June 1974 LMSC D403168

1

27

SPAR

REFERENCE MANUAL

by

W. D. Whetstone

(NASA-CR-120504) SPAR REFERENCE MANUAL (Lockheed Missiles and Space Co.) 159 p HC $6.25 CSCL 20K

N75-10495

Unclas
G3/39 53342

Lockheed Missiles & Space Company
Palo Alto, California

FOREWORD

The SPAR system of computer programs was designed and developed by Lockheed Missiles and Space Company for the National Aeronautics and Space Administration under Contract NAS8-26352. This contract was funded jointly by NASA's Marshall and Langley Centers, and the U.S. Army Missile Command. The Contracting Officers' representatives were L. A. Kiefling of the MSFC Aero-Astrodynamics Laboratory and J. C. Robinson of the Structures Division at Langley.

CONTENTS

Section

Section 1

INTRODUCTION

SPAR is a system of related programs which may be operated either in batch or Demand (teletype) mode. Information exchange between programs is automatically accomplished through one or more direct access libraries, known collectively as the data complex. Card input is command-oriented, in free-field form. Capabilities available in the first production release of the system include the following.

- Highly automated data management. Data complex size is unlimited.
- Linear stress analysis. Pressure, inertia, and thermal loading is permitted, in addition to concentrated forces and specified motion.
- Linear bifurcation buckling analysis, including effects of prestress.
- Linear vibrational analysis, including effects of prestress.

Low computer execution costs and large size capacity are achieved using sparse matrix techniques substantially the same as those described in Reference (1), and implemented in several earlier programs, including SNAP (Reference 2). Minimal core requirements permit the system to be operated economically in Demand (teletype) mode. Dynamic modification of core size is permitted. Most SPAR executions rarely require more than $25,000_{10}$-$30,000_{10}$ core locations (total of instruction + data storage). As an example of size capacity, static solutions to $10,000_{10}$ d.o.f. problems can be computed within a total field length of $35,000_{10}$. Much larger problems can be solved if more core is used. The size capacity of the eigensolver is similar.

On UNIVAC-1100 systems SPAR is comprised of an array of separate absolute programs. On CDC-6000 systems, the SPAR programs appear as primary overlays sheltered by a minimum executive. The SPAR CDC exec is configured to allow the same control card/data stream to be read on both UNIVAC and CDC systems. That is, CDC card input records appear as shown below:

```
@ XQT  PROGX
  Data cards read during execution of PROGX.

@ XQT PROGZ
  Data cards read during execution of PROGZ.

@ XQT PROGY
  -
  etc.
7/8/9
```

In the above example, each of the processors, PROGX, PROGZ, - - - - may obtain information from the data complex, and insert data into it. Each direct access library is a drum or disk "file" (in both UNIVAC and CDC terminology). Accordingly the analyst may discontinue computation at any point, and retain all results using either of two simple methods:

(1) By assigning the libraries as cataloged files using standard control card language provided for that purpose on both UNIVAC and CDC systems, or

(2) By copying the file(s) from disk to tape using either (a) system commands (e.g.@COPY,GM), or (b) simple copy commands available in SPAR's data complex utility program, DCU. To restart, the files are copied back to disk.

The analyst has extensive control of which libraries are used as data sources for each processor, and into which libraries processor output data flows.

In problems of small-to-moderate size, the usual practice is to keep all data in one library (i.e., the data complex is a single file). Ultimate simplicity of operation is obtained in this way, especially if the file is cataloged. To resume an earlier analysis, for example, it is sufficient to @ASG (or ATTACH) the file, then SPAR processors may be executed as though the previous run had not been terminated.

Within the SPAR processors, the libraries are identified by logical unit numbers 1, 2, 3, - - - 26. These are not Fortran unit numbers. The UNIVAC internal file names corresponding to libraries 1, 2, - - - are SPAR-A, SPAR-B, - - -. On CDC the corresponding file names are SPARLA, SPARLB, - - -. SPAR reserves units 21-26 for temporary (scratch) use by processors. On both CDC and UNIVAC systems, SPAR dynamically assigns new files as they are needed. The analyst assigns (or attaches) existing files, if needed.

Each library has imbedded in it a Table of Contents (TOC), which is automatically maintained by the SPAR information management system (IMSYS), through which processors communicate with the data complex. When a processor inserts a new data set into a library, IMSYS adds to the TOC a line containing the following information:

- The four-word name, such as VIBR EVAL 1 0 , identifying data set. Usually the names are automatically assigned by the originating processor, although the analyst may occasionally assign part or all of a name through control card input.
- The date and time the data set was created.
- Data type and error codes. If an abnormal condition (e.g. excessive round-off error) is detected by the processor originating the data set, the error code is set to indicate (1) warning, or (2) fatal error, as appropriate. This makes it possible for processors to avoid wasted effort by automatically declining to operate on data marked in error.
- Several words describing the size and configuration of the data set. A single data set may be millions of words long, comprised of many physical records, or it may be a small single-block entry, or anything in between.
- Disc address information enabling IMSYS to retrieve the data upon request.

TOC printouts, such as the one shown on Figure 1-1, are very useful in helping analysts keep track of the progress of large complicated studies. They comprise a complete record, in chronological order, of the creation of all data sets.

Before IMSYS inserts a data set into a library, it scans the TOC and disables the existing data set (if present) having the same name. In TOC printouts, disabled data sets are indicated by a minus sign preceding "RR", as shown on Figure 1-1. Disabled data sets can be re-enabled, if required, through the data complex utility program, DCU.

If any of the four words of a data set name are replaced by MASK, those words are not used by IMSYS in scanning the TOC. That is, a data set named GY B 7 412 could be referenced as GY MASK 7 MASK.

Some of the primary data sets generated by SPAR processors are indicated below.

- Individual tables used in constructing the basic definition of the structure, e.g. material constants, section properties, nonstructural distributed weight, joint locations, rigid lumped masses, element definitions in primary form, etc.

- Detailed element information packets. Depending upon the type of element, each packet usually contains the following:

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

(2) Material constants.

(3) Detailed geometrical data, e.g. dimensions, transformations.

(4) Section properties.

(5) Intrinsic stiffness matrix, or equivalent.

(6) Intrinsic stress matrix, or equivalent.

- System M, K, Kg, matrices, and linear combinations of them, in SPAR's sparse matrix format.
- Factored system matrices in sparse matrix format.
- Applied loading and constraint data.
- Solution data such as displacements, stresses, eigensolutions, etc.

```
TABLE OF CONTENTS, DAL  1

SATURN V LAUNCHER UMBILICAL TOWER (LUT)

                                          E               ROWS    BLK   T    DATA SET NAME
SEQ        RR   DATE    TIME    R    WORDS  /BLK   STZE   Y    N1   N2      N3    N4
  1        11 061074 112901  0       18      1     18   0 JDF1 BTAB     1     8
  2        12 061074 112901  0      372    372    372   0 JREF BTAB     2     6
  3        26 061074 112901  0       12      1     12   1 ALTR BTAB     2     4
  4        27 061074 112901  0       19      1     19   4 NDAL          0     0
  5        28 061074 112902  0       80      4     80   3 TEXT BTAB     2     1
  6        31 061074 112902  0       10      1     10   1 MATC BTAB     2     2
  7        32 061074 112903  0      372    372    372   0 CON           1     0
  8        46 061074 112903  0       10      2     10   1 MREF BTAB     2     7
  9        47 061074 112906  0     2418     78   2418   1 BA   BTAB     2     9
 10       134 061074 112909  0     1116    372   1116   1 JLOC BTAB     2     5
 11       174 061074 112909  0     2232    372   2232   1 RMAS BTAB     2    18
 12       254 061074 112909  0     3348    372   3348   1 QJJT BTAB     2    19
 13       374 061074 112921  0    15104    944    896   0 DEF  E21      1     2
 14       918 061074 112921  0        8      4      8   0 GD   E21      1     2
 15       919 061074 112921  0       60      4     60   0 GTIT E21      1     2
 16       922 061074 112921  0       12     12     12   0 NELZ BTAB     1    11
 17       923 061074 112921  0        5      1      5   0 KE            0     0
 18       924 061074 112921  0        6      1      6   0 NS            0     0
 19       925 061074 112921  0        1      1      1   3 ELTS NAME     0     0
 20       926 061074 112921  0        1      1      1   0 ELTS LTYP     0     0
 21       936 061074 112921  0        1      1      1   0 ELTS NNOD     0     0
 22       937 061074 112921  0        1      1      1   0 ELTS ISCT     0     0
 23       938 061074 112921  0        1      1      1   0 ELTS NELS     0     0
 24       939 061074 112921  0        1      1      1   0 ELTS LE3      0     0
 25       940 061074 112929  0    10752    372   1344   0 KMAP       1316    20
 26      1324 061074 112932  0    12096    372   1344   0 AMAP *00) 2223    71
 27      1756 061074 112957  0   132160    944    140   4 E21           1     2
 28      6476 061074 112943  0       20     20     20   0 DIR  E21      1     2
 29      6477 061074 113057  0    51520    372   2240   1 K    SPAR    36  1316
 30      8317 061074 113129  0    90944    372   3136   1 INV  K        1  2223
 31     11565 061074 113242  0    51520    372   2240   1 M    SPAR    36  1316
 32     13405 061074 113249  0    51520    372   2240   1 M+RM SPAR    36  1316
 33     15245 061074 113258  0     2232    372   2232  -1 M+RM DIAG     0     0
 34     15325 061074 113303  0     2232    372   2232  -1 RBMX VEC      1     0
 35     15405 061074 113306  0     2232    372   2232  -1 RBMX VEC      2     0
 36     15485 061074 113309  0     2232    372   2232  -1 RBMX VEC      3     0
 37     15565 061074 113310  0     2232    372   2232  -1 NFM           0     1
 38     15645 061074 113310  0       15      1     15   4 CASE ****     0     1
 39     15646 061074 113310  0        0      0      0   0 LDIR          1     1
 40     15648 061074 113331  0     2232    372   2232   1 SSOL U        1     1
 41     15735 061074 213213  0    51520    372   2240   1 KG   SPAR    36  1316
 42     17575 061074 213215  0    51520    372   2240   1 K+KG SPAR    36  1316
 43     19415 061074 213303  0    90944    372   3136   1 INV  K+KG     1  2223
 44     22063 061074 213311  0        5      5      5  -1 VIBR EVAL     1     0
 45     22064 061074 213312  0    11160    372   2232  -1 VIBR U        1     0
```

Fig. 1-1

Section 2

GENERAL INFORMATION

It is assumed that the reader is familiar with linear finite element theory, as applied to static, vibrational, and bifurcation buckling analysis of structures.

Table ELD-1, near the beginning of Section 3.2, contains a brief summary of the types of elements currently available in SPAR. Appendix B gives details of the formulations on which individual element K, M, and K_g calculations are based. Reference 3 contains a series of annotated example problems.

Before reading further, the new user should examine Table ELD-1, and briefly scan Reference 3 to get a general idea of SPAR's control card - data card structure.

2.1 Reference Frames

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 Card Input Rules

The SPAR free-form input decoder recognizes three types of words: integer, floating-point, and typeless. Leading blanks are ignored. Each word is ended by a blank, comma, equal sign, slash, 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. Typeless 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, ≤ on CDC) terminates one record and initiates a new record on the same card. For example,

```
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 Processor Execution

The short summary of processor functions given at the beginning of Section 3 should be thoroughly examined before reading further in the present discussion.

Typical execution sequences are illustrated below.

Basic Preparations

```
@ XQT TAB
  input cards

@ XQT ELD
  input cards

@ XQT TOPO

@ XQT E

@ XQT  EKS

@ XQT  K

@ XQT M;  if
  required

@ XQT INV
```

Static Solutions

```
@ XQT  Q
  input cards

@ XQT DSOL

@ XQT LCD
  input cards

@ XQT GSF

@ XQT PSF
```

Form System Kg

```
@ XQT KG
```

Buckling or Vibrational Modes

```
@ XQT EIG
  control cards
```

Of the above, only TAB, ELD, and Q read significant amounts of data through input cards. In TAB, the analyst is permitted to make corrections to individual items in tables of joint locations, section and material properties, etc. This is done by modifying previously created data sets resident in direct access libraries, thereby greatly reducing the card input load on the host system. It is expected that similar pin-point editing capability will be released in the near future for data sets produced by ELD, Q, and other processors.

In preparing finite element models of complicated structures, it is almost invariably best to begin by preparing separate models of individual components and analyzing them to see that they perform reasonably; e.g., by computing response to representative static loadings. Models of components are then merged, perhaps in several stages, with additional testing performed after each merger. In this way, input errors are easily and systematically identified. The analyst working on each individual model usually finds it convenient to use reference frames, joint numbers, section property tables, etc., local to the particular component. In TAB and ELD, various data modification commands are provided to assist in merging models. It is suggested that the new user carefully examine the NREF and MOD commands in TAB, and the MOD and INC commands in ELD. Additional devices to further automate the merging of models are planned for future releases.

Most of the processors generate little or no printout. In the few that do, some degree of control of the quantity is provided by inserting a data card containing the following command:

ONLINE = n,

where n = 0 for minimum (or no) printout, n = 1 for normal printout, n = 2 for more.

2.4 Reset Controls and Core Size Modification

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.

2.5 Processor Output Grouping

The RESET controls and the data complex utility program, DCU, provide the analyst total control over the distribution of data sets among libraries. However, the following rules should be observed uniformly.

- Output from TAB, ELD, E, and EKS should always be in Library 1. Library 1 should be assigned when any processor (except DCU) is executed, since it contains several small directories which are widely used.

- For each associated set of quasi-static load cases, the output of Q, DSOL, GSF and LCD (if used) must be in the same library. It is usually best not to put more than a few dozen independent load cases into a single library. Since GSF output data sets are comparatively large, it is often best to save only the Q and DSOL data sets on archive tapes, in order to cut down on the amount of stored data. Stresses can be recalculated very cheaply, if necessary.

2.6 Data Salvage

If it is necessary to make limited changes in the model (e.g. to study effects of parameter variations, or to patch errors), it is often possible to reuse much of the existing data complex, provided the analyst is sufficiently aware of how information flows among the processors (see Figure 2-1). A few examples are illustrated below:

- Vibrational modes have been computed for a given configuration. It is now required to compute modes for the same structure with different rigid masses attached. The procedure is: (1) re-execute TAB sub-processor RMASS to read the new rigid mass data, (2) re-execute MRM to add the new mass data to the existing system M representing element mass, and (3) re-execute EIG, using the previously computed modes as initial approximations.

- After a system matrix has been factored, it is discovered that incorrect constraint conditions were used. The procedure is: (1) re-execute TAB sub-processor CON to patch the constraint case definition, and (2) re-execute INV.

- After computing vibrational modes, it is discovered that there is an error in cross-section property data resulting in a bad system K, but not affecting M. The procedure is

(1) re-execute the appropriate TAB sub-processor to correct the erroneous item, (2) re-execute E, EKS, K, INV, and (3) re-execute EIG, using the previously computed modes as initial eigenvector approximations, if appropriate.

In selecting processor execution sequences in cases such as those indicated above, users who are not familiar with the system should proceed with caution.

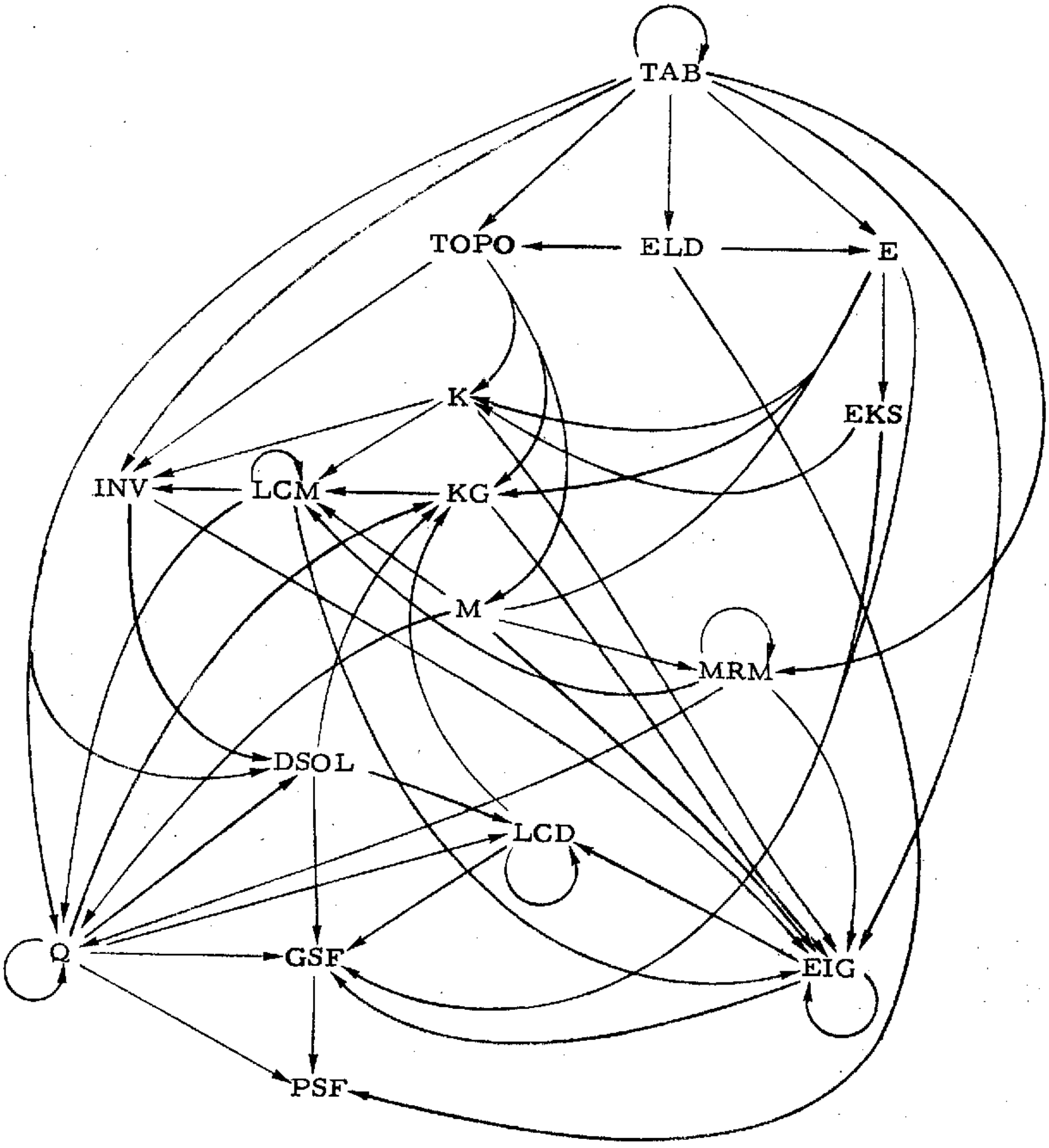


Figure 2-1 **Flow of Information Among Processors**

2.7 Sparse Matrix Solution Method

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 Reference 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 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. It will be necessary for users of SPAR to become familiar with Reference 1 ; especially with regard to the effects of joint numbering sequence on execution costs and data storage requirements.

2.8 Error Messages

Various error messages, usually self-explanatory, may originate in SPAR processors. Messages indicating insufficient core space refer to working core 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.

The most frequently encountered error messages indicate that either a required input data set is not present (IERR=-1), or that it is marked in error (IERR = a positive integr). Both the causes and remedies for such messages are usually obvious.

The messages XLIB = n and ASG = n indicate an attempt to refer to a nonexistent library, n.

Section 3

PROCESSORS

SPAR processors currently available are summarized below.

Section	Processor Name	Function
3.1	TAB	Through an array of sub-processors, card input is translated into tables of material constants, section properties, joint locations, rigid mass data, constraint conditions (any number of separate cases are allowed), joint elimination sequence, etc.
3.2	ELD	Element definitions are read from cards. The definitions refer to entries in the TAB-generated tables of section and material constants, etc. Various automatic mesh generation options are provided.
3.3	TOPO	Element interconnection is analyzed and two data sets produced: one guides assembly of system K, M, and Kg matrices in SPAR's sparse matrix format; the other guides factoring of sparse matrices by processor INV. Size and cost indices are computed permitting accurate estimates to be made of the execution costs which will be required by processors INV and DSOL.

Processors (continued)

Section	Processor Name	Function
3.4	E	Using data sets created by TAB and ELD, information packets are formed to describe each element in detail. The packets contain geometrical details, section and material properties, and other information.
3.5	EKS	Element intrinsic stiffness and stress matrices (or their equivalent) are computed and inserted into the element information packets. These matrices are based on current values of section properties, etc., imbedded in the element information packets.
3.6	K	System K is formed in sparse matrix format.
3.7	M	System M is formed in sparse matrix format.
3.8	KG	System Kg is formed in sparse matrix format.
3.9	LCM	Any required linear combinations of system matrices in sparse format are formed. Examples: K+Kg, K-rM, aK_1+bK_2+ - - - etc.
3.10	MRM	Adds a diagonal matrix to a SPAR-format matrix. Normally used to add effects of rigid mass to a system M.

Processors (continued)

Section	Processor Name	Function
3.11	INV	SPAR-format system matrices such as K, K+Kg, K-rM, etc., are factored using a sparse matrix procedure substantially the same as the one described in Reference 1.
3.12	Q	Through sub-processors, card input is read defining applied quasi-static loading. The following types of loading are included: ● Inertia loads ● Pressure distribution ● Temperature distribution ● Specified nodal motions ● Concentrated loads. Q creates data sets defining load case details, but does not execute the corresponding solutions.
3.13	DSOL	Solutions (joint motions) due to selected load cases are computed.
3.14	LCD	Linear combinations of (joint motion) solutions are formed.
3.15	GSF	Data sets containing element stresses and internal loads are generated.
3.16	PSF	Stress data printout is generated.

Processors (continued)

Section	Processor Name	Function
3.17	EIG	Vibrational and buckling eigenproblems are solved. High-order problems represented by SPAR format M, K, Kg, etc., matrices are solved directly.
4.0	DCU	The data complex utility program, used to copy data sets from one library to another, change data set names, disable (or re-enable) data sets, print data set contents, print library tables of contents, etc.

3.1 TAB

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 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 Reference 1). 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) Both integer and real data may be given in either fixed or floating point format (i.e. the integer 5 may be given as 5. , and the real variable 5. may be given as 5).

(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, JREF, CON and JSEQ. In JLOC, 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.

<u>Correcting existing data sets</u>

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 one 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$
```

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

Table TAB-1, TAB Sub-Processors

Short Name	Table Length*	Long Form Name, Function
TEXT	20 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.

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

Table TAB-1, continued

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.

3.1.1 TEXT

The function of the TEXT sub-processor is to allow the analyst to imbed 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 76 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. In the above,

E = modulus of elasticity,

Nu = Poisson's ratio,

Rho = Weight per unit volume, and

Alpha = Linear thermal expansion coefficient.

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

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

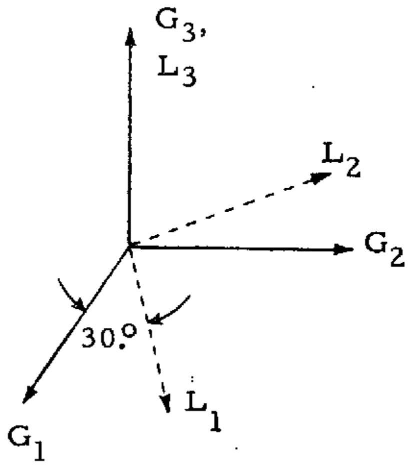
In the above x_1, x_2, and x_3 are position coordinates, relative to the global frame, of the origin of frame k. The x's need not be given if only the orientation of frame k is of significance, which often is the case.

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.

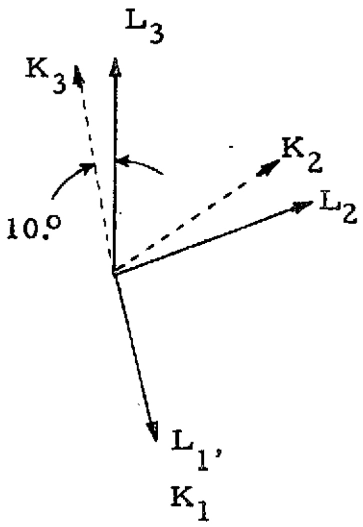
If FORMAT=1(default), the sequence is: (1) rotate the local frame a_1 degrees about local axis i_1, then (2) from the new position, rotate the local frame a_2 degrees about axis i_2, then (3) from the resulting position, rotate the local frame a_3 degrees about axis i_3. If FORMAT=2, the i's and a's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

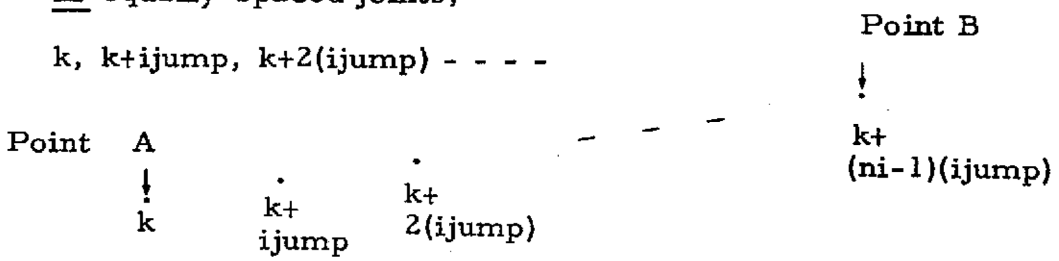
jjump, xc_1, xc_2, xc_3, xd_1, xd_2, xd_3

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.

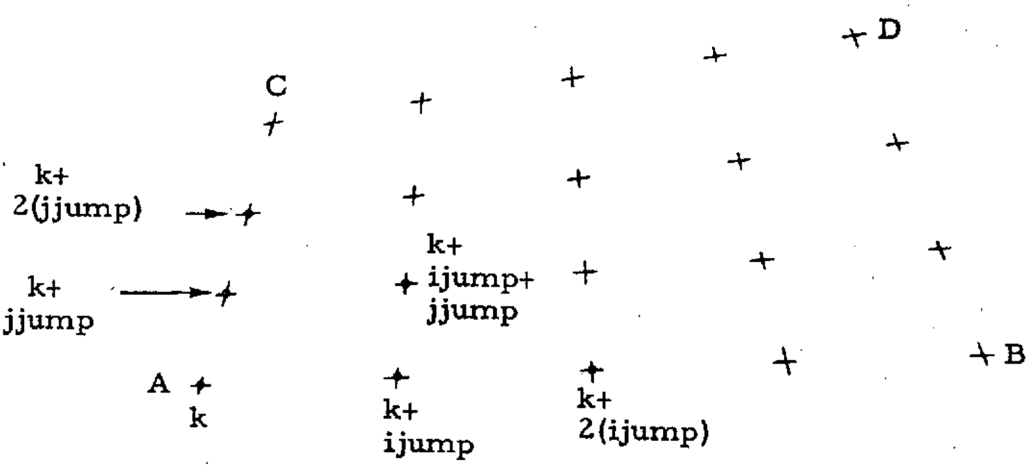
(2) If k, xa_1, - - - - xb_3, ni, and ijump are given, the xa's and xb's are coordinates of points A and B terminating a string of <u>ni</u> equally-spaced joints,

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



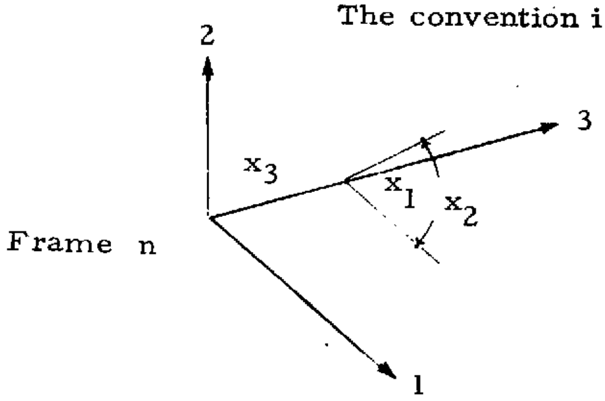
The default value of ijump is 1.

(3) If nj is given, a linearly interpolated two-dimensional mesh of (ni)(nj) joints are 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.



x_1 = radial distance from 3-axis.

x_2 = angle, in degrees, from the 1-3 plane.

x_3 = linear distance from the 1-2 plane.

Control	Meaning
FORMAT =1	Switch back to rectangular coordinates.

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, cylinder 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 processor Q).

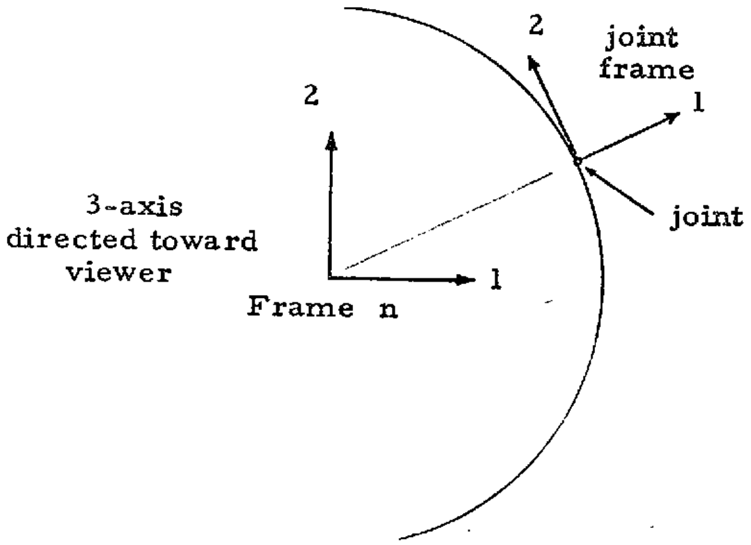
To identify a set of joints with reference frames parallel to frame n (see ALTREF), an input card containing NREF=n is followed by a sequence of records in 'loop limit' format naming the joints. 'Loop limit' format is: j_1, j_2, inc, meaning joints j_1, j_1+inc, j_1+2(inc), - - - j_2. Default for j_2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

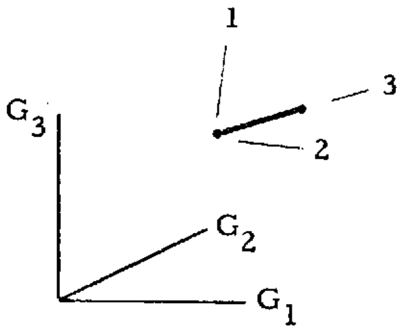
Each two-node element has an "element reference frame" associated with it (see processor ELD discussion). Beam section properties, stresses, etc., are defined relative to these frames. The 3-axis of the frame is directed from the beam's origin to its terminus. The origin and terminus are the ends connected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

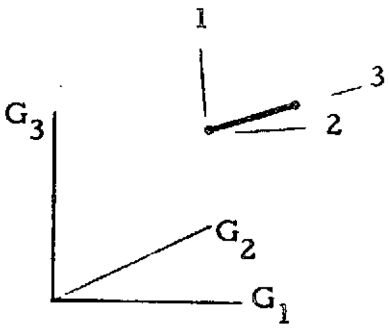
If FORMAT=1 (default), the data sequence defining table entry k is

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.

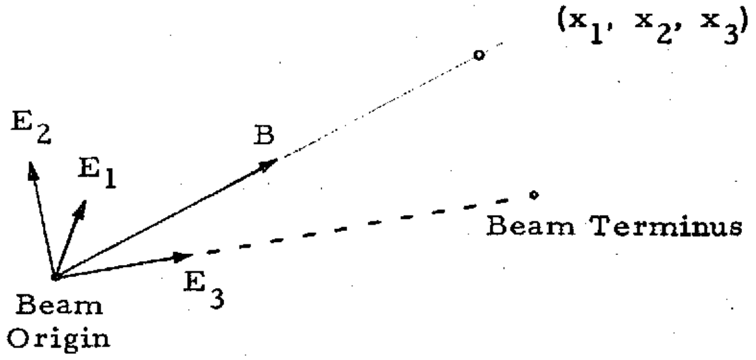
In about 95% of the cases c is either 1.0 or .0.

Care should be taken to supply meaningful information. For example, if the beam 3 axis 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.

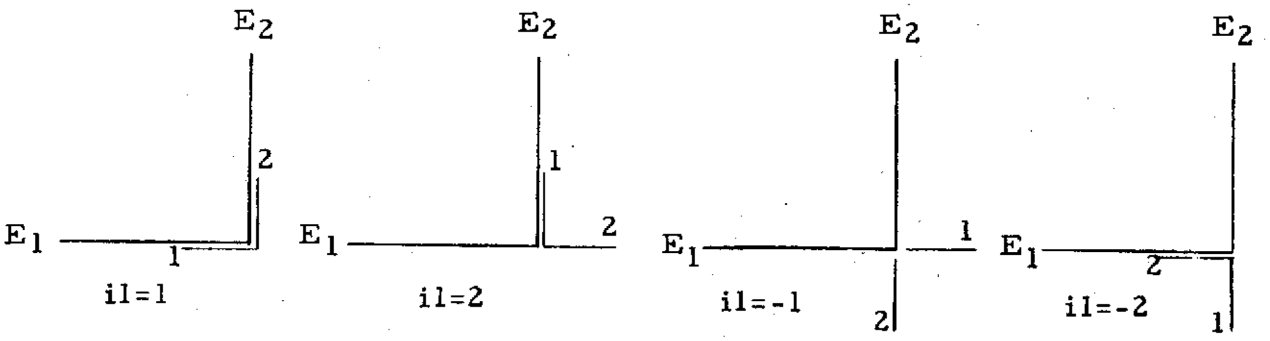
In <u>FORMAT=2</u>, the data sequence defining table entry K is

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

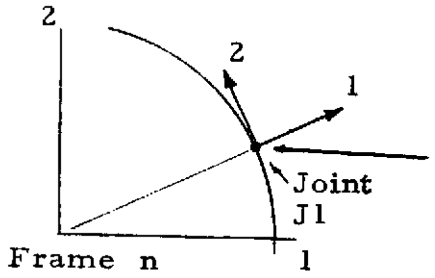
$$k,\ i^1,\ x_1^1\ \ x_2^1\ \ x_3^1 \qquad i^2,\ x_1^2\ \ x_2^2\ \ x_3^2$$

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.



x_2^1, x_2^1, x_3^1
relative to
this frame

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 equal to zero.

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).

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

$C_1 = Ef_1$

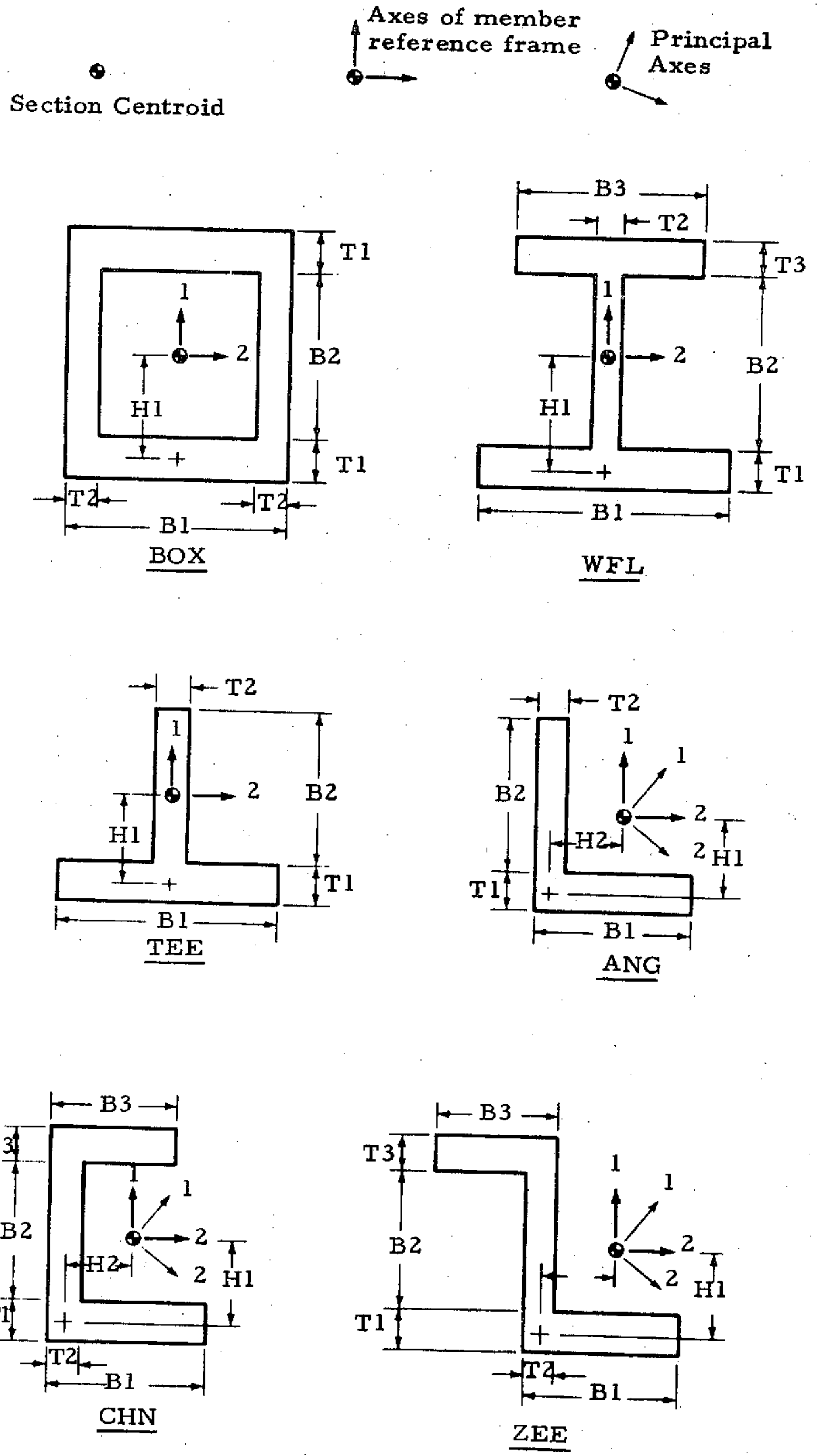

Section Centroid
Axes of member reference frame
Principal Axes
T1
1
2
B2
H1
T1
T2
T2
B1
BOX
B3
T2
T3
1
2
B2
H1
T1
B1
WFL
T2
1
2
B2
H1
T1
B1
TEE
T2
1
1
B2
2
H2
2
H1
T1
B1
ANG
B3
T3
1
1
B2
2
H2
2
H1
T1
T2
B1
CHN
B3
T3
1
1
B2
2
2
H1
T1
T2
B1
ZEE

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

$$k,\ a_t,\ a_c,\ s_t,\ s_c,\ a_{max},\ a_{min}.$$

In the above,

a_t = area effective if in tension,

a_c = area effective if in compression,

s_t = allowable tensile stress,

s_c = allowable compressive stress,

a_{max}= maximum area allowed during redesign, and

a_{min} = minimum area allowed during redesign.

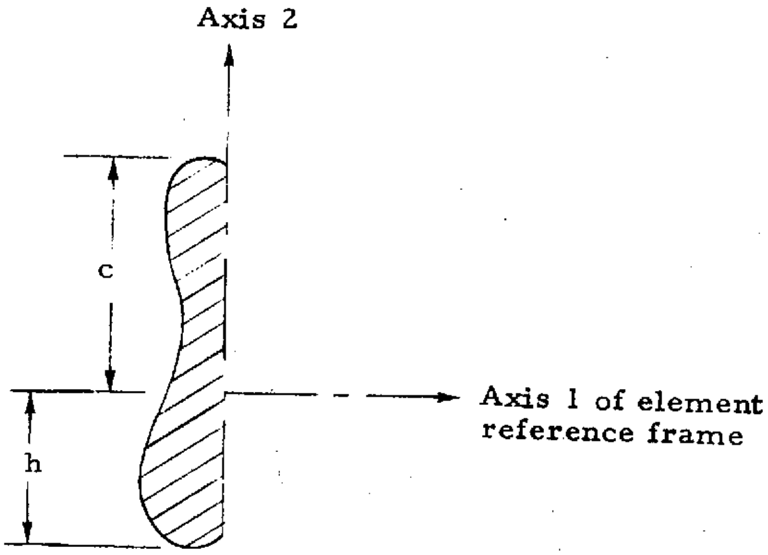
In SPAR level 7, no redesign or iterative analysis processors were released. Element cross-sectional area=a_t. The other parameters a_c, s_t, etc. are not used.

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 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, t, t_{11}, t_{22}, t_{12}.

In the above, t=thickness, and the constants t_{11}, t_{22}, and t_{12} have the same meaning as defined below for aeolotropic sections.

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}

c_{11}, c_{21}, c_{22}, c_{31}, c_{32}, c_{33}

d_{11}, d_{21}, d_{22}, d_{31}, d_{32}, d_{33}.

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

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

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

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

Where s_{11}, s_{22}, and s_{12} are membrane stresses,

$$s_{11} = N_{11}/t_{11}, \quad s_{22} = N_{22}/t_{22}, \quad s_{12} = N_{12}/t_{12}.$$

For isotropic sections, default $t_{11}=t_{22}=t_{12}=t$.

All of the above quantities are relative to the element reference frame (see ELD discussion).

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

$k, \quad t, \quad s, \quad t_{max}, \quad t_{min}.$

In the above,

$t =$ panel thickness, initial design.

$s =$ allowable shear stress.

$t_{max} =$ maximum thickness permitted during redesign.

$t_{min} =$ minimum thickness permitted during redesign.

In SPAR level 7 no redesign processors were released, and the parameters s, t_{max}, and t_{min} are not used.

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) or specified value, possibly not zero. In the second category, specific values of such components may vary from one load case to another, as defined in processor Q.

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 non-zero values assigned in processor Q, 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, 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 processor such as DSOL 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 do one of the following.

(1) Carefully study Reference 1 to learn the factors governing choice of favorable joint numbering sequences. This is not particularly difficult. The rules are less restrictive and in some 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 150 joints:

111, 112, 113, - - - - - 120,
101, 102, 103, - - - - - 110,
131, 132, 133, - - - - - 140,
121, 122, 123, - - - - - 130,
99, 98, 97, - - - - - - - - 3, 2, 1.

The above is represented by the following JSEQ input.

```
REPEAT 2, 20, 100: 11/20, 1/10:
REPEAT 0, 0, 0: 99/1$
```

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 where the data has the same meaning as above, except that Mi 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 Qand EIG.

3.2 ELD

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 should 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:

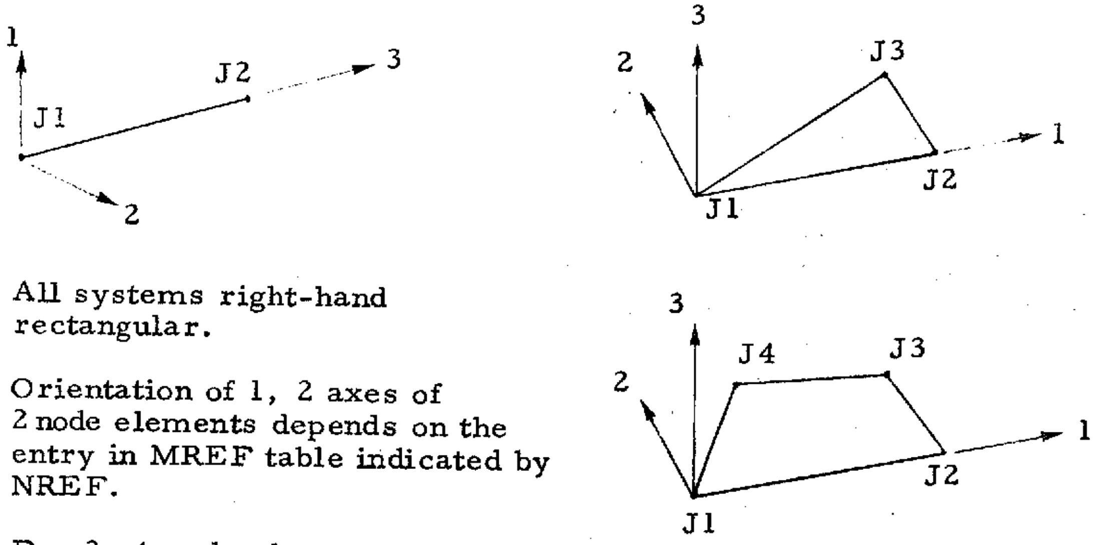
J1, J2, J3, J4 NETOPT, NET(1), NET(2), - - -

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 parameter, 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.



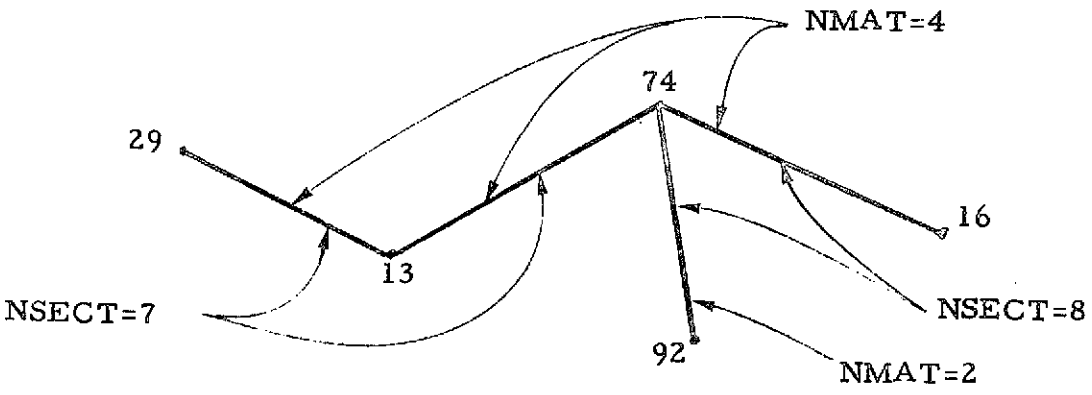
(1) All systems right-hand rectangular.

(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,

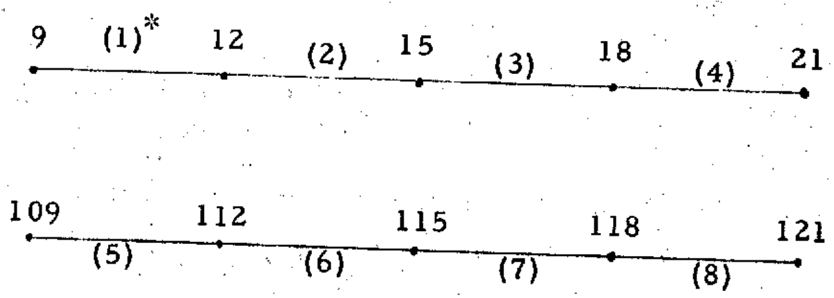
NI= NET(1) (default=1),
NJ= NET(2) (default=1),
JINC= NET(3).

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 define

Two-node element network generators (continued).

<u>If NETOPT=2</u>,

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

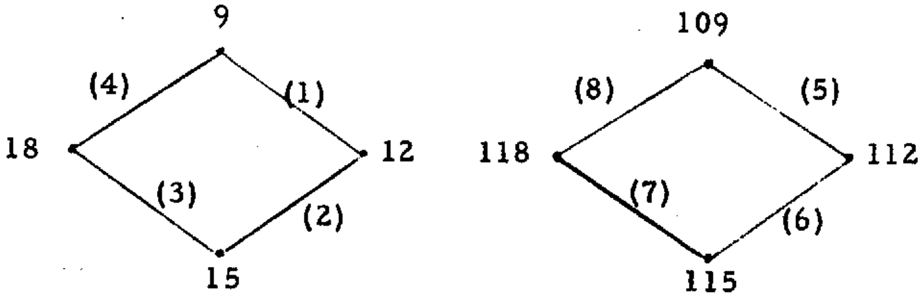
Implied sequence:

```
        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,

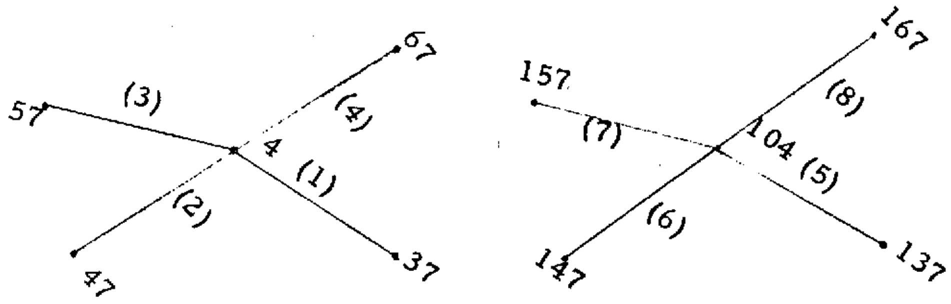
```
NI=    NET(1),
IINC=  NET(2),
NJ=    NET(3)  (default=1)
JINC=  NET(4).
```

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
 200   N2 = J2  + JINC
```

Example:

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



Three-node element network generators.

If Netopt=1,

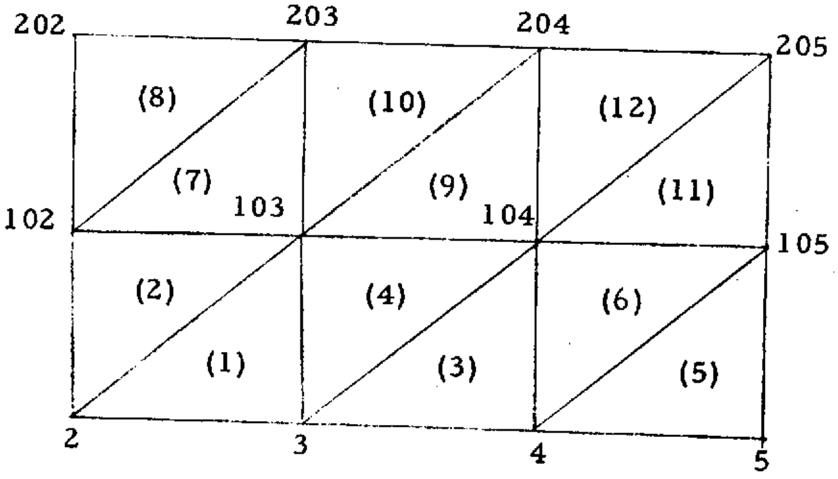
NI= NET(1),
NJ= NET(2) (default=1).

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,

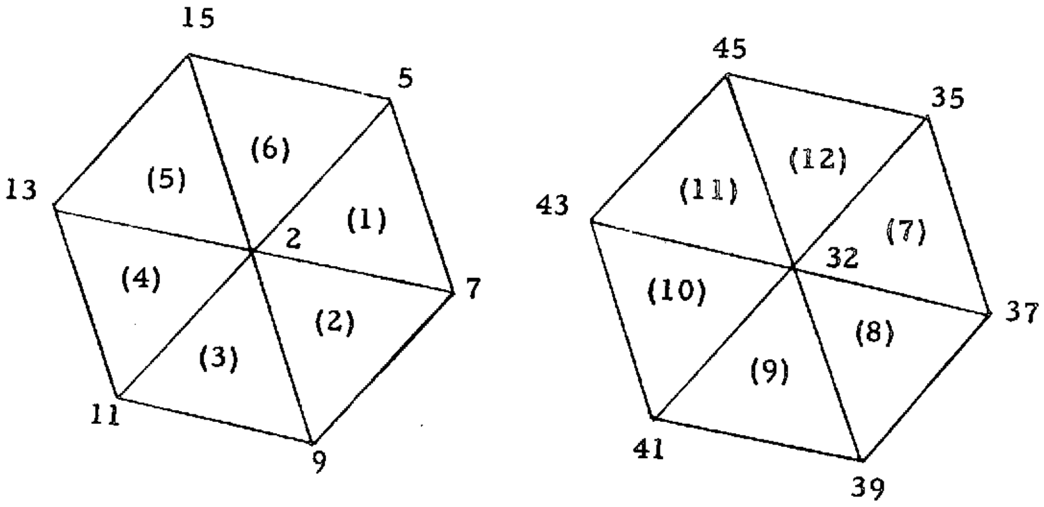
NI= NET(1),
NJ= NET(2) (default=1),
JINC= NET(3).

Implied sequence:

```
        IINC= J3-J2
        N1= J1
        N2= J2
        DO  200  J=1, NJ
        DO  100  I=1, NI
        N3= N2 + IINC (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	2	6	2	30



Four-node element network generators.

If Netopt=1,

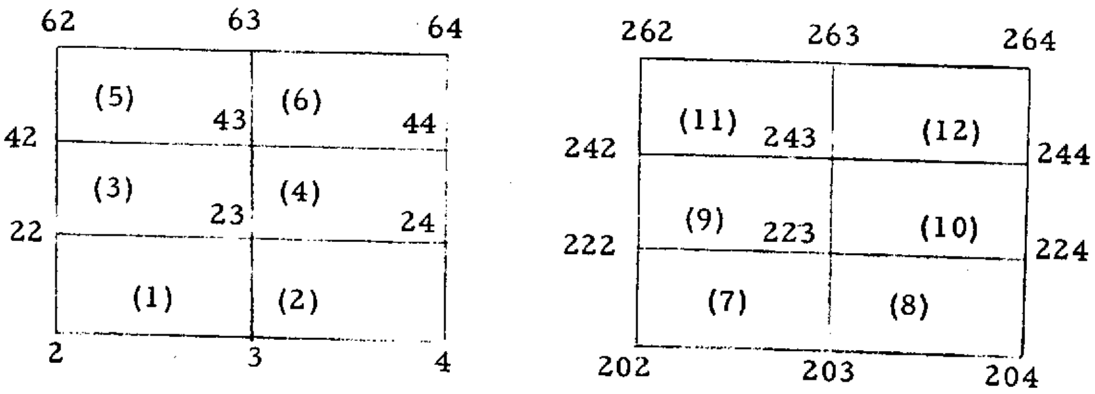
```
NI=    NET(1),
NJ=    NET(2) (default=1),
NK=    NET(3) (default=1),
KINC=  NET(4).
```

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,

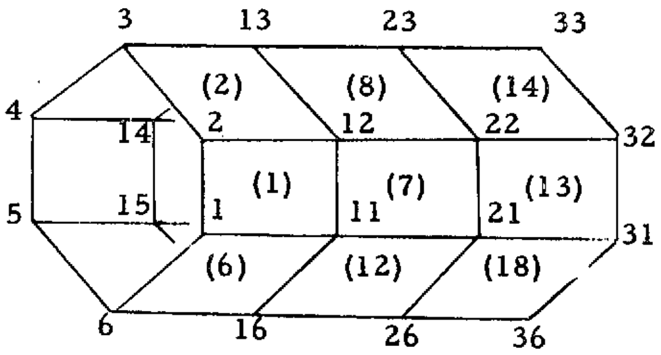
NI= NET(1),
NJ= NET(2) (default=1).

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 TOPO.

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 sub-processor JSEQ.

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	1344	KMAP block length
LRAMAP	1344	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).
ILMAX	(none)	Maximum joint connectivity to be permitted in allocating space for topological tables.

Name	Default Value	Meaning
TIME	0	Nonzero value causes printout of intermediate CP and wall clock times.

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

3.4 E.

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, and

(2) DIR Exx = an associated directory.

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^tR must equal the identity matrix, within a tolerance of RCH.
TIME	0	Nonzero value causes printout of intermediate CP and wall clock times.

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

3.5 EKS.

Function. Based on the dimensions, section properties, etc., currently imbedded 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.

3.6 K

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.
KLIB	1	Destination library for K output data.
LREC	2240	Block length of output data set.

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

- One block of AMAP
- 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 much less than I_s submatrix storage blocks (see Ref. 1).

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.7 M

Function. M assembles unconstrained system consistent mass matrices, considering only the structural and nonstructural distributed mass associated with the elements. That is, rigid masses lumped at joints are not included in M SPAR, the output data set produced by processor M. The first production release of M (level 7) ignores rigid link offsets in its calculations.

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 M 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

	Default	
Name	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.

3.8 KG

Function. KG assembles unconstrained system initial-stress (geometric) stiffness matrices, K_g, in the standard SPAR sparse matrix format. The first production release of KG (level 7) ignores rigid link offsets in its calculations. The output data set is named KG SPAR. Required input data sets are the same as for processor K, plus one additional data set containing the pre-stress solution upon which K_g is to be based. Core requirements are the same as for K, plus additional space (not more than 7 times the number of joints) in which the pre-stress solution is resident.

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

Name	Default Value	Meaning
QLIB	1	Library containing the pre-stress solution on which Kg is to be based.
QSEQ	1	Sequence number identifying the solution case on which Kg is to be based.
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.
ZEROI	$10.^{-10}$	Zero-test parameter used to detect identically-zero beam moments of inertia.
ZEROA	$10.^{-10}$	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.

3.9 LCM

Function. LCM forms linear combinations of system matrices in the SPAR standard sparse matrix format. That is, where Q1 and Q2 are existing system matrices, and C1 and C2 and constants, LCM forms the data set R:

$$R = C1*Q1 + C2*Q2.$$

The default operation of LCM is to add K to Kg.

RESET Controls.

Name	Default Value	Meaning
Q1LIB	1	Library containing Q1.
Q2LIB	1	Library containing Q2.
RLIB	1	Destination library for R.
C1	1.0	Value of C1
C2	1.0	Value of C2
Q1	K	First word in name of Q1.
Q2	KG	First word in name of Q2.
R	K+KG	First word in name of R.

Example. Calculate 3.7 M + 20.0K, where M SPAR is in library 5 and K SPAR is in library 7. The output is to be named CMK SPAR, and stored in library 2.

```
@XQT LCM
RESET  C1 = 3.7, Q1 = M, C2 = 20., Q2 = K
RESET  R = CMK
RESET  Q1LIB = 5, Q2LIB = 7, RLIB = 2
```

Core Requirements. The working core needed is 3 times the block length of Q1. Q2's block length must be equal to that of Q1.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.10 MRM

Function. The function of MRM is to add a diagonal matrix to a system matrix in SPAR's standard sparse matrix format, resulting in the production of two data sets (1) the sum of the two matrices in SPAR format, and (2) the diagonal of the sum of the two matrices.

The normal application of MRM is to add a diagonal mass matrix named RMASS (produced by TAB sub-processor RMASS) to the assembled system consistent mass matrix M SPAR, naming the sum M+RM SPAR. The data set containing the diagonal of the sum is M+RM DIAG.

RESET Controls.

Name	Default Value	Meaning
MLIB	1	Library containing M SPAR.
RMLIB	1	Library containing RMASS.
LIB	1	Destination library, output data sets.

Core Requirements. Where J is the number of joints, and L is the block length of M SPAR, the core required is J + L.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.11 INV

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. 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	1	Library containing topological data set AMAP.

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	3136	Output data set block length.
DZERO	.00001	Zero-test parameter used to detect identically zero diagonal terms during factoring of the system matrix.

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 and

n = no. of degrees of freedom per joint (usually 3, 4, 5 or 6), and

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

$$J + L_1 + L_2 + L_3 + n^2 I_s.$$

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.12 Q

Function. Q reads card input defining static load cases. The card data is translated into various data sets which are subsequently stored in library 1, or any other library selected by the user via RESET Control. The corresponding static solutions are subsequently computed by other processors. The definition of a static loading case is initiated by a command in the following form:

CASE n' ALPHA-NUMERIC TITLE-------------,

where n is an integer uniquely identifying the load CASE. It is advisable, though not required, for CASE 1 to appear first, CASE 2 second, etc. Following the CASE card there are one or more packets of cards, each packet defining one of several types of loading (e.g. pressure, thermal, etc.) which together comprise the CASE. A packet is initiated by any of the following multi-word commands:

NODAL FORCES
NODAL MOTIONS
NODAL TEMPERATURES
NODAL PRESSURES

A particular type of packet may not appear more than once within each CASE. There is no limit on the number of data cards in each packet. The order of individual data items on each card is defined below. Omitted trailing items are assumed zero.

NODAL FORCES:

jo, k, q, ni, iinc, dqi, nj, jinc, dqj.

Interpretation: If k = 1, 2, or 3, q is a concentrated force acting on joint jo in direction k, relative to the joint reference frame associated with joint jo. If k = 4, 5, or 6, q is a moment in direction k - 3.

If given, ni, iinc, etc., imply the following:

> For J = 1 through nj and I = 1 through ni, a direction k load of magnitude F is applied to joint N, where
> N = jo + (J - 1) * jinc + (I - 1) * iinc, and
> F = q + (J - 1) * dqj + (I - 1) * dqi.

Note that the orientation of individual joint reference frames is prescribed via the TAB processor.

NODAL MOTIONS:

Card input is exactly the same as for NODAL FORCES, except that q is interpreted as a specified nodal displacement or rotation. Only motion components declared to be NONZERO in the CONSTRAINT subprocessor of TAB may have non-zero prescribed values. Any motion components declared NONZERO in TAB, but not assigned a value for a particular CASE, are set equal to zero.

NODAL TEMPERATURES:

jo, q, ni, iinc, dqi, nj, jinc, dqj.

Interpretation: q is the temperature of joint jo. Parameters ni, iinc, etc. serve the same function as in the NODAL FORCE input. Note that through RESET control, temperatures may be shifted and scaled.

NODAL PRESSURES:

Data card interpretation is the same as for NODAL TEMPERATURES, except that q is nodal pressure. The positive direction of pressure acting on a 3 or 4 node element is determined by the value of NREF assigned when the element was defined in ELD. If NREF is zero, no pressure acts on the element. If NREF = 1, positive pressure acts in direction 3 of the local element reference frame. If NREF = -1, positive pressure acts in the opposite direction. Pressure scaling and shifting are available through RESET controls.

In addition to the above, each CASE may include inertia loading defined by one or more commands of the following form:

INERTIA FORCE, k, acc.

The above indicates linear acceleration, acc (not a "g" factor), in global direction k (k = 1, 2, or 3). Note that k must be an integer, and acc must be real.

Inertia loads are formed by linearly combining inertia load influence vectors named RBMX VEC (n = 1, 2, 3). If these vectors do not already reside in the Q output library, it will be necessary to RESET MLIB = k, where k is the library containing the system mass matrix to be used in calculating the inertia load influence vectors. In the default mode, inertia forces are based on the assembled system mass matrix M SPAR (see Processor M). Through RESET control other choices are available.

In future releases, a variety of additional methods are planned for defining all of the above, and other types of loading.

For load CASE n, Q produces the following data sets.

Name	Contents
CASE -- n	Alphanumeric title supplied on the input card initiating CASE n.
NTMP -- n	Nodal temperatures (if given).
NMOT -- n	Nodal Motions (if given).
NFM -- n	Total equivalent nodal forces and moments from all sources.

In addition, a data set named LDIR is produced, containing the smallest and largest CASE number processed in the execution. LDIR is used in the default operation of several processors.

RESET Controls.

Name	Default Value	Meaning
ELIB	1	Library containing the element information packets generated by processors E and EKS.
MLIB	0	Library containing the system mass matrix to be used in calculating inertia forces.
QLIB	1	Destination library for all output data sets.
TMULT	1.0	Input nodal temperatures are scaled (multiplied) by TMULT.
TNULL	.0	TNULL is subtracted from scaled nodal temperatures.
PMULT	1.0	Input nodal pressure are scaled (multiplied) by PMULT.

Name	Default Value	Meaning
PNULL	.0	PNULL is subtracted from scaled nodal pressures.
M	M	See below.
MTYP	SPAR	The default name of the mass matrix input is M SPAR. For example, if you elect to use consistent mass matrix M+RM, RESET M = M+RM. If you elect to use only the rigid mass data input in TAB, RESET M = RMASS, MTYP = MASK.

Core Requirements. Working core requirements normally will not substantially exceed 12 to 15 times the number of joints in the structure.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.13 DSOL

Function. DSOL computes static solutions in the form of joint motion vectors. Input to DSOL consists of the following.

- An assembled system matrix (default is K SPAR), and its pseudo-inverse (default is INV K 1), calculated by INV.
- Data sets comprised of equivalent nodal forces and moments, and/or specified nodal motions. These are normally data sets produced by processor Q, bearing the names NFM --n, and NMOT -- n, where n is the load CASE identifier.
- The data set LDIR produced by Processor Q, identifying the smallest and largest CASE numbers processed in the last execution of Q.

The joint motion data output corresponding to CASE n is named SSOL -- n.

RESET Controls.

Name	Default Value	Meaning
KLIB	1	Library containing systems stiffness matrix.
KILIB	1	Library containing factored system stiffness matrix.
QLIB	1	Library containing loading data produced by processor Q, into which DSOL will insert its results.
K	K	First word of the name of the system stiffness matrix.
CON	1	Constraint case used in factoring the system stiffness matrix.
L1	None	See below.

Name	Default Value	Meaning
L2	None	If both L1 and L2 are given, load CASES L1 through L2 are processed. If L1 alone is given, only that case is processed. If neither L1 nor L2 is given, all CASES generated in the last execution of Q are processed.
ROUNDOFF	.0001	Maximum normalized error to be permitted in displacement solutions. Data sets corresponding to non-qualifying solutions are markedin error.
MAXIT	1	Maximum number of iterations allowed in attempting to recover from round-off error. A value of zero causes nodal reaction - force check calculations to be omitted.
EP	1	Nonzero value results in printout of normalized errors.
UP	0	Nonzero value results in printout of joint motions.
FP	0	Nonzero value results in printout of joint force summations (reactions, nodal force balance errors).
TIME	1	Nonzero value causes printout of intermediate CP and wall clock times.

Core Requirements. Working core required will not significantly exceed 12 times the number of joints, plus the block length of the factored system stiffness matrix.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.14 LCD

Function. The primary function of LCD is to form linear combinations of solutions in the form produced by processors Q and DSOL. Card input to LCD consists of a sequence of packets. Each packet may be in the following form.

CASE j' -- alphanumeric title for case j ----
L_1, N_1, C_1
L_2, N_2, C_2
- - -
L_r, N_r, C_r

The above is interpreted as follows: Form a new solution case, identified by the integer j, comprised of

C_1 times solution case N_1 from library L_1
$+C_2$ times solution case N_2 from library L_2
$+$ - - -
$+C_r$ times solution case N_r from library L_r.

If C is omitted, it is assumed to be 1.0. The alphanumeric label given on the first card of the packet serves the same purpose as in processor Q; i.e. to identify CASE j. LCD can be used to combine results produced by LCD.

The following more general packet form is also permitted.

CASES j, k' title (applies to cases j through k)

L_1 N_1 C_1 I_1

L_2 N_2 C_2 I_2

- -

L_r N_r C_r I_r

The above language results in generation of a succession of combined cases: j, j + 1, ---- k. Case j is constructed as previously defined. Then I_i is added to N_i (for i = 1, 2, ---r), and case j + 1 is formed by the same rule, etc. The default value of I_i is 1. Each case, j, j + 1 ---k, is given the same title from the label field of the CASE card.

For example, suppose it is desired to define combined cases 40, 41, ----- 47 as follows:

Case 40 = 4.5 (case 130) + 2.1 (case 22) + 6.7 (case 204)

Case 41 = 4.5 (case 131) + 2.1 (case 24) + 6.7 (case 204)

-

-

Case 47 = 4.5 (case 137) + 2.1 (case 36) + 6.7 (case 204)

If cases 130 - 137 are in library 3, cases 22 - 36 in library 2, and case 204 in library 5, the following packet produces the desired result.

```
CASES  40, 47' ----- TITLE -----
  3   130  4.5
  2    22  2.1  2
  5   204  6.7  0
```

The above language is useful in executing extensive sequences of combined loads (e.g. symmetric + antisymmetric, thermal + mechanical, static + dynamic, etc.)

The title of existing case n, originated either in Q and DSOL or in LCD, may be changed by a card in the following form.

CASE n' ------- New Title -------

RESET Controls.

Name	Default Value	Meaning
OUTLIB	1	Output data set destination library.
THERM	1	If no thermal loading is present, setting ITHERM = 0 will slightly reduce execution costs.

Core Requirements. Working core requirements will not significantly, exceed 13 times the number of joints in the structure.

Code ReleaseData. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.15 GSF

Function. GSF generates element stresses. Input to GSF normally consists of the data sets generated by Q and DSOL, or combined solutions produced by LCD. GSF also accesses the element information packets originated by processors E and EKS. Output data sets have names of the form STRS EXXX N, where EXXX identifies the element type, and N identifies the solution case. If no input cards follow the @XQT GSF command (and any RESET cards), stresses are computed for all elements in the structure. If it is desired to limit stress calculation to designated types of elements, one or more input cards in the following form must appear.

ELTS EXX_1 EXX_2 ------,

where EXX_i is the name of an element type for which stress data is to be computed.

RESET Control.

Name	Default Value	Meaning
ELIB	1	Library containing element information packets, etc (from ELD, E, EKS).
QLIB	1	Library containing solution cases for which stress data is to be computed and into which stresses are to be stored.
L1	None	Same as in DSOL.
L2	None	Same as in DSOL.
LREC	5600	Block length of output data sets.

Name	Default Value	Meaning
TIME	0	Nonzero value causes printout of intermediate CP and wall clock time.

Core Requirements. Working core requirements will not significantly exceed (1) seven times the number of joints plus (2) the block length of the output data (default = 5600), plus (3) a buffer area for handling the element information packets. 5000-10000 is usually a suitable size for the buffer.

Code Release Data. Level 7 (Univac, CDC), July 1974, routines PBSR3, PMSR4, PBSR4 coded by C. L. Yen, all others coded by W. D. Whetstone.

3.16 PSF

Function. PSF prints stress data generated by GSF. Input normally consists of the data sets generated by Q, GSF, and ELD. If no input cards (other than RESET controls, if any) are given PSF will print stress data for all elements. If it is desired to limit stress print to designated element types, one or more input cards in the following form must appear.

ELTS $\quad EXX_1 \quad EXX_2 \quad$ --,

where EXX_i is the name of an element type for which stress data is to be printed.

RESET Controls.

Name	Default Value	Meaning
QLIB	1	Library containing output from Q and DSOL.
L1	None	Same as DSOL and GSF.
L2	None	Same as DSOL and GSF.
LINES	56	Lines per printed page.
IEA	1	Zero value will cause an abort if input errors are detected.
FORMAT	1	Print format controller. A value of 2 results in short-form printout for certain t of elements.

Core Requirements. Working core space does not substantially exceed the block length of GSF output.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

3.17 EIG

Function. EIG solves linear vibration and bifurcation buckling eigenproblems of the types indicated by eqs. (1) and (2).

$$rMX - KX = 0 \qquad (1)$$

$$rKgX + KX = 0 \qquad (2)$$

K and Kg must be in the SPAR standard sparse matrix format. M may be in either SPAR sparse matrix or diagonal format. Although K must be nonsingular, it need not be positive definite.

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:

$$Z = q_1 Y_1 + q_2 Y_2 + \ldots + q_m Y_m$$

$$= HQ, \text{ where}$$

$$H = (Y_1\ Y_2\ ----\ Y_m),\ \text{and}$$

$$Q = (q_1\ q_2\ ----\ q_m)^*. \tag{3}$$

The Rayleigh-Ritz procedure consists of replacing X with Z in eq. (1); that is, substituting (3) into (1) and pre-multiplying by H^*,

$$r\ (H^*MH)Q + (H^*KH)Q = 0 \tag{4}$$

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,

$$\begin{aligned} Z_1 &= HQ_1 \\ Z_2 &= HQ_2 \\ - &\quad - \\ - &\quad - \\ Z_m &= HQ_m \end{aligned} \tag{5}$$

The Stodola (matrix iteration) step is as follows. From each Z of eq. (5) a new Y is computed, subject to the requirement that

$$MZ = KY. \tag{6}$$

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

$$b_1 = 2.0 a_i,$$
$$b_7 = .1 a_7, \text{ and}$$
$$b_{25} = .001 a_{25}.$$

Since K need not be positive definite, EIG can be used to solve "shifted" 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 LCM 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.

If a vibrational eigenproblem is solved, output from EIG consists of two data sets: (1) VIBR U, the final approximations (whether converged or not) of all system eigenvectors being iterated on, and (2) VIBR EVAL, the associated eigenvalue approximations (<u>not</u> in HZ). If a buckling eigenproblem is solved, the output is (1) vectors in STAB U, and eigenvalues in STAB EVAL. Eigenvectors are normalized to X*MX = 1 or $X^*K_G X = 1$.

RESET Controls.

Name	Default Value	Meaning
PROB	VIBR	VIBR indicates solution of eq. (1). STAB indicates solution of eq. (2).
NDYN	5	Maximum number of passes through the EIG iterative process.
CONV	.0001	See below.
LOWER	1	See below.
UPPER	.8 m	For iteration j, the measure of convergence of eigenvalue i is e_i = $\left\| (r_i^j - r_i^{j-1}) / r_i^j \right\|$. EIG terminates when e_i is less than CONV for eigenvalues LOWER through UPPER. The default value of UPPER is 80% of the number of functions being used in the iterative process.
K	K	First word of the name of the data set to be used as K in eq. (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	M	First word of the name of the data set to be used as M in eq. (1). A common RESET is M = M+RM.
MTYP	SPAR	Second word of the data set to be used as M in eq. (1). If element masses are to be ignored, RESET M = RMASS, MTYP = DIAG.
CON	1	Constraint case applied in factoring of K, or the K surrogate, by INV.

RESET Controls.

Name	Default Value	Meaning
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.
INLIB	1	Library containing an existing data set (named either VIBR U or STAB U) from which vectors L1 through L2 (see below) are to be extracted for use in the iterative procedure.
L1	-	See below.
L2	-	See below.
INIT	0	In addition to, or instead of, initial vectors obtained from data sets in INLIB, EIG generates INIT vectors containing random numbers. That is, the total number of functions being iterated is generally m = INIT + L2 - L1 + 1. If INIT = 0, and L1 and L2 are not specified via RESET, EIG will use as initial vectors all vectors from VIBR U (or STAB U, if PROB = STABILITY), in INLIB. Note that it is allowed to have OUTLIB = INLIB, in which case results from previous executions are not overwritten; they are only disabled by the creation of a new data set of the same name (e.g. VIBR U), and can be re-enabled via DCU.
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 STAB EVAL.

RESET Controls.

Name	Default Value	Meaning
MAXIT	1	Same as in processor DSOL.
ROUNDOFF	.0001	Same as in processor DSOL. If RESET, ROUNDOFF and CONV should be equal.
REPRT	0	Non-zero value results in printout of roundoff error measures during linear equation solutions.
HIST	4	Non-zero values, 4 or 8, result in printout of eigenvalue convergence history, displayed as HIST columns per page.
TIME	0	Non-zero values result in printout of intermediate CP and wall clock times.
RRCH	0	Non-zero values result in execution of a check of the accuracy of each Rayleigh Ritz analysis.

Following any RESET cards, two additional types of commands may appeai

(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 result

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 guidelines of fairly general applicability 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 thar 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 too parallel requirements, the second of which will dominate in almost all cases:

(1) $3\ m^2 + 16n$, and

(2) $2\ m\ + L + B$,

where B is a pool area usable only in integral multiples of 12J. The IO activity of EIG will diminish linearly as B is increased in multiples of 12 J.

Code Release Data. Level 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

Section 4

DCU, THE DATA COMPLEX UTILITY PROGRAM

Function - The term "data complex" refers collectively to all SPAR direct access libraries involved in an analysis. 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 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, i_1, i_2, j_1, j_2, v_1, v_2. The identified data set(s) ar printed. Not all SPAR-generated data sets can be printed via this command. Only data such as tables, stresses, eigenvectors, displacement vectors, etc., in regular "table" form should be printed in this way. Data such as topological maps (from TOPO), sparse matrices (from K, M, KG, INV, LCM, etc.), and element information packets (from E, EKS) should not be named on a PRINT command.

The i_1,i_2, - - - items, if given, must be the 7th, 8th, - - - items on the card. For eigenvector printout these parameters indicate component i_1 through i_2, joints j_1 through j_2, vector (blocks) v_1 through v_2. These iter have similar meaning for other multi-block data sets.

- COPY Lib_1, Lib_2, Id$. Copy the indicated data sets from Lib_1 to Lib_2. Disabled data sets are not copied.

- XCOPY Lib_1, Lib_2, Id\$. The indicated data (in Lib_1) is written as an ordinary sequential file, Lib_2, comprised of physical records identical to individual blocks of the data set as it resides in Lib_1. As an example, XCOPY 1, 5, VIBR U\$ causes eigenvectors to be written onto a file known externally as SPAR-E (UNIVAC) or SPARLE (CDC), corresponding to Lib_2=5. The output file will contain one physical record for each eigenvector.

- SCALE C_1, C_2\$. This command may precede a COPY or XCOPY command. If it does, the output data sets will = C_1 (input) + 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 imbedded 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 7 (Univac, CDC), July 1974, coded by W. D. Whetstone.

Section 5

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, December 1970.

3. Yen, C. L., Moore, R. A., Whetstone, W. D., "SPAR Demonstration Problems," LMSC D403169, June 1974.

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.

TAB Sub-processors

TEXT:	4/8 - - - 76 characters of text.
MATC:	k, E, Nu, Rho, Alpha, Stress.
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, 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)

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_t, a_c, s_t, s_c, a_{max}, a_{min}.

BD: k, A, I_1, α_1, h, c, q_1

SA: k, t, t_{11}, t_{22}, t_{12}. (Format=1, isotropic)

k, t_{11}, t_{22}, t_{12}. (Format=2, aeolotropic)

$c_{11}, c_{12}, c_{22}, c_{31}, c_{32}, c_{33}$. (membrane stiffness)

$d_{11}, d_{12}, d_{22}, d_{31}, d_{32}, d_{33}$. (bending stiffness)

SB: k, t, s, t_{max}, t_{min}.

CON n: (constraint case n)

ZERO $m_1, m_2,$ - -: $j_1^{1}, j_2^{1},$ inc^{1}: $j_1^{2}, j_2^{2},$ 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 'fans').

4-node elements:

j_1, j_2, j_3, j_4

j_1, j_2, j_3, j_4, 1, ni, nk, kinc. (nk rectangular meshes)

j_1, j_2, j_3, j_4, 2, ni, nj. (cylinder, nj bays)

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
QLIB	1	Library containing the prestress solution.
QSEQ	1	Load casc required number identifying the prestress solution.

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).

LCM

Forms R=C1 * Q1 + C2 * Q2

Reset Name	Default	Meaning
R	K+KG	Name of output matrix.
Q1	K	Name of Q1 data set.
Q2	KG	Name of Q2 data set.

Reset		
Name	Default	Meaning
C1	1.0	
C2	1.0	
RLIB	1	R destination library.
Q1LIB	1	Q1 source library.
Q2LIB	1	Q2 source library.

Q

Reset Name	Default	Meaning
QLIB	1	Destination library for Q output
MLIB	1	Source library, system M
M	M	System M, first word of name
MTYPE	SPAR	System M, second word of name
TMULT	1.	T = TMULT * (T Input) - TNULL
TNULL	.0	
PMULT	1.	P = PMULT * (P Input) - PNULL
PNULL	.0	

Commands:

CASE n'- - - title for case n - - -

NODAL FORCES: - - data - -

NODAL MOTIONS: - - data - -

NODAL TEMPERATURES: - - data - -

NODAL PRESSURES: - - data - -

INERTIA FORCE direction, acceleration (not g's)

Nodal force, motion data cards:

jo, k, q, ni, iinc, dqi, nj, jinc, dqj

Nodal temperature, pressure data cards:

jo, q, ni, iinc, dqi, nj, jinc, dqj

DSOL

Reset Name	Default	Meaning
QLIB	1	Source library for data from Q, destination library for output.

Reset Name	Default	Meaning
CON	1	Constraint case
L1		Process load cases L1 through L2
L2		
K	K	Name of system K
KLIB	1	Library containing system K
ROUNDOFF	10^{-4}	Roundoff error tolerance
MAXIT	1	Maximum iterative accuracy improvement passes
EP	1	Nonzero: print error measures
UP	0	Nonzero: print displacements
FP	0	Nonzero: print reactions, nodal balance

LCD

Cases n, m' - - title for cases n through m - -

Lib_1, seq_1, increment_1

Lib_2, seq_2, increment_2

-

-

Lib_f, seq_f, increment_f

GSF

Reset controls QLIB, L1, L2 same as in DSOL

Optional data cards:

ELTS E23, E42, etc.

PSF

Same as for GSF. Also reset LINES = lines of printout/page.

EIG

Reset Name	Default	Meaning
PROB	VIBR	VIBR or STAB
CON	1	Constraint case
K	K	
KG	KG	
M	M	
MTYPE	SPAR	
NDYN	5	Max eigenvector iterations
CONV	10^{-4}	Eigenvalue convergence test
LOWER	1	Terminate when eigenvalues LOWER through UPPER have converged
UPPER	.8 x no. fcts.	
L1		Select vectors L1 through L2 from previous solution for further iteration
L2		
INIT	0	Generate INIT random vectors
SHIFT	.0	Add SHIFT to computed eigenvalues
REPRT	0	Nonzero: Print linear equation solution error measures
MAXIT	1	Same as DSOL
ROUNDOFF	10^{-4}	Same as DSOL

Data Cards:

PRINT	k_1, k_2, k_3, k_4, k_5. (k_5 = final eigenvectors)
RSCALE	s_1, s_2, s_3, s_4, s_5, s_6

Random vector element u_j^i (direction j) is multiplied by s_j

DCU

TITLE	Lib Title - -
STATUS	Lib
TOC	Lib, Id
DISABLE	Lib, Id
ENABLE	Lib, Id
COPY	Lib_1, Lib_2, Id
XCOPY	Lib_1, Lib_2, Id
SCALE	C_1, C_2 Output = C_1 x (input) + C_2
CHANGE	Lib, Id_{old}, Id_{new}
DUPLICATE	Lib_1, Lib_2
NTAPE	nt
TWRITE	Lib
TREAD	Lib
LIBLIB	nl
STORE	Lib, Id
RETRIEVE	Lib, Id
PRINT	Lib, Id, i_1, i_2, j_1, j_2, v_1, v_2
ABORT	n

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

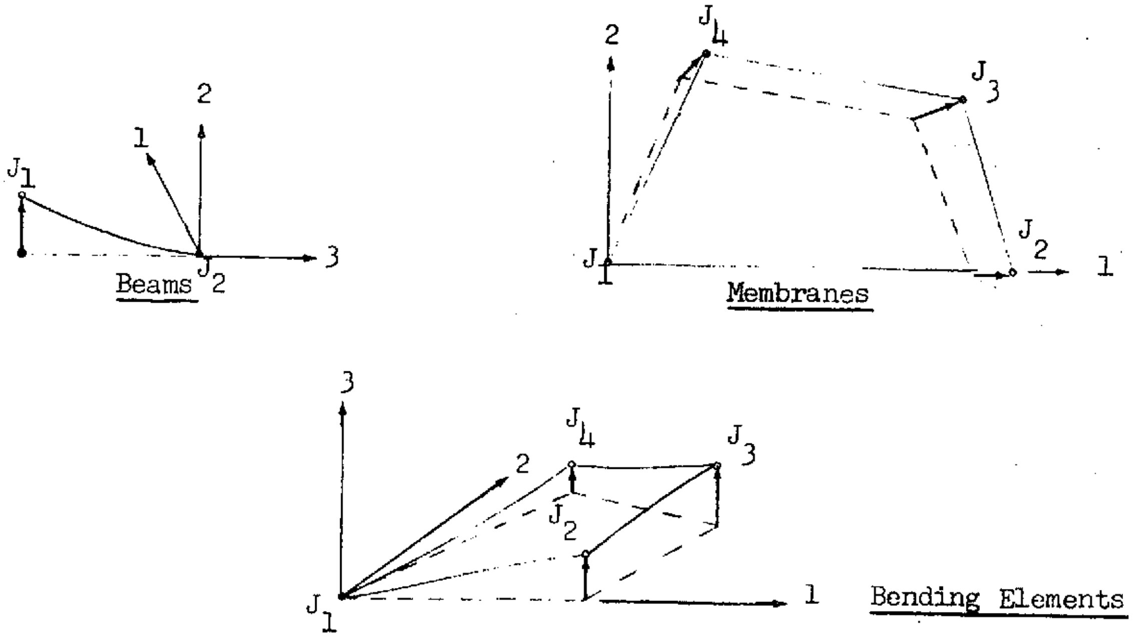
Element Type		K	M	Kg
E21	Gen.beam	K21	M62	BEAMKG
E22	Direct K	(see text)	none	BARKG
E23	Bar	(see text)	M32	BARKG
E24	Pl. beam	(see text)	M62	BEAMKG
E25	Direct K, L=0	(see text)	none	none
E31	Membrane	TM	M33	GTM
E32	Bending	TPB7	M63	none
E33	Mem.+bend.	TM+TPB7	M63	GTP
E41	Membrane	QMB5	M34	GQM
E42	Bending	QPB11	M64	none
E43	Mem.+bend.	QMB5+QPB11	M64	GQP
E44	Shear Panel	QMB1	M34	none

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., can 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,

$$\begin{aligned}
&\text{Beams:} && U = \left[u^1_1\ u^1_2\ u^1_3\ \theta^1_1\ \theta^1_2\ \theta^1_3\right]^t \\
&\text{3-node membranes:} && U = \left[u^2_1\ u^3_1\ u^3_2\right]^t \\
&\text{4-node membranes:} && U = \left[u^2_1\ u^3_1\ u^3_2\ u^4_1\ u^4_2\right]^t \\
&\text{3-node bending:} && U = \left[u^2_3\ \theta^2_1\ \theta^2_2\ u^3_3\ \theta^3_1\ \theta^3_2\right]^t \\
&\text{4-node bending:} && U = \left[u^2_3\ \theta^2_1\ \theta^2_2\ u^3_3\ \theta^3_1\ \theta^3_2\ u^4_3\ \theta^4_1\ \theta^4_2\right]^t \qquad (1\text{-}1)
\end{aligned}$$

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 matrice 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

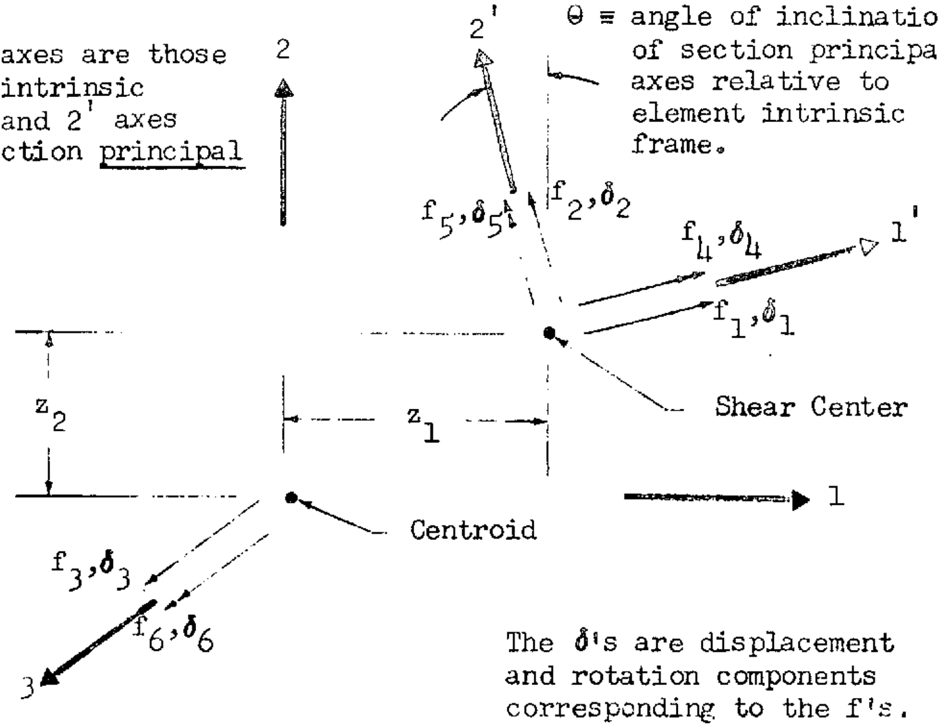
$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} s_{11} & 0 & 0 & 0 & s_{15} & 0 \\ & s_{22} & 0 & s_{24} & 0 & 0 \\ & & s_{33} & 0 & 0 & 0 \\ & & & s_{44} & 0 & 0 \\ & & & & s_{55} & 0 \\ \text{symmetric} & & & & & s_{66} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} , \qquad (1\text{-}3)$$

or,

$$f = s\,\boldsymbol{\delta} \quad , \qquad (1\text{-}4)$$

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.

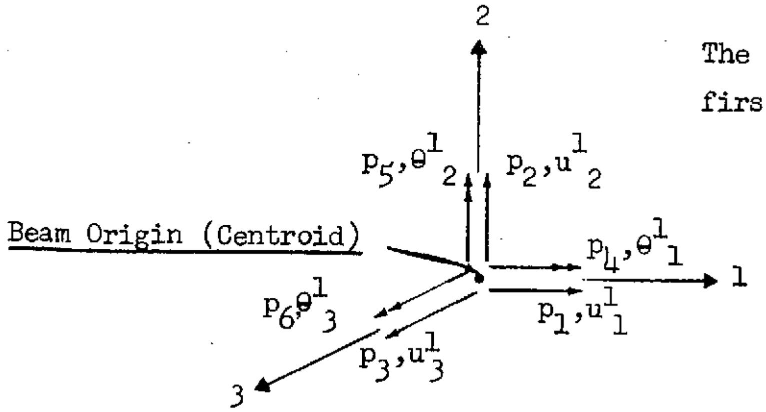
θ ≡ angle of inclination of section principal axes relative to element intrinsic frame.

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 θ, s = sin θ.

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -z_2 & z_1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 & 0 & 0 \\ s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -s & 0 \\ 0 & 0 & 0 & s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \qquad (1\text{-}5)$$

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

$$s_{66} = \frac{CG}{L}\left[1 - \frac{\tanh\left(\frac{L}{2}\right)\sqrt{\frac{C}{C_1}}}{\frac{L}{2}\sqrt{\frac{C}{C_1}}}\right],$$

which assumes no warping of end cross sections.

Under the usual assumptions of Timoshenko beam theory,

$$\delta_1 = f_1\,\frac{L^3}{3\,EI_2} - f_5\,\frac{L^2}{2\,EI_2} + f_1\,\frac{\alpha_1 L}{GA}$$

$$\delta_5 = -f_1\,\frac{L^2}{2\,EI_2} + f_5\,\frac{L}{EI_2};$$

or where $e_1 \equiv \frac{EI_2}{GA} \alpha_1$ and $k_1 \equiv \frac{EI_2}{e_1 + L^2/12}$,

$$\begin{bmatrix} \delta_1 \\ \\ \delta_5 \end{bmatrix} = \frac{L^2}{EI_2} \begin{bmatrix} \left(\frac{L}{3} + \frac{e_1}{L}\right) & -\frac{1}{2} \\ \\ -\frac{1}{2} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} f_1 \\ \\ f_5 \end{bmatrix}$$

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 and 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 volume

$$V = \frac{1}{2}\int_{Vol} \sigma^{t}\,N\,\sigma\,d(Vol)$$

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,

$$\Pi_c = \frac{1}{2}\,B^{t}\,H\,B - B\,T\,q,$$

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 $i = 1,2,\ldots,$

$$H\,B = T\,q,$$
$$B = H^{-1}\,T\,q, \qquad (1.2\text{-}8)$$

and, from equation (1.2-4),

$$V = \tfrac{1}{2} B^t T^t H^{-1} T B,$$

from which it is concluded that

$$K = T^t H^{-1} T. \qquad (1.2\text{-}9)$$

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.

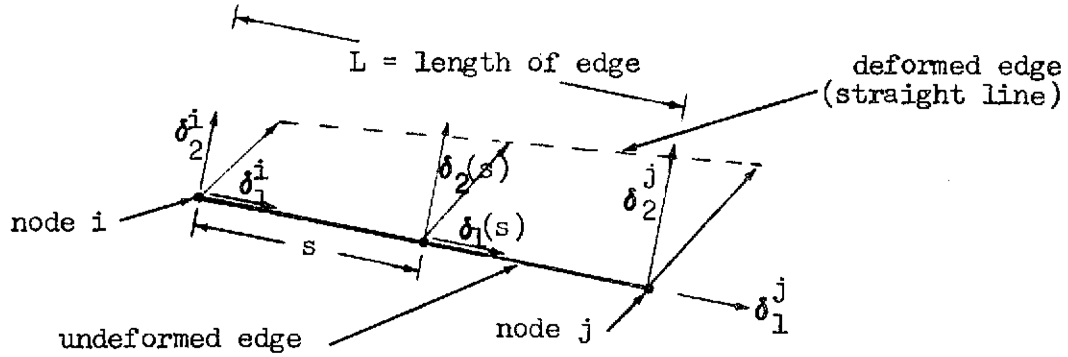
To summarize, the following are the two basic assumptions involved in application of the assumed stress field - minimum complementary energy method.

(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).

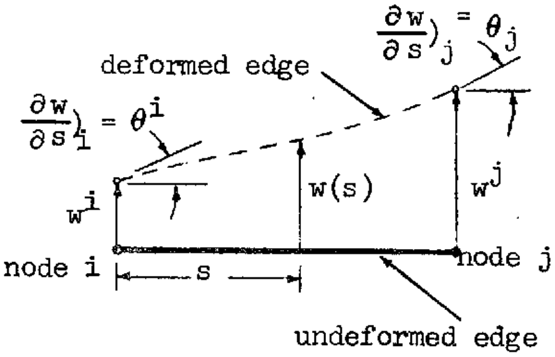
To implement the above assumptions, the principal computational tasks are the calculation of H from equation (1.2-5), and T from equation (1.2-7). From H and T the element stress and stiffness matrices, $H^{-1}T$ and $T^t H^{-1} T$ are readily determined.

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.



$$\delta_k = \delta_k^i + \frac{s}{L}(\delta_k^j - \delta_k^i), \text{ for } k = 1, 2$$

FIGURE B1.2-1 IN-PLANE EDGE MOTION ASSUMED IN TM, QMB5, QMB1


$\frac{\partial w}{\partial s})_j = \theta_j$
deformed edge
$\frac{\partial w}{\partial s})_i = \theta^i$
w^i
w(s)
w^j
node i
s
node j
undeformed edge

w is normal to the element surface.

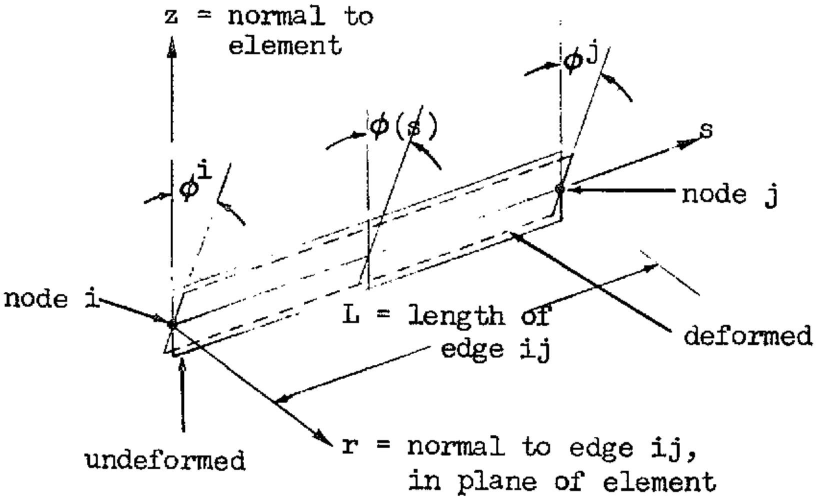
w = a cubic function of s such that:

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

$$\alpha_1 = -(2D_{12} + 4D_{33})/D_{11},$$

$$\alpha_2 = 2D_{23}/D_{11},$$

$$\alpha_3 = 2D_{13}/D_{11},$$

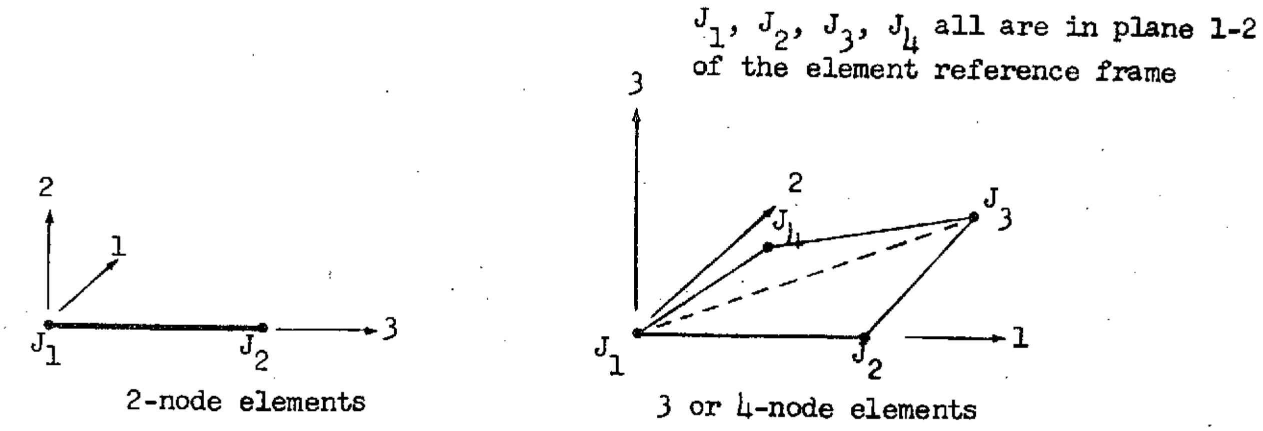
$$\alpha_4 = -D_{22}/D_{11}.$$

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 $u^i = \lfloor u^i_1 \;\; u^i_2 \;\; u^i_3 \rfloor$,

for bars (M32),

$$q = \lfloor u^1 \;\; u^2 \rfloor^t .$$

For triangular membranes (M33),

$$q = \lfloor u^1 \;\; u^2 \;\; u^3 \rfloor^t .$$

For beams (M62),

$$q = \lfloor u^1_1 \;\; u^1_2 \;\; u^1_3 \;\; \theta^1_1 \;\; \theta^1_2 \;\; \theta^1_3 \;\; u^2_1 \;\; u^2_2 \;\; u^2_3 \;\; \theta^2_1 \;\; \theta^2_2 \;\; \theta^2_3 \rfloor^t .$$

Where $\psi^i = \lfloor u^i_3 \;\; \theta^i_1 \;\; \theta^i_2 \rfloor$,

for triangular bending elements (MTP),

$$q = \lfloor \psi^1 \;\; \psi^2 \;\; \psi^3 \rfloor^t .$$

For quadrilateral bending elements (MQP),

$$q = \left[\psi^1 \quad \psi^2 \quad \psi^3 \quad \psi^4\right]^t.$$

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, $a_1, a_2, \ldots a_n$, 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

$$T = \frac{1}{2} \int_{Vol} m\,(\dot{u}^2 + \dot{v}^2 + \dot{w}^2)\,d(Vol). \qquad (2\text{-}2)$$

Where $A = \left[a_1 \quad a_2 \ldots a_n\right]^t$, substitution of (2-1) into (2-2) gives

$$T = \frac{1}{2}\,\dot{A}^t \left[\int_{Vol} m \quad G^t\,G\,d(Vol)\right] \dot{A}. \qquad (2\text{-}3)$$

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),

$$\begin{aligned} u &= a_1 + a_2 z, \\ v &= a_3 + a_4 z, \\ w &= a_5 + a_6 z. \end{aligned} \qquad (2\text{-}6)$$

For triangular membranes (M33),

$$\begin{aligned} u &= a_1 + a_2 x + a_3 y, \\ v &= a_4 + a_5 x + a_6 y, \\ w &= a_7 + a_8 x + a_9 y. \end{aligned} \qquad (2\text{-}7)$$

For beams (M62), where u, v, and w are displacements of the central axis, and where ϕ is the total twist angle,

$$\begin{aligned} u &= a_1 + a_2 z + a_3 z^2 + a_4 z^3, \\ v &= a_5 + a_6 z + a_7 z^2 + a_8 z^3, \\ w &= a_9 + a_{10} z, \\ \phi &= a_{11} + a_{12} z. \end{aligned} \qquad (2\text{-}8)$$

For triangular bending elements (MTP)

$$\begin{aligned} w = {} & a_1 + a_2 x + a_3 x^2 \\ & + a_4 y + a_5 xy + a_6 x^2 y \\ & + a_7 y^2 + a_8 xy^2 + a_9 x^2 y^2. \end{aligned} \qquad (2\text{-}9)$$

For quadrilateral bending elements (MQP)

$$\begin{aligned} w = {} & a_1 + a_2 x + a_3 x^2 + a_4 x^3 \\ & + a_5 y + a_6 xy + a_7 x^2 y + a_8 x^3 y \\ & + a_9 y^2 + a_{10} xy^2 \\ & + a_{11} y^3 + a_{12} xy^3. \end{aligned} \qquad (2\text{-}10)$$

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.) used will be the same as was used in the discussion of element mass matrices.

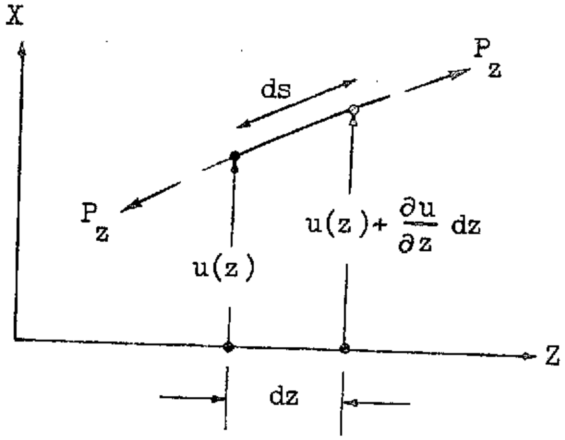
For 2-node elements, the principal contributions to K_g are due to changes in axial strain due to lateral deformation, u and v,

$$\epsilon_z = \frac{1}{2}\left(\frac{\partial u}{\partial z}\right)^2 + \frac{1}{2}\left(\frac{\partial v}{\partial z}\right)^2 .$$

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.



$$\epsilon_z = \frac{ds-dz}{dz} = \left[1 + \left(\frac{\partial u}{\partial z}\right)^2 \right]^{\frac{1}{2}} = 1 + \tfrac{1}{2}\left(\frac{\partial u}{\partial z}\right)^2 + \ldots$$

FIGURE B3-1 K_g EFFECTS IN A PLANE BEAM

Similarly, in two-dimensional elements the changes in membrane strains due to lateral displacement produce the following change in strain energy.

$$U_1 = \frac{1}{2} \int_{Area} \left[N_x \left(\frac{\partial w}{\partial x}\right)^2 + N_y \left(\frac{\partial w}{\partial y}\right)^2 + 2N_{xy} \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \right] d\,(\text{Area}) \qquad (3\text{-}2)$$

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 similar to those used in calculating element mass matrices. Lateral displacement functions are assumed in terms of n undetermined coefficients, $a_1, a_2, \ldots a_n$. Except for the quadrilateral membrane, GQM, each of the K_g formulations assumes the same displacement field as a corresponding mass matrix, as tabulated below.

K_g Type	Assumed Displacements
BARKG	Eq (2-6), axial element.
BEAMKG	Eq (2-8), general beam.
GTM	Eq (2-7), triangular membrane.
GTP	Eq (2-9), triangular bending + membrane.
GQP	Eq (2-10), quadrilateral bending + membrane.

For the quadrilateral membrane, GQM, lateral displacement is assumed in the following form.

$$w = a_1 + a_2 x + a_3 y + a_4 xy.$$

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.