear.

Command	Meaning
ZC= Data set i.d.[1,2]	Identifies Z_c. $Z_c = 0$ if this command does not appear.

T= Data set i.d.[1,2]
Y= Data set i.d.[1,2]
	T identifies a transformation, T_k. This command must be immediately followed by a Y= Data set i.d. command. Any number of such pairs of commands is allowed.

Z= Data set i.d.[3,2]	Name of output data set containing the full time history of Z. This data set will not be produced if this command is not present.
EXT= Data set i.d.[3,2]	Name of output data set containing extreme values of Z elements. This data set will not be produced unless this command is present.

(1) Input data set i.d.'s are in the following form (underlined words are optional):

C, Lib, N1, N2, n3, n4, nskip

The floating-point constant C, which multiplies the named data set, may be omitted (i.e., default C= 1.0). If Lib is omitted, the data set is in the nominal SOURCE library designated in the BACK command. For Y = ---, nskip means that matrix Y is defined to begin after the first nskip vectors (time points) in the data set.

(2) Omitted trailing parts of data set names are MASK-filled.

(3) Output data set i.d.'s are in the following form:

Lib N1, N2, n3, n4

If Lib is omitted, the output goes to the library designated by the DESTINATION parameter in the BACK command.

The following control parameters may be defined within the BACK(p_1= v_1,---) statement:

Parameter Name	Default Value	Meaning
SOURCE	1	Nominal data source library.
DEST	1	Nominal destination library for output data sets.
M1	1	Z's are computed for time points M1, M1+1,---M2. Default M2 is the last time point, which must be the same in each Y_k.
M2		
TSTART	.0	The time-of-occurrence data associated with maximum values of Z elements is controlled by these parameters. TSTART is the time corresponding to M1, and DT is the time increment between successive time points. If DT is not given here, an attempt will be made to locate it in a data set named DT N2 MASK MASK, where N2 is the second word of the name of the data set appearing on the last Y= --- command.
DT	1.0	
PRINT	1	Any nonzero value causes direct printout of max/min and time-of-occurrence data.
BIG	$1. \times 10^{20}$	Number used to initialize max/min scan.
FMAX	$-1. \times 10^{20}$	If max/min data are printed, data for an element will be displayed only if the maximum exceeds FMAX or the minimum is less than FMIN.
FMIN	$1. \times 10^{20}$	
LRZ	896	Nominal block length for Z output, if any.
N3REPEAT	1	The effect of these parameters is to cause the entire transformation process to be repeated for multiple response cases. Each repetition involves incrementing (by one) n3 and n4 of the names of all input and output data set names, <u>except</u> for the T_k's.
N4REPEAT	1	

On successive executions of BACK (within the same DR execution), the parameters defined above retain the values they had during the immediately preceding execution except for FMAX, FMIN, BIG, TSTART, and DT, which revert to the default values indicated above.

Core Requirements. The required data space is approximately

$$\begin{aligned} & 5 \text{ times nz} \\ & + \sum_{k=1}^{f} (\text{Block length of } T_k + \text{block length of } Y_k) \\ & + \text{Block length of } Z \text{ output, if any.} \end{aligned}$$

It should be noted that BACK is designed to function efficiently when transforming a limited number of quantities; i.e., nz should not often exceed a few hundred, and should be held as small as possible if many time points are involved.

Example:

Track inertia forces.

```
BACK: Z= INERTIA  FORCES 1 1: T= MX: Y= QX2$
```

Section 10

GRAPHICS

Figure 10-1 illustrates the function of plot programs PLTA and PLTB, which are described in detail in Section 10.1 and 10.2. It is suggested that the new user examine the examples shown in Section 10.3 before reading 10.1 and 10.2.

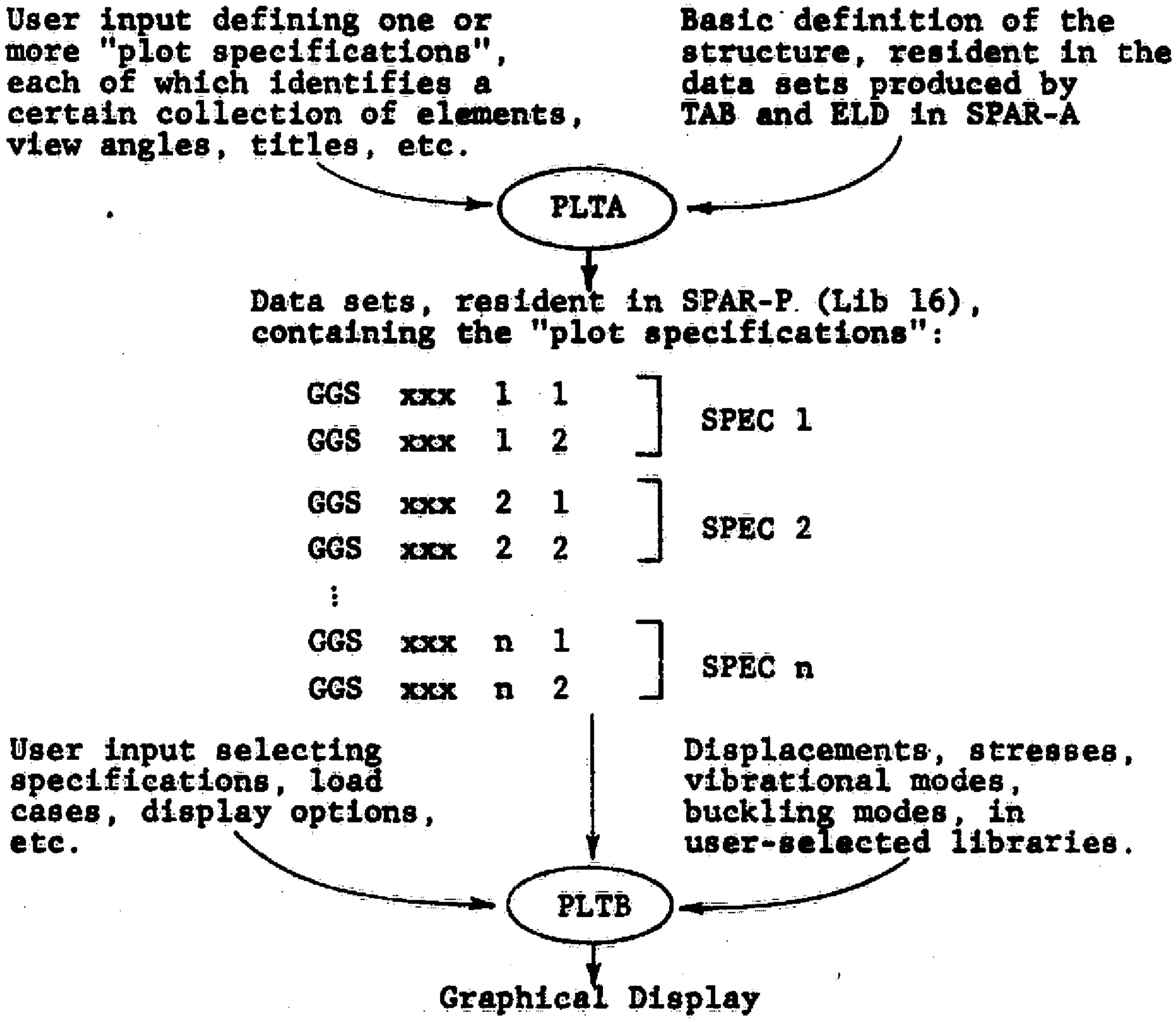

User input defining one or more "plot specifications", each of which identifies a certain collection of elements, view angles, titles, etc.
Basic definition of the structure, resident in the data sets produced by TAB and ELD in SPAR-A
PLTA
Data sets, resident in SPAR-P (Lib 16), containing the "plot specifications":
GGS xxx 1 1
GGS xxx 1 2
SPEC 1
GGS xxx 2 1
GGS xxx 2 2
SPEC 2
⋮
GGS xxx n 1
GGS xxx n 2
SPEC n
User input selecting specifications, load cases, display options, etc.
Displacements, stresses, vibrational modes, buckling modes, in user-selected libraries.
PLTB
Graphical Display

Fig. 10-1 PLTA - PLTB Data Exchange

10.1 PLTA - PLOT SPECIFICATION GENERATOR

Function. As shown on Fig. 10-1, PLTA is used to produce data sets containing plot specifications. The data input sequence is as follows:

@XQT PLTA

SPEC $nspec_1$

Optional control statements establishing view angles, joint labels, titles, etc.

Geometric composition commands establishing which elements are to be shown and/or which joints are to be interconnected by lines

Input defining plot specification $nspec_1$

SPEC $nspec_2$

Optional control statements

Geometric composition commands

Specification $nspec_2$

.
.
.

The optional control statements and the geometric composition commands are explained in Sections 10.1.1 and 10.1.2, respectively. The SPEC command may also have the form SPEC, nspec, n, in which case the group title (see ELD) for group n of the last type of element named in the geometric composition commands will appear as line 1 of the plot specification title. (See also the STITLE command in Section 10.1.1).

Table PLTA-1 summarizes all PLTA commands.

Table PLTA-1: Summary of PLTA Commands

SPEC	nspec, n

Optional control statements:

STITLE'	Title line 1
S2TITLE'	Title line 2
TEXT'	Text line
SYM	iaxis
ANTISYM	iaxis
ROTATE	a_1, i_1, a_2, i_2, a_3, i_3
LROTATE	a_1, i_1, a_2, i_2, a_3, i_3
VIEWS	v_1, v_2, v_3, n
LOCLABEL	iquad, ix, iy
JLABEL	j_1, i_1, j_2, i_2, j_3, i_3, - - -
AXES	o_1, o_2, o_3, ℓ_1, ℓ_2, ℓ_3
MARGIN	percent

Geometric Compostion:

ALL		
FORMAT 0:	EIJ	index_1 index_2 ijump group_1 group_2 gjump
FORMAT 1:	EIJ	$\text{group}_1/\text{group}_2$, list of indexes
FORMAT 2:	EIJ	$\text{index}_1/\text{index}_2$, list of groups

CONNECT	j_1 j_2 j_3 - - -

LCONTROL	jshift, nrepeat, inc
LINES	j_1 j_2 j_3 - - -

Reset Controls.

Name	Default Value	Meaning
GGSL	16	Destination Library, plot specifications
SDEF	21	Destination Library, temporary data sets produced by PLTA

Core Requirements. The working data space requirement is approximately the following.

3500 + 8 * (the number of joints in the structure)
+ the number of words in the ALTREF data set (from TAB)

Code Release Data. Level 10, March, 1976, Univac, CDC.

10.1.1 Optional Control Parameters

Statement Form	Meaning
STITLE' 40 characters S2TITLE' 40 characters	Lines 1 and 2 of the plot specification title.
TEXT' 70 characters	A line of text to appear on a separate frame preceeding the frame(s) generated by this specification. A maximum of 41 TEXT statements are permitted. Default is no TEXT.
SYM iaxis ANTISYM iaxis	These commands may be used if the finite element model is a half-model of a symmetric structure, with the symmetry plane normal to the global axis iaxis. To plot symmetrical modes or static deformations of the complete structure, select SYM= iaxis. To plot antisymmetrical deformations, select ANTI= iaxis.
VIEW= view LROTATE d_1 i_1 d_2 i_2 d_3 i_3 (d_2 i_2 and d_3 i_3 are optional.) *(See also the ROTATE command on Page 10.1-6)*	These commands control view angles. Through the LROTATE command, the user specifies the orientation of a plot reference frame, as follows: Beginning with the plot reference frame coincident with the system global frame (1) rotate the plot reference frame d_1 degrees about its i_1 axis, (2) from the resulting position, rotate the plot reference frame d_2 degrees about its i_2 axis, then (3) from the resulting position, rotate the plot reference frame d_3 degrees about its i_3 axis. All of the d's are integer degrees.

Statement Form (Cont.)

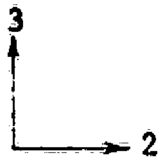
Meaning (Cont.)

For example,

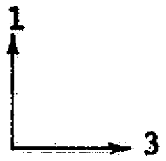
LROTATE 15,3, 20,1, 70,2 means:

- Rotate 15 degrees about the 3-axis, then
- Rotate 20 degrees about the 1-axis, then
- Rotate 70 degrees about the 2-axis, to establish the orientation of the plot reference frame.

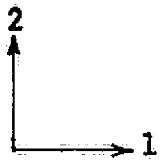
The VIEW parameter, which may assume values of 1,2,3,-1,-2 or -3, designates which of the following six views is to be displayed. The axes shown are those of the plot reference frame:



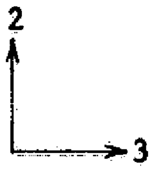
VIEW=1



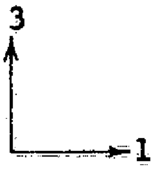
VIEW=2



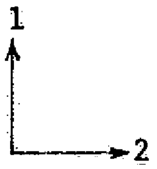
VIEW=3



VIEW=-1



VIEW=-2



VIEW=-3

To obtain 2 views on separate frames, the following command is used:

VIEWS= view_1, view_2

For 3 views, each on a separate frame, the command is

VIEWS= view_1, view_2, view_3

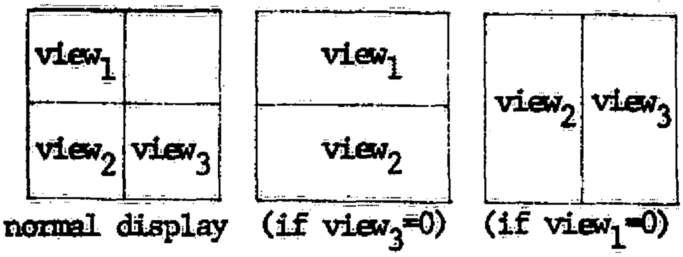
Statement Form (Cont.)

Meaning (Cont.)

To obtain multiple views on a single frame, use the following command:

VIEWS= view_1, view_2, view_3, 1

The resulting form of display is as follows:



normal display (if view_3=0) (if view_1=0)

The default is VIEWS= 1, 2, 3.

ROTATE d_1 i_1 $\underline{d_2\ i_2}$ $\underline{d_3\ i_3}$
(d_2 i_2 and d_3 i_3 are optional.)

This is an alternate method of prescribing the orientation of a plot reference frame, as follows:

Beginning with the plot reference frame coincident with the system global reference frame (1) rotate the global reference frame d_1 degrees about its i_1 axis, (2) from the resulting position rotate the global reference frame d_2 degrees about its i_2 axis, (3) from the resulting position, rotate the global reference frame d_3 degrees about its i_3 axis. All of the d's are integer degrees.

AXES o_1 o_2 o_3 ℓ_1 ℓ_2 ℓ_3

A set of coordinate axes are plotted parallel to the global reference frame axes. The origin will be located at (o_1, o_2, o_3). The length of the plotted axes will be ℓ_1, ℓ_2, and ℓ_3.

MARGIN= percent (integer)

Percentage of the plotted frame allotted for margin.
Default is MARGIN= 12

Statement Form (Cont.)

```
LOCLABEL    iquad, ix, iy
JLABEL      j1 l1, j2 l2, ---
```

Meaning (Cont.)

An alphameric label, l_1, will be displayed next to joint j_1, l_2 next to j_2, and so forth. Each label must begin with a letter, and may not exceed 4 characters. The following shows the position of a joint label, ABCD, relative to the joint (✦), if ix and iy are zero.

```
iquad=2  ABCD✦ABCD  iquad=1
         ABCD ABCD
iquad=3             iquad=4
```

Labels may be shifted by raster counts ix and iy from the nominal positions shown above. Default LOCLABEL= 4, 0, 0.

LOCLABEL and JLABEL commands may be repeated, and remain in effect throughout PLTA execution. For example:

```
   :
SPEC     3
JLABEL   10, X400
   :
SPEC     17
```

(the label X400 will still be be attached to joint 10).

To remove all joint labels previously defined, insert the command JLABEL , followed by no j_1 l_1 - - - parameters.

10.1.2 Geometric Composition Commands

The command ALL indicates that the entire finite element model is to be included in the specification. Otherwise, selected finite elements to be included are indentified by element type, group and index. Optional formats are provided for selecting specific elements. The command statement, FORMAT= f, designates the prevailing format option. Default is FORMAT= 1. A format designation will remain in effect for subsequent specifications until a FORMAT command is encountered. Switching back and forth among formats is unrestricted.

Where EIJ indicates the element type (E21, E43, etc.) and group and index refer to group and index numbers, element selection is achieved through the following commands:

FORMAT=0: EIJ index_1, index_2, ijump, group_1, group_2, gjump
FORMAT=1: EIJ $\text{group}_1/\text{group}_2$, (list of indexes)
FORMAT=2: EIJ $\text{index}_1/\text{index}_2$, (list of groups).

Example:

Geometric Composition Commands	Selected Group Nos.	Element Index Nos.
FORMAT=0	2	3, 6, 9, 12
E33 3, 12, 3, 2, 10, 4	6	3, 6, 9, 12
	10	3, 6, 9, 12
FORMAT=1	5	16, 27, 28, 29, 30
E21 5/7, 16, 27/30	6	16, 27, 28, 29, 30
	7	16, 27, 28, 29, 30
E43 4	4	all group 4 E43's
E31 1, 7, 8, 10, 12	1	7, 8, 10, 12
FORMAT=2 E41 26/30, 6	6	26, 27, 28, 29, 30
E23 7, 3/4	3	7
	4	7

Two additional geometric composition commands are provided for generating lines interconnecting specified joints.

Lines Command	Meaning
CONNECT j_1 j_2 . . . j_n	Line images are constructed connecting joint j_1 to j_2, j_2 to j_3, . . . j_{n-1} to j_n.
LCONTROL jshift nrepeat inc LINES j_1 j_2 . . . j_n	One or more LINES commands build up a joint number sequence which is manipulated by an LCONTROL command (shifted, repeated, incremented) to construct a joint number array to be interconnected on the plot.

Example:

Commands	Interconnected Joints
CONNECT 402/404, 411	402, 403, 404, 411
LCONTROL 10, 3, 100 LINES 2/4, 9, 11 LINES 22/24	sequence 1: 12 13 14 19 21 32 33 34 sequence 2: 112 113 114 119 121 132 133 134 sequence 3: 212 213 214 219 221 232 233 234
CONNECT 128/130, 140 CONNECT 152, 178	128, 129, 130, 140, 152, 178

10.2 PLTB - PRODUCTION OF GRAPHICAL DISPLAYS

Function. As shown on Figure 10-1, PLTB (or PLTB/TEK when using Tektronix scopes on the U-1110), is used to produce graphical displays. To cause images corresponding to plot specifications $spec_1$ through $spec_2$ to be displayed, the following command is given:

PLOT $spec_1$, $spec_2$

The form of display resulting from a PLOT command will depend on the current values of an array of execution control parameters which the user selects through the control statements summarized below. The PLOT statement and all control statements may appear any number of times during the single PLTB execution.

Control Statement	Meaning
DISPLAY=UNDEformed, STATic deformation, VIBRational mode, BUCKling mode, or	Display mode selection. Default is DISPlay=UNDEformed.
DISPLAY=SX, TXY, . . .	Selected stress or internal load data is displayed. See examples in Section 10.3. A complete list of available stress quantity display symbols is given in Table 10.2-1. The following form is also permitted (underlined quantities may be omitted): DISPLAY=SX/div, node, loc, TXY ... SX is divided by div. Div must be greater than or equal to 1.

Control Statement (Cont.)	Meaning (Cont.)
	node indicates the element node (1, 2, etc.) at which the stress is to be evaluated. For 3 and 4 node elements, node 0 is the center of the element. For 3 and 4 node elements, loc values of 0, 1, and -1 indicate mid, outer, and inner surfaces, corresponding to points C, A, and B (in order) on Figure 7.3-1 (PSF). NOTE: Stress data may be plotted only for complete element stress data sets produced by GSF. That is, the user must not restrict GSF output to a limited number of element groups, if it is to be read by PLTB.
DNORM=dnorm	When plotting deformed structures, (i.e., if DISPLAY=STAT, or VIBR, or BUCK), joint displacements are normalized to dnorm. This command must be given, since there is no default value.

INLIB=inlib
SET=nset
CON=ncon
CASES=case_1, case_2
or
VECTORS=vect_1, vect_2
(The control statements CASES and VECTORS are synonymous.)
	The following source data, if needed as a result of the prevailing DISPLAY statement, will reside in inlib.
	STAT DISP nset ncon
	VIBR MODE nset ncon
	BUCK MODE nset ncon
	STRS EIJ nset i,
	for i=case_1, case_1+1, --- case_2
	Default values are INLIB=1, SET=1, CON=1, CASES=1.

OPTIONS=n_1, n_2 - - -	List of options. See Table 10.2-2

Table 10.2-1 Summary of Available Stress Display Symbols

		Applicable Element Type				
			E31	E32	E33	
Symbol	Meaning	E21	E41	E42	E43	E44
SMAX	Maximum P/A + My/I beam stress	X				
SMIN	Minimum " " " "	X				
P/A	Axial beam stress	X				
S1	Dir. 1 beam shear stress	X				
S2	Dir. 2 " " "	X				
TS	Beam twisting stress	X				
SX	Normal stress, element x-dir.		X	X	X	
SY	" " " y-dir.		X	X	X	
TXY	In-plane shear stress		X	X	X	X
PS1	Maximum principal stress		X	X	X	
PS2	Minimum " "		X	X	X	
TMAX	Maximum shear stress		X	X	X	
ANG	Angle between x-axis and PS1 vector		X	X	X	
NX	Normal stress resultant, x-dir.		X		X	
NY	" " " y-dir.		X		X	
NXY	In-plane shear stress resultant		X		X	X
PN1	Maximum principal stress resultant		X		X	
PN2	Minimum " " "		X		X	
NMAX	Maximum shear stress resultant		X		X	
NANG	Angle between x-axis and PN1 vector		X		X	
MX	Bending stress resultant, x-dir.			X	X	
MY	" " " y-dir.			X	X	
MXY	Twisting stress resultant			X	X	
QX	Transverse shear resultant, x-dir.			X	X	
QY	" " " y-dir.			X	X	

Table 10.2-2 Meaning of OPTION Numerical Codes

Description of Option	Option Numeric Code	
Specification Control		
Plot error free specifications only	1	
Plot all specifications ignoring error status	2	
Plot all specifications appearing on a single PLOT command to the same scale	3	
Frame Labeling		
Omit deformation identification label	4	
Omit specification titles	5	
Omit "SPEC" identification	6	
Omit "SCALE"	7	
Omit all frame labeling	8	
Collapse margin and omit all labels	9	
Geometric Construction		
Dotted deformed structure	24	
Curved lines, deformed structure	25	
Superimpose deformed/undeformed structures	26	
Dotted undeformed structure	27	
Plot Content	Large Char.	Small Char.
Joint numbers displayed	10	11
Joint elimination order displayed	12	13
Joint labels displayed	14	15
Element index numbers displayed	16	17
Element group-index numbers displayed	18	19
Element section property group displayed	20	21
Element stress display size	28	29
3-Node elements are shaded	22	(no size control)
4-Node elements are shaded	23	(no size control)

Notes:

Select no more than one from options 10,11,12,13.
Select no more than one from options 14,15.
Select no more than one from options 16,17,18,19,20,21,28,29.
Options 16-21 may only be used in conjunction with DISPLAY=UNDEformed.

Reset Controls. There are no special reset controls.

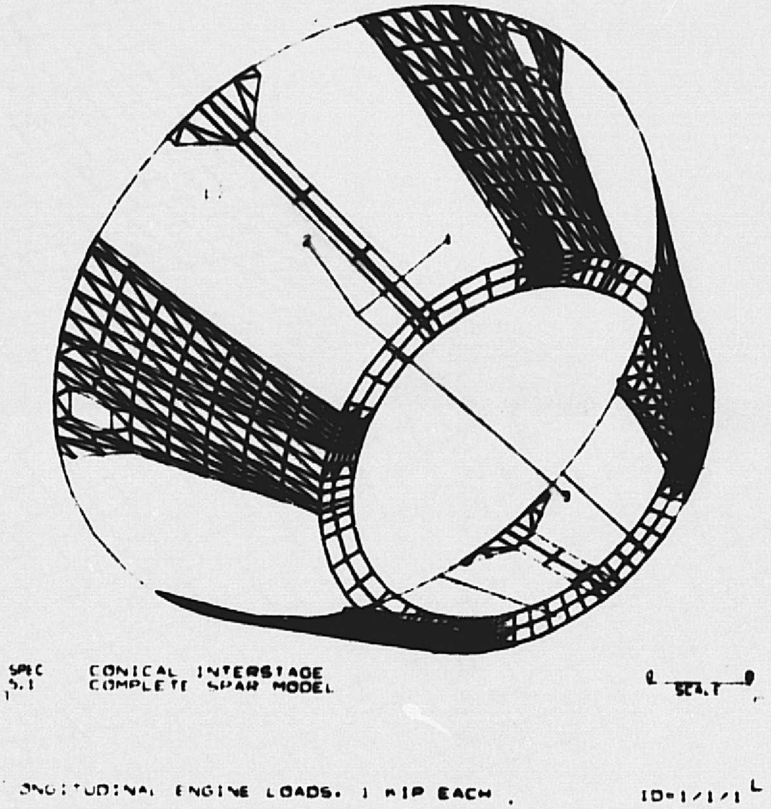
Core Requirements. Where J is the number of joints in the structure, the working data space requirements are as follow.

For plotting undeformed structures:	2000 + J
For plotting deformed structures:	2000 + 13 J
For plotting stresses:	2000 + J + the length of one block of stress input data.

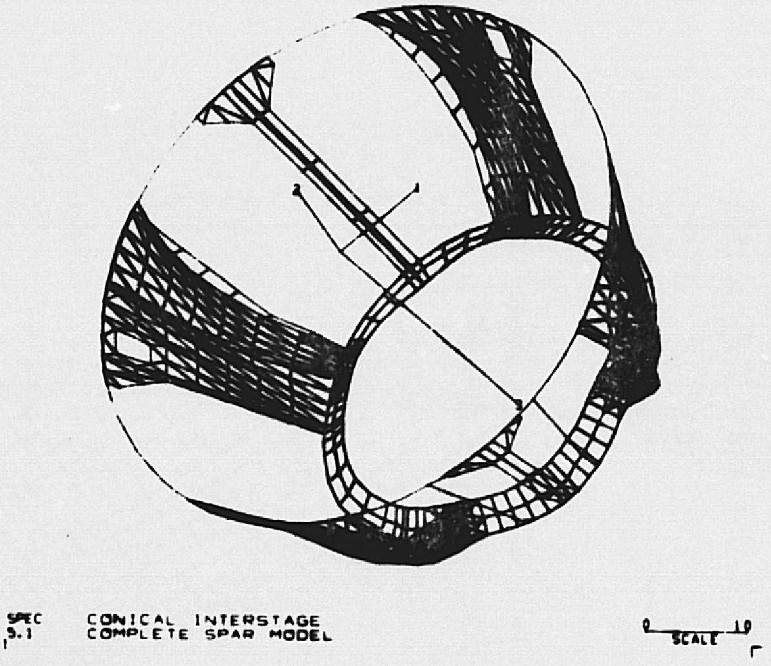
Code Release Data. Level 10, March, 1976, Univac, CDC.

10.3 GRAPHICS EXAMPLES

The following plots illustrate various examples of PLTA and PLTB executions. The plotted images were created with a COMp 80 microfilm system.



A

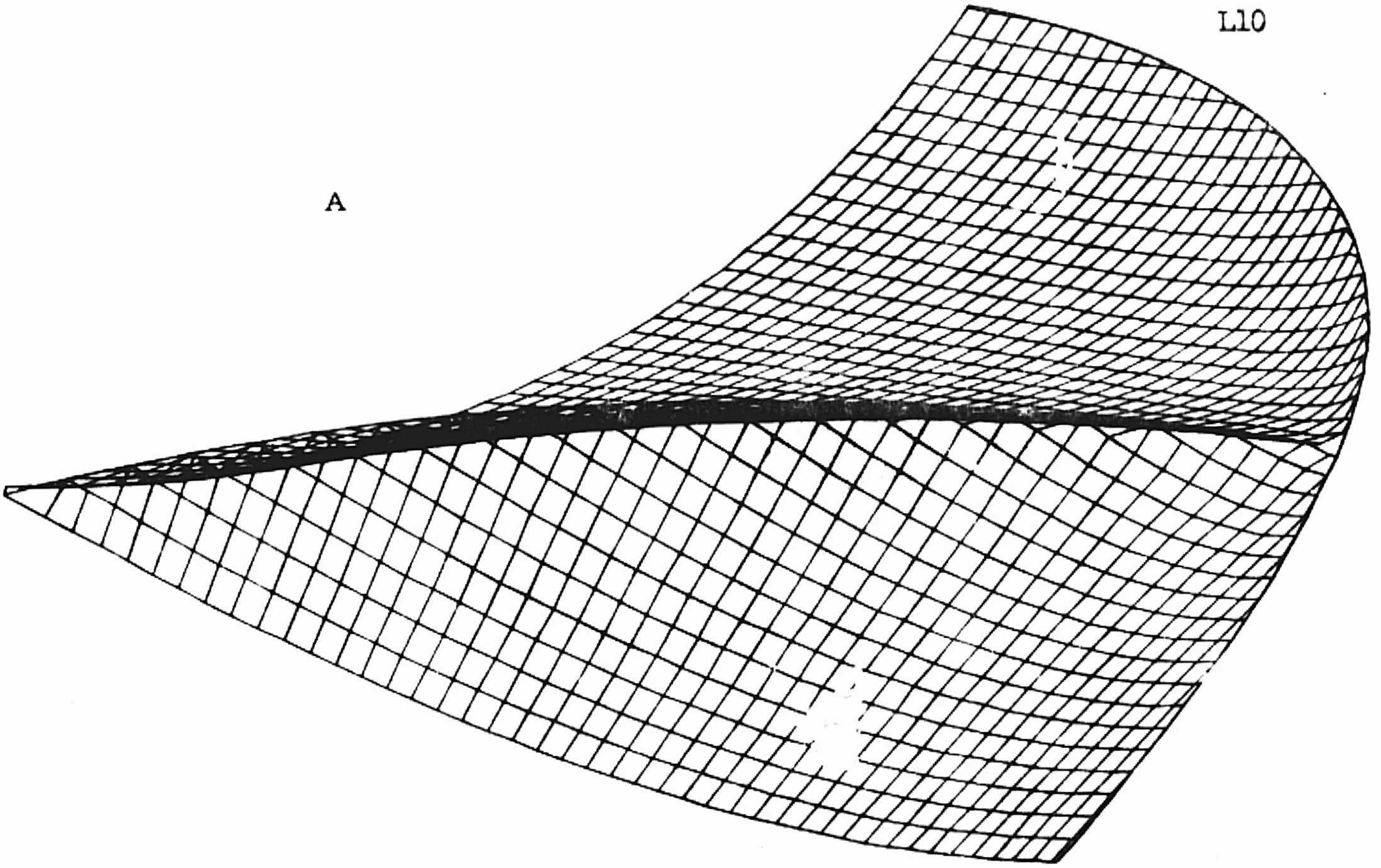


B

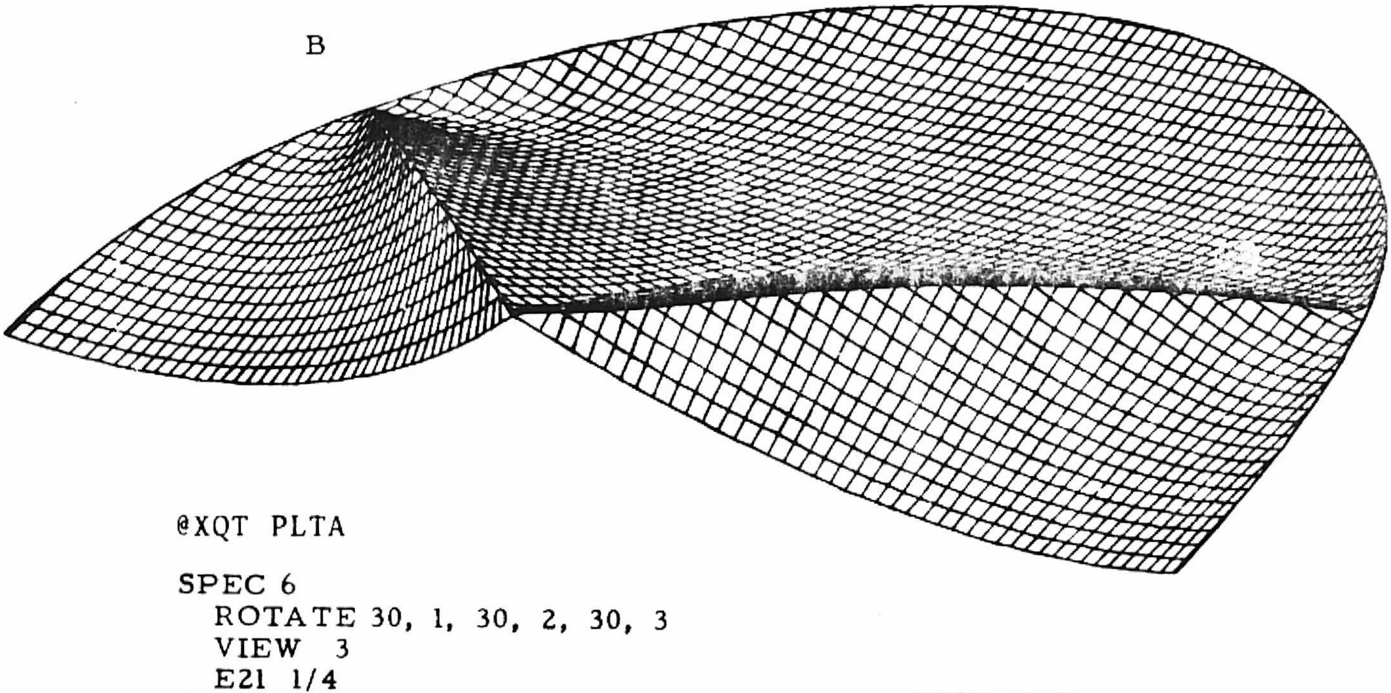
```
@XQT PLTA

SPEC 5
  STITLE' CONICAL INTERSTAGE
  S2TITLE' COMPLETE SPAR MODEL
  ROTATE 30, 1, 30. 2, 30, 3
  VIEW  3
  AXES 0. 0. 382. 10. 10. 35.
  ALL
```

```
@XQT PLTB

  RESET CORE= 40000
  DISPLAY= UNDEFORMED
PLOT 5 $ Frame A
  DISPLAY= STATIC DEFORMATION
  CASES= 1
  DNORM= 5.
PLOT 5 $ Frame B
```

A



B

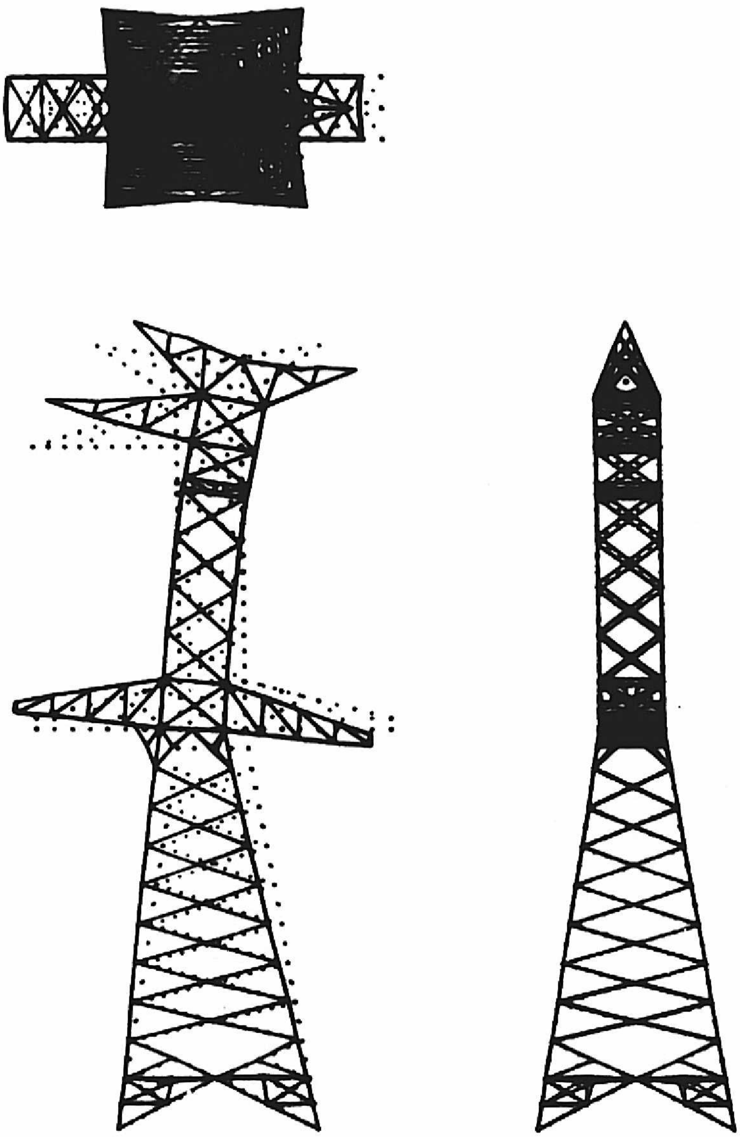


```
@XQT  PLTA

SPEC 6
  ROTATE 30, 1, 30, 2, 30, 3
  VIEW  3
  E21  1/4
SPEC 7
  ROTATF 30, 1, 30, 2, 30, 3
  VIEW  3
  SYM 2
  E21  1/4
```

```
@XOT  PLTB

  OPTIONS 9
  PLOT 6, 7
```

L10

```
VIBRATIONAL MODE.   FREQ (HZ)    .619982X10+01   ID=100/1/12
```

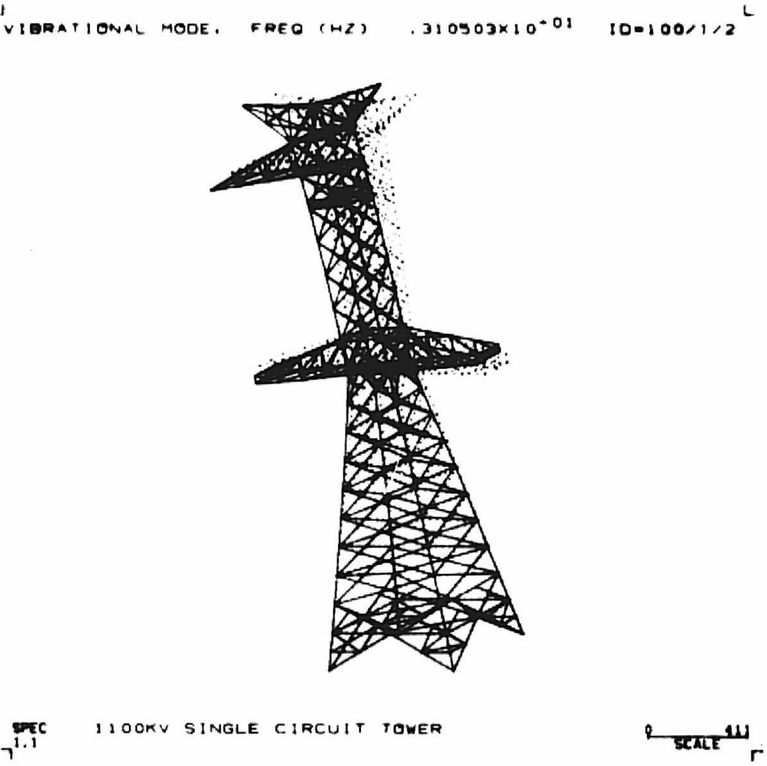


```
SPEC      1100KV  SINGLE  CIRCUIT  TOWER
2.1
```

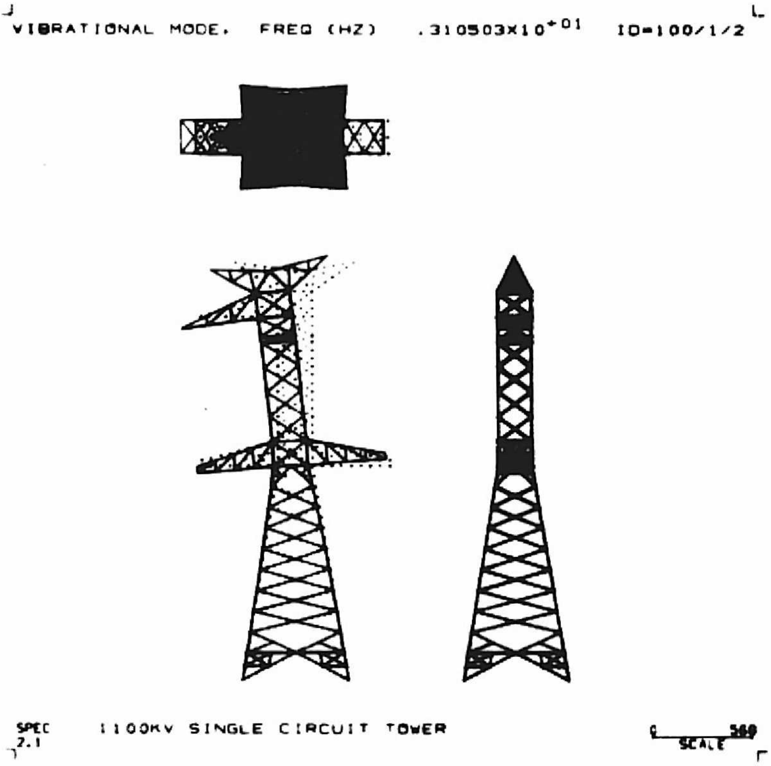
```
@XQT PLTA

SPEC 2
 STITLE' 1100KV  SINGLE CIRCUIT  TOWER
 VIEW  3, -2, 1, 1
 ALL
```

```
@XQT PLTB

 DISPLAY=  VIBRATIONAL MODES
 SET= 100
 CASES= 12
 DNORM= 50.
 OPTIONS= 26, 27
PLOT 2
```

A

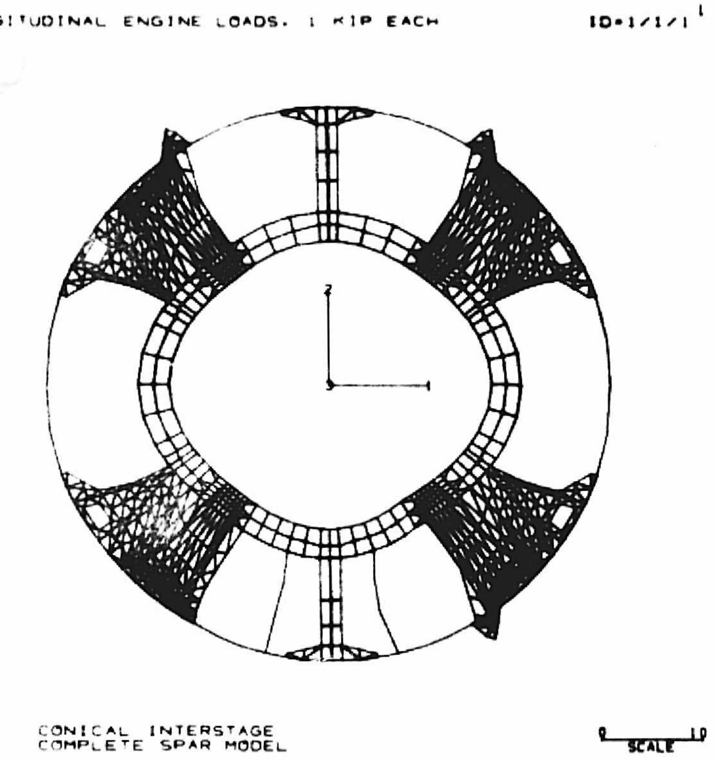

VIBRATIONAL MODE. FREQ (HZ) .310503X10+01 ID=100/1/2
SPEC 1.1
1100KV SINGLE CIRCUIT TOWER
0 SCALE 411

B


VIBRATIONAL MODE. FREQ (HZ) .310503X10+01 ID=100/1/2
SPEC 2.1
1100KV SINGLE CIRCUIT TOWER
0 SCALE 569

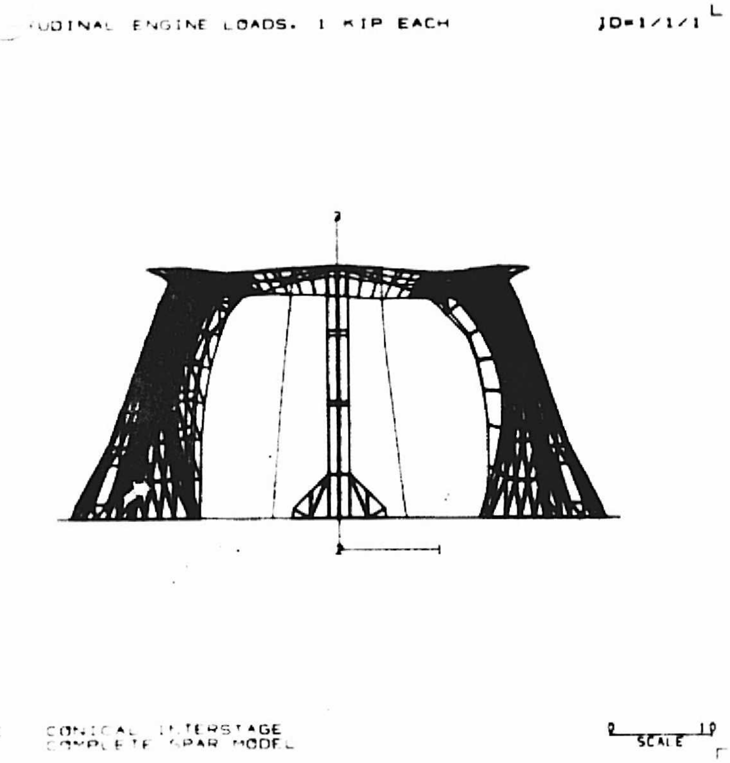
```
@XQT PLTA

SPEC 1
  STITLE' 1100KV SINGLE CIRCUIT TOWER
  ROTATE - 30, 3, 15, 1
  VIEW  - 2
  ALL
SPEC 2
  STITLE' 1100KV SINGLE CIRCUIT TOWER
  VIEW  3, -2, 1, 1
  ALL

@XQT PLTB

 DISPLAY= VIBRATIONAL MODES
 SET= 100
 CASES= 2
 DNORM= 50.
 OPTIONS= 26, 27
PLOT 1, 2 $ Frames A & B
```

A

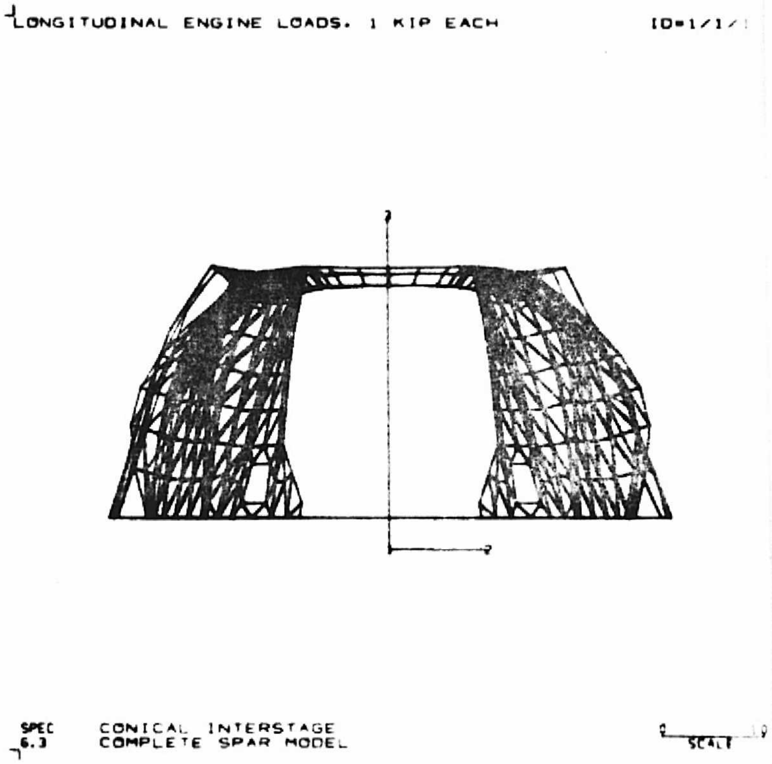


```
@XQT PLTA
SPEC 6
  STITLE' CONICAL INTERSTAGE
  S2TITLE' COMPLETE SPAR MODEL
  VIEW  3, -2, 1
  AXES 0.  0.  382.  10.  10.  35.
  ALL

@XQT PLTB
  RESET CORE=  40000
  DISPLAY=  STATIC DEFORMATION
  CASES=  1
  DNORM=  5.
 PLOT 6  $  3 VIEWS, Frames A, B, & C
```

B



C

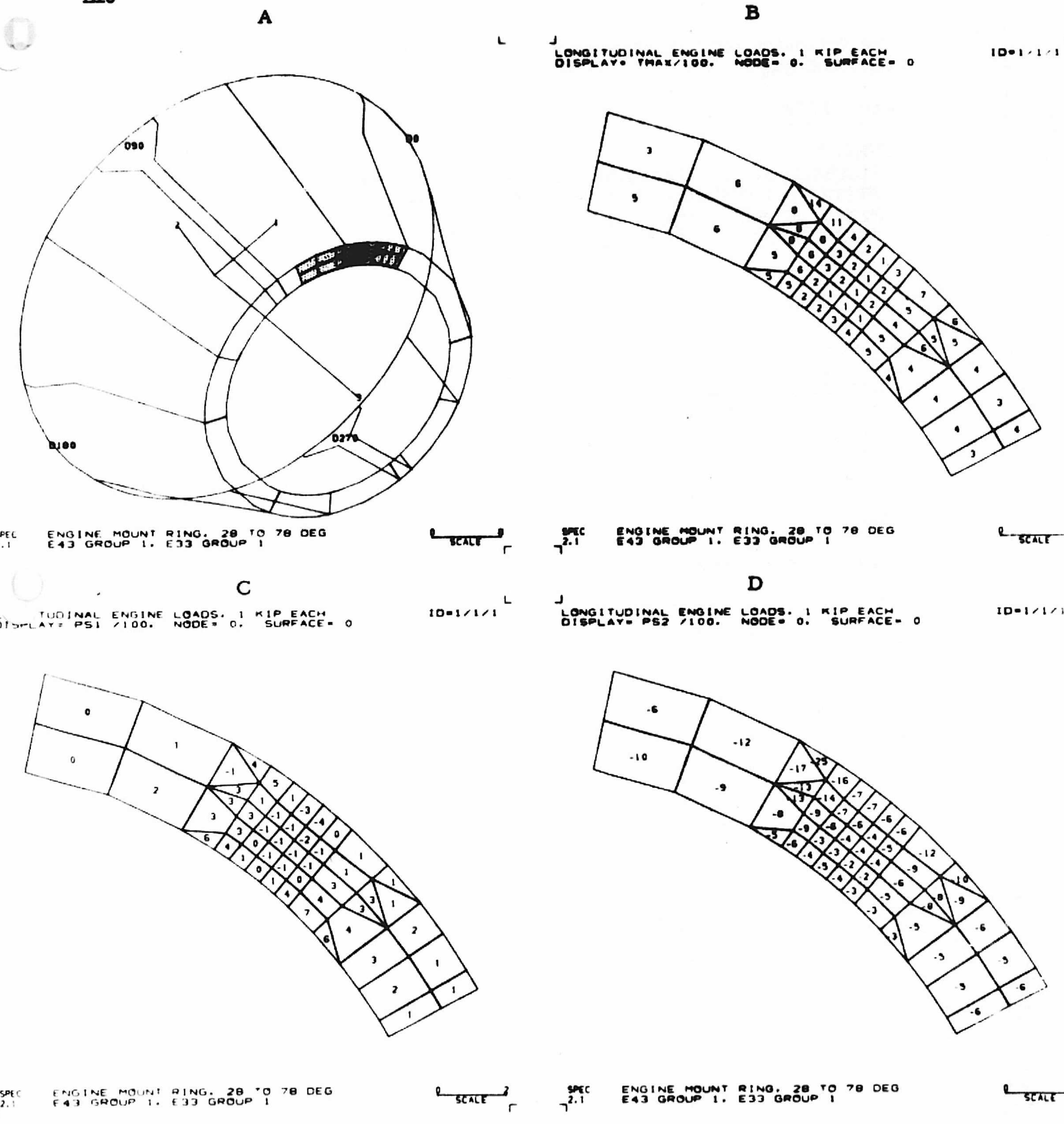


The following data input sequence produced the plots on the adjacent page:

```
@XQT PLTA

SPEC 1
 JLABEL 13 D0 211 D90 428 D180 637 D270
 STITLE' ENGINE MOUNT RING, 28 TO 78 DEG
 S2TITLE'E43 GROUP 1, E33 GROUP 1
 AXES 0. 0. 382. 10. 10. 35.
 ROTATE 30, 1, 30, 2, 30, 3
 VIEW 3
 add outline data*
 FORMAT 1
 E43 1, 9/50
 E33 1
SPEC 2
 STITLE' ENGINE MOUNT RING, 28 TO 78 DEG
 S2TITLE' E43 GROUP 1, E33 GROUP 1
 VIEW 3
 FORMAT 1
 E43 1, 9/50
 E33 1

@XQT PLTB

 DISPLAY= UNDEFORMED STRUCTURE
 OPTIONS= 14, 22, 23
PLOT 1 $ Frame A
 DISPLAY= TMAX/100., PS1/100., PS2/100.
 OPTIONS= 28
PLOT 2 $ Frames B, C and D
```

*Outline data consists of a sequence of "LINES" and "CONNECT" commands.

L10



SPEC .1 ENGINE MOUNT RING. 28 TO 78 DEG
E43 GROUP 1. E33 GROUP 1

LONGITUDINAL ENGINE LOADS. 1 KIP EACH ID=1/1/1
DISPLAY= TMAX/100. NODE= 0. SURFACE= 0

SPEC 2.1 ENGINE MOUNT RING. 28 TO 78 DEG
E43 GROUP 1. E33 GROUP 1

TUDINAL ENGINE LOADS. 1 KIP EACH ID=1/1/1
ISPLAY= PS1 /100. NODE= 0. SURFACE= 0

SPEC 2.1 ENGINE MOUNT RING. 28 TO 78 DEG
E43 GROUP 1. E33 GROUP 1

LONGITUDINAL ENGINE LOADS. 1 KIP EACH ID=1/1/1
DISPLAY= PS2 /100. NODE= 0. SURFACE= 0

SPEC 2.1 ENGINE MOUNT RING. 28 TO 78 DEG
E43 GROUP 1. E33 GROUP 1

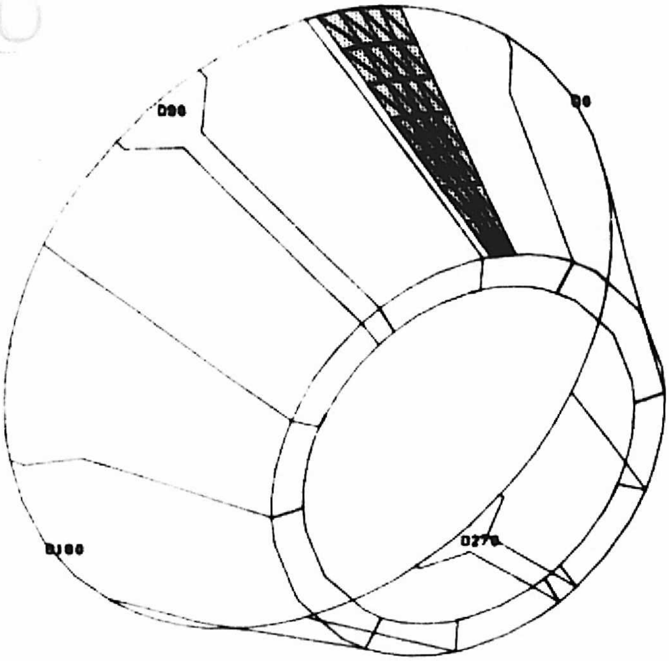
The following data input sequence produced the plots on the adjacent page:

```
@XQT  PLTA

SPEC 3
 JLABEL  13  D0  211  D90  428  D180  637  D270
 STITLE' CONE SURFACE,SEGMENT AT 50 DEG
 S2TITLE' E33 GROUP 10
 ROTATE 30, 1, 30, 2, 30, 3
 VIEW  3
 add outline data*
 FORMAT 1
 E33  10, 158/213
SPEC 4
 STITLE' CONE SURFACE,SEGMENT AT 50 DEG
 S2TITLE' E33 GROUP 10
  ROTATE  -140   3
 VIEW   3
 FORMAT 1
 E33  10, 158/213

@XQT  PLTB

 DISPLAY=  UNDEFORMED STRUCTURE
 OPTIONS=  14, 22
PLOT 3  $  Frame  A
 OPTIONS=  10, 16
PLOT 4  $  Frame  B
 OPTIONS=  12, 20
PLOT 4  $  Frame  C
 OPTIONS=  10, 20
PLOT  4  $  Frame  D
```

*Outline data consists of a sequence of "LINES" and "CONNECT" commands.

L10

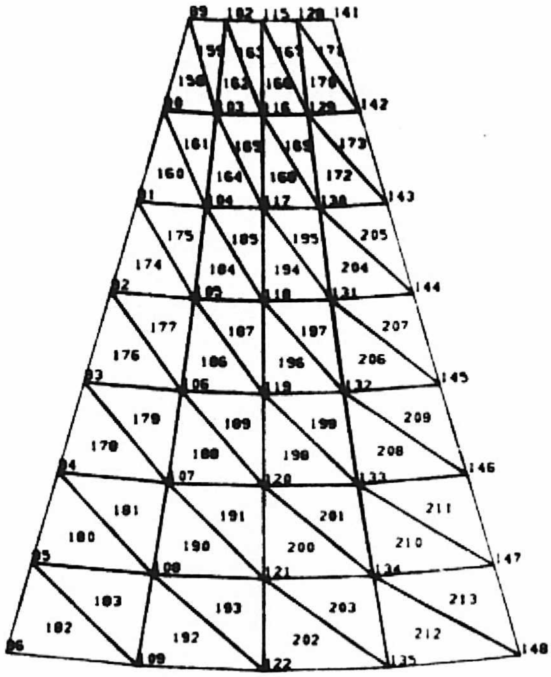
A



SPEC 4.1 CONE SURFACE, SEGMENT AT 50 DEG
E33 GROUP 10

0 SCALE 8

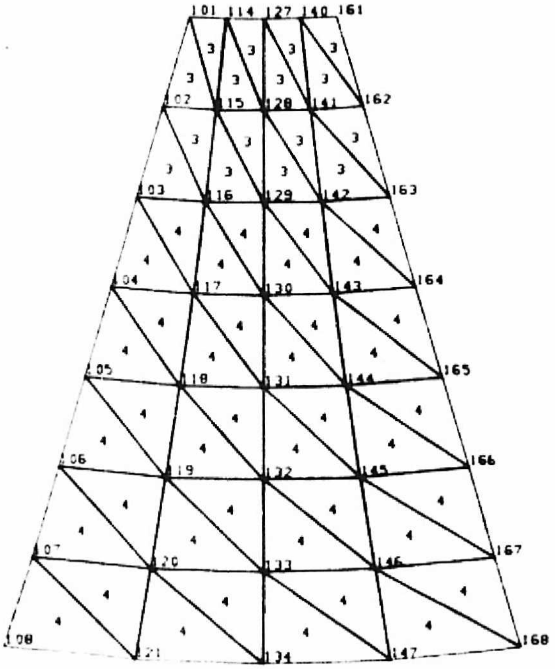
B



SPEC 4.1 CONE SURFACE, SEGMENT AT 50 DEG
E33 GROUP 10

0 SCALE

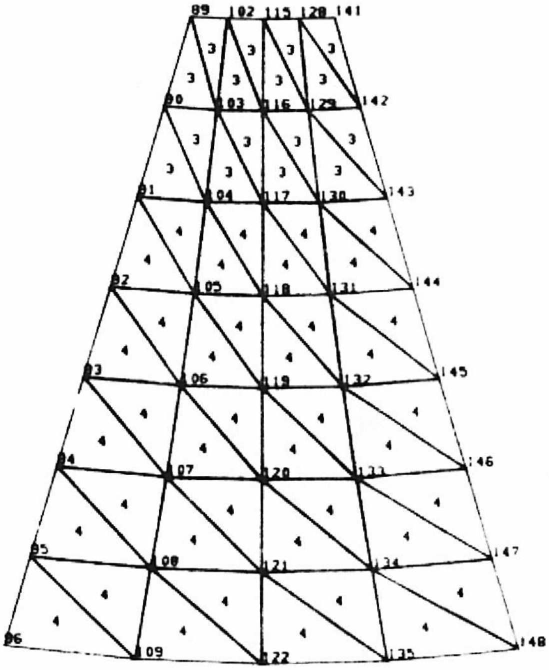
C



SPEC 4.1 CONE SURFACE, SEGMENT AT 50 DEG
E33 GROUP 10

0 SCALE 2

D



SPEC 4.1 CONE SURFACE, SEGMENT AT 50 DEG
E33 GROUP 10

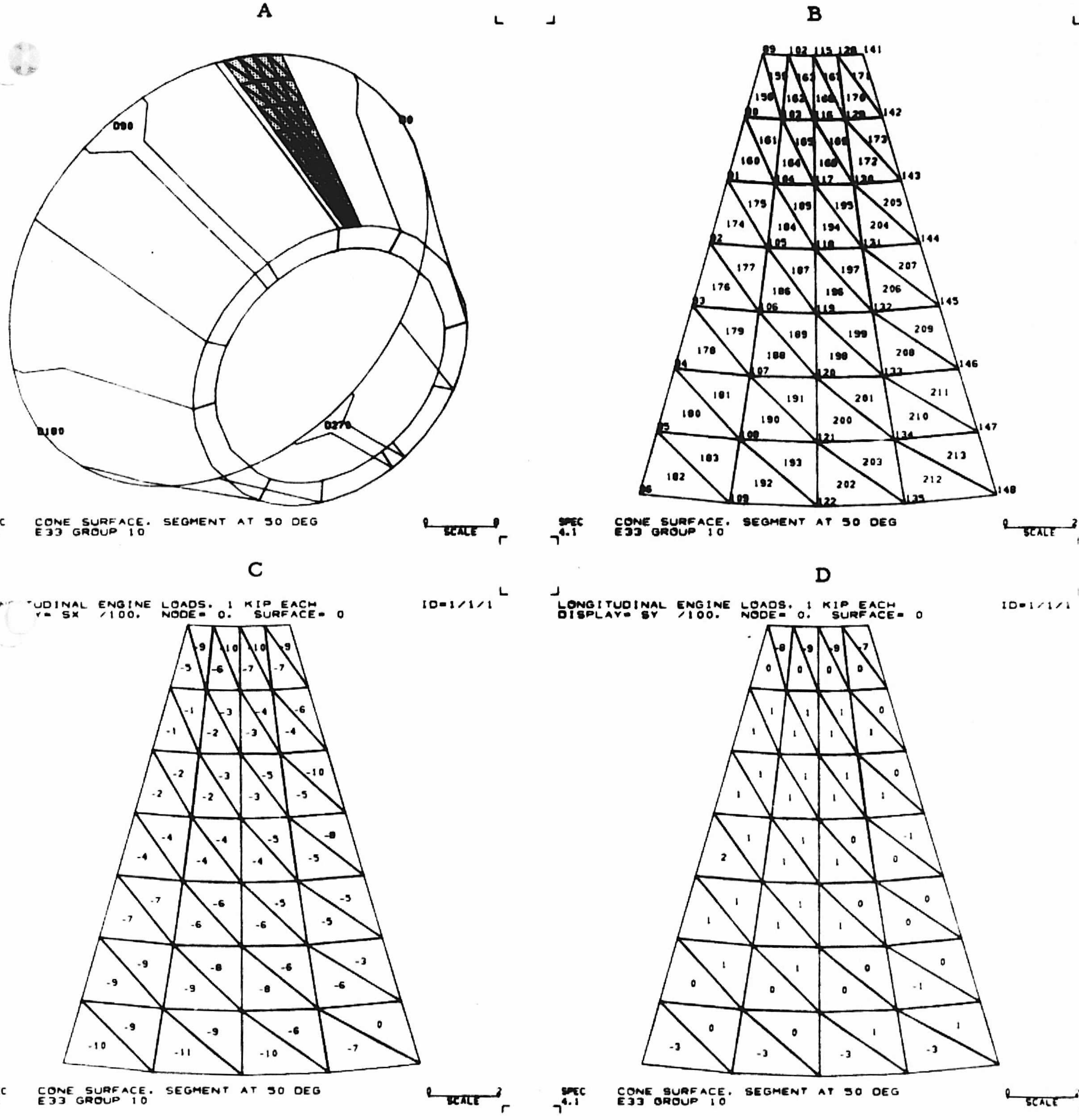
0 SCALE

The following data input sequence produced the plots on the adjacent page:

```
@XQT PLTA

SPEC 3
 JLABEL 13  D0  211  D90  428  D180  637  D270
 STITLE' CONE SURFACE,SEGMENT AT 50 DEG
 S2TITLE' E33 GROUP 10
 ROTATE 30, 1, 30, 2, 30, 3
 VIEW  3
 add outline data*
 FORMAT 1
 E33 10, 158/213
SPEC 4
 STITLE' CONE SURFACE,SEGMENT AT 50 DEG
 S2TITLE' E33 GROUP 10
 ROTATE  -140, 3
 VIEW  3
 FORMAT 1
 E33 10, 158/213

@XQT PLTB

 DISPLAY=  UNDEFORMED STRUCTURE
 OPTIONS=  14, 22
PLOT 3  $  Frame  A
 OPTIONS=  10, 16
PLOT 4 $  Frame  B
 DISPLAY=  SX/100., SY/100.
 OPTIONS=  28
PLOT 4  $  Frames C and D
```

*Outline data consists of a sequence of "LINES" and "CONNECT" commands.

L10


A
D90
D0
D180
D270
C CONE SURFACE. SEGMENT AT 50 DEG
E33 GROUP 10
SCALE
B
SPEC 4.1
CONE SURFACE. SEGMENT AT 50 DEG
E33 GROUP 10
SCALE
C
-TUDINAL ENGINE LOADS. 1 KIP EACH
-Y= SX /100. NODE= 0. SURFACE= 0
ID=1/1/1
C CONE SURFACE. SEGMENT AT 50 DEG
E33 GROUP 10
SCALE
D
LONGITUDINAL ENGINE LOADS. 1 KIP EACH
DISPLAY= SY /100. NODE= 0. SURFACE= 0
ID=1/1/1
SPEC 4.1
CONE SURFACE. SEGMENT AT 50 DEG
E33 GROUP 10
SCALE

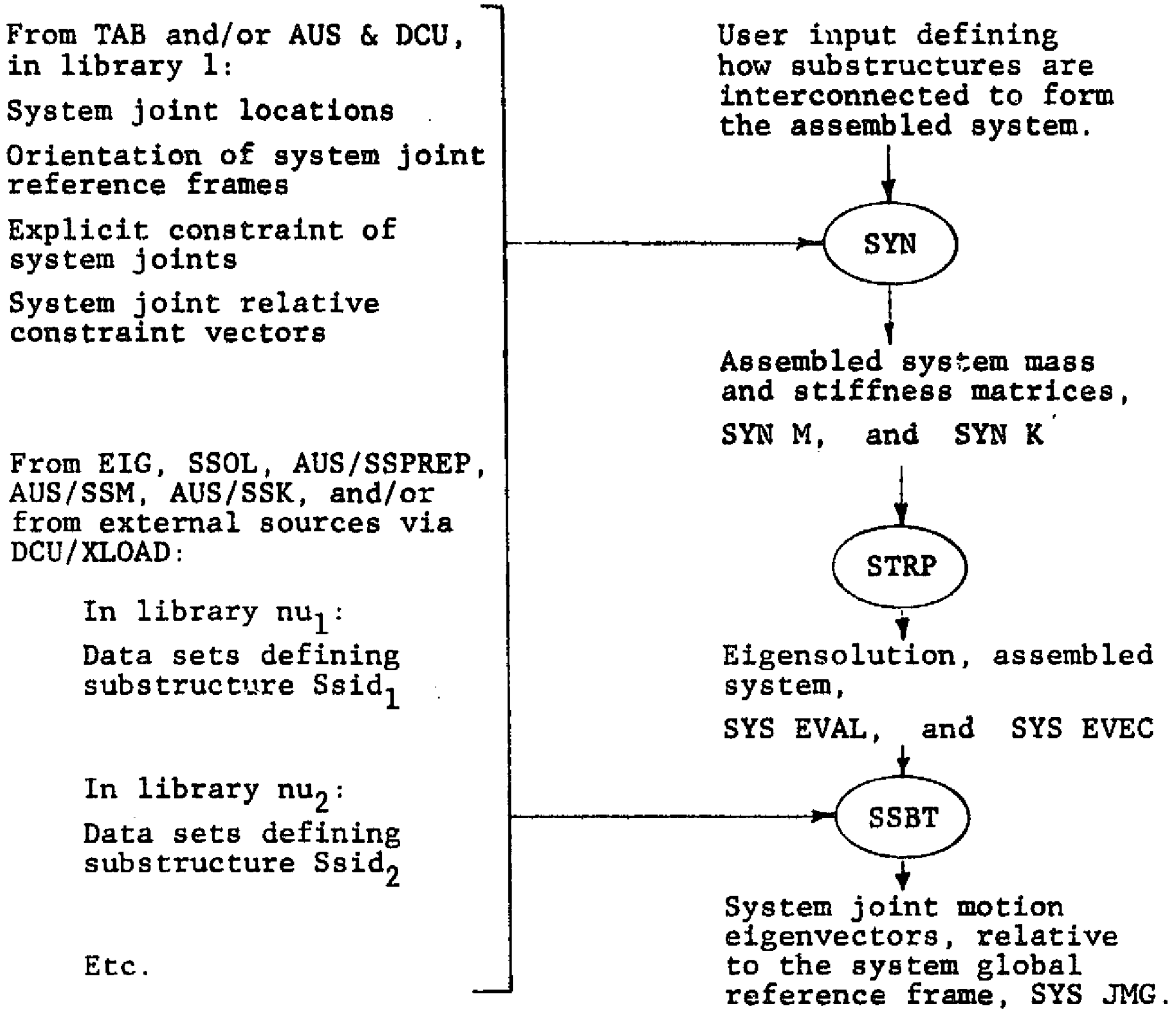
L10

Section 11

SUBSTRUCTURE PROCESSORS

It is assumed that the reader is familiar with substructure methods of vibrational and/or buckling analysis. Figure 11-1 illustrates typical flow of information among processors SYN, STRP, and SSBT, the functions of which are defined in Sections 11.2, 11.3, and 11.4. In Section 11.1, definitions are given of terms and symbols which are used in subsequent discussions of processor functions.

Assembled system mass and stiffness matrices, SYN M and SYN K, produced by SYN and operated on by STRP, are in LE (labelled element) format, which is sometimes called "strip format."



From TAB and/or AUS & DCU, in library 1:

System joint locations

Orientation of system joint reference frames

Explicit constraint of system joints

System joint relative constraint vectors

From EIG, SSOL, AUS/SSPREP, AUS/SSM, AUS/SSK, and/or from external sources via DCU/XLOAD:

In library nu_1:

Data sets defining substructure $Ssid_1$

In library nu_2:

Data sets defining substructure $Ssid_2$

Etc.

User input defining how substructures are interconnected to form the assembled system.

Assembled system mass and stiffness matrices,

SYN M, and SYN K

Eigensolution, assembled system,

SYS EVAL, and SYS EVEC

System joint motion eigenvectors, relative to the system global reference frame, SYS JMG.

Individual substructure eigenvectors in SYSVEC form:

USB $Ssid_1$

USB $Ssid_2$

Etc.

Also see Figures 11.1-1, 11.2-1, and 11.4-1

Figure 11-1 Typical Information Flow in Substructure Analysis

11.1 TERMINOLOGY

A system consists of two or more substructures interconnected at system joints, as illustrated on Figure 11.1-1. Each substructure has one or more boundary nodes, each of which must be connected to a system joint. If a boundary node does not coincide with the system joint to which it is connected, the connection is made through a rigid link.

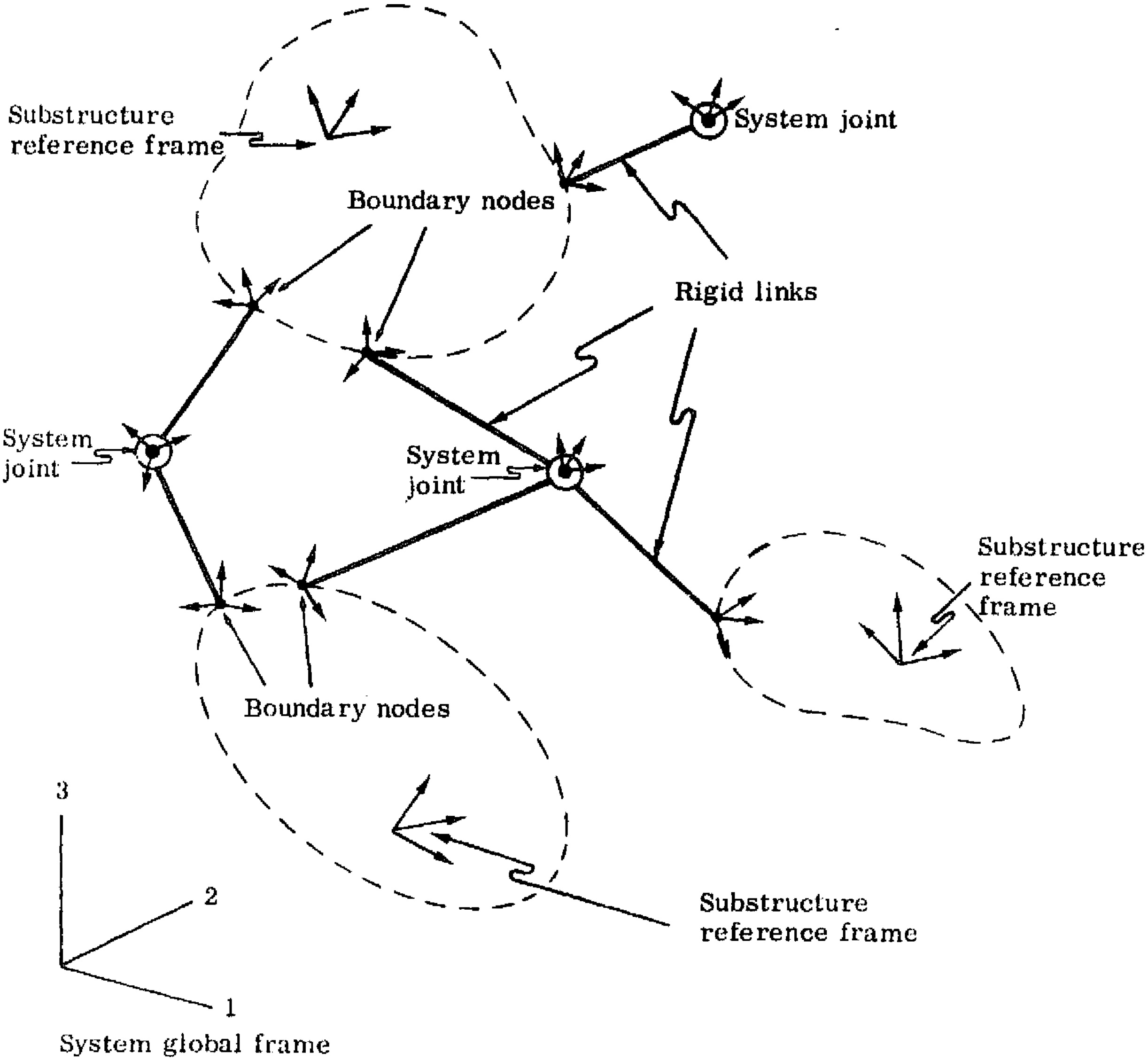


Figure 11.1-1 Assembled System.

11.1.1 Reference Frames

The motion of boundary node j of substructure i is b_j^i, a six-component vector relative to the boundary node reference frame uniquely associated with the boundary node.

Each substructure has an associated substructure reference frame, relative to which are defined (1) the locations of the boundary nodes, and (2) the orientations of the boundary node reference frames. Each substructure reference frame coincides with one of the system alternate reference frames, each of which is defined by specifying (1) the location of its origin, and (2) the orientation of its axes, relative to the system global frame.

The motion of system joint j is a_j, a six-component vector relative to the system joint reference frame uniquely associated with system joint j. System joint reference frame orientations are defined relative to the system global frame.

11.1.2 Substructure State

The deformation of substructure i is represented as a linear combination of boundary node unit motion functions, B_{jk}^i, and fixed boundary functions, F_m^i.

Each B_{jk}^i is a static displacement function produced by point forces and moments acting at boundary nodes. Where b_{jk}^i, for k= 1 through 6, is component k of the motion of boundary node j, B_{jk}^i is the function corresponding to $b_{jk}^i = 1.0$, with all other boundary node motion components identically zero.

For the F_m^i's, it is required that all boundary node motion components be identically zero. Typical F_m^i's are eigenfunctions and static displacement fields due to inertial loading corresponding to uniform acceleration.

The total motion of substructure i is approximated as

$$\sum_{j=1}^{\bar{j}} \sum_{k=1}^{6} B^i_{jk}\, b^i_{jk} + \sum_{m=1}^{\bar{m}} F^i_m\, f^i_m\,, \qquad (1)$$

where

$\bar{j}$ = the number of boundary nodes,

$\bar{m}$ = the number of fixed boundary functions, and

f^i_m = generalized coordinate associated with F^i_m.

The corresponding approximations of the kinetic and potential energies of substructure i are as follow.

$$T^i = \frac{1}{2} \begin{bmatrix} \dot{q}^i_b \\ \dot{q}^i_f \end{bmatrix}^t \begin{bmatrix} M^i_{bb} & M^i_{bf} \\ \left(M^i_{bf}\right)^t & M^i_{ff} \end{bmatrix} \begin{bmatrix} \dot{q}^i_b \\ \dot{q}^i_f \end{bmatrix}, \text{ and} \qquad (2)$$

$$V^i = \frac{1}{2} \begin{bmatrix} q^i_b \\ q^i_f \end{bmatrix}^t \begin{bmatrix} K^i_{bb} & 0 \\ 0 & K^i_{ff} \end{bmatrix} \begin{bmatrix} q^i_b \\ q^i_f \end{bmatrix}, \text{ where} \qquad (3)$$

$$q^i_b \equiv \begin{bmatrix} b^i_1 \\ b^i_2 \\ \cdot \\ \cdot \\ b^i_{\bar{j}} \end{bmatrix}, \quad \text{and} \quad q^i_f \equiv \begin{bmatrix} f^i_1 \\ \cdot \\ \cdot \\ \cdot \\ f^i_{\bar{m}} \end{bmatrix}. \qquad (4)$$

The K's and M's of Eqs. (2) and (3) are called substructure mass and stiffness matrices.

11.1.3 System State

The deformation of the system is defined by (1) all of the system joint motions, and (2) all of the substructure fixed boundary function coefficients, resident in the q_f^i's. Where n is the number of substructures, the system state vector is

$$q= \begin{bmatrix} q_r \\ q_e \\ q_f^1 \\ q_f^2 \\ \vdots \\ q_f^n \end{bmatrix} \tag{5}$$

The q_f^i's have the same meaning as in Section 11.1.1. System joint motion is defined by q_r and q_e, as described subsequently. System kinetic and potential energies are expressed in terms of system mass and stiffness matrices M and K as follows.

$$T= \frac{1}{2}\dot{q}^t M\dot{q} \quad ; \qquad V= \frac{1}{2}q^t Kq \tag{6}$$

Where a_j has the same meaning as in Section 11.1.1, and p is the number of system joints,

$$\begin{aligned} u &\equiv \begin{bmatrix} a_1^t & a_2^t & \qquad & a_p^t \end{bmatrix}^t \\ &= \begin{bmatrix} u_1 & u_2 & \qquad & u_{(6\times p)} \end{bmatrix}^t \end{aligned} \tag{7}$$

In Eq. (7), each u_i corresponds to a single component of an a_j; for example, u_7 is component 1 of a_2. The relation between system joint motion, u, and vectors q_r and q_e of Eq. (5) is as follows:

$$u = Rq_r + Eq_e \, , \tag{8}$$

In Eq. (8), R is a matrix of x relative constraint vectors,

$$R = [r_1 \quad r_2 \quad \dots \quad r_x] \, , \tag{9}$$

and elements of q_r are the associated coefficients. The composition of q_e is

$$q_e = \begin{bmatrix} u_{i_1} \\ u_{i_2} \\ \vdots \\ u_{i_y} \end{bmatrix} \, ,$$

where the u_{i_k}'s are the subset of components of u for which (1) the corresponding system joint motion component has not been explicitly constrained, and (2) every element in row i_k of R is equal to zero. Column k of E contains all zeros, except for a 1.0 in row i_k.

The analyst determines the content of q_r and q_e by (1) defining relative constraint vectors, if appropriate, and (2) specifying that certain system joint motion components be identically constrained, if appropriate.

11.2 SYN – SYSTEM SYNTHESIS

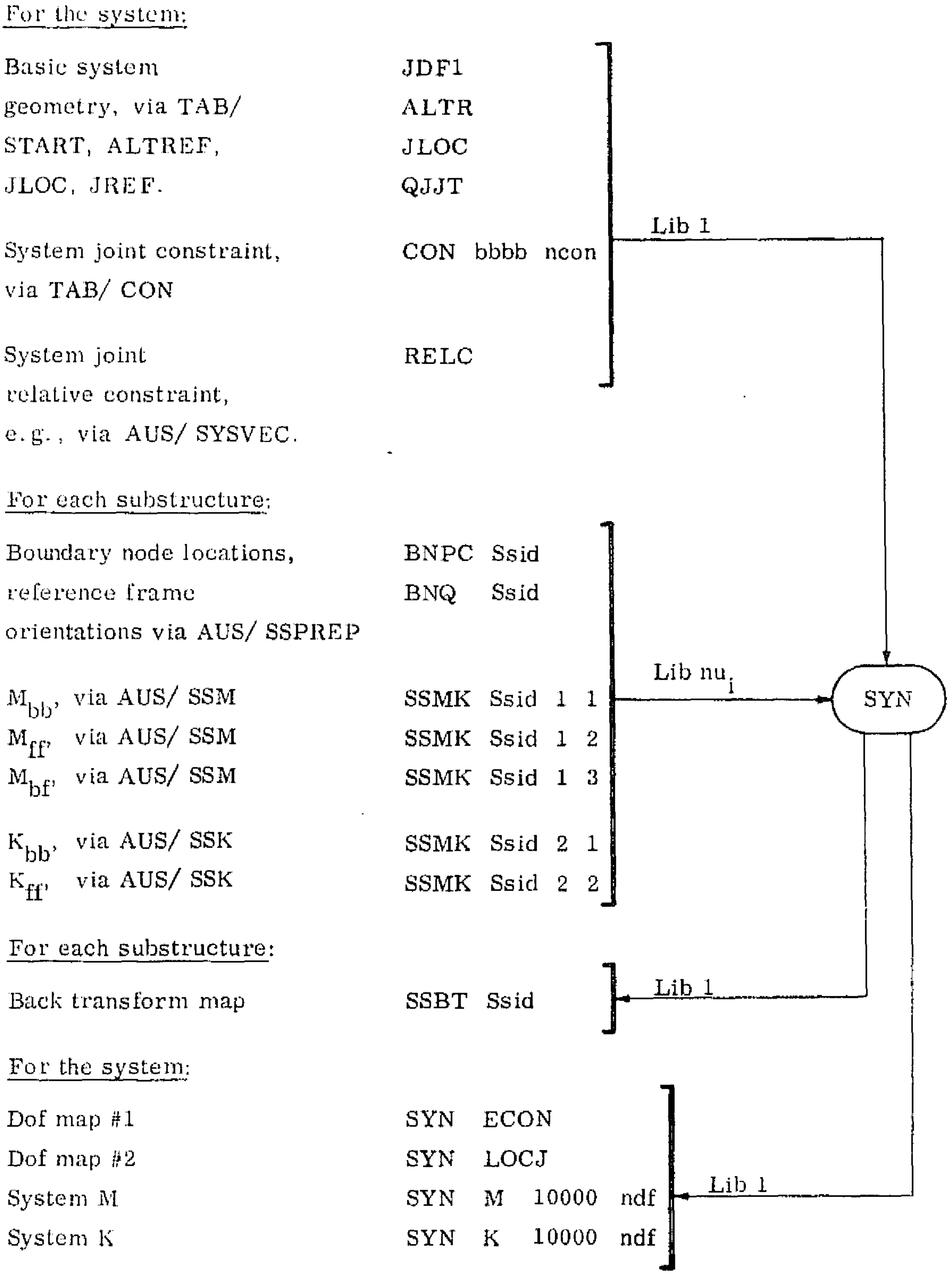
Function. SYN produces mass and stiffness matrices for systems comprised of interconnected substructures. SYN information flow, as illustrated on Figure 11.2-1, should be examined before reading the rest of this section.

Before executing SYN, TAB subprocessors must be executed as indicated below, to produce in Library 1 certain system-related data sets.

- Execute TAB/ ALTREF, to define the location and orientation, relative to the system global frame, of alternate reference frames 2, 3, 4, ---. As described later in this section, the location and orientation of each substructure is established by reference to one of these alternate frames.
- Execute TAB/ JLOC, to define system joint locations.
- Execute TAB/ JREF, to define system joint reference frame orientations.
- Execute TAB/ CON, to define system joint constraint conditions.

SYN requires six degrees of freedom per system joint; i.e., in the TAB execution outlined above, the START command must not exclude any degrees of freedom. The same restriction applies to all substructures assembled by SYN.

Relative constraint vectors, if any (see Section 11.1.2), must reside in library 1 in a SYSVEC-format data set named RELC. If no such data set exists, SYN assumes that there are no relative constraints.



For the system:

Basic system geometry, via TAB/ START, ALTREF, JLOC, JREF.	JDF1 ALTR JLOC QJJT
System joint constraint, via TAB/ CON	CON bbbb ncon
System joint relative constraint, e.g., via AUS/ SYSVEC.	RELC

For each substructure:

Boundary node locations, reference frame orientations via AUS/ SSPREP	BNPC Ssid BNQ Ssid
M_{bb}, via AUS/ SSM	SSMK Ssid 1 1
M_{ff}, via AUS/ SSM	SSMK Ssid 1 2
M_{bf}, via AUS/ SSM	SSMK Ssid 1 3
K_{bb}, via AUS/ SSK	SSMK Ssid 2 1
K_{ff}, via AUS/ SSK	SSMK Ssid 2 2

For each substructure:

Back transform map	SSBT Ssid

For the system:

Dof map #1	SYN ECON
Dof map #2	SYN LOCJ
System M	SYN M 10000 ndf
System K	SYN K 10000 ndf

Figure 11.2-1 SYN Data Transactions.

After any Reset commands, the following input sequence is given for each substructure.

$$\text{Ssid, nu, nrf}$$

$$j_1\text{:}\ \ j_2\text{:}\ \ j_3\text{: ---- }j_n\$^*$$

In the above, nu is the library containing the data sets which describe the substructure. These data sets are usually produced by means of the AUS commands SSID, SSPREP, SSM, and SSK. Ssid is the alphanumeric identifier assigned via AUS/SSID. The substructure reference frame (i.e., the "global frame" used in defining the basic finite element mesh representation of the substructure preparatory to AUS/SSPREP, etc.) coincides with alternate frame nrf of the ALTREF data set resident in library 1. Subsequent records, j_1: j_2: etc., indicate that substructure boundary node 1 is connected to system joint j_1, boundary node 2 to system joint j_2, etc. Where n is the number of boundary nodes, there must be exactly n j's in this list.

The input sequence described below may be used repeatedly, if required, to cause additional terms to be added to the system mass and/or stiffness matrix. The sequence indicates a 6 × 6 submatrix, coupling system joint i to system joint j, in the system mass matrix. The term in row k, column ℓ of the submatrix is $m_{k\ell}$.

$$\text{M}\quad i,\ j$$

$$\underbrace{m_{11}\ \ m_{21}\ \cdot\ \cdot\ m_{61},}_{\text{column 1}}\ \ \underbrace{m_{12}\ \ m_{22}\ \cdot\ \cdot\ m_{62},}_{\text{column 2}}\ \cdot\ \cdot\ m_{66}$$

*Loop-limit format also is permitted.

The string of 36 terms, $m_{11} \cdots m_{66}$, may extend over several cards. Any omitted trailing $m_{k\ell}$'s are assumed to be zero.

To add a 6 × 6 submatrix to the system stiffness matrix, replace M with K in the above sequence.

Reset Controls.

Name	Default Value	Meaning
CON	1	Constraint case, system joint constraint
TOLM	$10.^{-30}$	Terms in system mass matrix less than TOLM are neglected.
TOLK	$10.^{-30}$	Terms in system stiffness matrix less than TOLK are neglected.
TOLR	$10.^{-5}$	Elements of relative constraint vectors are assumed to be zero if their absolute value is less than TOLR.
LR	896	System M, K output block length.
SYSL	1	Library containing all system-related input, output.

Core Requirements. Data space is required for:

All system related input data sets (see Figure 11.2-1)

+ one block each of system M and K output data sets,

+ the largest value of L_i, the space required for data sets associated with substructure i (see Figure 11.2-1),

+ several thousand words of scratch storage space.

Code Release Data. Level 9 (Univac), July 1975. Synthesis routines coded by C. E. Jones, under NASA Contract NAS8-30520. SPAR interface coded by W. D. Whetstone.

11.3 STRP – SUBSTRUCTURE EIGENSOLVER

Function. STRP computes eigenvalues and eigenvectors of the $\lambda MX = KX$ problem. M must be positive definite, and K positive semi-definite. STRP input consists of two data sets, normally produced by processor SYN (see Section 11.2),

SYN M ncode ndof, and

SYN K ncode ndof.

The third and fourth words of the above data set names are automatically established by SYN (ncode is the row/column encoding constant, and ndof is the number of degrees of freedom).

Eigenvalues are stored in a data set named SYS EVAL. Eigenvectors are stored in SYS EVEC.

STRP uses the Cholesky-Householder method, and assumes, in principal effect that all terms of M and K are non-zero. Accordingly, execution costs rise steeply as the number of degrees of freedom is increased. STRP has no explicit degree of freedom limitation; however, a large amount if I/O activity will result if too little working core space is provided when attempting to solve large (e.g., more than 150 d. o. f.) problems.

Reset Controls.

Name	Default Value	Meaning
FRQ1	$-10.^{10}$	FRQ1 and FRQ2 indicate lower and upper limits (in Hz) of the range in which eigenvalues are to be computed.
FRQ2	$+10.^{10}$	
SOURCE	1	Data source library.
DEST	1	Data destination library.

Core Requirement. There is no explicit working core requirement. As more core space is provided, I/O activity will automatically dimish.

Code Release Data. Level 9 (Univac), July 1975. Internal coding by R. A. Moore and M. L. Pearson. SPAR interface coded by W. D. Whetstone.

11.4 SSBT SUBSTRUCTURE BACK TRANSFORMATION

Function. System eigenvectors, residing in data sets named SYS EVEC generated by processor STRP, have the internal form indicated by Eq. (5) of Section 11.1.3. As shown on Figure 11.4-1, SSBT back-transforms SYS EVEC data sets in two phases, as discussed below.

In Phase-1, a data set named SYS JMG is produced. Each block of SYS JMG is of length 6 x J, where J is the number of system joints. Components of the j-th six-element subvector of each block are motion components (3 displacements, 3 rotations), relative to the system global frame, of system joint j. Each block of SYS JMG is derived from the corresponding block (mode) of SYS EVEC.

Phase-2 is controlled by the following input sequence:

Ssid_1, nu_1, ndest_1\$ substructure_1
Ssid_2, nu_2, ndest_2\$ substructure_2
Etc.

For each substructure designated by the user as indicated above, a data set named USB Ssid is generated and stored in library ndest. If ndest is omitted, default ndest= nu. Each block of USB Ssid is of length 6 x N, where N is the total number of joints in the substructure (not just the boundary joints). Components of the j-th six-element subvector of each block are motion components (3 displacements, 3 rotations) of joint j of the substructure, relative to the associated joint reference frame. Each block of USB Ssid is derived from the corresponding block of SYS EVEC, unless the following command is used:

MODES= n_1, n_2

For all (Ssid, nu, ndest) commands following a MODES command, block 1 of USB Ssid will correspond to block (mode) n_1 of SYS EVEC, block 2 to mode n_1+1, - - - and block (n_2- n_1+ 1) to mode n_2. It should be noted that the USB Ssid data sets are in SYSVEC format for the associated substructures, and may be operated on by VPRT, GSF, etc.

Reset Controls.

	Default	
Name	Value	Meaning
JMG	1	RESET JMG= 0 to omit Phase-1 if SYS JMG already exists.

Core Requirements. For Phase-1, the core required is fourteen times the number of system joints, plus one block of SYS EVEC (i.e. 14J + the number of system degrees of freedom).

For Phase-2, data space is required to concurrently hold in core one block of each input and output data set (see Fig 11.4-1).

Code Release Data. Level 10 (Univac, CDC), March 1976, coded by W. D. Whetstone.

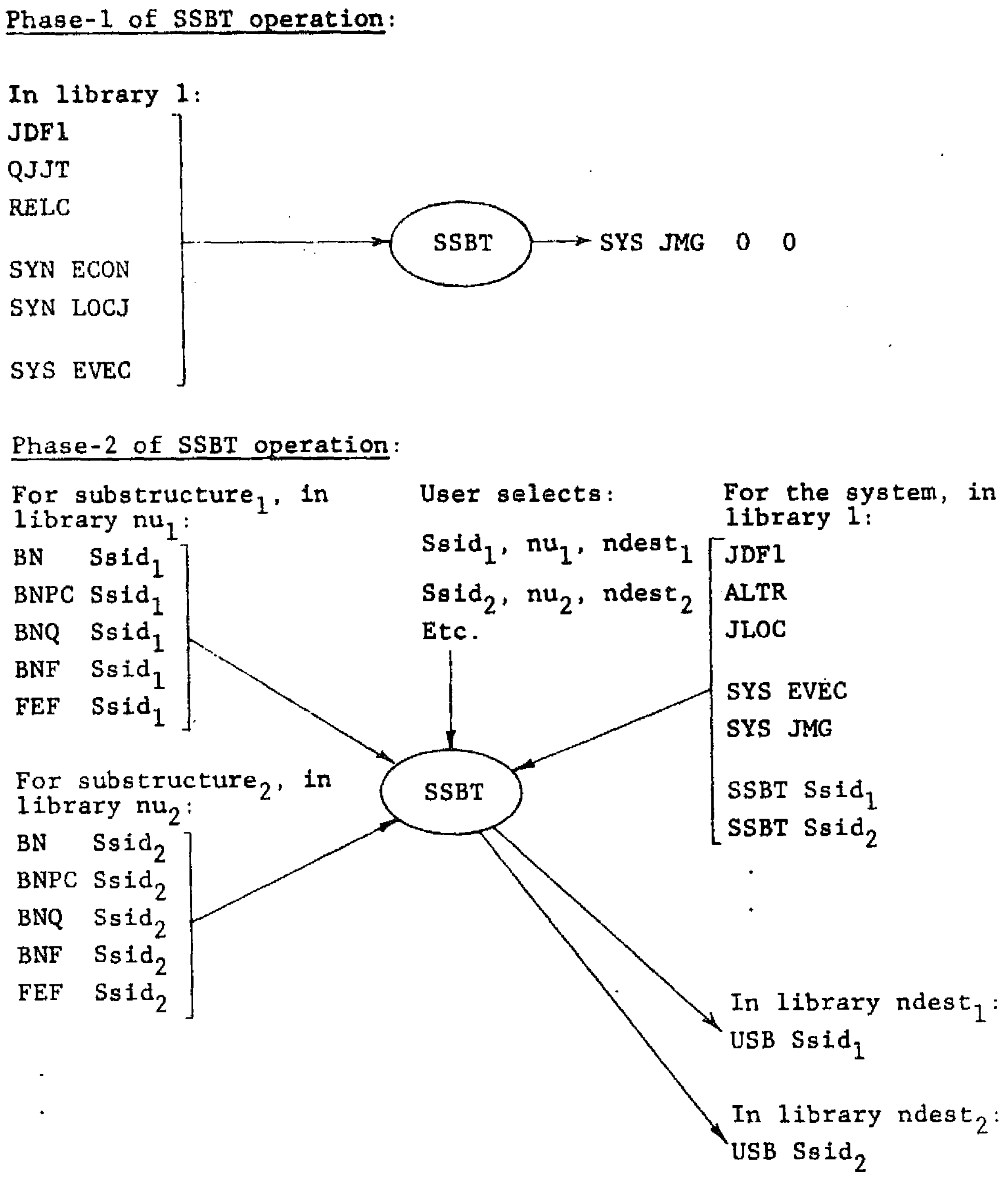


Figure 11.4-1 SSBT Data Transactions

APPENDIX A

Short List*of Commands and Data Sequences

TAB

START ` number of joints, list of excluded joint motion components.

UPDATE = 1 to enter update mode, 0 to leave.

FORMAT = i, to select alternate format i.

NREF = n, to identify reference frame n.

MOD = m, to modify table entry numbers or joint numbers by m.

After each sub-processor execution, TAB returns automatically to FORMAT=1, NREF=1, and MOD=0.

*Only frequently used reset controls, etc., are included in this summary. See the main text for full information.

TAB Sub-processors

TEXT:	4/8 - - - 76 characters of text.
MATC:	k, E, Nu, Rho, Alpha_1, Alpha_2
NSW:	k, weight/(area or length).
ALTREF:	k, i_1, a_1, i_2, a_2, i_3, a_3, x_1, x_2, x_3
JLOC:	k, xa_1, xa_2, xa_3, xb_1, xb_2, xb_3, ni, ijump, nj. jjump, xc_1, xc_2, xc_3, xd_1, xd_2, xd_3. Formats 1, 2 = rectangular, cylindrical. NREF = reference frame.
JREF:	NREF= n: j_1, j_2, inc: k_1, k_2, inc: - - - -
MREF:	k, nb, ng, isign, c (FORMAT 1) k, i_1, x_1, x_2, x_3 (FORMAT 2) NREF = n allowed under Format 2.
BRL:	k, i^1, x_1^1, x_2^1, x_3^1, i^2, x_1^2, x_2^2, x_3^2 .

BA: BOX, k, b_1, t_1, b_2, t_2.

TEE, k, b_1, t_1, b_2, t_2.

ANG, k, b_1, t_1, b_2, t_2.

WFL, k, b_1, t_1, b_2, t_2, b_3, t_3.

CHN, k, b_1, t_1, b_2, t_2, b_3, t_3.

ZEE, k, b_1, t_1, b_2, t_2, b_3, t_3.

TUBE, k, inner r, outer r.

GIVN, k, I_1, α_1, I_2, α_2, a, f, f_1, z_1, z_2, θ.

DSY, k, l_1, α_1, I_2, α_2, a, f, f_1. (card 1)

q_1, q_2, q_3, y_{11}, y_{12}, - - - y_{41}, y_{42}. (card 2)

BB: k, s_{11}.

s_{21}, s_{22}.

s_{31}, s_{32}, s_{33}.

s_{41}, s_{42}, s_{43}, s_{44}.

s_{51}, s_{52}, s_{53}, s_{54}, s_{55}.

s_{61}, s_{62}, s_{63}, s_{64}, s_{65}, s_{66}.

BC: k, a

BD: k, a, I_1, α_1, h, c, q_1

SA: k, T. (Format 1)

k, T_{11}, T_{22}, T_{12}, F_1^a, F_2^a, F_1^b, F_2^b. (Format 2)

c_{11}, c_{12}, c_{22}, c_{31}, c_{32}, c_{33}. (membrane stiffness)

d_{11}, d_{12}, d_{22}, d_{31}, d_{32}, d_{33}. (bending stiffness)

SB: k, t

CON n: (constraint case n)

ZERO m_1, m_2, - -: $j_1^1, j_2^1, \text{inc}^1$: $j_1^2, j_2^2, \text{inc}^2$: - - - -:

NONZERO (same as above)

RELEASE (same as above)

SYMMETRY PLANE = n

ANTISYMMETRY PLANE = n

FIXED PLANE = n

XZERO = x.

RZERO = r.

JSEQ: REPEAT n^1, inc^1, jmod^1:

$j_1/j_2, j_3, j_4/j_5$ - - :

$j_6, j_7, j_8/j_9$ - - - :

REPEAT n^2, inc^2, jmod^2:

$j_{10}, j_{11}, j_{12}/j_{13}$ - - -:

etc.

RMASS: k, M, I_1, I_2, I_3. (Format 1)

k, $M_1, M_2, M_3, I_1, I_2, I_3$. (Format 2)

REPEAT n, jump.

CM = (M multiplier), (I multiplier).

ELD

Sub-processor Name	Element type
E21	General beam
E22	Finite length, directly specified K
E23	Axial element
E24	Plane beam
E25	Zero-length, directly specified K
E31	Triangular membrane
E32	Triangular bending element
E33	Triangular membrane + bending element
E41	Quadrilateral membrane
E42	Quadrilateral bending element
E43	Quadrilateral membrane + bending element
E44	Quadrilateral shear panel

Pointer Name	Default Value	Associated Table
NMAT	1	Material constants (MATC)
NSECT	1	Section properties (BA - - - - -SB)
NOFF	0	Rigid link offsets (BRL)
NNSW	0	Nonstructural weight (NSW)
NREF	1	Beam orientation (MREF). Also indicates pressure sense for 3 and 4 node elements.

Data Modifiers:

MOD JOINT NUMBERS n, or MOD JOINT=n

MOD GROUP NUMBERS n, or MOD GROUP=n

MOD NMAT (or NSECT, etc.)=n

INC NMAT (or NSECT, etc.)=n

All MOD and INC parameters are reset to zero upon conclusion of each sub-processor execution.

2-node elements:

j_1, j_2

j_1, j_2, 1, ni, nj, jinc. (nj grid lines)

j_1, j_2, 2, ni, nj, jinc. (nj rings)

j_1, j_2, 3, ni, iinc, nj, jinc. (nj sets of spokes)

3-node elements:

j_1, j_2, j_3

j_1, j_2, j_3, 1, ni, nj. (rectangular mesh)

j_1, j_2, j_3, 2, ni, nj, jinc. (nj 'open fans')

j_1, j_2, j_3, 3, ni, nj, jinc. (nj 'closed fans')

4-node elements:

j_1, j_2, j_3, j_4

j_1, j_2, j_3, j_4, 1, ni, nj, nk, kinc. (nk rectangular meshes)

j_1, j_2, j_3, j_4, 2, ni, nj. (cylinder, nj bays)

E

Reset Name	Default	Meaning
G	1.0	Mass= Weight/G

K

Reset Name	Default	Meaning
SPDP	1	SPDP= 2 for double precision
OUTLIB	1	Output library

M

Reset Name	Default	Meaning
G	1.0	Mass = Weight/G
IBEAM	0	If not zero, beam rotatory inertia ignored.
INERT	0	If not zero, only displacement-dependent terms (as in bars, membranes) are included, for all element types.

KG

Reset Name	Default	Meaning
IKG2	0	If not zero, all beams treated as bars.
IKG34	0	If not zero, all shell elements treated as membranes.

INV

Reset Name	Default	Meaning
CON	1	Constraint case.
K	K	Name of matrix to be factored.
KILIB	1	Destination library for factored matrix.
ILIB	1	Library containing AMAP (from TOPO).
KLIB	1	Library containing K

AUS – Miscellaneous

INLIB= nin

OUTLIB= nout

DEFINE Z= Lib N1 N2 n3 n4

ZERO= value

FIND Lib N1 N2 n3 n4

AUS – Arithmetic

Command Forms	Meaning
Z= SUM(X, Y)	Z = X + Y (system matrices)
Z= PRODUCT(X, Y)	Z = X Y (system matrices)
Z= UNION(X_1, X_2, ---)	Z = $[X_1 \mid X_2 \mid X_3$ ---$]$
Z= XTY(X, Y)	Z = X^tY
Z= XTYSYM(X, Y)	Z = X^tY, symmetric
Z= XTYDIAG(X, Y)	Z = X^tY, diagonal
Z= NORM(X, j, k, v)	System vector renormalization
Z= RIGID(j)	Rigid body motion vectors
Z= RECIP(X)	Each element $z = 1./x$
Z= SQRT(X)	Each element $z = \text{sign}(x)\sqrt{\lvert x \rvert}$
Z= SQUARE(X)	Each element $z = x^2$
Z= RPROD(X, Y)	Z = X Y (rectangular matrices)
Z= RTRAN(X)	Z = X^t (rectangular matrices)
Z= RINV(X)	Z = X^{-1} (square matrices)

AUS/TABLE

TABLE(NI= ni, NJ= nj): N1 N2 n3 n4 $ = Output name

CASE n$ or BLOCK n

OPERATION= SUM$, or XSUM, or MULT, or DIVIDE

I= i_1, i_2, . . . i_m

DDATA= d_1, d_2, . . . d_m

J= j_1: j_2: j_3: . . $j_r$$ Loop limit format also allowed.

$e_{j_1}^{i^1}$, $e_{j_1}^{i^2}$ . . . $e_{j_1}^{i^m}$ $ Data for j= j_1

$e_{j_2}^{i^1}$, $e_{j_2}^{i^2}$ . . . $e_{j_2}^{i^m}$ $ Data for j= j_2

$\vdots$ $\vdots$ $\vdots$

$e_{j_r}^{i^1}$, $e_{j_r}^{i^2}$ . . . $e_{j_r}^{i^m}$ $ Data for j= j_r .

AUS/SYSVEC

Same as AUS/TABLE, except that NI and NJ are automatically fixed by SYSVEC.

AUS/ELDATA

ELDATA: N1 N2 n3$ 4th word not given here

CASE n4$ Results in output data set N1 N2 n3 n4. Other input is the same as in TABLE, except that J= j_1: j_2, --- is replaced by

G= g_1: g_2: - - - - $ List of groups

E= e_1: e_2: - - - - $ List of element indexes

AUS/ALPHA

ALPHA: N1 N2 n3 n4

1' Title string 1

2' Title string 2

etc.

DCU

TITLE	Lib' Title - -
STATUS	Lib
TOC	Lib, Id
DISABLE	Lib, Id
ENABLE	Lib, Id
COPY	Lib_1, Lib_2, Id
XCOPY	Lib, n, Id
XLOAD	n, Lib, nwords, nj, ninj, type, N1, N2, n3, n4
REWIND	n
CHANGE	Lib, Id_{old}, Id_{new}
DUPLICATE	Lib_1, Lib_2
NTAPE	nt
TWRITE	Lib
TREAD	Lib
LIBLIB	n1
STORE	Lib, Id
RETRIEVE	Lib, Id
PRINT	Lib, Id, j_1, j_2, i_1, i_2, b_1, b_2
ABORT	n

L10

EQNF

Reset Name	Default	Meaning
SET	1	Load set identifier
INLIB	1	Source library
FEFLIB	1	Destination library, equivalent nodal loads
ISLIB	1	Destination library, initial strains

SSOL

Reset Name	Default	Meaning
SET	1	Load set identifier
K	K	Name of stiffness matrix
CON	1	Constraint case
KLIB	1	Library containing stiffness matrix
KILIB	1	Library containing factored stiffness matrix

GSF

Reset Name	Default	Meaning
SET	1	Load set identifier
QLIB	1	Library containing displacement and initial strain data
L1	1	First case
L2	–	Last case
EMBED	0	Nonzero value causes stresses to be embedded in E-state.
ACCUM	0	Nonzero value causes stresses to be added to those already resident in the E-state.

Execution Control Parameters

SOURCE= N1, N2, n_3, n_4 \$ Replaces STAT DISP --

EIJ_1: g_1: g_2: -- : EIJ_2 - etc. -

PSF

Reset controls SET, QLIB, L1, and L2 are the same as in GSF.

Reset Name	Default	Meaning
LINES	56	Lines per page
DISPLAY	1	DISPLAY= 1 for standard stress print DISPLAY= 2 for forces, stress resultants DISPLAY= 3 for beam bending stress detail
NODES	1	Set = 0 to omit data at element corners.
CROSS	1	Set = 0 for mid-surface data, only.

EIG

Reset Name	Default	Meaning
PROB	VIBR	BUCK indicates buckling analysis.
NDYN	8	Iteration control.
CONV		" "
NREQ		" "
V1		" "
V2		" "
K	K	K name
KG	KG	Kg name
M	DEM	M name
CON	1	Constraint case
KLIB	1	Data source library, K
KILIB	1	" " , INV K
KGLIB	1	" " , KG
MLIB	1	" " , M
OUTLIB	1	Output destination library
NEWSET	1	Output set identifier
INIT	0	Number of random vectors.
L1		Designates source of initial functions — vectors L1 through L2 of VIBR MODE oldset, contained in inlib.
L2		
INLIB	1	
OLDSET	1	
SHIFT	.0	See Section 8.
HIST	4	Controls print-out of iteration history

DR

DTEX Parameter Name	Default Value
DT	—
NTERMS	10
N2	MASK
INLIB	1
OUTLIB	1

TR1 Parameter Name	Default Value
INLIB	1
N2	MASK
CASE	1
T1	—
T2	—
ALIB	1
QXLIB	0
QX1LIB	0
QX2LIB	1
QRLIB	0
QR1LIB	0
QR2LIB	1
LB	896

BACK Parameter Names	Default Value
M1	1
M2	—
DT	1.0
TSTART	.0
LRZ	896
PRINT	1
N3REPEAT	1
N4REPEAT	1
SOURCE	1
DEST	1
FMAX	$-1. \times 10^{20}$
FMIN	$1. \times 10^{20}$
BIG	$1. \times 10^{20}$

BACK Execution Control Statements:

ZC=	C	Lib	N1	N2	n3	n4
T=				"		
Y=				"		
Z=	Lib	N1	N2	n3	n4	
EXT=	Lib	N1	N2	n3	n4	

SYN

Reset Name	Default	Meaning
CON	1	Constraint case for system
LR	896	Output block length
SYSL	1	Library containing system JLOC, etc.

For each substructure,

Ssid, nu, nrf

j_1: j_2: j_3: - - - - $j_n$$

STRP

Reset Name	Default	Meaning
FRQ1	$-10.^{10}$	Frequency range is FRQ1
FRQ2	$10.^{10}$	through FRQ2
SOURCE	1	Data source library
DEST	1	Data destination library

SSBT

Reset Name	Default	Meaning
JMG	1	If zero, omit creation of the data set SYS JMG if SYS JMG already exists

Execution Control Statements

MODES= n_1, n_2

$SSID_i$, nu_i, $ndest_i$ *(Control for substructure "i")*

PLTA

Reset Name	Default	Meaning
GGSL	16	Destination library for plot specifications
SDEF	21	Destination library for temporary PLTA data sets

Execution Control Statement

SPEC	nspec, n

Optional Control Statements

STITLE'	Title line 1
S2TITLE'	Title line 2
TEXT'	Text line
SYM	iaxis
ANTISYM	iaxis
ROTATE	a_1 i_1 a_2 i_2 a_3 i_3
LROTATE	a_1 i_1 a_2 i_2 a_3 i_3
VIEWS	v_1 v_2 v_3 n
LOCLABEL	iquad ix iy
JLABEL	j_1 l_1 j_2 l_2 j_3 l_3 - - -
AXES	o_1 o_2 o_3 ℓ_1 ℓ_2 ℓ_3
MARGIN	percent

Geometric Composition Statements

ALL	
FORMAT 0:	ELJ $indx_1$ $indx_2$ ijump grp_1 grp_2 gjump
FORMAT 1:	ELJ $group_1$/$group_2$, list of indexes
FORMAT 2:	ELJ $index_1$/$index_2$, list of groups
CONNECT	j_1 j_2 j_3 - - -
LCONTROL	jshift nrepeat inc
LINES	j_1 j_2 j_3 - - -

PLTB

Execution Control Statements

DISPLAY=	UNDEformed STATic deformation VIBRrational mode BUCKling mode *Stress Quantity/div, node, loc* *(See Table 10.2-1 for summary of stress quantities.)*
DNORM=	dnorm
INLIB=	inlib
SET=	nset
CON=	ncon
CASES= or VECTORS=	$case_1$, $case_2$ or $vect_1$, $vect_2$
OPTIONS=	n_1 n_2 - - - *(See Table 10.2-2 for summary of options)*

APPENDIX B

ELEMENT FORMULATIONS

Sections 1, 2, and 3 of this appendix summarize the assumptions involved in the routines used by SPAR to compute individual element K, M, and Kg matrices, respectively, relative to local element reference frames. The correspondence between these routines (named K21, TM, M34, GQM, etc.) and the SPAR element type designations (E21, E22, etc.) is indicated in Table B-1.

TABLE B-1 SUMMARY OF K, M, and Kg ROUTINES

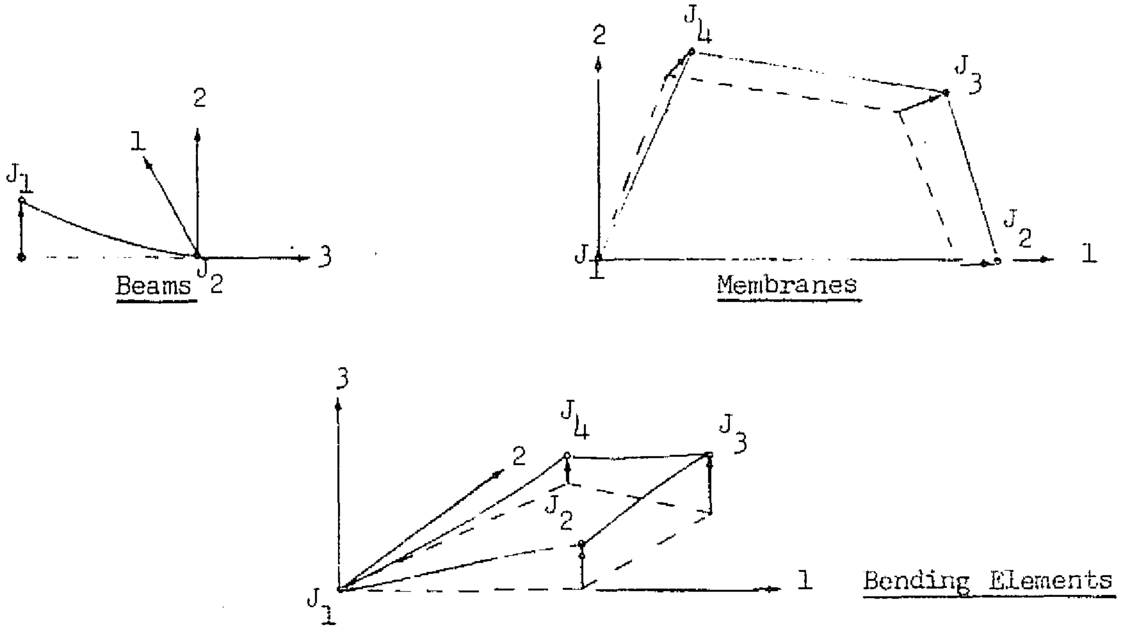
Element Type		K	M	Kg
E21	Gen.beam	K21	M62	BEAMKG
E22	Direct K	(see text)	none	BARKG
E23	Bar	(see text)	M32	BARKG
E24	Pl. beam	(see text)	M62	BEAMKG
E25	Direct K, L=0	(see text)	none	none
E31	Membrane	TM	M33	GTM
E32	Bending	TPB7	M63	none
E33	Mem.+bend.	TM+TPB7	M63	GTP
E41	Membrane	QMB5	M34	GQM
E42	Bending	QPB11	M64	none
E43	Mem.+bend.	QMB5+QPB11	M64	GQP
E44	Shear Panel	QMB1	M34	GQM

1. Stiffness Matrices

Individual element K's computed by the routines K21, TM, QPB11, etc. indicated in the first column of Table B-1 are relative to intrinsic reference frames imbedded in the elements. As indicated on Figure B-1, the intrinsic frames move with the elements as the structure deforms. Some of the principal reasons for using intrinsic frames are summarized below.

(1) K's relative to intrinsic frames totally represent each element's inherent flexibility characteristics in minimal (least data) form.

(2) Transformation, assembly into system K, etc., car be performed by very general routines which are totally unaffected by the details of specific formulations used in computing the element intrinsic K's. As a result, the cost and time required to develop routines to implement various element formulations are minimized, since their total function is to compute an intrinsic K, given the dimensions, section properties, and material constants of the element.



Beams

Membranes

Bending Elements

(1) Dashed lines indicate undeformed shape.

(2) For beams, the origin of the intrinsic frame is imbedded in (rotates with) joint J2, to which the beam terminus connects.

(3) For bending elements, the origin of the intrinsic frame is imbedded in J1.

(4) In the undeformed state, the intrinsic frames coincide with the local "element reference frames."

FIGURE B-1 INTRINSIC FRAMES

For each class of element, an intrinsic deformation vector, U, is defined as follows,

Beams: $U = \left[u^1_1 \; u^1_2 \; u^1_3 \; \theta^1_1 \; \theta^1_2 \; \theta^1_3\right]^t$

3-node membranes: $U = \left|u^2_1 \; u^3_1 \; u^3_2\right|^t$

4-node membranes: $U = \left[u^2_1 \; u^3_1 \; u^3_2 \; u^4_1 \; u^4_2\right]^t$

3-node bending: $U = \left|u^2_3 \; \theta^2_1 \; \theta^2_2 \; u^3_3 \; \theta^3_1 \; \theta^3_2\right|^t$

4-node bending: $U = \left[u^2_3 \; \theta^2_1 \; \theta^2_2 \; u^3_3 \; \theta^3_1 \; \theta^3_2 \; u^4_3 \; \theta^4_1 \; \theta^4_2\right]^t$ (1-1)

In the above, the u^j_i and θ^j_i are displacement and rotation, respectively, of point Jj (i.e. J1, J2, ...) in direction i, relative to the intrinsic frame. For each type of element, the associated intrinsic stiffness matrix, K, is defined by the following equation.

$$\text{Strain energy} = V = 1/2 \; U^t \, K \, U \,. \qquad (1\text{-}2)$$

The order of K is 6, 3, 5, 6, and 9, respectively, for the five classes of U indicated above. For linear elastic materials, K is symmetric.

The methods used to transform and assemble the intrinsic K's into system matrices, etc., will not be discussed here, since the purpose of the present discussion is to explain the basis of the intrinsic K's.

1.1 Beam Intrinsic Stiffness Matrices

For element types E22 and E25, the 6 x 6 intrinsic K, as defined by equations (1-1) and (1-2), is directly read as input. The following discussion defines the basis of E21 and (as special cases), E23 and E24. The characterizing feature of the E21 class of elements is that the force-deflection behavior may be expressed in the form indicated by equation (1-3). The relation between the δ's and the intrinsic deformation vector, U, will be established subsequently.

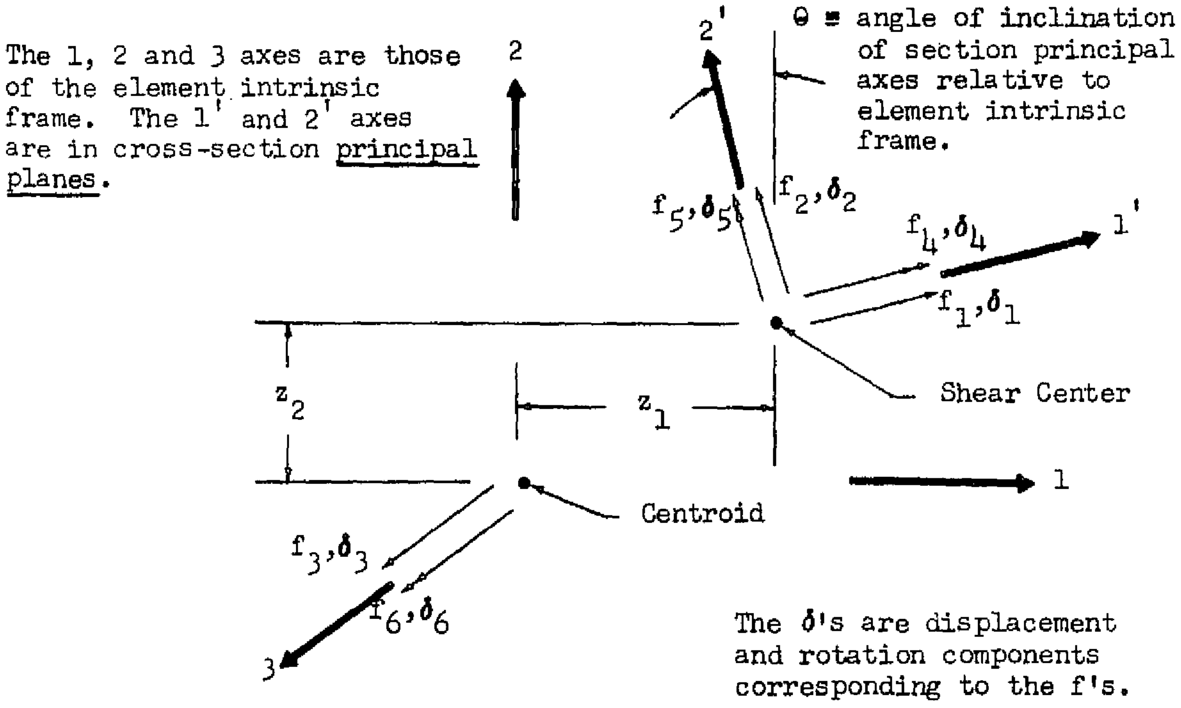
$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} s_{11} & 0 & 0 & 0 & s_{15} & 0 \\ & s_{22} & 0 & s_{24} & 0 & 0 \\ & & s_{33} & 0 & 0 & 0 \\ & & & s_{44} & 0 & 0 \\ & & & & s_{55} & 0 \\ \text{symmetric} & & & & & s_{66} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} , \qquad (1\text{-}3)$$

or,

$$f = s\,\delta \quad , \qquad (1\text{-}4)$$

In the above, the f_i's are forces (i=1,2,3) and moments (i=4,5,6) acting on a rigid plane* imbedded in the beam origin, with the beam terminus fixed. The δ_i's are corresponding displacements and rotations. As defined on Figure B-2, the transverse forces and bending moments do not act through the beam origin; instead they act in principal bending planes - hence the form of equation (1-3).

* The rigid plane is perpendicular to the 3-axis of the intrinsic frame.



The 1, 2 and 3 axes are those of the element intrinsic frame. The 1' and 2' axes are in cross-section principal planes.

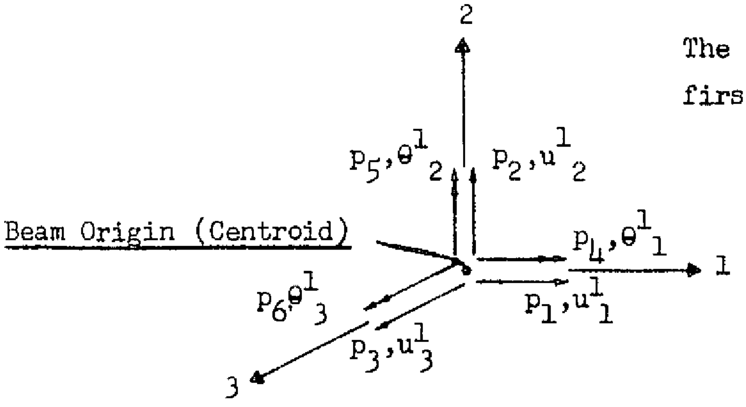
Θ = angle of inclination of section principal axes relative to element intrinsic frame.

The δ's are displacement and rotation components corresponding to the f's.

FIGURE B-2 FORCE AND MOMENT RESULTANTS EXERTED ON THE MEMBER END SURFACE BY THE TRANSVERSE RIGID PLANE

In equation (1-3), s_{11}, s_{15} and s_{55} are associated with bending about principal axis 2; s_{22}, s_{24}, and s_{44} with bending about principal axis-1.

On Figure B-3, components of the element intrinsic deformation vector, and the associated point forces, p_i , and moments are shown.



The u^i_j's correspond to the first of equation (1-1).

FIGURE B-3 FORCES, DEFORMATIONS RELATIVE TO INTRINSIC FRAME

In order to identify K, the equilibrium relation between the p_i's (Fig. B-3) and f_i's (Fig. B-2) will be established. In the following, θ, z_1, and z_2 are as defined on Figure B-2. Also, $c = \cos \theta$, $s = \sin \theta$.

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -z_2 & z_1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 & 0 & 0 \\ s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -s & 0 \\ 0 & 0 & 0 & s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} , \quad (1\text{-}5)$$

or.

$$p = T\, f, \qquad (1\text{-}6)$$

where T is defined by equation (1-5), and $p = \left[p_1 \; p_2 \cdots \; p_6 \right]^t$.

The kinematical relation indicated by equation (1-7) may readily be verified.

$$\delta = T^t\, U\,. \qquad (1\text{-}7)$$

From equations (1-4), (1-6), and (1-7),

$$p = T\ S\ T^{t}U, \text{ or}$$
$$K = T\ S\ T^{t}. \qquad (1\text{-}8)$$

If the shear center and centroid coincide, and the intrinsic frame axes are in principal planes,

$$K = S.$$

The following symbols will be used in identifying the eight non-zero s_{ij}'s.

L = length of beam.

A = cross sectional area,

E = modulus of elasticity,

G = shear modulus,

I_1, I_2 = cross-section moments of inertia about principal axes 1 and 2, respectively,

α_1, α_2 = transverse shear deflection constants (see Timoshenko "Strength of Materials," Part 1, p. 170),

C = uniform torsion constant, and

C_1 = non-uniform torsion constant (see Timoshenko "Strength of Materials," Part 2, p. 255-273).

The axial spring constant is

$$s_{33} = AE/L.$$

The torsional constant is

$$s_{66} = \frac{C}{L}\left(1 - \frac{\tanh b}{b}\right)^{-1}, \quad \text{where} \quad b = \frac{L}{2}\sqrt{C/C_1}$$

which assumes no warping of end cross sections.

Under the usual assumptions of Timoshenko beam theory,

$$\delta_1 = f_1 \frac{L^3}{3\ EI_2} - f_5 \frac{L^2}{2\ EI_2} + f_1 \frac{\alpha_1 L}{GA}$$

$$\delta_5 = -f_1 \frac{L^2}{2\ EI_2} + f_5 \frac{L}{EI_2};$$

L10

or where $e_1 = \frac{EI_2}{GA} \alpha_1$ and $k_1 = \frac{EI_2}{e_1 + L^2/12}$,

$$\begin{bmatrix} \delta_1 \\ \\ \delta_5 \end{bmatrix} = \frac{L^2}{EI_2} \begin{bmatrix} \left(\frac{L}{3} + \frac{e_1}{L}\right) & -\frac{1}{2} \\ \\ -\frac{1}{2} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} f_1 \\ \\ f_5 \end{bmatrix}$$

and

$$\begin{bmatrix} f_1 \\ \\ f_5 \end{bmatrix} = k_1 \begin{bmatrix} \frac{1}{L} & \frac{1}{2} \\ \\ \frac{1}{2} & \left(\frac{L}{3} + \frac{e_1}{L}\right) \end{bmatrix} \begin{bmatrix} \delta_1 \\ \\ \delta_5 \end{bmatrix}$$

From the above, the first set of bending stiffness constants may be identified as:

$$s_{11} = \frac{1}{L} k_1 ,$$

$$s_{15} = -\frac{1}{2} k_1 , \text{ and}$$

$$s_{55} = \left(\frac{L}{3} + \frac{e_1}{L}\right) k_1 .$$

Also,

$$s_{22} = \frac{1}{L} k_2,$$

$$s_{24} = -\frac{1}{2} k_2, \text{ and}$$

$$s_{44} = \left(\frac{L}{3} + \frac{e_2}{L}\right) k_2,$$

where the definitions of k_2 and e_2 are analogous to those of k_1 and e_1.

1.2 Membrane and Bending Elements

In the displacement formulation of linear finite element networks, the boundary motion of each element is written in terms of certain generalized displacement quantities, usually the displacement and rotation components of system joints to which the element is connected. Elements are said to be conforming if the relations between the system generalized displacements and element edge motions are such that displacements are continuous across element boundaries. The basic problem in determining the stiffness matrix of a conforming element may be stated as follows: given the boundary motions of the element, determine the interior stress state. Since the late 1950's, this problem has most commonly been approached through the assumed displacement field - minimum potential energy method. In the July, 1964 issue of the AIAA Journal, T.H.H. Pian proposed a simple, general, and highly effective alternative: the assumed stress field - minimum complementary energy method, which is outlined below.

In terms of generalized displacements, q, (e.g. joint motions), the strain energy of an element is

$$V = \tfrac{1}{2} q^t K q. \qquad (1.2\text{-}1)$$

The minimum complementary energy principal may be stated as

$$\Pi_c = V - \int_{\text{Area}} u_i S_i \, d(\text{Area}) = \text{minimum} \qquad (1.2\text{-}2)$$

where V is strain energy in terms of stress components, σ_{ij}. u_i and S_i are direction i components of prescribed boundary motion and boundary force on the exterior surface.

To apply the principal, the stress distribution, σ, is written in terms of m undetermined stress coefficients, $b_1, b_2, .. b_m$, as follows.

$$\begin{aligned} \sigma &= P B, \quad \text{where} \\ B &= \left[b_1, b_2 \ldots b_m\right]^t, \\ \sigma &= \left[\sigma_{11}, \sigma_{22} \ldots\right]^t, \end{aligned} \qquad (1.2\text{-}3)$$

and elements of P are functions of position coordinates, selected such that equations of equilibrium are identically satisfied within the element, regardless of the values of the b_i's.

Where the stress-strain relation is

$$\epsilon = N \sigma,$$

the internal strain energy within the element is

$$V = \frac{1}{2} \int_{vol} \sigma^t N \sigma \, d(Vol)$$

or,

$$V = \frac{1}{2} B' H B, \qquad (1.2\text{-}4)$$

where

$$H = \int_{Vol} P^t N P \, d(Vol) \qquad (1.2\text{-}5)$$

Where $u = [u_1 \;\; u_2 \;\; u_3]^t$, the relation between boundary motions and generalized displacements, q, is of the form

$$u = Lq, \qquad (1.2\text{-}6)$$

where elements of L are functions of position coordinates, determined by the location of the boundary.

Where $S = [s_1 \;\; s_2 \;\; s_3]^t$, the relation between surface forces and the stress field coefficients is of the form

$$S = R B,$$

where the elements of R are functions of position coordinates determined by the location of the boundary, and by the form of P.

Accordingly,

$$\Pi_c = \frac{1}{2} B^t H B - B^t T q,$$

where

$$T = \int_{Area} R^t L \, d(Area). \qquad (1.2\text{-}7)$$

For minimum Π_c, i.e. $\partial \Pi_c / \partial b_i = 0$, for $i = 1,2,\ldots,$

$$HB = T q,$$
$$B = H^{-1} T q, \qquad (1.2\text{-}8)$$

and, from equation (1.2-4),

$$V = \tfrac{1}{2} q^t T^t H^{-1} T q ,$$

from which it is concluded that

$$K = T^t H^{-1} T. \tag{1.2-9}$$

The procedure outlined above was proposed by Pian as a method of calculating K's approximating those of conforming elements. Where $\bar{K}$ is the stiffness matrix corresponding to the perfect conforming element, K approaches $\bar{K}$ from the soft direction (too much strain energy), as the number of assumed stress functions is increased. In the assumed displacement field - minimum potential energy approach, K approaches $\bar{K}$ from the stiff side as the number of assumed displacement functions is increased.

A network comprised of conforming elements will converge uniformly from the stiff side toward the exact solution, as the mesh is made progressively finer. Even if a mesh were comprised of perfect conforming elements (equilibrium, compatibility equations satisfied over the interior of each element), the network solution would be too stiff. Therefore, K's based on the assumed stress field procedure may be viewed as a means of systematically introducing compensating error (K's softer than $\bar{K}$), to obtain more accurate solutions.

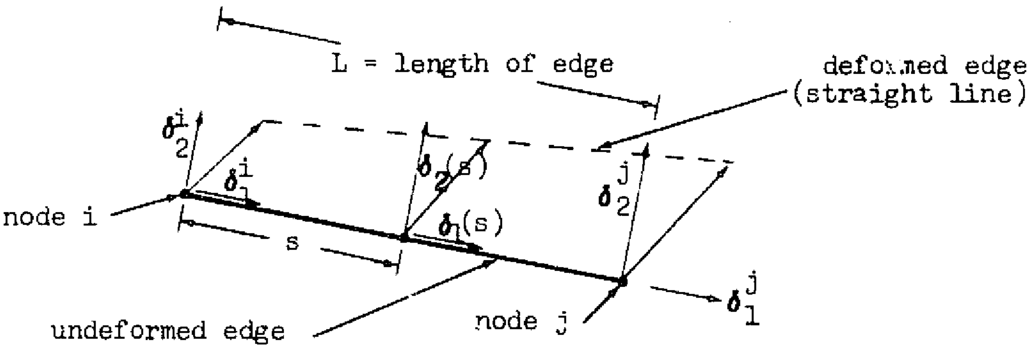
To summarize, the following are the two basic assumptions involved in application of the assumed stress field - minimum complementary energy method.

(1) The boundary motion of the element is assumed as a function of system generalized displacements (e.g. joint displacements, rotations, etc.), in a form which provides inter-element compatibility. This amounts to defining the matrix L in equation (1.2-6).

(2) An equilibrated stress field, expressed in terms of m undetermined coefficients, $b_1, b_2, \ldots b_m$, is assumed. This amounts to defining the matrix P in equation (1.2-3).

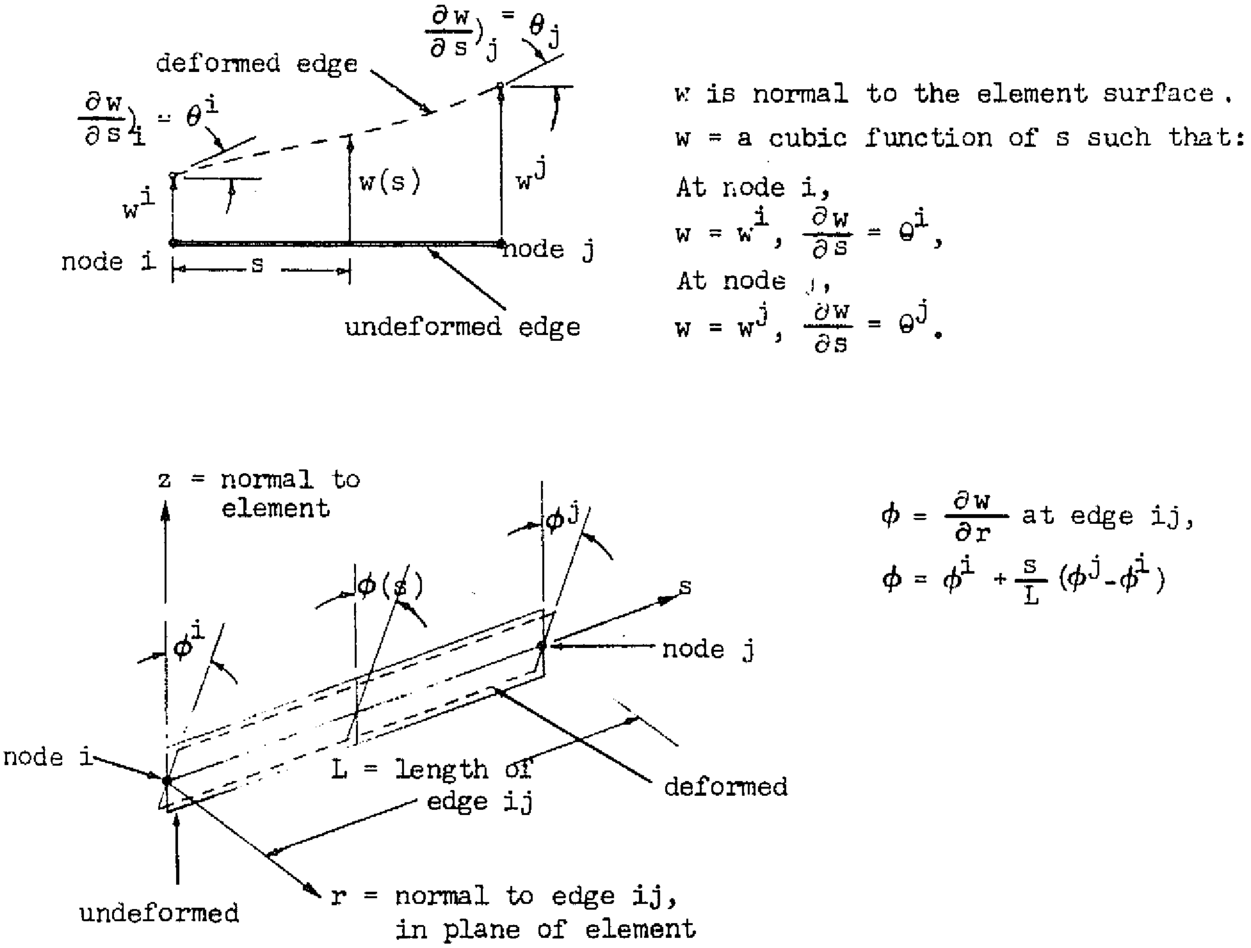
To implement the above assumptions, the principal computational tasks are the calculation of H from equation (1.2-5), and T from equation (1.2-7). From H and T the element stress and stiffness matrices, $H^{-1}T$ and $T^tH^{-1}T$ are readily determined.

The in-plane edge motion assumed for the TM (triangular membrane), QMB5 (quadrilateral membrane), and QMB1 (quadrilateral shear panel) elements is shown on Figure B1.2-1. The edge motion assumed for the TPB7 (triangle), and QPB11(quadrilateral) bending elements is shown on Figure B1.2-2.



$$\delta_k = \delta_k^i + \frac{s}{L}\left(\delta_k^j - \delta_k^i\right), \text{ for } k = 1, 2$$

FIGURE B1.2-1 IN-PLANE EDGE MOTION ASSUMED IN TM, QMB5, QMB1


deformed edge
$\left.\frac{\partial w}{\partial s}\right)_j = \theta_j$
$\left.\frac{\partial w}{\partial s}\right)_i = \theta^i$
w^i
$w(s)$
w^j
node i
s
node j
undeformed edge
z = normal to element
ϕ^j
$\phi(s)$
s
ϕ^i
node j
node i
L = length of edge ij
deformed
undeformed
r = normal to edge ij, in plane of element

w is normal to the element surface.

w = a cubic function of s such that:

At node i,

$w = w^i$, $\frac{\partial w}{\partial s} = \theta^i$,

At node j,

$w = w^j$, $\frac{\partial w}{\partial s} = \theta^j$.

$\phi = \frac{\partial w}{\partial r}$ at edge ij,

$\phi = \phi^i + \frac{s}{L}(\phi^j - \phi^i)$

FIGURE B1.2-2 EDGE MOTION ASSUMED IN TPB7, QPB11

Stress fields assumed for TM, QMB1, and QMB5 are indicated below. In the following, x and y correspond to directions 1 and 2 of the intrinsic frame defined on Figure B-1, and N_x, N_y and N_{xy} are in-plane stress resultants. For TM (constant-stress triangular membrane),

$$N_x = b_1,$$
$$N_y = b_2,$$
$$N_{xy} = b_3.$$

For QMB1 (quadrilateral shear panel),

$$N_x = 0,$$
$$N_y = 0,$$
$$N_{xy} = b_1.$$

For QMB5 (quadrilateral membrane),

$$N_x = b_1 + b_2 y,$$
$$N_y = b_3 + b_4 x,$$
$$N_{xy} = b_5.$$

In TPB7 and QPB11, the (Kirchoff) stress resultant-curvature relation is of the following form.

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{bmatrix} -\partial^2 w/\partial x^2 \\ -\partial^2 w/\partial y^2 \\ 2\ \partial^2 w/\partial x \partial y \end{bmatrix} \qquad (1.2\text{-}10)$$

The stress fields assumed in TPB7 and QPB11 are derived, using equation (1.2-10), from the displacement functions indicated below, which satisfy both equilibrium and compatibility* equations.

* This is not essential to application of the assumed stress field-minimum complementary energy method. It is necessary only that the assumed stresses satisfy all equations of equilibrium.

For TPB7 (triangle):

$$w = -b_1 \frac{x^2}{2} - b_2 \frac{y^2}{2} + b_3 \frac{xy}{2} - b_4 \frac{x^3}{6} - b_5 \frac{y^3}{6} - b_6 \frac{x^2y}{2} - b_7 \frac{xy^2}{2} \qquad (1.2\text{-}11)$$

For QPB11 (quadrilateral):

$$w = \text{same as for TPB7} - b_8\left(\frac{x^2y^2}{2} + \alpha_1 \frac{x^4}{12}\right) - b_9\left(\frac{xy^3}{6} + \alpha_2 \frac{x^4}{12}\right)$$

$$- b_{10}\left(\frac{x^3y}{6} + \alpha_3 \frac{x^4}{12}\right) - b_{11}\left(\frac{y^4}{12} + \alpha_4 \frac{x^4}{12}\right), \qquad (1.2\text{-}12)$$

where

$$\alpha_1 = -(2D_{12} + 4D_{33})/D_{11},$$

$$\alpha_2 = 2D_{23}/D_{11},$$

$$\alpha_3 = 2D_{13}/D_{11},$$

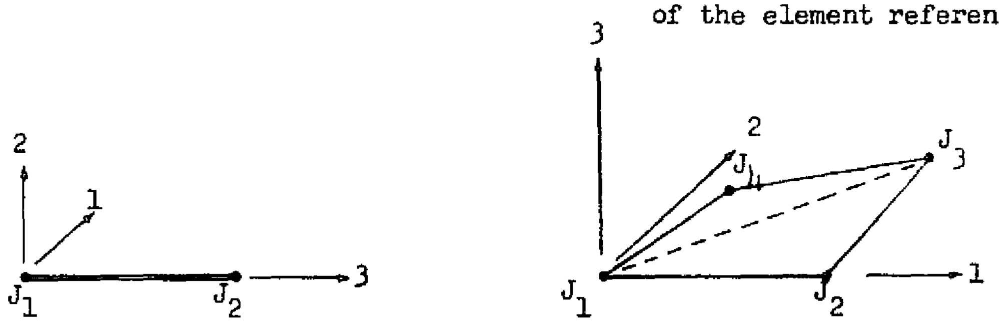
$$\alpha_4 = -D_{22}/D_{11}.$$

Calculation of H and T (see equations (1.2-5) and (1.2-7) for the above elements involves area and line integrals, respectively, of polynominals of varying degrees of complexity. For the simple terms, closed form expressions are readily determined. For the complicated forms, Gaussian integration of order sufficient to assure exact integrals is used.

The routines which compute intrinsic K's for the TM, QMB1, QMB5, TPB7, and QPB11 elements were written by C. L. Yen.

2. Element Mass Matrices

For immediate reference, the SPAR element reference frame convention is shown below.



J_1, J_2, J_3, J_4 all are in plane 1-2 of the element reference frame

2-node elements

3 or 4-node elements

FIGURE B2-1 ELEMENT REFERENCE FRAMES

In the following, u^i_k and θ^i_k are the total displacement and rotation, respectively, of node J_i in direction k, relative to the element reference frames. Local generalized displacement vectors, q, for individual element categories are defined as follows.

Where $u^i = \lfloor u^i_1 \quad u^i_2 \quad u^i_3 \rfloor$,

for bars (M32),

$$q = \lfloor u^1 \quad u^2 \rfloor^t .$$

For triangular membranes (M33),

$$q = \lfloor u^1 \quad u^2 \quad u^3 \rfloor^t .$$

For beams (M62),

$$q = \lfloor u^1_1 \quad u^1_2 \quad u^1_3 \quad \theta^1_1 \quad \theta^1_2 \quad \theta^1_3 \quad u^2_1 \quad u^2_2 \quad u^2_3 \quad \theta^2_1 \quad \theta^2_2 \quad \theta^2_3 \rfloor^t .$$

Where $\psi^i = \lfloor u^i_3 \quad \theta^i_1 \quad \theta^i_2 \rfloor$,

for triangular bending elements (MTP),

$$q = \lfloor \psi^1 \quad \psi^2 \quad \psi^3 \rfloor^t .$$

For quadrilateral bending elements (MQP),

$$q = \left\lfloor \psi^1 \quad \psi^2 \quad \psi^3 \quad \psi^4 \right\rfloor^t.$$

Where q is as defined above, the kinetic energy of an element is expressed in terms of a local element mass matrix, M, as $\frac{1}{2}\dot{q}^t M \dot{q}$. Transformations to system coordinates, assembly into system M, etc., are performed in SPAR by general purpose routines which are independent of the source of the element M's.

In the following discussion, x, y and z are position coordinates in directions 1, 2, and 3 of the element reference frame; and u, v and w are displacement components in directions x, y, and z. In order to compute element M's, displacement functions in terms of n undetermined coefficients, a_1, a_2, . . . a_n, are assumed in the following form.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = G \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \qquad (2\text{-}1)$$

where elements of G are functions of x, y, and z. Where m is mass/volume, the kinetic energy of the element is

$$T = \tfrac{1}{2} \int_{Vol} m\,(\dot{u}^2 + \dot{v}^2 + \dot{w}^2)\,d(Vol). \qquad (2\text{-}2)$$

Where $A = \left\lfloor a_1 \quad a_2 \; . \; . \; . \; a_n \right\rfloor^t$, substitution of (2-1) into (2-2) gives

$$T = \tfrac{1}{2}\, \dot{A}^t \left[\int_{Vol} m \quad G^t\, G\, d(Vol) \right] \dot{A}. \qquad (2\text{-}3)$$

For each element type, n is selected equal to the number of elements in q, so that a unique kinematic relation between A and q of the form indicated by equation (2-4) is readily determined.

$$q = C\,A, \qquad (2\text{-}4)$$

where the elements of C depend on the locations of the node points. Substitution of equation (2-4) into (2-3) identifies the element mass matrix as

$$M = (C^{-1})^t \left[\int_{Vol} m\, G^t\, G\, d(Vol) \right] C^{-1}. \qquad (2\text{-}5)$$

The displacement fields assumed for each type of element are indicated below.

For bars (M32),

$$\begin{aligned} u &= a_1 + a_2 z, \\ v &= a_3 + a_4 z, \\ w &= a_5 + a_6 z. \end{aligned} \tag{2-6}$$

For triangular membranes (M33),

$$\begin{aligned} u &= a_1 + a_2 x + a_3 y, \\ v &= a_4 + a_5 x + a_6 y, \\ w &= a_7 + a_8 x + a_9 y. \end{aligned} \tag{2-7}$$

For beams (M62), where u, v, and w are displacements of the central axis, and where φ is the total twist angle,

$$\begin{aligned} u &= a_1 + a_2 z + a_3 z^2 + a_4 z^3, \\ v &= a_5 + a_6 z + a_7 z^2 + a_8 z^3, \\ w &= a_9 + a_{10} z, \\ \phi &= a_{11} + a_{12} z. \end{aligned} \tag{2-8}$$

For triangular bending elements (MTP)

$$\begin{aligned} w = {} & a_1 + a_2 x + a_3 x^2 \\ & + a_4 y + a_5 xy + a_6 x^2 y \\ & + a_7 y^2 + a_8 xy^2 + a_9 x^2 y^2. \end{aligned} \tag{2-9}$$

For quadrilateral bending elements (MQP)

$$\begin{aligned} w = {} & a_1 + a_2 x + a_3 x^2 + a_4 x^3 \\ & + a_5 y + a_6 xy + a_7 x^2 y + a_8 x^3 y \\ & + a_9 y^2 + a_{10} xy^2 \\ & + a_{11} y^3 + a_{12} xy^3. \end{aligned} \tag{2-10}$$

Using the displacement fields indicated above, the integration indicated by equation (2-5) is readily performed in closed form, except for MTP and MQP, in which cases Gaussian integration is used. In the case of M62, the rotatory inertia terms associated with all three cross section moments of inertia may be included, if desired.

The correspondence between the above and the designations used in Table B-1 is as follows:

M32: As defined above.
M33: As defined above.
M34: Two overlaid pairs of M33's.
M62: As defined above.
M63: M33 + MTP.
M64: M34 + MQP.

It is noted that the above are generally not "consistent" mass matrices, according to the usual definitions since they are derived on the basis of arbitrarily selected displacement fieldswhich only in special cases coincide with the displacement fields associated with the corresponding element K's. They usually give a much better approximation of system kinetic energy than do lumped mass models, however.

The routines which compute the local element mass matrices designated above as M33, M34, M63 and M64 were written by C. L. Yen.

3. Geometric (Initial Stress) Stiffness Matrices

The meaning of the initial stress stiffness matrix, K_g, may be viewed in the following way. Assume that a structure in a given equilibrium configuration undergoes an additional deformation, q. The pre-stress dependent quadratic terms in the associated change in strain energy are $U_1 = \frac{1}{2} q^t K_g q$. The ordinary quadratic terms are $\frac{1}{2} q^t K q$. Accordingly the total change in strain energy associated with the perturbation, q, is approximately $\frac{1}{2} q^t (K_g + K) q$. This provides means of (1) including effects of pre-stress on deformation of linear systems, and (2) performing piecewise linear analysis of geometrically nonlinear problems. The application of K_g to linear bifurcation buckling is based on the following: if K_g corresponds to an applied force system, F, the change in strain energy due to perturbation of a structure to which a load λF is applied is $q^t(\lambda K_G + K)q = 0$. For stationary strain energy, $(\lambda K_g + K)q = 0$.

In the following discussion of the basis of individual element K_g's, the notation (u,v,w, etc.) will be the same as in the discussion of element mass matrices.

For 2-node elements, the principal contributions to K_g are due to changes in axial strain due to lateral deformation, u and v,

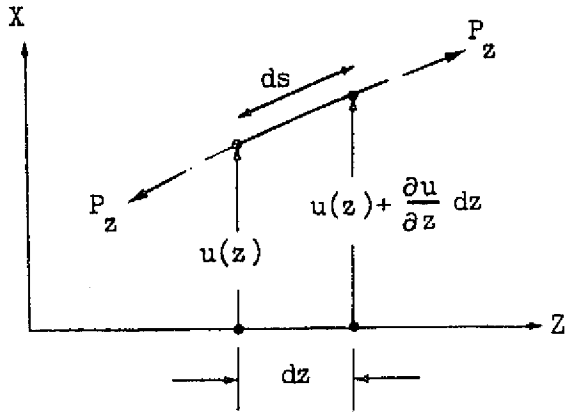
$$\epsilon_z = \frac{1}{2}\left(\frac{\partial u}{\partial z}\right)^2 + \frac{1}{2}\left(\frac{\partial v}{\partial z}\right)^2 .$$

Where P_z is the pre-stress axial force in the element, the corresponding change in strain energy within the element is

$$U_1 = \int_0^L P_z \epsilon_z \, dz = \frac{1}{2} P_z \int_0^L \left[\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right] dz. \qquad (3\text{-}1)$$

A planar example is shown on Figure B3-1.

The BEAMKG formulation used in SPAR for general beam elements also contains terms which include the effects of transverse shear forces, bending moments and torque. The terms produced by BEAMKG are identical to those described in Section 7.2.3 of the Nastran Theoretical Manual, as corrected on 12-15-72.



$$\epsilon_z = \frac{ds-dz}{dz} = \left[1 + \left(\frac{\partial u}{\partial z}\right)^2\right]^{\frac{1}{2}} -1 = \tfrac{1}{2}\left(\frac{\partial u}{\partial z}\right)^2 + \ldots$$

FIGURE B3-1 K_g EFFECTS IN A PLANE BEAM

In two-dimensional elements the changes in membrane strains due to lateral and in-plane displacements produce the following change in strain energy:

$$U_1 = 1/2 \int_{\text{Area}} \left[N_x \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 \right) + N_y \left(\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2 \right) + 2N_{xy} \left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial w}{\partial y} \right) \right] d(\text{Area}) \qquad (3\text{-}2)$$

In the above, N_x, N_y and N_{xy} are the pre-load membrane stress resultants.

The procedures used to calculate K_g are similiar to those used in calculating element mass matrices. Lateral and in-plane displacements are assumed as functions of n undetermined coefficients, $a_1, a_2, \ldots a_n$. For 2- and 3-node elements, each of the K_g fornulations assumes the same displacement field as a corresponding mass matrix. For the quadrilaterial membrane, GQM, displacements are assumed of the following form:

$$\begin{aligned} u &= a_1 + a_2 x + a_3 y + a_4 xy \\ v &= a_5 + a_6 x + a_7 y + a_8 xy \\ w &= a_9 + a_{10} x + a_{11} y + a_{12} xy. \end{aligned} \qquad (3\text{-}3)$$

The assumed displacement fields used for K_g formulations are tabulated below:

	Assumed Displacements			
K_g Type	u	v	w & ϕ	Element Type
BARKG	Eq(2-6)	Eq(2-6)	Eq(2-6)	E22,E23
BEAMKG	Eq(2-8)	Eq(2-8)	Eq(2-8)	E21,E24
GTM	Eq(2-7)	Eq(2-7)	Eq(2-7)	E31
GTP	Eq(2-7)	Eq(2-7)	Eq(2-7)	E33
GQM	Eq(3-3)	Eq(3-3)	Eq(3-3)	E41,E44
GQP	Eq(3-3)	Eq(3-3)	Eq(2-10)	E43

The membrane stress distribution used in computing K_g for quadrilaterals GQM and GQP is the same as indicated for the QMB5 element (i.e. linearly varying). BEAMKG includes, in addition to the principal terms indicated in Eq.(3-1), other secondary terms involving initial shear stresses and bending moments. As before, the integrations indicated by equations (3-1) and (3-2) in some cases are readily determined in closed form; and in others are evaluated using Gaussian integration. These routines were coded by C. L. Yen.