

FOR ERRATA

AD 400280

THE FOLLOWING PAGES ARE CHANGES

TO BASIC DOCUMENT

400280

400280

ERRATA SHEET NO. 2
for
LINCOLN MANUAL NO. 48
STAIR
(STRUCTURAL ANALYSIS INTERPRETIVE ROUTINE)
INSTRUCTION MANUAL

Experience with the STAIR program over the past year has disclosed a number of errors in the printed manual. Rather than list the errors on a sheet and rely on each holder of the manual to make the appropriate corrections, corrected pages for insertion into the manual are being provided for all copies as follows.

- Page iii: Add - Error Halts in the BSS Loader 66a
- Page 10: Program description changed to account for the Y reaction component at joint 0305.
- Page 15: Y coordinates at joints 0102, 0103, 0105, 0106, 0202, 0203, 0302, 0303, 0305, 0306, changed to agree with card listing on Page 25.
- Page 36: Heading of a column in the table changed.
- Page 57: Z coordinates of joints 101, 102, 103 added.
- Page 66a: New page of Error Halts in the BSS Loader.
- Page 70: Program description changed and a duplicated footnote removed.
- Page 79: Changes in variable names and block names.
- Page 80: Size restriction relocated from Page 79.
- Page 82: Change in the incrementing of a "do" loop.
- Page 84: Change in the calculation of the variable NRAS.

If any other errors are found, please notify Mr. William R. Fanning, Room D-250, Ext. 7592, Lincoln Laboratory, M.I.T., P.O. Box 73, Lexington, Massachusetts 02173.

4 August 1964

Publications Office
M.I.T. Lincoln Laboratory
P.O. Box 73
Lexington, Massachusetts 02173

**Best
Available
Copy**

TABLE OF CONTENTS

Acknowledgments	ii
Chapter 1 – INTRODUCTION	1
Chapter 2 – USER'S MANUAL	3
A. Introduction	3
B. The Stiffness Matrix Method	7
C. Input Data	10
D. Input Data for the STAIR Load Program (SLOP)	11
E. The STAIR Operations	30
F. Input Data for the STAIR System	35
G. STAIR System Error Stops	41
Chapter 3 – OPERATOR'S MANUAL	65
A. Introduction	65
B. Input-Output	65
C. Machine Operation	65
D. Sense Switches and Indicator Lights	65
E. Program Stops	66
F. Interruption of a Run	66
G. Error Halts in the BSS Loader	66a
Chapter 4 – PROGRAMMER'S MANUAL	67
A. Introduction	67
B. Definition of Matrix Terms	67
C. Core Storage Allocation	68
D. Definition of Common Variables	69
E. Tape Storage Formats	69
F. STAIR Operating Routines	70
STRIX	70
EFFRIX	78
MATRO	95
MADD	105
MATOUT	112
MATIN	115
DEFSOL	117
BARSOL	137
CHECK	142
G. STAIR Service Routines	145
Master Routine	145
FIXTPE	147
MAVERT	149
MISLAM for IBM-7090	149
H. STAIR Load Programs	157
SLOP 1	157
SLOP 2	162
Chapter 5 – LIMITATIONS AND EXTENSIONS	171

The new matrix will represent the same structure as the original matrix. This operation, which amounts to the partitioning of the original stiffness matrix, is equivalent to the elimination process used to reduce the number of unknowns in a set of simultaneous equations such as Eq. (2), and is referred to as "Elimination." The new matrix is called an "Effective Matrix."

To illustrate the elimination process consider the force-displacement equation for unit 01 in Fig. 2:

$$\begin{bmatrix} F_{0101} \\ F_{0102} \\ F_{0103} \\ F_{0104} \\ F_{0105} \\ F_{0106} \\ F_{0304} \\ F_{0305} \\ F_{0306} \end{bmatrix} = \begin{bmatrix} K \end{bmatrix} \begin{bmatrix} U_{0101} \\ U_{0102} \\ U_{0103} \\ U_{0104} \\ U_{0105} \\ U_{0106} \\ U_{0304} \\ U_{0305} \\ U_{0306} \end{bmatrix} \quad (4)$$

where F and U represent three components of joint force and joint displacement, respectively. Displacements U_{0101} , U_{0102} and U_{0103} are now eliminated from the equations (these are at the joints not common to other units) and Eq. (4) is reduced to

$$\begin{bmatrix} F'_{0104} \\ F'_{0105} \\ F'_{0106} \\ F'_{0304} \\ F'_{0305} \\ F'_{0306} \end{bmatrix} = \begin{bmatrix} K^e \end{bmatrix} \begin{bmatrix} U_{0104} \\ U_{0105} \\ U_{0106} \\ U_{0304} \\ U_{0305} \\ U_{0306} \end{bmatrix} \quad (5)$$

The original matrix (K) of size 27×27 is thus reduced to the effective matrix (K^e) of size 18×18 . During this process, the force vector F is also modified to F'. The details of this modification are described in Chapter 4.

A similar effective matrix can be formed for unit 2 and superimposed on that for unit 1. The resulting matrix represents the structure shown in Fig. 5. At this point, the fact that displacements y_{0106} and z_{0106} must equal zero is taken into account by discarding the columns and rows in the matrix corresponding to these displacements. The matrix for the two-unit combination is

then 25×25 . The displacements at joints 0105 and 0104 and the remaining displacement at 0106 are now eliminated, leaving an effective 18×18 matrix.

Finally, the effective matrix for unit 3 is superimposed. This does not increase the size of the matrix since no new joints are introduced. Joints 0204, 0205 and 0206 are now eliminated leaving a 9×9 matrix. Displacements x_{0306} and y_{0305} are all zero and the corresponding terms in the matrix may be discarded. This leaves only a 5×5 matrix, which is finally solved for the displacements at joint 0304 and the remaining displacements at joint 0305.

By the simple process of substitution, all other displacements may be determined. Having displacements, all bar stresses and reactions are easily obtained.

In comparison to the method described above, a direct solution without subdivision of the structure would have required the solution of 48 simultaneous equations. The great advantage of the method is now obvious when applied to large structures having many joints.

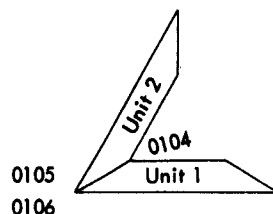


Fig. 5.

C. INPUT DATA

It is desirable to prepare the input data on format input data sheets before punching them onto cards. The restrictions of the STAIR ANALYSIS PROGRAM and the items of data required for each step of a problem are listed and discussed in the following sections.

Two decks of cards have to be prepared for the STAIR System: a structural data deck and a load data deck. These decks are handled differently during the running of the problem.

The structural data deck is input directly to the STAIR System. This deck begins with a block of cards punched from an "Initial Data Sheet." Subsequent blocks of cards correspond to consecutive steps in the ANALYSIS PROGRAM; the first card of each block calls the routine which executes the step, and the following cards provide the data needed for the execution.

The load data deck is input to the STAIR Load Program (SLOP). This program, which must be run before the ANALYSIS PROGRAM, receives and/or computes concentrated joint loads for various types of loading on the structure. The loads are assembled into groups of loads applied to individual units of the structure. SLOP is stored on tape for use by the STAIR System.

Each item of data entered on the format sheets is either an integer number (such as a quantity of joints or bars) or a fixed number (e.g., a joint coordinate or a bar area). The format sheets prescribe a field for each item of data. A field is the maximum number of characters an item may contain.

Integer numbers are always entered at the right of their field and are never written with a decimal point. Fixed numbers may be entered in one of two ways: with or without a decimal point. If the decimal point is specified as a character in the item of data, the number may be written anywhere within its field. If the decimal point is omitted, the number must be positioned about the dashed line in its field, with the fractional part to the right of the dashed line. The location of dashed lines is written into the machine program as indicated on the data sheets contained herein. If there is no dashed line in the field of a fixed number, it may be assumed at the right of the field.

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA I

PROBLEM 200

PAGE 2/11

LIST JOINT COORDINATES PUNCH ONE LINE PER CARD																																								(14,3F12.6)																		
JT. NO.				X (FT.)												Y (FT.)												Z (FT.)																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40																			
0	1	0	1							0	0											0	0										0																									
0	1	0	2							0	0									-	3	4	6	4	1	0	2						2	0																								
0	1	0	3							0	0									-	3	4	6	4	1	0	2						-	2	0																							
0	1	0	4							-	6	0									0	0											0	0																								
0	1	0	5							-	1	2	0								-	3	4	6	4	1	0	2					-	2	0																							
0	1	0	6							-	1	2	0								-	3	4	6	4	1	0	2					2	0																								
0	2	0	1							-	3	0										5	1	9	6	1	5	2					0	0																								
0	2	0	2							-	6	0										6	9	2	8	2	0	3					2	0																								
0	2	0	3							-	6	0										6	9	2	8	2	0	3					-	2	0																							
0	2	0	4							0	0											1	0	3	9	2	3	0	5					0	0																							
0	2	0	5							0	0											1	7	3	2	0	5	0	8					-	2	0																						
0	2	0	6							0	0											1	7	3	2	0	5	0	8					2	0																							
0	3	0	1							3	0												5	1	9	6	1	5	2					0	0																							
0	3	0	2							6	0												6	9	2	8	2	0	3					2	0																							
0	3	0	3							6	0												6	9	2	8	2	0	3					-	2	0																						
0	3	0	4							6	0												0	0										0	0																							
0	3	0	5							1	2	0											-	3	4	6	4	1	0	2					-	2	0																					
0	3	0	6							1	2	0											-	3	4	6	4	1	0	2					2	0																						

28 LINES

28 LINES

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA II

PROBLEM 200

PAGE 3/11

LIST BAR NUMBERS OF THE STRUCTURE (215, F12.6)																					
BAR NUMBER										AREA (SQ. IN)											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	0	1	0	1		0	1	0	2						3	1	0				
	0	1	0	1		0	1	0	3						3	1	0				
	0	1	0	1		0	1	0	4						3	1	0				
	0	1	0	1		0	3	0	4						3	1	0				
	0	1	0	2		0	1	0	3						3	1	0				
	0	1	0	2		0	1	0	4						3	1	0				
	0	1	0	2		0	1	0	5						3	1	0				
	0	1	0	2		0	1	0	6						3	1	0				
	0	1	0	2		0	3	0	4						3	1	0				
	0	1	0	2		0	3	0	6						3	1	0				
	0	1	0	3		0	1	0	4						3	1	0				
	0	1	0	3		0	1	0	5						3	1	0				
	0	1	0	3		0	3	0	4						3	1	0				
	0	1	0	3		0	3	0	5						3	1	0				
	0	1	0	3		0	3	0	6						3	1	0				
	0	1	0	4		0	1	0	5						3	1	0				
	0	1	0	4		0	1	0	6						3	1	0				
	0	1	0	4		0	2	0	1						3	1	0				
	0	1	0	4		0	2	0	2						3	1	0				
	0	1	0	4		0	2	0	3						3	1	0				
	0	1	0	5		0	1	0	6						3	1	0				
	0	1	0	5		0	2	0	3						3	1	0				
	0	1	0	6		0	2	0	2						3	1	0				
	0	1	0	6		0	2	0	3						3	1	0				
	0	2	0	1		0	2	0	2						3	1	0				
	0	2	0	1		0	2	0	3						3	1	0				
	0	2	0	1		0	2	0	4						3	1	0				
	0	2	0	2		0	2	0	3						3	1	0				
	0	2	0	2		0	2	0	4						3	1	0				
	0	2	0	2		0	2	0	5						3	1	0				
	0	2	0	2		0	2	0	6						3	1	0				
	0	2	0	3		0	2	0	4						3	1	0				
	0	2	0	3		0	2	0	5						3	1	0				
	0	2	0	4		0	2	0	5						3	1	0				
	0	2	0	4		0	2	0	6						3	1	0				
	0	2	0	4		0	3	0	1						3	1	0				
	0	2	0	4		0	3	0	2						3	1	0				
	0	2	0	4		0	3	0	3						3	1	0				
	0	2	0	5		0	2	0	6						3	1	0				
	0	2	0	5		0	3	0	3						3	1	0				

PUNCH ONE LINE
PER CARD

40 LINES

F. INPUT DATA FOR THE STAIR SYSTEM

The input parameters required by each step of the STAIR operations are described below. The restrictions on the magnitude of these parameters and operations are given where applicable. The formats in which these data parameters are prepared will be illustrated in the following data sheets. Data sheets for each step are arranged in the same order as the example for the ANALYSIS PROGRAM. The same example structure of Fig. 2 is used.

Initial Input (refer to data sheet, p. 44)

- (a) Problem identification number (IDP).
- (b) Number of units into which the given structure is divided (NUT).
- (c) Number of different loading conditions to which this given structure is subjected (NK).
- (d) Number of reaction joints or supports in structure.
- (e) Elastic modulus of bar material E kips per sq. in.
- (f) List of reaction joint numbers and the coordinate directions in which displacement is restricted (1-immovable; 0-free).

Size Restrictions

- (a) Number of units ≤ 99 .
- (b) Number of reaction joints ≤ 125 .
- (c) Loading conditions ≤ 28 .

STRIX (refer to data sheet, p. 45)

- (a) Step number in ANALYSIS PROGRAM. It precedes every operation code, as the 001 on p. 45 precedes 01 STRIX, and the 005 precedes 04 MADD on p. 49. It is the first card of a card block calling the routine which executes the step as mentioned in the section on input data. (Step numbers must start at 01 and proceed sequentially to the end.)
- (b) Identification number of the unit (NUU) to which this STRIX operation applies. On p. 45 NUU is 01; later, units 02 and 03 will be formed by MATRO. In case some of the units are not identical, NUU is used to identify the different STRIX operations for different units when STRIX is called more than once.
- (c) Number of joints in the unit (NJTU).
- (d) Number of bars in the unit (NMU).
- (e) Joint sequence. This serves as a reminder only, and is not a part of the input data.
- (f) Joint cards giving the joint numbers [see Operation Restriction (b)] contained in the unit in sequence* and the coordinates of the joints in feet (reproduced from SLOP).
- (g) Bar cards are reproduced from SLOP. Bars which are common to two units are listed separately for this input (see p. 46).

*The sequence can be of any convenient form, but the joints that are to be eliminated by EFFRIX must be placed first in the sequence.

Size Restrictions

If $N = 3 \times$ (number of joints in the unit), $NK =$ number of loading conditions.

(a) $\frac{Nx(N+1)}{2} + N \times NK \leq 16000$ (see table).

NJTU	NK
59	1
58	4
57	7
56	10
55	13
54	17
53	20
52	24
51	27

- (b) Number of bars ≤ 600 .
(c) Unit number $\leq$ total number of units in the structure.

Operation Restrictions

- (a) In no case may all the bars meeting at a joint lie in one plane unless the joint is a reaction joint. If a planar joint exists, a new bar out of the plane (an imaginary member) may be inserted which will carry no stress but will allow the analysis to proceed.* The new bar may connect the planar joint with any other joint not lying in the plane.
- (b) If an EFFRIX operation immediately follows STRIX in the sequence of the ANALYSIS PROGRAM, those joints which are going to be eliminated have to be placed first in the row of joint cards. In the example, joints 0101, 0102 and 0103 are eliminated by the following EFFRIX.

EFFRIX (refer to p. 47)

- (a) Step number in ANALYSIS PROGRAM.
(b) Number of joints (NE) to be eliminated from the row sequence established by the previous operation (STRIX or MADD).

Size Restrictions

NE = number of joints to be eliminated.

NJ = number of joints in original row sequence (number of joints in the unit).

NR = number of reaction joints among joints being eliminated.

NC = number of reaction components acting at the eliminated reaction joints.

* See p. 41 for a more detailed explanation of a planar joint.

01207DEFSOL

060606

013000ARSOL

0005400010

0101 0102 30

0101 0103 30

0101 0104 30

0101 0304 30

0102 0103 30

0102 0104 30

0102 0105 30

0102 0106 30

0102 0304 30

0102 0306 30

0103 0104 30

0103 0105 30

0103 0304 30

0103 0305 30

0103 0306 30

0104 0105 30

0104 0106 30

0104 0201 30

0104 0202 30

0201 0204 30

0202 0203 30

0202 0204 30

0202 0205 30

0203 0204 30

0203 0205 30

0204 0205 30

0204 0206 30

0204 0301 30

0204 0302 30

0204 0303 30

0205 0206 30

0205 0303 30

0206 0302 30

0206 0303 30

0301 0302 30

0301 0303 30

0301 0304 30

0302 0303 30

0104 0203 30

0105 0106 30

0105 0203 30

0104 0203 30

0106 0202 30

0201 0202 30

0201 0203 30

0302 0304 30

0302 0305 30

0302 0306 30

0303 0304 30

0303 0305 30

0304 0305 30

0304 0306 30

0305 0306 30

0202 0206 30

101 00 00 00

102 00 -3464102 20

103 00 -3464102 -20

104	-60	00	00
105	-120	-3464102	-20
106	-120	-3464102	20
201	-30	5196162	00
202	-60	6928203	20
203	-60	6928203	-20
204	00	10392305	00
205	00	17320508	-20
206	00	17320508	20
301	30	5196162	00
302	60	6928203	20
303	60	6928203	-20
304	60	00	00
305	120	-3464102	-20
306	120	-3464102	20
01009CHECK			
00540010			
01510STOP			

G. ERROR HALTS IN THE BSS LOADER

The STAIR system uses the old BSS loader (commonly called the 10-card loader) and not the newer BSS2 loader because of incompatible timing and binary patches that are made to the loader by the MISLAM routine at execution time. The BSS loader has known errors in it; however, these do not affect the STAIR system. Because this loader is essentially obsolete, information on its operation and error stops is difficult to obtain. For this reason, the following table (as contained in an early IBM 709 FORTRAN Operators Manual and Lincoln Memorandum No. 2C-0318) is reproduced here for the users convenience.

<u>Halt Location (Octal)</u>	<u>Reason for Halt</u>	<u>Procedure</u>
77525	Check sum error on cards.	Press START to accept information.
77755	Instructions overlap the Symbol Table of the Loader.	Terminate loading. Combination of program and transfer vectors too long. Rewrite program.
77400	Instructions and Data overlap.	Terminate loading. Combination of instructions and data too long. Rewrite program.
77731	More than 20 subroutines are missing.	If missing subroutines are immediately available, ready in the card reader, press START until a stop at 77747 occurs.
77437	EOF at the card reader. This condition does not produce a Halt. The Program Light is <u>not</u> turned on but the Read Select Light <u>is</u> turned on. The location counter will contain 77437.	Press START to read cards.
77747	Missing subroutines.	This stop indicates the TRANSFER CARD has been reached. It is caused by one of the two occurrences listed below: (1) Loading has been completed, but at least one of the subroutines called for is missing. Location 77400 contains the BCD name of the first missing subroutine, location 77401 the second, etc. If the missing subroutine(s) is immediately available, it may be loaded without starting the entire loading process over again. Place another TRANSFER CARD (9 punch in column 1) at the end of the routine(s), ready in the card reader, and press START. (2) The TRANSFER CARD encountered is really a premature one that simply has not been withdrawn. Be certain that a TRANSFER CARD is the last card at the end of the deck and press START.

D. DEFINITION OF COMMON VARIABLES

Common storage contains the following list of variables during execution of all the operating routines:

IDP	problem identification number
NSTEP	consecutive number of steps in ANALYSIS PROGRAM being executed
E	elastic modulus of bar material
NREF	number of residual and reaction matrices written on tape A6
NJREAC	number of reaction joints in structure
NUT	number of units in structure
NK	number of loading conditions
NU	number of unit whose matrix is in primary storage area
NJT	number of joints in primary row sequence
NJSEC	number of joints being eliminated from primary row sequence, or number of joints in secondary row sequence
NRJSEC	number of reaction joints among joints being eliminated
NRCSEC	number of reaction components among joints being eliminated
JREAC	list of all reactions acting on structure (four words per reaction joint giving joint number and code indicating whether or not components of reaction act in x-, y- or z-directions); (0) means no reaction, (1) means a reaction
NTAPE	tape reading error indicator (set by FIXTPE and sensed by routine calling FIXTPE)
NROW	list of joint numbers associated with matrix in primary storage area
STRIX	primary matrix storage area

The following arrays are included in common storage by all routines except STRIX (to be distinguished from array STRIX), BARSOL and CHECK.

NROSEC	secondary row sequence, associated with matrix in secondary storage area
LRSEC	list of reactions among joints being eliminated
STSEC	secondary matrix storage area

Additional parameters and arrays may be included in common storage by the use of certain operating routines, but these parameters do not affect the continuity of operation among routines.

E. TAPE STORAGE FORMATS

Storage formats for the input and buffer tapes attached to DSC A are given below. The format of the program tape B4 will be described with MISLAM.

Tape A6	Residuals and reaction matrices in two binary-record groups. Record 1 - IDOUT, NJT, NJSEC, NRJSEC, NRCSEC Record 2 - NROW, LRSEC, STSEC
Tape A7	Matrix equation storage in three binary-record groups. Record 1 - NHED, NJT, NJSEC, NRJSEC, NRCSEC, NU Record 2 - NROW, STRIX Record 3 - NROSEC, LRSEC, STSEC

Tape A8 - Load data with consecutive binary records containing the load matrix for consecutive units of the structure. The first word of each record is a heading NHED.

Tape A9 - Structural data in groups of BCD records.

Following an initial group of two data records, the records for the step number and the operation number follow, each of which is followed by the data records for the step.

Records 1 and 2 - General structure data

Record 3 - Step number, operation number

Record 4...n - Data for this step

Record n+1 - Step number, operation number

NHED identifies a record group u as 32767-u.

IDOUT identifies a two-record group as a residual (0) or reaction matrix (1).

F. STAIR OPERATING ROUTINES

The nine operating routines are described on the following pages. Each description has the following format:

- (a) Operation - definition of the matrix operation performed.
- (b) Mathematical method - brief description of technique for performing the operation.
- (c) Initial condition of core - common variables which must be present to maintain continuity.
- (d) Error stops.
- (e) Output - to tape only; on-line output consists of error diagnostics and tracing information.
- (f) Final condition of core - common variables which are changed, affecting continuity.
- (g) Addressing diagrams.
- (h) Block diagrams - flow charts.

1. STRIX

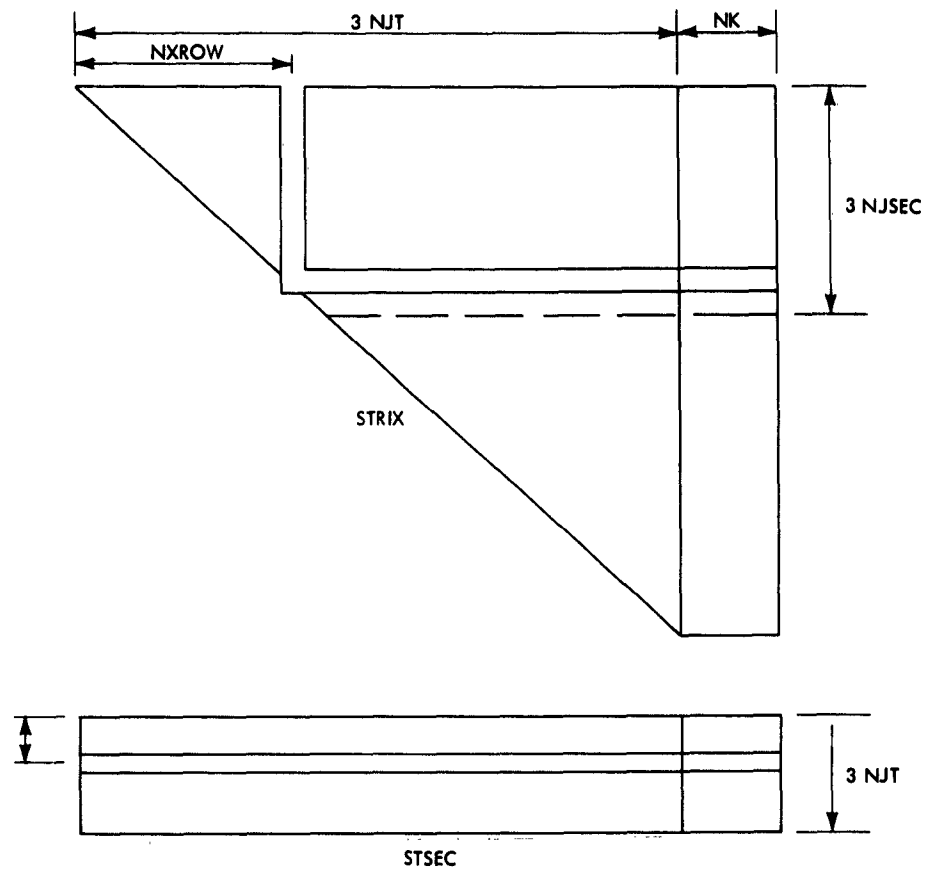
Operation: Forms a stiffness matrix equation in core from input defining the geometry of a structure and its load matrix.

Mathematical Method: Terms of the matrices for single bars are computed and superimposed on the array of the final matrix. The location of the final matrix array of the bar connecting joints a and b is determined by the positions of a and b in the list of joint numbers NROW. The load matrix for the structure is read directly from tape A8, and is stored immediately after the stiffness matrix. A vector check* indicates if all the bars meeting at any joint lie in a plane; such a geometry would result in a singular stiffness matrix.

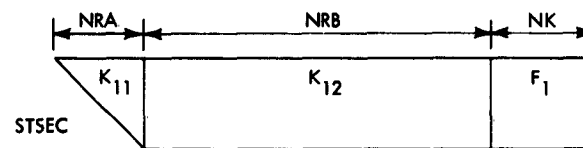
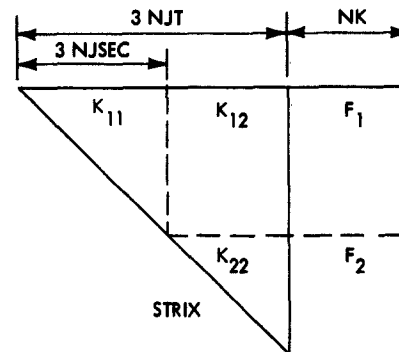
Initial Condition of Core: Common variables E, NUT and NK must be predetermined.

*See footnote on p. 41 for an explanation of this check.

EFFRIX ADDRESSING DIAGRAMS



REACTION EQUATION STORAGE BY EFFRIX



$$\text{NRA} = 3 \text{ NJSEC} - \text{NRCSEC}$$

$$\text{NRB} = 3 (\text{NJT} - \text{NJSEC})$$

SIZE RESTRICTIONS:

$$\text{NRA} (\text{NRA} + 1)/2 + \text{NRA} (\text{NRB} + \text{NK}) \leq 9500$$

$$\text{NRA} (\text{NRA} + 1) \leq 3 \text{ NJSEC} (3 \text{ NJSEC} + 1)/2 + 9 \text{ NJSEC} (\text{NJT} - \text{NJSEC})$$

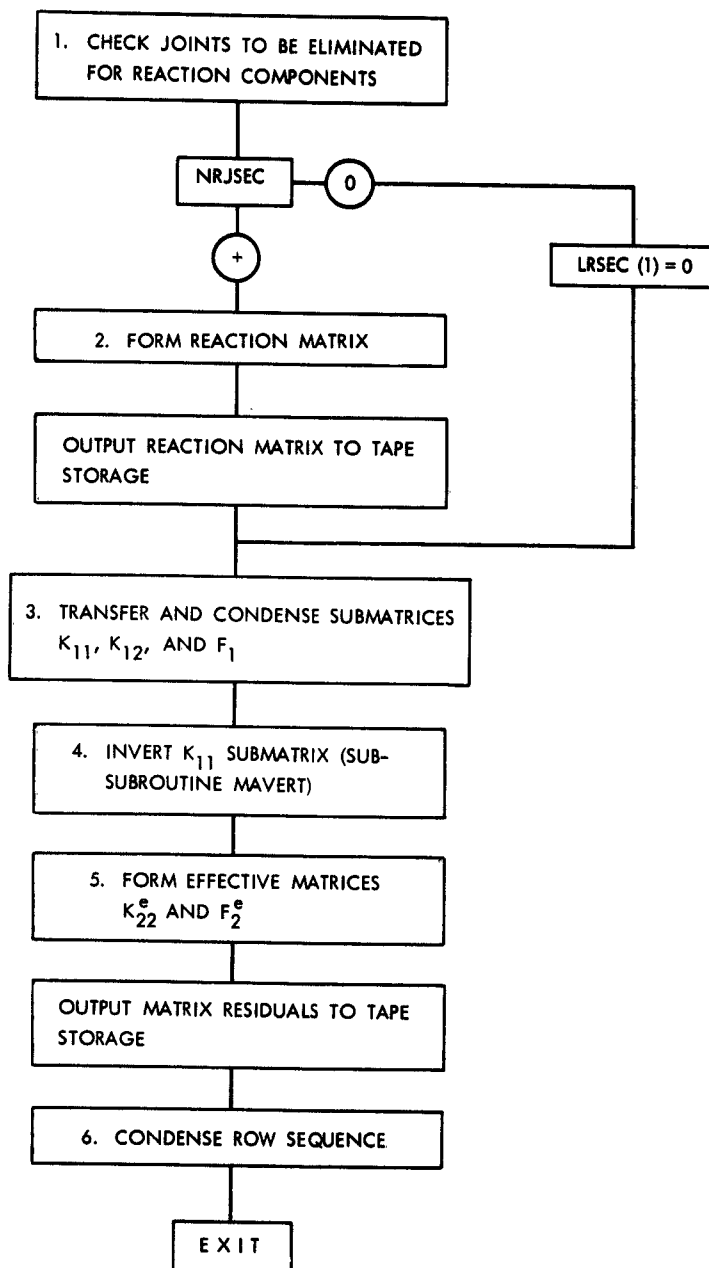
$$(3 \text{ NJT} + \text{NRB} + 1) 3 \text{ NJSEC}/2 > (\text{NRB} + 1) \text{NRB}/2$$

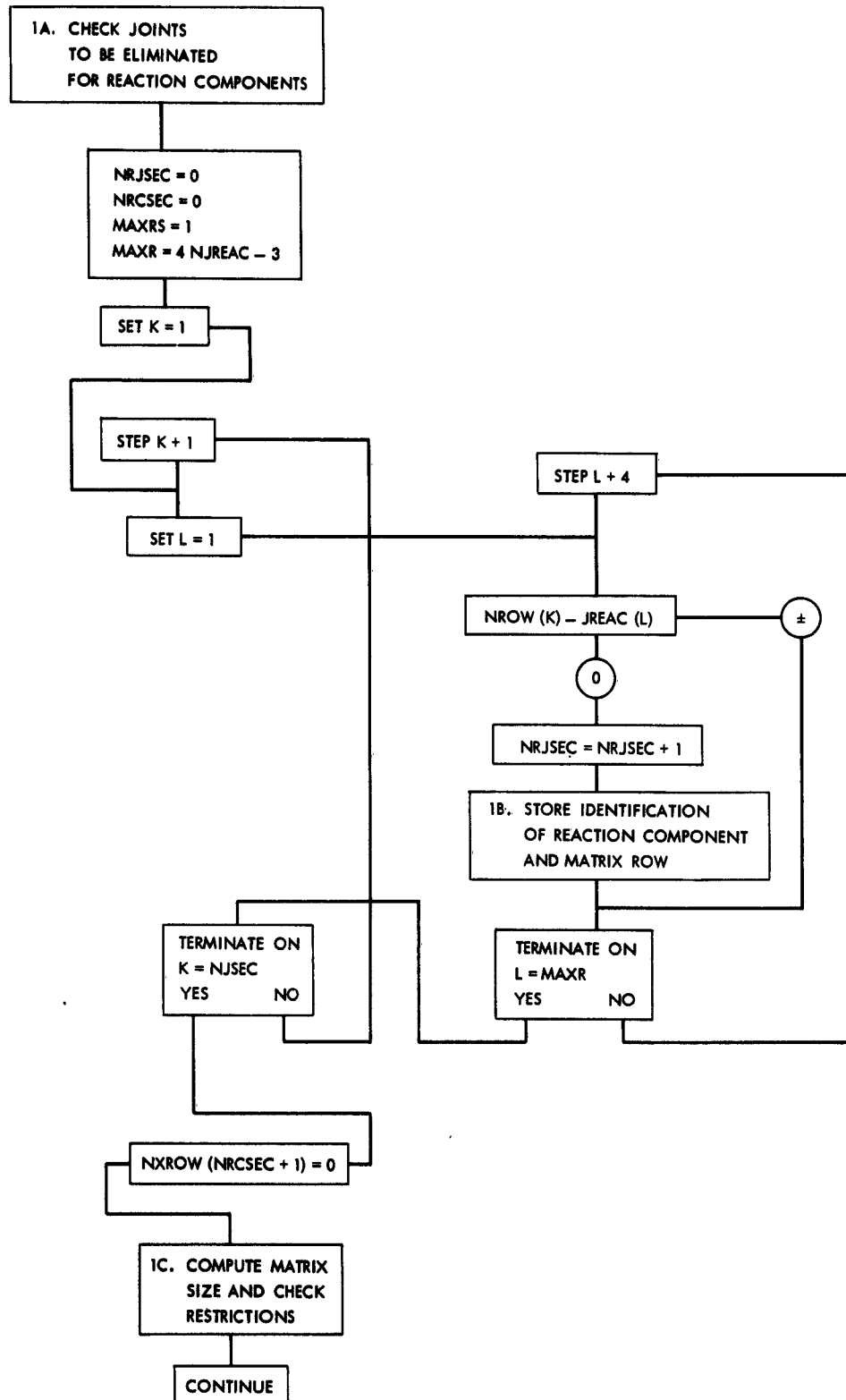
$$\text{NRA} > 0$$

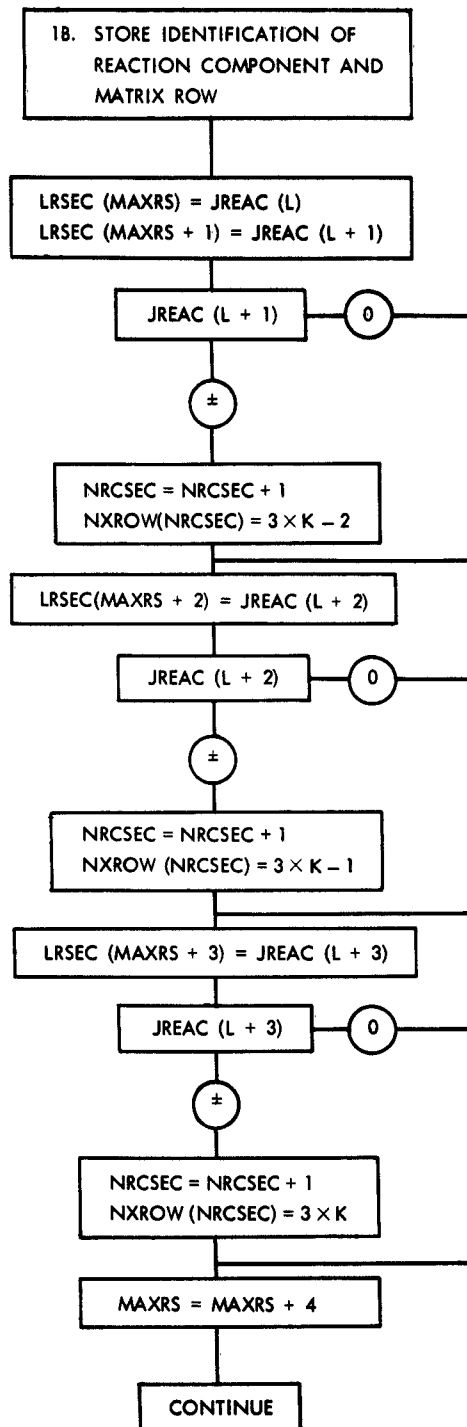
$$\text{NRJSEC} \leq 10$$

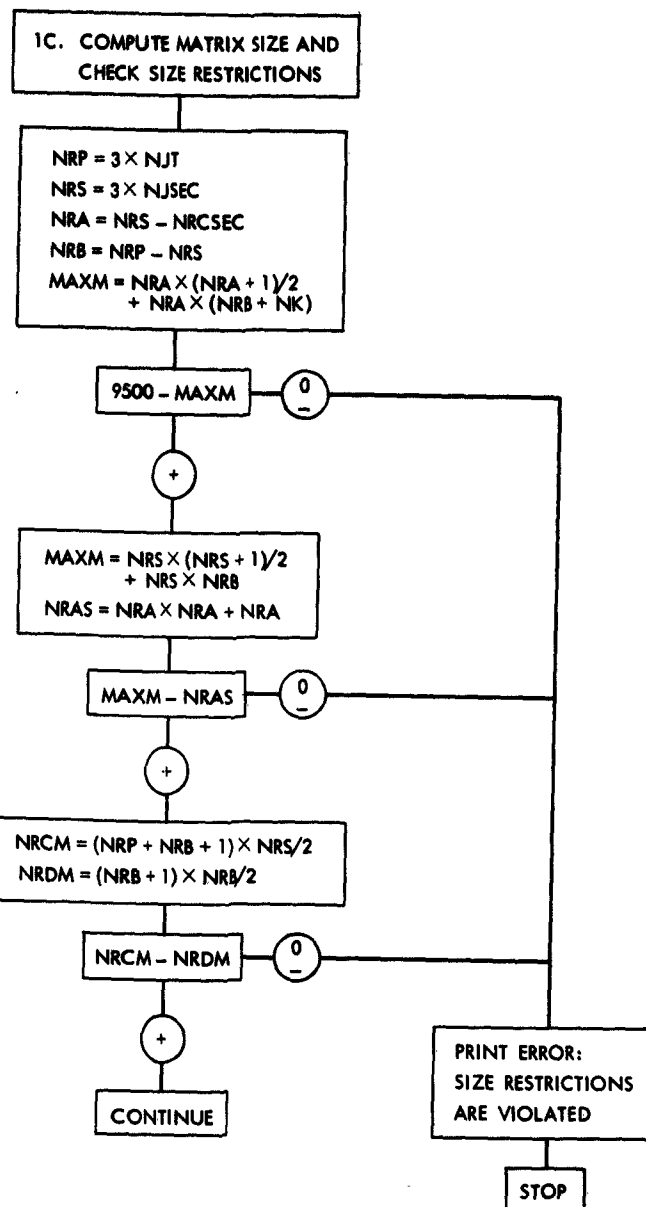
MATRIX RESIDUAL STORAGE BY EFFRIX

EFFRIX - GENERAL









AD 400280

~~END CHANGE PAGES~~

UNCLASSIFIED

AD 400 280

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

N-63-3-1

227

400280

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

STAIR

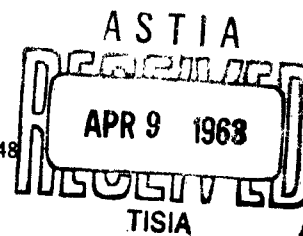
(STRUCTURAL ANALYSIS INTERPRETIVE ROUTINE)

INSTRUCTION MANUAL

MAINTAINED BY ASTIA

AW AD NO. _____

LINCOLN MANUAL NO. 48



MARCH 1962

AFESD - TDR - 63 - 29

400 280

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the joint support of the U.S. Army, Navy and Air Force under Air Force Contract AF 19(604)-7400.

LEXINGTON

MASSACHUSETTS

FOR ERRATA

AD _____

400280

THE FOLLOWING PAGES ARE CHANGES

TO BASIC DOCUMENT

AD-400280
400280

ERRATA SHEET
for
LINCOLN MANUAL NO. 48

STAIR
(STRUCTURAL ANALYSIS INTERPRETIVE ROUTINE)
INSTRUCTION MANUAL

Experience with the STAIR program over the past few months has disclosed a number of errors in the printed manual. Rather than list the errors on a sheet and rely on each holder of the manual to make the appropriate corrections, corrected sheets for insertion into the manual as follows are herein provided:

- Page 15: Problem 200, SLOP Input Data Sheet 'B,' Structural Data I: coordinate listing corrected. Remove original page 15.
- Page 16: Problem 200, SLOP Input Data Sheet 'B,' Structural Data II: area listing corrected. Remove original page 16.
- Page 17: Problem 200, SLOP Input Data Sheet 'B,' Structural Data II: area listing corrected. Line out page 17; save page 18.
- Page 21: Problem 200, SLOP Input Data Sheet 'C,' Loading Data, Case #3: loading corrected. Line out page 21; save page 22.
- Page 25: Printout: print error corrected. Remove original page 25.
- Page 26: Printout: print error corrected. Remove original page 26.
- Page 37: Program description: equations corrected. Line out page 37; save page 38.
- Page 39: Program description: equations corrected. Line out page 39; save page 40.
- Page 80: Program description: equations corrected. Line out page 80; save page 79.
- Page 84: Program block diagram: equations corrected. Line out page 84; save page 83.
- Page 118: DEFSOL Matrix Addressing Diagram: equations corrected. Line out page 118; save page 117.
- Page 119: DEFSOL Matrix Storage Diagram, Matrix Storage Area: equations corrected. Line out page 119; save page 120.
- Page 123: Program block diagram: equations corrected. Line out page 123; save page 124.

If any other errors come to notice, please notify Mr. William R. Fanning, Room D-255, Ext. 7592, M. I. T., Lincoln Laboratory, P.O. Box 73, Lexington 73, Massachusetts.

15 July 1963

Publications Office
M. I. T. Lincoln Laboratory
P. O. Box 73
Lexington 73, Massachusetts

PAGE 2/11

28 LINES

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA II

PROBLEM 200

PAGE 3/1

PUNCH ONE LINE
PER CARD

LIST BAR NUMBERS OF THE STRUCTURE (Z15, F12.6)																					
BAR NUMBER										AREA (SQ. IN)											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	0	1	0	1		0	1	0	2						3	1	0				
	0	1	0	1		0	1	0	3						3	1	0				
	0	1	0	1		0	1	0	4						3	1	0				
	0	1	0	1		0	3	0	4						3	1	0				
	0	1	0	2		0	1	0	3						3	1	0				
	0	1	0	2		0	1	0	4						3	1	0				
	0	1	0	2		0	1	0	5						3	1	0				
	0	1	0	2		0	1	0	6						3	1	0				
	0	1	0	2		0	3	0	4						3	1	0				
	0	1	0	2		0	3	0	6						3	1	0				
	0	1	0	2		0	1	0	4						3	1	0				
	0	1	0	2		0	1	0	5						3	1	0				
	0	1	0	3		0	3	0	4						3	1	0				
	0	1	0	3		0	3	0	5						3	1	0				
	0	1	0	3		0	3	0	6						3	1	0				
	0	1	0	4		0	1	0	5						3	1	0				
	0	1	0	4		0	1	0	6						3	1	0				
	0	1	0	4		0	2	0	1						3	1	0				
	0	1	0	4		0	2	0	2						3	1	0				
	0	1	0	4		0	2	0	3						3	1	0				
	0	1	0	5		0	1	0	6						3	1	0				
	0	1	0	5		0	2	0	3						3	1	0				
	0	1	0	6		0	2	0	2						3	1	0				
	0	1	0	6		0	2	0	3						3	1	0				
	0	2	0	1		0	2	0	2						3	1	0				
	0	2	0	1		0	2	0	3						3	1	0				
	0	2	0	1		0	2	0	4						3	1	0				
	0	2	0	2		0	2	0	3						3	1	0				
	0	2	0	2		0	2	0	4						3	1	0				
	0	2	0	2		0	2	0	5						3	1	0				
	0	2	0	2		0	2	0	6						3	1	0				
	0	2	0	3		0	2	0	4						3	1	0				
	0	2	0	3		0	2	0	5						3	1	0				
	0	2	0	4		0	2	0	5						3	1	0				
	0	2	0	4		0	2	0	6						3	1	0				
	0	2	0	4		0	3	0	1						3	1	0				
	0	2	0	4		0	3	0	2						3	1	0				
	0	2	0	4		0	3	0	3						3	1	0				
	0	2	0	5		0	2	0	6						3	1	0				
	0	2	0	5		0	2	0	3						3	1	0				

40 LINES

PAGE 8/11

PUNCH ONE LINE PER CARD

1	3	
2	0	3
IN	ND	
TP		

(11, 12)

1	5
00005	(15)

(I4,3F12.6)

JT No.	F _x (KIPS)	F _y (KIPS)	F _z (KIPS)
1	0		
2	1		
3	0		
4	3		
5			
6			
7			
8			
9	0		
10	5	-5	0
11	0		0
12	5		0
13	0		0
14	0		0
15			
16			
17			
18			
19			
20			
21		-10	-7.5
22			11
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			

20 LINES

20004030 0180034			
060606			
101	00	00	00
102	00	-3464102	20
103	00	-3464102	-20
104	-60	00	00
105	-120	-3464102	-20
106	-120	-3464102	20
201	-30	5196152	00
202	-60	6928203	20
203	-60	6928203	-20
204	00	10392305	00
205	00	17320508	-20
206	00	17320508	20
301	30	5196152	00
302	60	6928203	20
303	60	6928203	-20
304	60	00	00
305	120	-3464102	-20
306	120	-3464102	20
0101	0102	30	
0101	0103	30	
0101	0104	30	
0101	0304	30	
0102	0103	30	
0102	0104	30	
0102	0105	30	
0102	0106	30	
0102	0304	30	
0102	0306	30	
0103	0104	30	
0103	0105	30	
0103	0304	30	
0103	0305	30	
0103	0306	30	
0104	0105	30	
0104	0106	30	
0104	0201	30	
0104	0202	30	
0104	0203	30	
0105	0106	30	
0105	0203	30	
0106	0203	30	
0106	0202	30	
0201	0202	30	
0201	0203	30	
0201	0204	30	
0202	0203	30	
0202	0204	30	
0202	0205	30	
0202	0206	30	
0203	0204	30	
0203	0205	30	
0204	0205	30	
0204	0206	30	
0204	0301	30	
0204	0302	30	
0204	0303	30	
0205	0206	30	

0205	0303	30		
0206	0302	30		
0206	0303	30		
0301	0302	30		
0301	0303	30		
0301	0304	30		
0302	0303	30		
0302	0304	30		
0302	0305	30		
0302	0306	30		
0303	0304	30		
0303	0305	30		
0304	0305	30		
0304	0306	30		
0305	0306	30		
101				
1700		0	-1	0
102				
1700		0	-1	0
202				
00002				
0202	5	0		75
0303	5	0		-75
203				
00005				
0103	0	-5		0
0202	5	0		0
0204	0	-10		0
0206	0	0		-75
0303	5	0		1
204				
00001				
0101	1	2		3
3				
0109				
0101	0102	0103	0104	0105
0305	0306			
0209				
0201	0202	0203	0204	0205
0105	0106			
0309				
0301	0302	0303	0304	0305
0205	0206			

NK = number of different loading conditions.

$$A = 3NE - NC.$$

$$B = 3(NJ - NE).$$

$$C = 3NE.$$

- (a) $NR \leq 10.$
- (b) $\frac{A(A+1)}{2} + A(B + NK) \leq 9500.$
- (c) $A(A+1) < [C \times \frac{C+1}{2} + C \times B].$
- (d) $\frac{B(B+1)}{2} < [C \times \frac{C+1}{2} + C \times B].$
- (e) $A > 0$

MATOUT (refer to p. 47)

- (a) Step number.
- (b) Tape location (Loc.) in which a matrix is to be stored. Matrices are stored on tape in blocks. These are numbered consecutively, the tape location being the number given to the block.

Operation Restriction

Only one matrix may be stored in a tape location.

MATRO (refer to p. 48)

- (a) Step number.
- (b) Identification number of new unit (NUN).
- (c) Angle through which old unit* must be rotated in the x-y plane to arrive at the position of the new unit. Angle (EPS) measured in degrees and positive when clockwise, as viewed by an observer on the positive z-axis. In the STAIR system, the rotation must be about the z-axis.
- (d) Row sequence of the resulting matrix after rotation [NROW (1) to NROW (NJT)].
- (e) If the matrix being rotated is a reduced matrix, i.e., some joints have been eliminated from it by EFFRIX in a previous step, the inverse matrix corresponding to these eliminated joints is stored in the secondary matrix storage - then, not otherwise. The new row sequence of the eliminated joints, i.e., the new row sequence for the new inverse matrix, is NROSEC (1) to NROSEC (NE).

Size Restriction

- (a) New unit number $\leq$ number of units in the structure.

Operation Restriction

- (a) Matrices of identical units from which joints have been eliminated may be formed by MATRO except when one of the eliminated joints of the old unit has a corresponding joint in the new unit which is a reaction joint, or vice-versa.
- (b) MATRO can not follow a double EFFRIX sequence.

*Matrix of unit in core.

DEFSOL (refer to p. 44)

- (a) Step number.
- (b) List of the number of joints which are prefixed by each unit number (IUADR). For example, there are 9 joints in unit 01, but only 6 joints are prefixed by unit number 01. (Notes regarding this in SLOP apply here also.)

Size Restrictions

NJ = number of joints in final row sequence.
NR = number of reaction joints among final joints.
NC = number of reaction components among final joints.
NK = number of loading conditions.
NJT = total number of joints in structure.
NRT = total number of reaction joints in structure.
 $A = 3NJ - NC.$

- (a) $NR \leq 10.$
- (b) $A \times (A + 1)/2 + A \times NK \leq 9500.$
- (c) $A(A + 1) < 16,000.$
- (d) $4 \times NK(NJT + NRT) < 16,000.$

Operation Restriction

DEFSOL may be called only when the final matrix of the structure is stored in the computer core.

Output

- (a) Joint displacement (in inches).
- (b) Reactions (in kips).

BARSOL (refer to p. 54)

- (a) Step number.
- (b) Number of bars whose stresses are to be computed (NM).
- (c) Number of joints to which the bars listed in (b) are connected (NJ).
- (d) List of bar numbers of the bars whose stresses are to be computed (reproduced from SLOP).

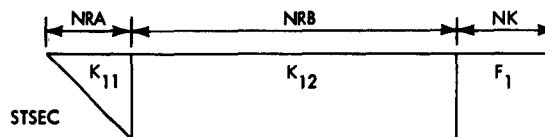
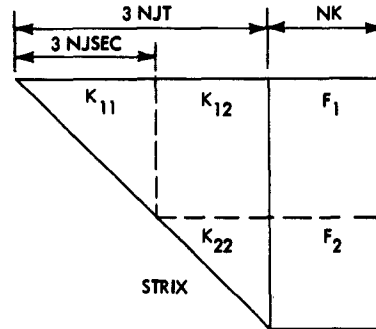
Note: STAIR will compute any desired number of bar stresses by repeated use of BARSOL.

- (e) Joint cards, giving the joint numbers and coordinates. Only those joints which appear in the list of bar numbers (p. 54) need be included (reproduced from SLOP).

Size Restrictions

NK = number of loading conditions.

- (a) $NJ \leq 800.$
- (b) $NM \leq 1000.$
- (c) $NK \times NM \leq 4300.$



$$\text{NRA} = 3 \text{ NJSEC} - \text{NRCSEC}$$

$$\text{NRB} = 3 (\text{NJT} - \text{NJSEC})$$

SIZE RESTRICTIONS:

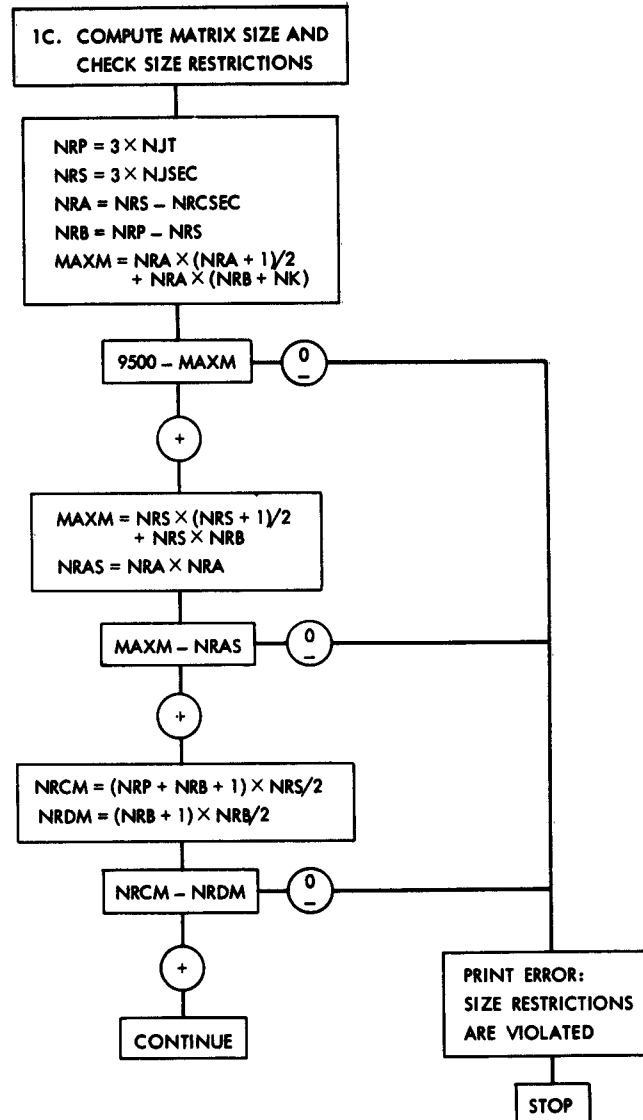
$$\text{NRA} (\text{NRA} + 1)/2 + \text{NRA} (\text{NRB} + \text{NK}) \leq 9500$$

$$\text{NRA} (\text{NRA} + 1) \leq 3 \text{ NJSEC} (3 \text{ NJSEC} + 1)/2 + 9 \text{ NJSEC} (\text{NJT} - \text{NJSEC})$$

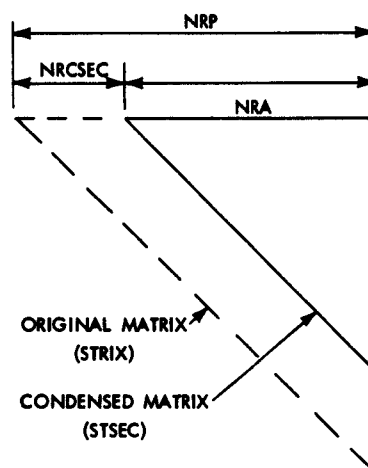
$$(3 \text{ NJT} + \text{NRB} + 1) 3 \text{ NJSEC}/2 > (\text{NRB} + 1) \text{NRB}/2$$

$$\text{NRA} > 0$$

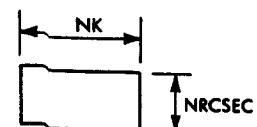
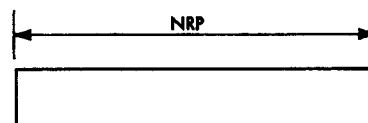
MATRIX RESIDUAL STORAGE BY EFFRIX



DEFSOL MATRIX ADDRESSING DIAGRAM



REACTION EQUATIONS (STSEC)



SIZE LIMITATIONS:

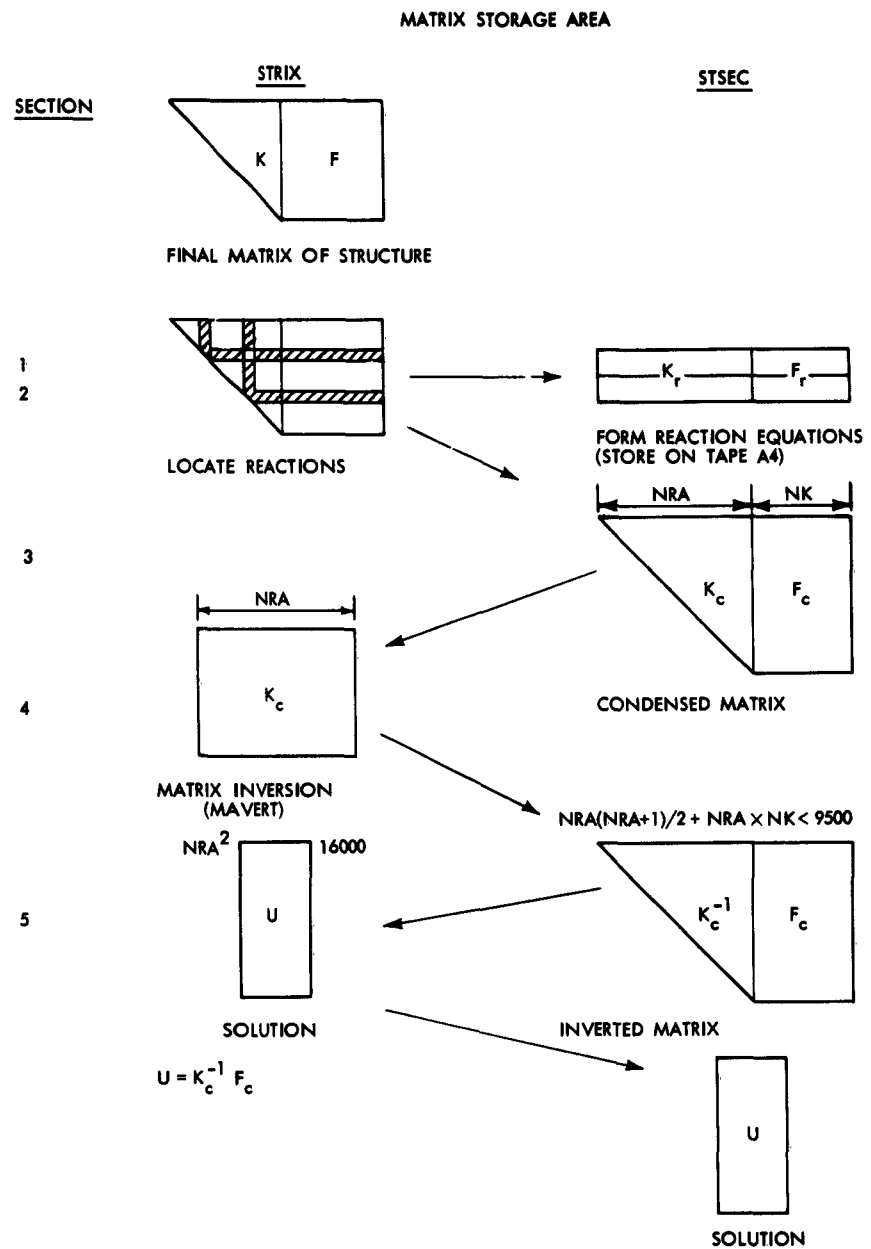
$$\frac{NRA(NRA + 1)}{2} + NRA \times NK \leq 9500$$

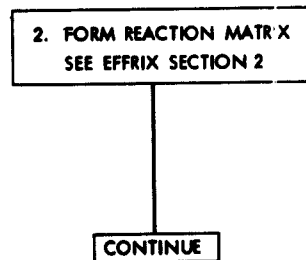
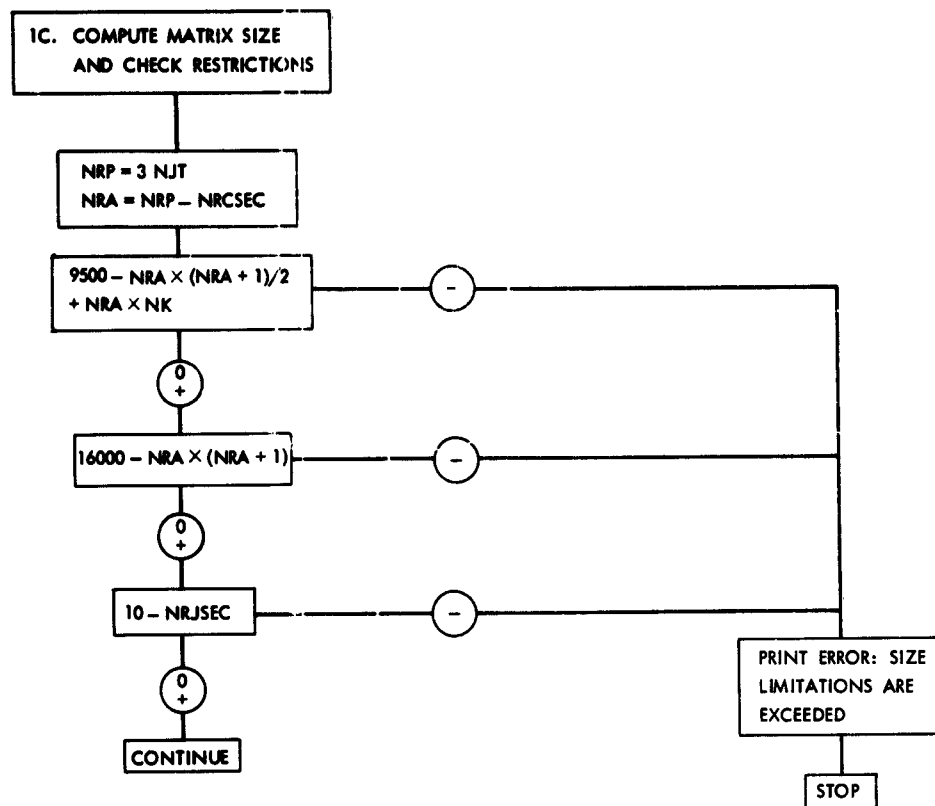
$$(NRA + 1) \times NRA \leq 16000$$

$$NRCSEC \leq 10$$

$$NRA > 0$$

DEFSOL MATRIX STORAGE DIAGRAM





ACKNOWLEDGMENTS

The STAIR program was originally developed in the Department of Civil and Sanitary Engineering, Massachusetts Institute of Technology, from 15 September 1958 through 30 June 1960, under purchase orders issued by Group 75, Construction Engineering. Mr. John M. Biggs, Associate Professor of Structural Engineering, and Mr. Saul Namyet, Assistant Professor of Structural Engineering, supervised the project. Mr. Richard Batchelder, Miss Peng-Chih Yang and Mr. Kirit Parikh, Research Assistants, worked on various phases of the project.

Modification of the program to adapt it to the IBM-7090 and final debugging were performed by Mr. Parikh, Mr. Edwin C. Rossow, a summer staff member of Group 75, and Mr. William Fanning, at that time of the firm of Cleverdon, Varney and Pike, Boston, while in residence at Lincoln Laboratory. General administrative control of the development of the program was exercised by Mr. David C. Moore, then Assistant Leader of Group 75.

TABLE OF CONTENTS

Acknowledgments	ii
Chapter 1 - INTRODUCTION	1
Chapter 2 - USER'S MANUAL	3
A. Introduction	3
B. The Stiffness Matrix Method	7
C. Input Data	10
D. Input Data for the STAIR Load Program (SLOP)	11
E. The STAIR Operations	30
F. Input Data for the STAIR System	35
G. STAIR System Error Stops	41
Chapter 3 - OPERATOR'S MANUAL	65
A. Introduction	65
B. Input-Output	65
C. Machine Operation	65
D. Sense Switches and Indicator Lights	65
E. Program Stops	66
F. Interruption of a Run	66
Chapter 4 - PROGRAMMER'S MANUAL	67
A. Introduction	67
B. Definition of Matrix Terms	67
C. Core Storage Allocation	68
D. Definition of Common Variables	69
E. Tape Storage Formats	69
F. STAIR Operating Routines	70
STRIX	70
EFFRIX	78
MATRO	95
MADD	105
MATOUT	112
MATIN	115
DEFSOL	117
BARSOL	137
CHECK	142
G. STAIR Service Routines	145
Master Routine	145
FIXTPE	147
MAVERT	149
MISLAM for IBM-7090	149
H. STAIR Load Programs	157
SLOP 1	157
SLOP 2	162
Chapter 5 - LIMITATIONS AND EXTENSIONS	171

CHAPTER 1

INTRODUCTION

An analysis of a structure is a determination of its deflections, reactions and internal stresses under load. For purposes of STAIR, a structure is defined as a three-dimensional system of prismatic bars, connected at frictionless pin joints and supported in a stable position by reactions. The only external loading considered is a system of concentrated forces applied to the joints. When the number of joints making up such a framework becomes large, the conventional methods of analysis become too cumbersome for slide rule or desk calculator solutions, and a high-speed machine analysis is necessary. If it is necessary to use electronic digital computers, it is desirable to consider less common methods of analysis, which may be more suitable than conventional procedures for use on a machine.

The stiffness matrix method of analysis, which is the basis of STAIR, is such a method. The method has two essential steps: (1) the forming of a stiffness matrix for the structure from the geometric properties of the structure, and (2) the solving of a matrix equation, which is a set of simultaneous equations formed from selected coefficients of the stiffness matrix and the applied loads. The set of equations describes the equilibrium of forces at the movable joints in terms of their unknown displacements. The solution of the equations is a set of displacements, which can, in turn, be used to solve for the bar stresses and reactions of the structure.

The characteristic of this method that makes it amenable to machine computation is its repetitive nature. The form of the stiffness matrix is the same for each member and a stiffness matrix for a structure is merely the sum of the matrices for the individual members. Solving a set of simultaneous equations is usually accomplished by means of some sort of algorithm, which is also a repetitive procedure. A second advantage of this method is its independence of the degree of redundancy. Additional redundancies do not require more machine time. Finally, the procedure yields all the joint displacements as the first available result, and, thus, is especially suited to displacement problems.

The one characteristic of the stiffness matrix method of analysis that is somewhat inconvenient and which makes it unsatisfactory for use in other than machine computation is the large number of simultaneous equations that must be handled. In general, there are three unknown displacements at each joint of a three-dimensional structure; therefore, three equations must be written at each joint. This becomes an inconvenience, even in machine computation, when larger structures with more joints are considered. The STAIR procedure has been developed to break down large structures into small parts because the simultaneous equations required for large structures do not fit into the available memory space in many computers and, more importantly, require an excessive amount of computer time if solved by direct methods. This enables the programmer to remain within the size-time restrictions of the modern computer.

The analysis of any three-dimensional framework by this method depends on the proper combination of nine "Basic Operations" which have been prepared as self-contained computer "sub-routines." The internal functioning or derivation of these operations need not concern the structural analyst, but he must have clearly in mind their functions and limitations.

Among the requirements of each subroutine are certain numbers which describe the geometry of the structure and parameters which are present for a given structure to control the manipulation of geometrical data by the subroutines.

The ANALYSIS PROGRAM described in Chapter 2 is the device for effecting an analysis by this procedure. It is essentially a list of the names of the necessary subroutines arranged in proper sequence for a particular structure. Associated with each subroutine is its data and control parameters. Rigid rules are presented for each subroutine and its data. If these are followed exactly, it is easy to analyze almost any three-dimensional pin-jointed framework.

In the following chapters, the stiffness matrix method of analysis is explained in some detail, the ANALYSIS PROGRAM is described, the rules for use of the subroutines are listed, the possible errors and pitfalls are listed for each operation and a sample solution is presented.

CHAPTER 2

USER'S MANUAL

A. INTRODUCTION

Various techniques exist for analyzing the structural behavior of a stable, three-dimensional, pin-jointed framework. Generally, statically determinate structures are analyzed by solving equations of equilibrium at joints or from cut sections, whereas indeterminate structures require the solution of additional simultaneous equations that are related to the redundancies of such structures.

These methods were developed for hand and slide rule computation; in many cases the results are at an acceptable level of precision. However, the introduction of high-speed, externally programmed, electronic computing machines has opened new channels for analysis. In particular, it has permitted the engineer to use rigorous mathematical models of structures, which formerly could not be used on large frameworks due to the sheer volume of computations required.

The STAIR system has been developed to use the stiffness matrix technique, which handles a structure in terms of the stiffness of its members and the deflections of its joints. This system is outlined in Sec. B.

The geometry of a structure is defined as the position and orientation of its joints and members relative to some fixed coordinate system, the cross-sectional area and modulus of elasticity of its members and the location and direction of its reactions. This is a necessary ingredient in any structural analysis technique.

The size of the core memory of the newest machines, while large, is still the limiting factor in the size of structure which can be analyzed. Since, in many cases, the structure under consideration may be several times larger than the limits of the memory, STAIR is arranged to permit the division of the structure into small, internally stable units, which can be partially analyzed and then combined for a final solution. This ability imposes a definite nomenclature on the geometrical arrangement of the structure.

The nomenclature used in STAIR is shown in Figs. 1 and 2. If the structure is small enough, it may be considered as a single unit (Fig. 1) and analyzed directly. However, as the number of joints (and therefore equations) increases, it is necessary to subdivide the structure, as shown in Fig. 2. It should be noted that STAIR is particularly useful with structures showing a rotational symmetry around the z-axis. Units are numbered consecutively starting with 01. Joints are given a four-digit number, the first two being the unit number with which it is identified. Thus, joint 0112 in Fig. 1 is the 12th joint in unit 01, and joint 0206 in Fig. 2 is the 6th joint in unit 02. To avoid duplication, a joint is given only one number even though it may appear in two units. Thus, in Fig. 2, joint 0104 is shown both in unit 01 and unit 02. Bars are designated by eight-digit numbers, which consist of the numbers of the joints which the bar connects. The joint with the lower number is listed first by convention. Thus, the bar connecting joints 0104 and 0105 is denoted as 0104-0105. In this case, where the bar is common to two units, the programming is handled by using one-half of the cross-sectional area in the input of each unit.

The procedure for solving a structure with STAIR is as follows:

1. Load data are prepared for as many cases as desired. The data, including dead load information, the geometry of the structure and the various live load conditions, are introduced

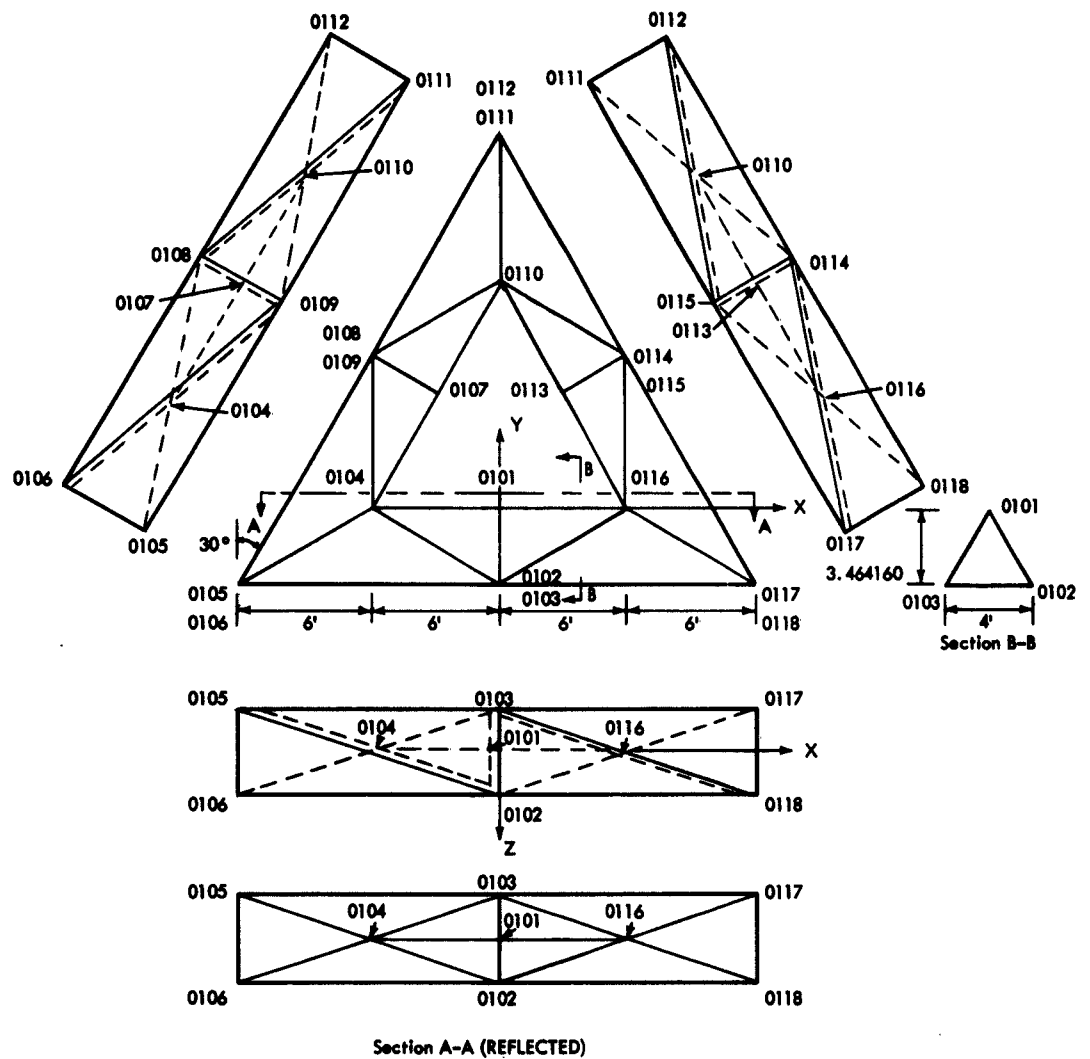


Fig. 1. Example Structure
cross section area of bars = 3 sq. in.
elastic modulus = 10,000 kips per sq. in.

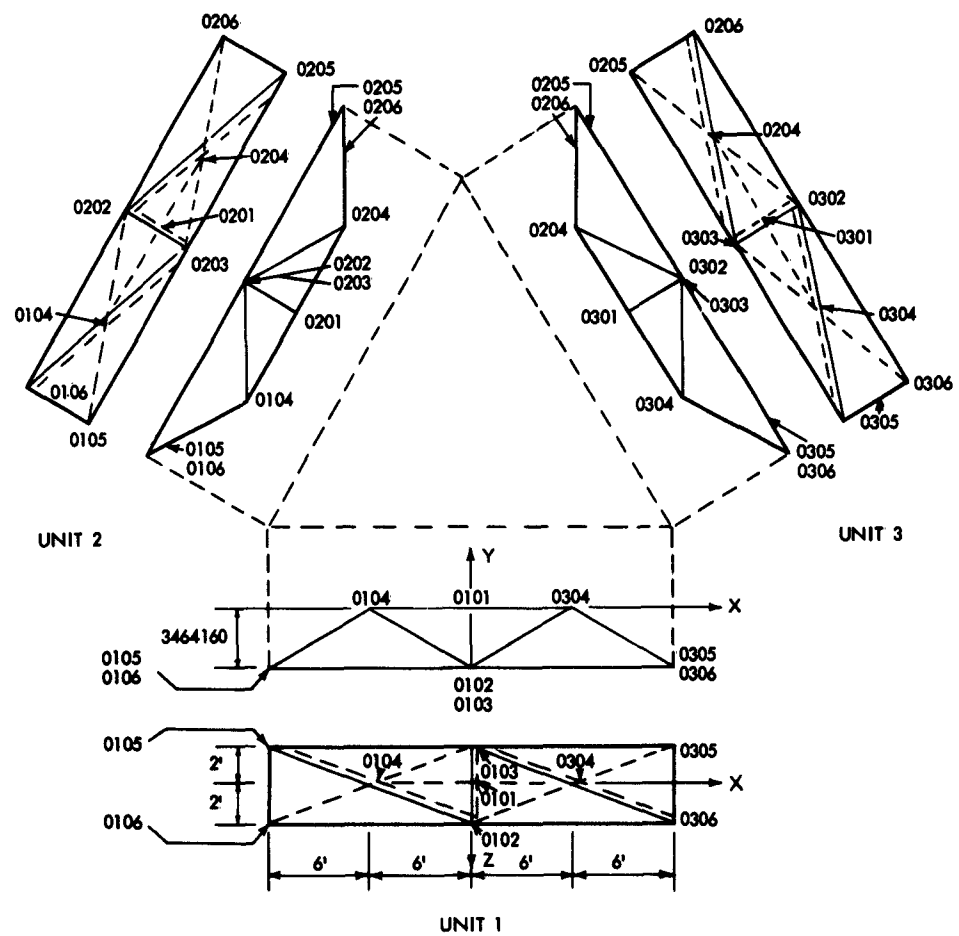


Fig. 2. Example structure divided into three units.

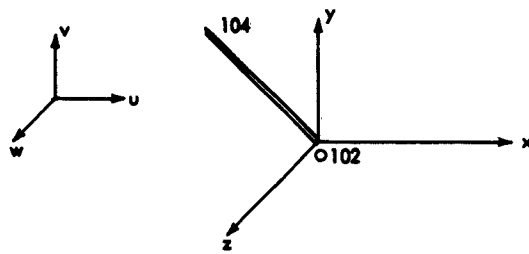


Fig. 3.

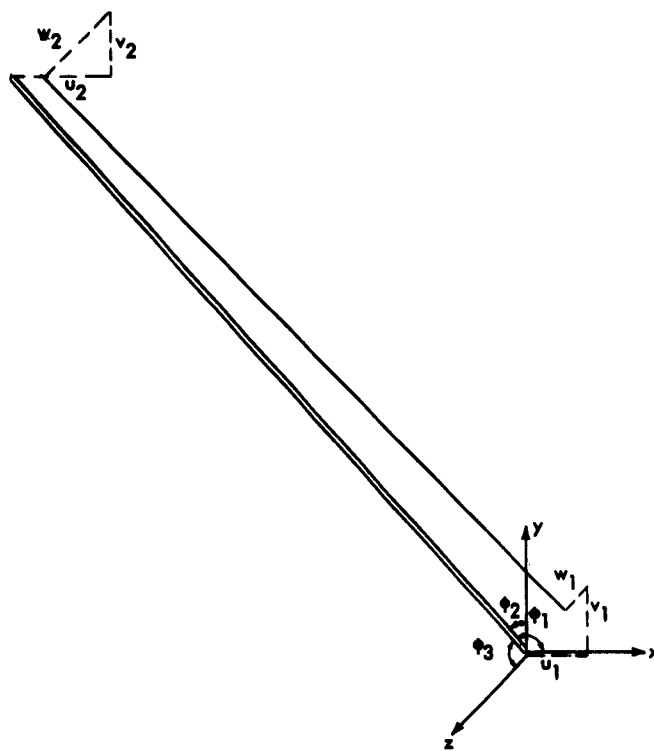


Fig. 4.

into the STAIR Load Program (SLOP); the output of this program is a data tape which serves as input to the STAIR ANALYSIS PROGRAM.

2. The ANALYSIS PROGRAM is prepared, listing the various subroutines required for solution. Geometrical data for the structure are again introduced, including the division of the structure into smaller units. When possible, units should have identical bar arrangements with rotational geometric symmetry, since this will reduce input data requirements and computational time. Loads and reaction joints for these units need not be symmetric.

3. THE ANALYSIS PROGRAM, the geometrical data and the SLOP output tape are introduced to the computer, which will form, modify and solve the stiffness matrices, and then back-substitute to get the final structural solution. The results of a complete run consist of the joint displacements, bar stresses, reactions and summation of internal bar forces at each joint for each loading condition.

In the following sections, a complete program for the example structure of Fig. 2 is set up and solved on the IBM-7090. Descriptions of the input data and program arrangement and discussion of the size restrictions are presented at each step. Although the structure used is small enough to permit solving as a single unit, it has been divided into rotationally symmetric units to demonstrate the use of the various subroutines.

B. THE STIFFNESS MATRIX METHOD

The stiffness matrix method used in structural analysis is the solution of the force equilibrium equations in the form of a matrix equation. The STAIR System is mainly a matrix operation, such as matrix formation, addition, multiplication, inversion, elimination, etc. When a final stiffness matrix representing the structure is obtained, it is solved for the joint displacements (U) corresponding to a particular loading condition (F).

A stiffness matrix (K) is a numerical representation of a structure. It relates the joint load (F) and joint displacement (U) in the form $(F) = (K)(U)$. Consider the bar, 0102-0104, of the example structure. Figure 3 shows its relation to the xyz coordinate system and the uvw displacement system. Figure 4 shows the bar after it has undergone positive u, v and w displacements. Note that the subscript 1 refers to joint 0102, and subscript 2 refers to joint 0104.

From Fig. 4,

$$\left. \begin{aligned} f_{x1} &= F_1 \cos \varphi_1 = F_1 \alpha & f_{x2} &= -f_{x1} \\ f_{y1} &= F_1 \cos \varphi_2 = F_1 \beta & f_{y2} &= -f_{y1} \\ f_{z1} &= F_1 \cos \varphi_3 = F_1 \gamma & f_{z2} &= -f_{z1} \end{aligned} \right\} \quad (1)$$

where f_{x1} , f_{y1} and f_{z1} are the x, y and z forces at joint 1 (0102), f_{x2} , f_{y2} and f_{z2} are the x, y and z forces at joint 2 (0104),

$$\alpha = \frac{\Delta x}{L} \quad \beta = \frac{\Delta y}{L} \quad \gamma = \frac{\Delta z}{L}$$

are the direction cosines and L is the length of the bar. F_1 can now be related to the uvw displacements in the following manner:

$$F_1 = \frac{AE}{L} [(u_1 - u_2) \cos \varphi_1 + (v_1 - v_2) \cos \varphi_2 + (w_1 - w_2) \cos \varphi_3]$$

$$F_1 = \frac{AE}{L} [(u_1 - u_2)\alpha + (v_1 - v_2)\beta + (w_1 - w_2)\gamma] ,$$

where A is the cross-sectional area of the bar and E is its modulus of elasticity.

Substitution into Eq. (1) gives

$$\begin{aligned} f_{x1} &= AE/L[\alpha^2 u_1 + \alpha\beta v_1 + \alpha\gamma w_1 - \alpha^2 u_2 - \alpha\beta v_2 - \alpha\gamma w_2] \\ f_{y1} &= AE/L[\alpha\beta u_1 + \beta^2 v_1 + \beta\gamma w_1 - \alpha\beta u_2 - \beta^2 v_2 - \beta\gamma w_2] \\ f_{z1} &= AE/L[\alpha\gamma u_1 + \beta\gamma v_1 + \gamma^2 w_1 - \alpha\gamma u_2 - \beta\gamma v_2 - \gamma^2 w_2] \\ f_{x2} &= AE/L[-\alpha^2 u_1 - \alpha\beta v_1 - \alpha\gamma w_1 + \alpha^2 u_2 + \alpha\beta v_2 + \alpha\gamma w_2] \\ f_{y2} &= AE/L[-\alpha\beta u_1 - \beta^2 v_1 - \beta\gamma w_1 + \alpha\beta u_2 + \beta^2 v_2 + \beta\gamma w_2] \\ f_{z2} &= AE/L[-\alpha\gamma u_1 - \beta\gamma v_1 - \gamma^2 w_1 + \alpha\gamma u_2 + \beta\gamma v_2 + \gamma^2 w_2] \end{aligned} \quad (2)$$

A convenient way of writing Eq. (2) is to employ matrix notation as follows:

$$\begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{z1} \\ f_{x2} \\ f_{y2} \\ f_{z2} \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} \alpha^2 & \alpha\beta & \alpha\gamma & -\alpha^2 & -\alpha\beta & -\alpha\gamma \\ \alpha\beta & \beta^2 & \beta\gamma & -\alpha\beta & -\beta^2 & -\beta\gamma \\ \alpha\gamma & \beta\gamma & \gamma^2 & -\alpha\gamma & -\beta\gamma & -\gamma^2 \\ -\alpha^2 & -\alpha\beta & -\alpha\gamma & \alpha^2 & \alpha\beta & \alpha\gamma \\ -\alpha\beta & -\beta^2 & -\beta\gamma & \alpha\beta & \beta^2 & \beta\gamma \\ -\alpha\gamma & -\beta\gamma & -\gamma^2 & \alpha\gamma & \beta\gamma & \gamma^2 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \end{bmatrix} \quad (3)$$

or $\{f\} = \{k\} \{u\}$.

The complete representation of the whole structure or unit can be obtained by superimposing $\{f\} = \{k\} \cdot \{u\}$ for all bars, giving $\{F\} = \{K\} \cdot \{U\}$ for the whole structure or unit. Then, stiffness matrices representing adjacent units of a structure may be superimposed to give a stiffness matrix representing a combined structure.

The formation of a stiffness matrix for a new unit, which is rotationally symmetric to another unit, requires only a transformation of the stiffness matrix of the latter unit. The transformation in the present STAIR System is limited to rotation of the plane containing two coordinate axes (the x and y) about the third (the z-axis).

The size of the stiffness matrix for a unit or combination of units can be reduced by eliminating from the equation $\{F\} = \{K\} \cdot \{U\}$ joints which are not part of units yet to be considered.

The new matrix will represent the same structure as the original matrix. This operation, which amounts to the partitioning of the original stiffness matrix, is equivalent to the elimination process used to reduce the number of unknowns in a set of simultaneous equations such as Eq. (2), and is referred to as "Elimination." The new matrix is called an "Effective Matrix."

To illustrate the elimination process consider the force-displacement equation for unit 01 in Fig. 2:

$$\begin{bmatrix} F_{0101} \\ F_{0102} \\ F_{0103} \\ F_{0104} \\ F_{0105} \\ F_{0106} \\ F_{0304} \\ F_{0305} \\ F_{0306} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} K \begin{bmatrix} U_{0101} \\ U_{0102} \\ U_{0103} \\ U_{0104} \\ U_{0105} \\ U_{0106} \\ U_{0304} \\ U_{0305} \\ U_{0306} \end{bmatrix} \quad (4)$$

where F and U represent three components of joint force and joint displacement, respectively. Displacements U_{0101} , U_{0102} and U_{0103} are now eliminated from the equations (these are at the joints not common to other units) and Eq. (4) is reduced to

$$\begin{bmatrix} F'_{0104} \\ F'_{0105} \\ F'_{0106} \\ F'_{0304} \\ F'_{0305} \\ F'_{0306} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix} K^e \begin{bmatrix} U_{0104} \\ U_{0105} \\ U_{0106} \\ U_{0304} \\ U_{0305} \\ U_{0306} \end{bmatrix} \quad (5)$$

The original matrix (K) of size 27×27 is thus reduced to the effective matrix (K^e) of size 18×18 . During this process, the force vector F is also modified to F' . The details of this modification are described in Chapter 4.

A similar effective matrix can be formed for unit 2 and superimposed on that for unit 1. The resulting matrix represents the structure shown in Fig. 5. At this point, the fact that displacements y_{0106} and z_{0106} must equal zero is taken into account by discarding the columns and rows in the matrix corresponding to these displacements. The matrix for the two-unit combination is

then 25×25 . The displacements at joints 0105 and 0104 and the remaining displacement at 0106 are now eliminated, leaving an effective 18×18 matrix.

Finally, the effective matrix for unit 3 is superimposed. This does not increase the size of the matrix since no new joints are introduced. Joints 0204, 0205 and 0206 are now eliminated leaving a 9×9 matrix. The displacements at joint 0306 are all zero and the corresponding terms in the matrix may be discarded. This leaves only a 6×6 matrix, which is finally solved for the displacements at joints 0304 and 0305.

By the simple process of substitution, all other displacements may be determined. Having displacements, all bar stresses and reactions are easily obtained.

In comparison to the method described above, a direct solution without subdivision of the structure would have required the solution of 48 simultaneous equations. The great advantage of the method is now obvious when applied to large structures having many joints.

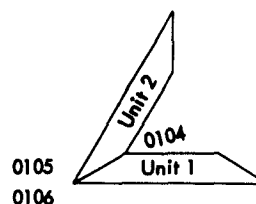


Fig. 5.

C. INPUT DATA

It is desirable to prepare the input data on format input data sheets before punching them onto cards. The restrictions of the STAIR ANALYSIS PROGRAM and the items of data required for each step of a problem are listed and discussed in the following sections.

Two decks of cards have to be prepared for the STAIR System: a structural data deck and a load data deck. These decks are handled differently during the running of the problem.

The structural data deck is input directly to the STAIR System. This deck begins with a block of cards punched from an "Initial Data Sheet." Subsequent blocks of cards correspond to consecutive steps in the ANALYSIS PROGRAM; the first card of each block calls the routine which executes the step, and the following cards provide the data needed for the execution.

The load data deck is input to the STAIR Load Program (SLOP). This program, which must be run before the ANALYSIS PROGRAM, receives and/or computes concentrated joint loads for various types of loading on the structure. The loads are assembled into groups of loads applied to individual units of the structure. SLOP is stored on tape for use by the STAIR System.

Each item of data entered on the format sheets is either an integer number (such as a quantity of joints or bars) or a fixed number (e.g., a joint coordinate or a bar area). The format sheets prescribe a field for each item of data. A field is the maximum number of characters an item may contain.

Integer numbers are always entered at the right of their field and are never written with a decimal point. Fixed numbers may be entered in one of two ways: with or without a decimal point. If the decimal point is specified as a character in the item of data, the number may be written anywhere within its field. If the decimal point is omitted, the number must be positioned about the dashed line in its field, with the fractional part to the right of the dashed line. The location of dashed lines is written into the machine program as indicated on the data sheets contained herein. If there is no dashed line in the field of a fixed number, it may be assumed at the right of the field.

Minus signs must always be specified for negative numbers and must always be written immediately before the first nonzero digit in the field. Plus signs need not be written, since any unsigned number is assumed positive.

The following examples illustrate these points:

<u>Integer Number</u>	<u>Format Sheet</u>	<u>Read by Computer</u>								
	<table><tr><td></td><td></td><td></td><td>1</td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>				1	1				+1
			1							
1										
	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>					1				+1000
1										
<u>Fixed Number</u>	<table><tr><td></td><td></td><td>1</td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>			1		1				+1.0
		1								
1										
	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>					1				+100.0
1										
	<table><tr><td></td><td></td><td>0</td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>			0		1				+1.0
		0								
1										
	<table><tr><td></td><td></td><td>1</td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>			1						-1.0
		1								
	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td>-</td><td>1</td><td></td><td>5</td></tr></table>					-	1		5	-1.5
-	1		5							
	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>1</td><td>5</td></tr></table>							1	5	+1.5
		1	5							
	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>1</td><td>5</td></tr></table>							1	5	+1.5
		1	5							

*

--	--	--	--

 is the field for the particular items whose maximum number of characters is four.

D. INPUT DATA FOR THE STAIR LOAD PROGRAM (SLOP)

The STAIR Load Program (SLOP) is written independently of the STAIR System. It enables one to analyze a structure subjected to more than one loading condition without repeating the complete process. The input data of SLOP is stored on tape and is ready to be called for analysis by the STAIR System. The data sheets, which are prepared by the structural analyst, are given on pp. 14 to 29.

Two types (INTP)[†] of loading are assembled by SLOP:

- Type (1) - automatically computed joint loads, e.g., dead load,
- Type (2) - manual input joint loads, e.g., live load.

Each loading condition may be Type (1) or Type (2) or a combination of both types. Each condition is identified by an NLD (loading condition number); if the same NLD number is given for both a dead and a live load condition, the analysis will be made for the combined effect of these loads. For example, on pp. 19 and 20, NLD = 2 is a combined dead and live condition, and NLD = 3 or 4 are conditions of live load only.

Loading condition Type (1) may be computed by SLOP for any orientation of the structure; the direction of the pull of gravity with respect to the coordinate system of the structure is specified by Gx, Gy and Gz, the direction cosines of the gravity vector. The coordinates of the center of gravity and the total weight of the structure are output from each dead load computation. The computed joint dead loads are stored on tape for the STAIR System.

The input data is composed of four groups of cards:

1. Initial data (refer to sheet A, p. 14).
2. Structural data (refer to sheet B, pp. 15 to 17).
3. Loading data (refer to sheet C, pp. 18 to 22).
 - (a) Dead load
 - (b) Live load

[†]See Loading Data, pp. 18 to 22.

4. Write-tape data (refer to sheet D, p. 23).

BARSOL data and some of the STRIX data can be obtained by duplicating the input data of Group 2 in the SLOP System. SLOP is stored on tape separately from the STAIR System, and structural data are required for dead load calculation and live load assembly; therefore, this input data must be repeated.

Initial Data (sheet A, p. 14)

- (a) Identification problem number (IDP), for example, 200.
- (b) Number of different loading conditions to which the structure is subjected (NK).
- (c) Number of units into which the structure is divided (NUT).
- (d) Number of total joints in the structure (NJT).*
- (e) Number of total bars in the structure (NBAR).*
- (f) Number of joints prefixed with each unit number (IUADR).† For example, each unit of the example structure in Fig. 2 has 9 joints, but only 6 joints are prefixed with the unit number, i.e., for unit 01, joints 0101 to 0106.

NUT

$$\sum_{i=1} (IUADR)_i = \text{Total number of joints in structure}$$

Structural Data (sheet B, pp. 15 to 17)

- (a) Joint card.
- (b) Bar card.

Loading Data

Dead Load (sheet C, p. 18).

- (a) INTP = 1. INTP is used to distinguish the following three blocks of cards: INTP = 1 for dead load input block; INTP = 2 for live load block; and INTP = 3 for write-tape input block, i.e., this instructs the computer to write loads on tape in the order listed on the cards.
- (b) Loading condition number (NLD).
- (c) Density of material used for the structure (DENS), in lbs. per cu. ft.
- (d) Direction cosines of the pull of gravity relative to the coordinate system used for structure (Gx, Gy and Gz).

Note: Items (a), (b), (c) and (d) are repeated for each loading condition containing dead load.

Live Load (sheet C, pp. 20 to 22).

- (a) INTP = 2.
- (b) Loading condition number (NLD).
- (c) Total number of joints loaded (NJLD).

*This is used to read in the correct number of joints and bars. NJT and NBAR can be zero if no dead load calculations are to be made.

†Joints are always consecutively numbered by the computer. However, if the programmer numbers joints 0101, 0103, 0105, 0107, then IUADR = 7, not 4. The displacement of joints 0102, 0104 and 0106 will all be zero.

- (d) Joint number with its joint load in components (Jt. No., Fx, Fy and Fz).

Note: Items (a), (b), (c) and (d) are repeated for each loading condition containing live load.

Write-Tape Data (sheet D, p. 23)

- (a) INTP = 3.
(b) Unit number of a unit (NUCK).
(c) Number of joints in the unit (NJU).
(d) Joint card giving the joint numbers contained in the unit in row sequence. Make sure the joint numbers are in the same sequence as the joint numbers in STRIX.

Note: Item (a) is used once, whereas items (b), (c) and (d) are repeated for each unit. List all units of the structure in sequence.

Size Restrictions

NK = number of loadings.
NUT = number of units.
NJT = total number of joints where a dead load is to be computed.
NJU = number of joints in any unit.
NB = number of bars input in this step.
IUAT = total number of joints in structure.

- (a) $NUT \leq 99$.
(b) $NB \leq 1800$.^{*}
(c) $NJT \leq 800$.[†]
(d) $3 \times IUAT \times NK \leq 19000$.
(e) $3 \times NJU \times NK \leq 2000$.

Output: Dead Load Computation

- (a) Coordinates of center of gravity.
(b) Total weight of structure.

Notes: 1. These outputs should be the same for any orientation of the structure, unless the density is changed.
2. The joint loads are assembled on tape A8 in binary form for input to STAIR.
3. If Sense Switch 3 on the computer console is down, the joint loads assembled by the program are stored on the output tape (B2) for listing (print out).

A listing of the input deck for SLOP for this problem can be found on p. 25. The on-line print out and the output of the SLOP are shown in pp. 27 to 29.

^{*}NB is a restriction on the number of bars whose weight can be calculated in any one SLOP run. It only applies if gravity loads are to be computed.

[†]NJT is a restriction on the number of joint coordinates that can be stored for any one SLOP run. It only applies if gravity loads are to be computed.

PROBLEM 200

PAGE 1/1

[illegible]

(16, 212, 214)

IUADR IN SEQUENCE OF UNIT NUMBER
(1012)

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA I

PROBLEM 200

PAGE 2/11

LIST JOINT COORDINATES
PUNCH ONE LINE PER CARD

(14,3F12.6)

JT. NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
0101										0												0																				
0102										0												-3464			160																	
0103										0												-3464			160																	
0104									-6													0																				
0105								-12														-3464			160																	
0106								-12														-3464			160																	
0201								-3														5196			152																	
0202								-6														6928			320																	
0203								-6														6928			320																	
0204								0														10392			305																	
0205								0														17132			0508																	
0206								0														17132			0508																	
0301								3														5196			152																	
0302								6														6928			320																	
0303								6														6928			320																	
0304								6														0																				
0305								12														-3464			160																	
0306								12														-3464			160																	

28 LINES

28 LINES

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA II

PROBLEM 200

PAGE 311

PUNCH ONE LINE
PER CARD

LIST BAR NUMBERS OF THE STRUCTURE (Z15, F12.6)																					
BAR NUMBER										AREA (SQ. IN)											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	0	1	0	1		0	1	0	2					3							
	0	1	0	1		0	1	0	3					3							
	0	1	0	1		0	1	0	4					3							
	0	1	0	1		0	3	0	4					3							
	0	1	0	2		0	1	0	3					3							
	0	1	0	2		0	1	0	4					3							
	0	1	0	2		0	1	0	5					3							
	0	1	0	2		0	1	0	6					3							
	0	1	0	2		0	3	0	4					3							
	0	1	0	2		0	3	0	6					3							
	0	1	0	2		0	1	0	4					3							
	0	1	0	2		0	1	0	5					3							
	0	1	0	2		0	3	0	4					3							
	0	1	0	2		0	3	0	5					3							
	0	1	0	2		0	3	0	6					3							
	0	1	0	4		0	1	0	5					3							
	0	1	0	4		0	1	0	6					3							
	0	1	0	4		0	2	0	1					3							
	0	1	0	4		0	2	0	2					3							
	0	1	0	4		0	2	0	3					3							
	0	1	0	5		0	1	0	6					3							
	0	1	0	5		0	2	0	3					3							
	0	1	0	6		0	2	0	2					3							
	0	1	0	6		0	2	0	3					3							
	0	2	0	1		0	2	0	2					3							
	0	2	0	1		0	2	0	3					3							
	0	2	0	1		0	2	0	4					3							
	0	2	0	2		0	2	0	3					3							
	0	2	0	2		0	2	0	4					3							
	0	2	0	2		0	2	0	5					3							
	0	2	0	2		0	2	0	6					3							
	0	2	0	3		0	2	0	4					3							
	0	2	0	3		0	2	0	5					3							
	0	2	0	4		0	2	0	5					3							
	0	2	0	4		0	2	0	6					3							
	0	2	0	4		0	3	0	1					3							
	0	2	0	4		0	3	0	2					3							
	0	2	0	4		0	3	0	3					3							
	0	2	0	5		0	2	0	6					3							
	0	2	0	5		0	2	0	8					3							

40 LINES

SLOP INPUT DATA SHEET 'B'
STRUCTURAL DATA II

PROBLEM 200

PAGE 4/11

**PUNCH ONE LINE
PER CARD**

BAR NUMBER										AREA (SQ. IN)											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	0	2	0	6		0	3	0	2						3						
	0	2	0	6		0	3	0	3						3						
	0	3	0	1		0	3	0	2						3						
	0	3	0	1		0	3	0	3						3						
	0	3	0	1		0	3	0	4						3						
	0	3	0	2		0	3	0	3						3						
	0	3	0	2		0	3	0	4						3						
	0	3	0	2		0	3	0	5						3						
	0	3	0	2		0	3	0	6						3						
	0	3	0	3		0	3	0	4						3						
	0	3	0	3		0	3	0	5						3						
	0	3	0	4		0	3	0	5						3						
	0	3	0	4		0	3	0	6						3						
	0	3	0	5		0	3	0	6						3						

40 LINES

LOADING ['A
CASE# / (Sh / of 5)

PROBLEM 200

PUNCH ONE LINE PER CARD

11 3
1 01
THINLD
TP
(11, 12)

67	1213	1819	2425	3031	3637	4243	48
1700				-1		0	
Density g/f3		GX		GY		GZ	(4F12.6)

Direction cosines of the gravity vector

SLOP INPUT DATA SHEET 'C'

PAGE 5/11

SLOP INPUT DATA SHEET 'C'

PAGE 6/11

LOADING I A
CASE 2 (Sh 2 of 5)

PROBLEM 200

PUNCH ONE LINE PER CARD

11 31
11 012
MINUB
TP (I1, I2)

67	1213	1819	2425	3031	3637	4243	48
1700				-1		0	
Density #/ft3		GX		GY		GZ	(4F12.6)

Direction cosines of the gravity vector

PAGE 7/11

PUNCH ONE LINE PER CARD

(I1, I2)

(15)

(I4,3F12.6)

F_z (KIPS)

F_v (KIPS)

F_u (KIPS)

५३.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
J1 N6:										X										Y										Z									

512



2020

52-

[illegible]

5

30

20 LINES

SLOP INPUT DATA SHEET 'C'
LOADING DATA
CASE #3 (Sh 4 of 5)

PROBLEM 200

PAGE 8/11

PUNCH ONE LINE PER CARD

(11, 12)

(15)

(14, 3F12.6)

JT No.	F _x (KIPS)	F _y (KIPS)	F _z (KIPS)
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0
30	0	0	0
31	0	0	0
32	0	0	0
33	0	0	0
34	0	0	0
35	0	0	0
36	0	0	0
37	0	0	0
38	0	0	0
39	0	0	0
40	0	0	0

20 LINES

SLOP INPUT DATA SHEET 'C'
LOADING DATA
CASE #4 (Sh 5 of 5)

PROBLEM 200

PAGE 9/11

PUNCH ONE LINE PER CARD

1 3
2 0 4
IN IN ID
IP

(11, 12)

1 5
0 0 0 0 1
NJID

(15)

(14, 3F12.6)

JT No.	F _x (KIPS)					F _y (KIPS)					F _z (KIPS)				
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
22	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
25	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
29	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
33	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
34	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
35	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
36	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
37	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
39	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
40	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

20 LINES

PROBLEM 200

PAGE 14/11

1 } needed only once at
3 } beginning of this
 } operation

(11)

**PUNCH ONE LINE
PER CARD**

INTP

2	4
0.1	0.9

(212)

NuckNJU

(715)

[illegible]

2	4
0.2	0.9

(212)

Nuck NJU

(715)

[illegible]

**NOTE: ORDER OF THE JOINTS MUST BE THE SAME
AS IN STRIX**

PROBLEM 200

PAGE 1411

NOTE: ORDER OF THE JOINTS MUST BE THE SAME
AS IN STRIX

200040300180054

60606

101	00	00	0
102	00	-3464102	20
103	00	-3464102	-20
104	-60	00	00
105	-120	-3464102	-20
106	-120	-3464102	20
201	-30	5196152	00
202	-60	6928203	20
203	-60	6928203	-20
204	00	10392305	00
205	00	17320508	-20
206	00	17320508	20
301	30	5196152	00
302	60	6928203	20
303	60	6928203	-20
304	60	00	00
305	120	-3464102	-20
306	120	-3464102	20
0101	0102	30	
0101	0103	30	
0101	0104	30	
0101	0304	30	
0102	0103	30	
0102	0104	30	
0102	0105	30	
0102	0106	30	
0102	0304	30	
0102	0306	30	
0103	0104	30	
0103	0105	30	
0103	0304	30	
0103	0305	30	
0103	0306	30	
0104	0105	30	
0104	0106	30	
0104	0201	30	
0104	0202	30	
0104	0203	30	
0105	0106	30	
0105	0203	30	
0106	0203	30	
0106	0202	30	
0201	0202	30	
0201	0203	30	
0201	0204	30	
0202	0203	30	
0202	0204	30	
0202	0205	30	
0202	0206	30	
0203	0204	30	
0203	0205	30	
0204	0205	30	
0204	0206	30	
0204	0301	30	
0204	0302	30	
0204	0303	30	
0205	0206	30	

0205	0303	30			
0206	0302	30			
0206	0303	30			
0301	0302	30			
0301	0303	30			
0301	0304	30			
0302	0303	30			
0302	0304	30			
0302	0305	30			
0302	0306	30			
0303	0304	30			
0303	0305	30			
0304	0305	30			
0304	0306	30			
0305	0306	30			
101					
1700		0	-1		0
102					
1700		0	-1		0
202					
0002					
202	5	0		75	
303	5	0		-75	
203					
0005					
103	0	-5		0	
202	5	0		0	
204	0	-10		0	
206	0	0		-75	
303	5	0		1	
204					
0001					
101	1	2		3	
3					
109					
0101	0102	0103	0104	0105	0106
0305	0306				
209					
0201	0202	0203	0204	0205	0206
0105	0106				
309					
0301	0302	0303	0304	0305	0306
0205	0206				

M504 STAIR- SLOP2 LOAD ASSEMBLY FOR PROBLEM 200
FOR INPUT BLOCK 1, TOTAL WEIGHT OF STRUCTURE IS 1.536 KIPS
COMPONENTS OF THE GRAVITY VECTOR ARE 0. -1.00000 0.
COORDINATES OF THE CENTER OF GRAVITY ARE 0.00000 3.46410 -0.00000
M504 STAIR- SLOP2 LOAD ASSEMBLY FOR PROBLEM 200
FOR INPUT BLOCK 2, TOTAL WEIGHT OF STRUCTURE IS 1.536 KIPS
COMPONENTS OF THE GRAVITY VECTOR ARE 0. -1.00000 0.
COORDINATES OF THE CENTER OF GRAVITY ARE 0.00000 3.46410 -0.00000
M504 STAIR- SLOP2 LOAD ASSEMBLY FOR PROBLEM 200
COMPLETE

M504 STAIR- SLOP2 LOAD ASSEMBLY FOR PROBLEM 200

LOADS FOR UNIT 1, LOADING 1

101	0.	-0.035417	0.
102	0.	-0.104605	0.
103	0.	-0.104605	0.
104	0.	-0.097868	0.
105	0.	-0.084752	0.
106	0.	-0.084752	0.
304	0.	-0.097868	0.
305	0.	-0.084752	0.
306	0.	-0.084752	0.

LOADS FOR UNIT 1, LOADING 2

101	0.	-0.035417	0.
102	0.	-0.104605	0.
103	0.	-0.104605	0.
104	0.	-0.097868	0.
105	0.	-0.084752	0.
106	0.	-0.084752	0.
304	0.	-0.097868	0.
305	0.	-0.084752	0.
306	0.	-0.084752	0.

LOADS FOR UNIT 1, LOADING 3

101	0.	0.	0.
102	0.	0.	0.
103	0.	-5.000000	0.
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.

LOADS FOR UNIT 1, LOADING 4

101	1.000000	2.000000	3.000000
102	0.	0.	0.
103	0.	0.	0.
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.

LOADS FOR UNIT 2, LOADING 1

201	0.	-0.035417	0.
202	0.	-0.104605	0.
203	0.	-0.104605	0.
204	0.	-0.097868	0.
205	0.	-0.084752	0.
206	0.	-0.084752	0.
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.

LOADS FOR UNIT 2, LOADING 2

201	0.	-0.035417	0.
202	5.000000	-0.104605	7.500000
203	0.	-0.104605	0.
204	0.	-0.097868	0.
205	0.	-0.084752	0.
206	0.	-0.084752	0.
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.

LOADS FOR UNIT 2, LOADING 3

201	0.	0.	0.
202	5.000000	0.	0.
203	0.	0.	0.
204	0.	-10.000000	0.
205	0.	0.	0.
206	0.	0.	-7.500000
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.

LOADS FOR UNIT 2, LOADING 4

201	0.	0.	0.
202	0.	0.	0.
203	0.	0.	0.
204	0.	0.	0.
205	0.	0.	0.
206	0.	0.	0.
104	0.	0.	0.
105	0.	0.	0.
106	0.	0.	0.

LOADS FOR UNIT 3, LOADING 1

301	0.	-0.035417	0.
302	0.	-0.104605	0.
303	0.	-0.104605	0.
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.
204	0.	0.	0.
205	0.	0.	0.
206	0.	0.	0.

LOADS FOR UNIT 3, LOADING 2

301	0.	-0.035417	0.
302	0.	-0.104605	0.
303	5.000000	-0.104605	-7.500000
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.
204	0.	0.	0.
205	0.	0.	0.
206	0.	0.	0.

LOADS FOR UNIT 3, LOADING 3

301	0.	0.	0.
302	0.	0.	0.
303	5.000000	0.	1.000000
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.
204	0.	0.	0.
205	0.	0.	0.
206	0.	0.	0.

LOADS FOR UNIT 3, LOADING 4

301	0.	0.	0.
302	0.	0.	0.
303	0.	0.	0.
304	0.	0.	0.
305	0.	0.	0.
306	0.	0.	0.
204	0.	0.	0.
205	0.	0.	0.
206	0.	0.	0.

E. THE STAIR OPERATIONS

The STAIR System is divided into ten different operations:

Code No. and Name*	Operation
1 STRIX	Forms a <u>stiffness matrix</u> from an input of joint coordinates, bar areas and elastic modulus.
2 EFFRIX	<u>Eliminates</u> a joint or a group of joints from a <u>stiffness matrix</u> equation and the row sequence which identifies the stiffness matrix.
3 MATRO	Transforms the coordinates of a stiffness matrix to give a new <u>matrix</u> of another identical unit. This transformation consists of the <u>rotation</u> of two axes about the third axis.
4 MADD	(<u>Matrix addition</u>) Superimposes two stiffness matrices: one in the computer core and the other on tape.
5 MATOUT	(<u>Matrix out</u>) Transfers a matrix from the operating core of the computer to magnetic tape.
6 MATIN	(<u>Matrix in</u>) Transfers a matrix stored on magnetic tape back into the core of the computer.
7 DEFSOL	(<u>Deflection solution</u>) Computes joint displacements from a stiffness matrix which completely represents the structure under a given set of loading conditions.
8 BARSOL	(<u>Bar stress solution</u>) Computes the stress in any bar from the displacements of its end.
9 CHECK	Sums all the bar forces at each joint in each coordinate direction.
10 STOP	Ends the analysis sequence.

*Whenever an operation is given on a data sheet, both the code number and name must appear.

Any three-dimensional pin-jointed structure can be analyzed by certain combinations of these operations. Such an operation sequence is called an ANALYSIS PROGRAM. The particular ANALYSIS PROGRAM used should perform the required analysis in a minimum of computer time without exceeding the size limitation on matrices imposed by the computer capacity. The computer time required for each operation is different. For the details of computer time accumulation, an experienced programmer should be consulted.

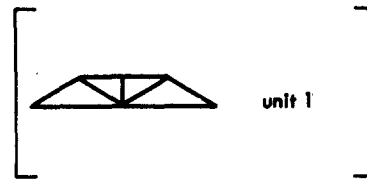
The size of a stiffness matrix in operation is limited by the capacity of the computer used. The size of matrix which represents a structure increases with the square of three times the number of joints; the largest number of joints which can be handled by an ANALYSIS PROGRAM at any time in the IBM-7090 computer is 59. This is the main reason for dividing the structure into units and is also why EFFRIX is used to eliminate the joints (or reduce the size of matrix) as soon as possible in the ANALYSIS PROGRAM.

Several other parameters are limited in magnitude in the STAIR System. These limitations are listed in the next section. The STAIR System detects violations of these restrictions and prints an indication of the error.

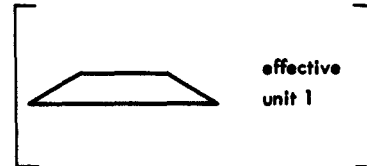
A STAIR ANALYSIS PROGRAM SHEET with a routine to analyze the example structure in Fig. 2 is presented. This sheet should be completed before the formation of input data for each step of the analysis. The following symbols are used in this description:

- [] contained in computer core.
- () contained in tape location - 1.
- (()) contained in tape location - 2.

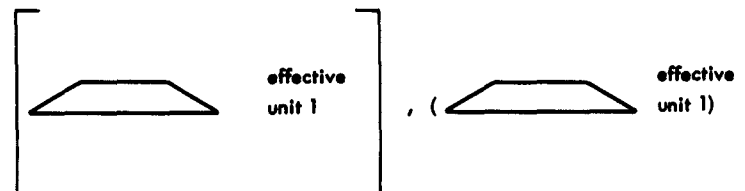
Step 1 – STRIX forms the stiffness matrix for unit 01 which is identified by row sequence 0101 – 0106, 0304 – 0306.



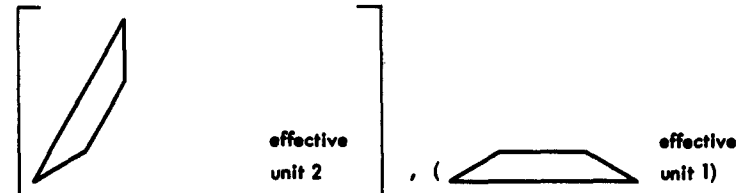
Step 2 – EFFRIX eliminates joints 0101 – 0103 from unit 1.



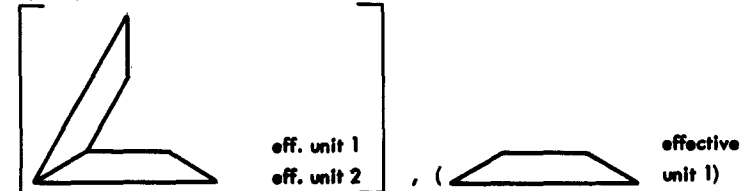
Step 3 – MATOUT transfers the matrix in core to magnetic tape loc. 1.



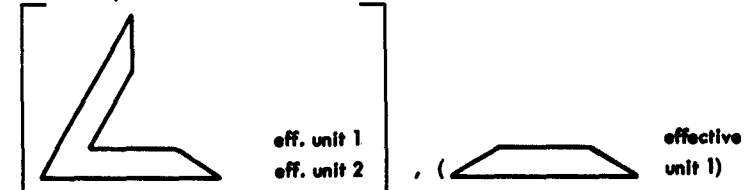
Step 4 – MATRO transforms the matrix in core into matrix for unit 2.



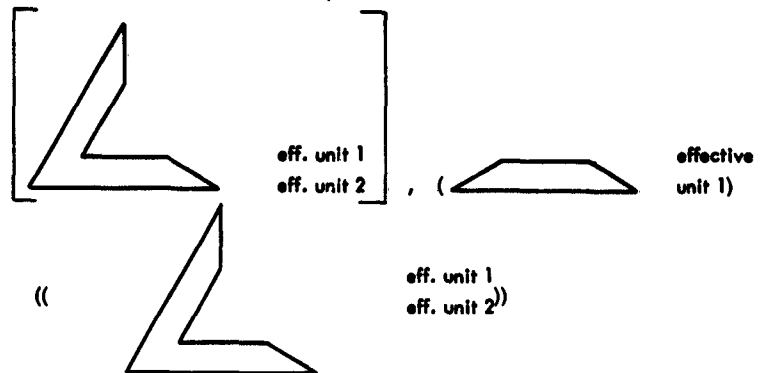
Step 5 – MADD superimposes the matrix in core and matrix in tape loc. 1.



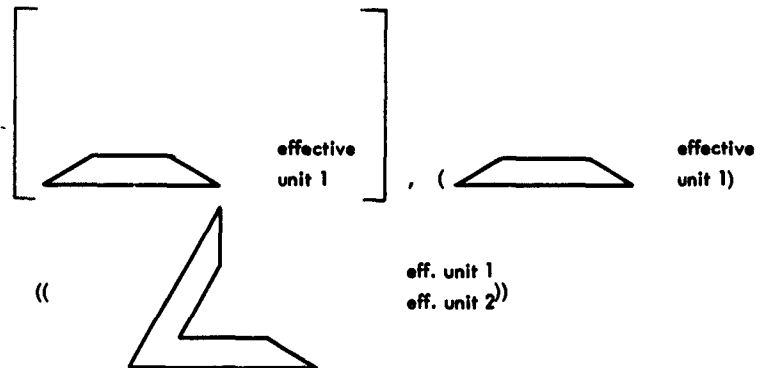
Step 6 – EFFRIX eliminates joint 0104 – 0106 from step 5.



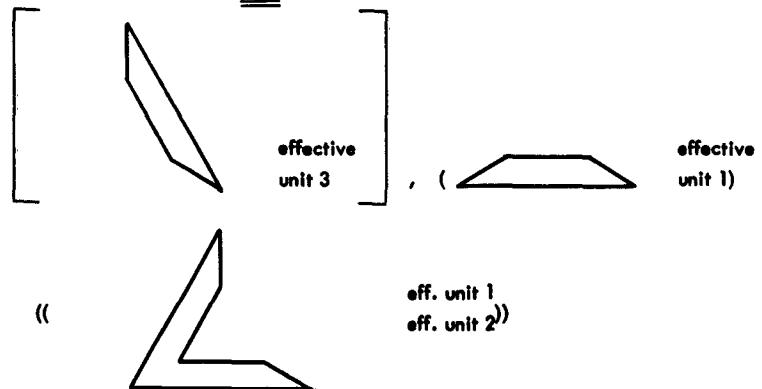
Step 7 - MATOUT transfers the matrix in core to tape location 2.



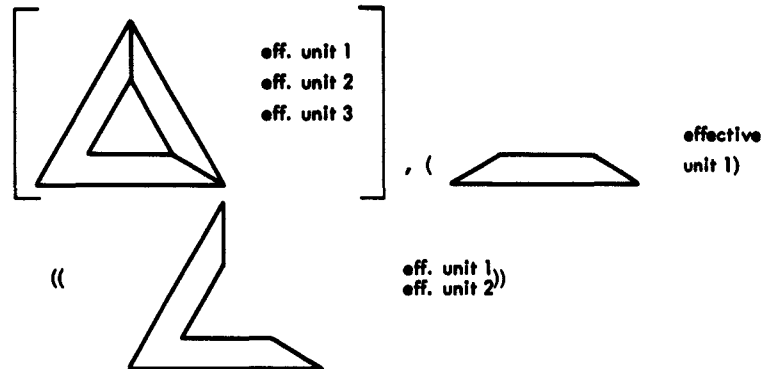
Step 8 - MATIN transfers the matrix in tape location 1 to the core.



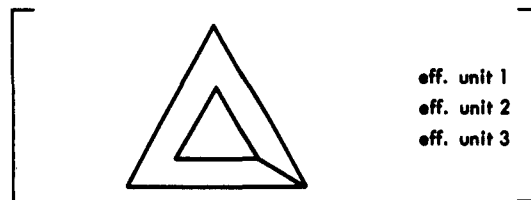
Step 9 - MATRO transforms the matrix in core into matrix for unit 3.



Step 10 – MADD superimposes the matrix in core and the matrix in tape location 2 in the row sequence 0204 – 0206, 0304 – 0306.



Step 11 – EFFRIX eliminates the joint 0204 – 0206 from Step 10.



Step 12 – DEFSOL computes the joint displacements of all joints and reactions.

Step 13 – BARSOL computes the bar stresses.

Step 14 – CHECK gives summation of internal forces at each joint.

Step 15 – STOP ends the analysis program.

F. INPUT DATA FOR THE STAIR SYSTEM

The input parameters required by each step of the STAIR operations are described below. The restrictions on the magnitude of these parameters and operations are given where applicable. The formats in which these data parameters are prepared will be illustrated in the following data sheets. Data sheets for each step are arranged in the same order as the example for the ANALYSIS PROGRAM. The same example structure of Fig. 2 is used.

Initial Input (refer to data sheet, p. 44)

- (a) Problem identification number (IDP).
- (b) Number of units into which the given structure is divided (NUT).
- (c) Number of different loading conditions to which this given structure is subjected (NK).
- (d) Number of reaction joints or supports in structure.
- (e) Elastic modulus of bar material E kips per sq. in.
- (f) List of reaction joint numbers and the coordinate directions in which displacement is restricted (1-immovable; 0-free).

Size Restrictions

- (a) Number of units ≤ 99 .
- (b) Number of reaction joints ≤ 125 .
- (c) Loading conditions ≤ 28 .

STRIX (refer to data sheet, p. 45)

- (a) Step number in ANALYSIS PROGRAM. It precedes every operation code, as the 001 on p. 45 precedes 01 STRIX, and the 005 precedes 04 MADD on p. 49. It is the first card of a card block calling the routine which executes the step as mentioned in the section on input data. (Step numbers must start at 01 and proceed sequentially to the end.)
- (b) Identification number of the unit (NUU) to which this STRIX operation applies. On p. 45 NUU is 01; later, units 02 and 03 will be formed by MATRO. In case some of the units are not identical, NUU is used to identify the different STRIX operations for different units when STRIX is called more than once.
- (c) Number of joints in the unit (NJTU).
- (d) Number of bars in the unit (NMU).
- (e) Joint sequence. This serves as a reminder only, and is not a part of the input data.
- (f) Joint cards giving the joint numbers [see Operation Restriction (b)] contained in the unit in sequence* and the coordinates of the joints in feet (reproduced from SLOP).
- (g) Bar cards are reproduced from SLOP. Bars which are common to two units are listed separately for this input (see p. 46).

*The sequence can be of any convenient form, but the joints that are to be eliminated by EFFRIX must be placed first in the sequence.

Size Restrictions

If $N = 3 \times$ (number of joints in the unit), $NK =$ number of loading conditions.

(a) $\frac{Nx(N+1)}{2} + N \times NK \leq 16000$ (see table).

N	NK
59	1
58	4
57	7
56	10
55	13
54	17
53	20
52	24
51	27

(b) Number of bars ≤ 600 .

(c) Unit number $\leq$ total number of units in the structure.

Operation Restrictions

- (a) In no case may all the bars meeting at a joint lie in one plane unless the joint is a reaction joint. If a planar joint exists, a new bar out of the plane (an imaginary member) may be inserted which will carry no stress but will allow the analysis to proceed.* The new bar may connect the planar joint with any other joint not lying in the plane.
- (b) If an EFFRIX operation immediately follows STRIX in the sequence of the ANALYSIS PROGRAM, those joints which are going to be eliminated have to be placed first in the row of joint cards. In the example, joints 0101, 0102 and 0103 are eliminated by the following EFFRIX.

EFFRIX (refer to p. 47)

- (a) Step number in ANALYSIS PROGRAM.
- (b) Number of joints (NE) to be eliminated from the row sequence established by the previous operation (STRIX or MADD).

Size Restrictions

NE = number of joints to be eliminated.

NJ = number of joints in original row sequence (number of joints in the unit).

NR = number of reaction joints among joints being eliminated.

NC = number of reaction components acting at the eliminated reaction joints.

* See p. 41 for a more detailed explanation of a planar joint.

NK = number of different loading conditions.

$$A = 3NE - NC.$$

$$B = 3(NJ - NE).$$

$$C = 3NE.$$

- (a) $NR \leq 10.$
- (b) $\frac{A(A+1)}{2} + A(B + NK) \leq 9500.$
- (c) $A^2 < [C \times \frac{C+1}{2} + C \times B].$
- (d) $\frac{B(B+1)}{2} < [C \times \frac{C+1}{2} + C \times B].$

MATOUT (refer to p. 47)

- (a) Step number.
- (b) Tape location (Loc.) in which a matrix is to be stored. Matrices are stored on tape in blocks. These are numbered consecutively, the tape location being the number given to the block.

Operation Restriction

Only one matrix may be stored in a tape location.

MATRO (refer to p. 48)

- (a) Step number.
- (b) Identification number of new unit (NUN).
- (c) Angle through which old unit* must be rotated in the x-y plane to arrive at the position of the new unit. Angle (EPS) measured in degrees and positive when clockwise, as viewed by an observer on the positive z-axis. In the STAIR system, the rotation must be about the z-axis.
- (d) Row sequence of the resulting matrix after rotation [NROW (1) to NROW (NJT)].
- (e) If the matrix being rotated is a reduced matrix, i.e., some joints have been eliminated from it by EFFRIX in a previous step, the inverse matrix corresponding to these eliminated joints is stored in the secondary matrix storage - then, not otherwise. The new row sequence of the eliminated joints, i.e., the new row sequence for the new inverse matrix, is NROSEC (1) to NROSEC (NE).

Size Restriction

- (a) New unit number $\leq$ number of units in the structure.

Operation Restriction

- (a) Matrices of identical units from which joints have been eliminated may be formed by MATRO except when one of the eliminated joints of the old unit has a corresponding joint in the new unit which is a reaction joint, or vice-versa.
- (b) MATRO can not follow a double EFFRIX sequence.

* Matrix of unit in core.

MADD (refer to p. 49)

- (a) Step number.
- (b) Number of joints in the resulting structure (NF) not including those previously eliminated.
- (c) Tape location (Loc.) in which a matrix has been stored by MATOUT. The matrix in this tape location is superimposed on the matrix in the computer core by this operation.
- (d) List of row sequence of the resulting matrix. Caution should be taken in setting up the row sequence if EFFRIX is planned to follow the MADD operation. Those joints which are being eliminated in EFFRIX should be placed first in the row sequence.

Size Restriction

NF = number of joints in resulting structure.

NA = larger number of joints in either unit being added.

NK = number of loading conditions.

(a) $3NF(3NF + 1)/2 + 3NF \times NK \leq 16000$ (see table).

(b) $3NA(3NA + 1)/2 + 3NA \times NK \leq 9500$ (see table).

(c) $NA < 45$.

NF	NK	NA	NK
59	1	45	2
58	4	44	5
57	7	43	8
56	10	42	11
55	13	41	15
54	17	40	18
53	20	39	22
52	24	38	25
51	27	37	28

EFFRIX (refer to p. 50, similar to Step 2)

MATOUT (refer to p. 50, similar to Step 3)

MATIN (refer to p. 50)

- (a) Step number.
- (b) Tape location (Loc.) of the matrix to be read into computer core.

Operation Restriction

None.

MATRO (refer to p. 51, similar to Step 4)

MADD (refer to p. 52, similar to Step 5)

EFFRIX (refer to p. 53, similar to Step 2)

DEFSOL (refer to p. 54)

- (a) Step number.
- (b) List of the number of joints which are prefixed by each unit number (IUADR). For example, there are 9 joints in unit 01, but only 6 joints are prefixed by unit number 01. (Notes regarding this in SLOP apply here also.)

Size Restrictions

NJ = number of joints in final row sequence.
NR = number of reaction joints among final joints.
NC = number of reaction components among final joints.
NK = number of loading conditions.
NJT = total number of joints in structure.
NRT = total number of reaction joints in structure.
 $A = 3NJ - NC.$

- (a) $NR \leq 10.$
- (b) $A \times (A + 1)/2 + A \times NK \leq 9408.$
- (c) $A^2 < 16,000.$
- (d) $4 \cdot NK(NJT + NRT) < 16,000.$

Operation Restriction

DEFSOL may be called only when the final matrix of the structure is stored in the computer core.

Output

- (a) Joint displacement (in inches).
- (b) Reactions (in kips).

BARSOL (refer to p. 54)

- (a) Step number.
- (b) Number of bars whose stresses are to be computed (NM).
- (c) Number of joints to which the bars listed in (b) are connected (NJ).
- (d) List of bar numbers of the bars whose stresses are to be computed (reproduced from SLOP).

Note: STAIR will compute any desired number of bar stresses by repeated use of BARSOL.

- (e) Joint cards, giving the joint numbers and coordinates. Only those joints which appear in the list of bar numbers (p. 54) need be included (reproduced from SLOP).

Size Restrictions

NK = number of loading conditions.

- (a) $NJ \leq 800.$
- (b) $NM \leq 1000.$
- (c) $NK \times NM \leq 4300.$

Operation Restriction

BARSOL may be called only after the DEFSOL operation.

Output

(a) Bar stresses (kips per sq. in.) for all loading conditions.

CHECK (refer to p.55)

- (a) Step number.
- (b) Total number of bars in structure.
- (c) Total number of joints in structure.

Size Restriction

None.

Output

Sum of all bar forces at each joint (in kips) for all loading conditions.

STOP (refer to p.55)

- (a) Step number.

Notes on Input Data Units

The above description assumed that the input data are as follows:

Joint coordinates (x, y, z) (in ft)
Bar areas (A) (in sq. in.)
Modulus of elasticity (E) (in kips per sq. in.)
Loads (F) (in kips)

The results are given as follows:

Deflections (in in.)
Reactions (in kips)
Bar stresses (in kips per sq. in.)
Equilibrium check (in kips)

The input data could be in any units, but in such a case, the output would be in terms of the following units:

The deflections would be in the units of $\frac{FL}{AE}$, and bar stresses would be in the units of E.

Reactions would be in terms of F, and equilibrium check in terms of F.

A listing of the input cards for the ANALYSIS PROGRAM is shown on p. 56, and the output from a successful run for the sample problem is shown on p.59.

G. STAIR SYSTEM ERROR STOPS

The STAIR system detects several errors, in addition to violation of size restrictions. When an error is detected, the problem number and the step of the ANALYSIS PROGRAM at which the error occurs are printed by the computer, as well as a diagnosis of the trouble.

Some error stops may be bypassed by setting Sense Switch 1 down; this may be necessary under unusual circumstances, but care should be exercised. Depressing Sense Switch 1 on the machine console causes the following error stops to be bypassed:

- STRIX error (a) (see below),
- SLOP errors (a), (b) and (d) (see below).

In each case, the error diagnosis is printed before the program continues.

Some errors are not caused by mistakes in input, but are machine errors. A common complaint arises from the malfunctioning of the tape units. When machine error is suspected, a qualified programmer should be consulted before rerunning the problem.

Error diagnoses written by the various STAIR routines are given below.

MISLAM*

A FORTRAN II MISLAM diagnosis indicates a machine error in reading the STAIR system tape. It may be necessary to rewrite this tape from the binary card deck containing all the STAIR routines.

CONTROL

- (a) Illegal subroutine code used; incorrect data for a preceding step or a misspunched call card.
- (b) Input data file out of order; a wrong step number on a call card or incorrect data for a preceding step.
- (c) Size limitations are exceeded; mistake on data sheet.
- (d) Program pause; used for program testing, probably indicates an illegal subroutine code [see (a) above].

STRIX

- (a) The following joints are planar joints; either the wrong coordinate data are supplied or the bars at the joints listed lie in a plane.† If there is no load acting normal to the plane of the bars, a new bar may be inserted which will carry no stress but which will allow the analysis to proceed.
- (b) Space limitations have been exceeded; check input parameters.
- (c) Joint N is used in a bar number, but is not in the row sequence; check bar number and row sequence.
- (d) Loads can not be found; tape error. (Tape A8 should contain the loads.)

*See Chapter 4 for description of MISLAM.

†The check for a planar joint is performed in the following manner: Let L_1 = unit vector in direction of bar 1; L_2 = unit vector in direction of bar 2; L = unit vector in direction of any other bar. First, the vector product $C = L_1 \times L_2$ is executed. If each component of C is less than 0.000001, the bars are considered collinear, and the unit vector associated with the next bar is used as L_2 to form a new C . Then, the scalar product $C \cdot L$ is executed. If $|C \cdot L| < 0.00001$ for every remaining bar, the joint is considered planar.

EFFRIX

- (a) Space limitations have been exceeded; check input parameters.
- (b) Computer stops without diagnosis; possibly the matrix is singular (unstable structure). Check the previous STRIX stops for planar joints now being EFFRIXed.

MATRO

- (a) An attempt has been made to rotate the effective matrix of a unit with internal reaction joints; the ANALYSIS PROGRAM must be changed. Either the old or the new unit contains reactions at the joints which were eliminated.
- (b) The number of the new unit is wrong; check input to this step and the total number of units as given in the initial input block.
- (c) Loads can not be found; tape error (see STRIX).

MADD

- (a) Space limitations have been exceeded; check input parameters.
- (b) Joint N is not included in the final row sequence; either the final row sequence input at this step is wrong, or the row sequence of one of the matrices being added is wrong.
- (c) Matrix on tape can not be found; tape error (check location number).

MATIN or MATOUT

- (a) Record group can not be found; tape error.

DEFSOL

- (a) Space limitations are exceeded; check input.
- (b) No diagnosis; see EFFRIX and previous comments on machine errors.

BARSOL

- (a) Space limitations have been exceeded; check input.
- (b) Joint coordinate not defined for this bar, the coordinates of one end of the bar were not supplied, the bar number is illegal or a joint number in the input or coordinates is illegal. (This error is printed in the output, but does not stop the program.)

SLOP

- (a) Size limitations are exceeded; check input.
- (b) Illegal joint numbers or bar numbers (list); error in the input of joint coordinates, bar numbers, loads or unit row sequences.
- (c) Illegal input heading; an input-type code has been read which is not 1, 2 or 3.
- (d) Joint list out of consecutive order; row sequence of units is out of order or joint number is wrong.

Illegal joint numbers in BARSOL and SLOP are detected when the first two digits of a joint number are zero or greater than the number of units, or when the last two digits are greater than the number of joints in the unit which contains the joint.

STAIR ANALYSIS PROGRAM SHEET

PROBLEM 200

PAGE 1/13

[illegible]

STAIR INITIAL DATA SHEET

PROBLEM 200

PAGE 2/13

PUNCH ONE LINE PER CARD

6			8		10		13		19			25			(I6, 2I2, I3, F12.6)					
0	0	0	2	0	0	0	3	0	4	0	0	3	1	0		0	0	0	0	0
Problem No.			No. of Units		No. of Loads		No. of React.		No. of Joints		E (KIPS per sq. in.)									

[4(15, 311)]

JOINT				X	Y	Z	JOINT				X	Y	Z	JOINT				X	Y	Z	JOINT				X	Y	Z					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
0	1	0	6	0	1	1				0	3	0	5	0	1	0				0	3	0	6	1	1	1						

32 LINES

STAIR STRIX DATA SHEET 'A'

PROBLEM 200

PAGE 3 / 13

0	0	1	0	1	S	T	R	I	X
Step No.		Operation							

(13,12,64)

**PUNCH AS FIRST TWO
CARDS OF STEP ONLY**

	2	4	7
0.1	0.9	0.2	1
NUU	NJTU	NMU	

(212, 13)

**JOINT CARDS TO BE REPRODUCED (For one unit of each type)
FROM THE SLOP JOINT CARDS.**

THE SEQUENCE OF STRIX CARDS MUST BE AS FOLLOWS FOR THE DESIRED FORMATION OF THE STIFFNESS MATRIX.

[illegible]

DO NOT PUNCH

STAIR EFFRIX DATA SHEET

PROBLEM 200

PAGE 5/13

PUNCH ONE LINE PER CARD

3	5	11
002	02	EFFRIX
Step No	Operation	

(I3, I2, 6H)

2
02
NE

(I2)

STAIR MATOUT DATA SHEET

PROBLEM 200

PAGE /

PUNCH ONE LINE PER CARD

3	5	11
003	05	MATOUT
Step No	Operation	

(I3, I2, 6H)

2
01
Loc.

(I2)

STAIR MATIN DATA SHEET

PROBLEM

PAGE /

PUNCH ONE LINE PER CARD

3	5	11
	06	MATIN
Step No	Operation	

(I3, I2, 6H)

2
Loc.

(I2)

	3	5	11
004	03	MATR	0
Step No		Operation	

2	5	12
0.2	12.00	
NUN	EPS	

[illegible][illegible]

48

STAIR MADD DATA SHEET

PROBLEM 200

PAGE 7/23

PUNCH ONE LINE PER CARD

	3	5		11					
0	0	5	0	4	M	A	D	D	
Step No.	Operation								

(13, 12, 6H)

2	4
0901	
NF	Loc.

(212)

(715)

[illegible]

List row sequence of resulting matrix. Punch row-wise one line per card.

STAIR EFFRIX DATA SHEET

PROBLEM 200

PAGE 8/13

PUNCH ONE LINE PER CARD										
3		5		11						
0	0	6	0	2	E	F	F	R	I	X
Step No				Operation						
2		(12)								
0	3	NE								

STAIR MATOUT DATA SHEET

PROBLEM 200

PAGE /

PUNCH ONE LINE PER CARD										
3		5		11						
0	0	7	0	5	M	A	T	O	U	T
Step No				Operation						
2		(12)								
0	2	Loc.								

STAIR MATIN DATA SHEET

PROBLEM 200

PAGE /

PUNCH ONE LINE PER CARD										
3		5		11						
0	0	8	0	6	M	A	T	I	N	
Step No				Operation						
2		(12)								
0	1	Loc.								

PUNCH ONE LINE PER CARD

3	5	11
00903	MATR	0
Step No	Operation	

(I3, I2, 6H)

2	5	12
0.3240	0	
NUN	EPS	

(I2, F10.7)

[illegible]

NROW - Row sequence for the resulting matrix **NROW (I)**
to **NROW (NJ T)**

[illegible]

NROSEC - Row sequence for the new inverse matrix
NROSEC (1) to NROSEC (NE)

STAIR MADD DATA SHEET

PROBLEM

PAGE 10/13

PUNCH ONE LINE PER CARD

010	04	M	A	D	D
Step No.		Operation			

(13, 12, 6H)

2	4
06	02
NF	Loc.

(212)

(715)

[illegible]

List row sequence of resulting matrix. Punch row-wise one line per card.

STAIR EFFRIX DATA SHEET

PROBLEM 200

PAGE 11/13

PUNCH ONE LINE PER CARD

3	5	11
0 1 1	0 2	E F F R I X
Step No	Operation	
2	(I2)	
0 3		
NE		

STAIR MATOUT DATA SHEET

PROBLEM

PAGE /

PUNCH ONE LINE PER CARD

3	5	11
	0 5	M A T O U T
Step No	Operation	
2	(I2)	
Loc.		

STAIR MATIN DATA SHEET

PROBLEM

PAGE /

PUNCH ONE LINE PER CARD

3	5	11
	0 6	M A T I N
Step No	Operation	
2	(I2)	
Loc.		

STAIR CHECK DATA SHEET

PROBLEM 200

PAGE 13/13

PUNCH ONE LINE PER CARD

3	5	11
0	1	4
0	9	C
H	E	C
K		
Step No.	Operation	

(I3, I2, 6H)

4	8
0	0
5	4
0	0
1	8
No. of Bars	No. of JTS

STAIR STOP DATA SHEET

PROBLEM 200

PAGE /

PUNCH ONE LINE PER CARD

3	5	11
0	1	5
1	0	S
T	0	P
Step No.	Operation	

(I3, I2, 6H)

```

0002000304003 100000
0106011 0305010 0306111
00101STRIX
0109021
101      00      00      00
102      00      -3464102    20
103      00      -3464102   -20
104     -60      00      00
105    -120     -3464102   -20
106    -120     -3464102    20
304      60      00      00
305     120     -3464102   -20
306     120     -3464102    20
0101 0102      30
0101 0103      30
0101 0104      30
0101 0304      30
0102 0103      30
0102 0104      30
0102 0105      30
0102 0106      30
0102 0304      30
0102 0306      30
0103 0104      30
0103 0105      30
0103 0304      30
0103 0305      30
0103 0306      30
0104 0105      15
0104 0106      15
0105 0106      15
0304 0305      15
0304 0306      15
0305 0306      15
00202EFFRIX
03
00305MATOUT
01
00403MATRO
021200
0204 0205 0206 0104 0105 0106
0201 0202 0203
00504MADD
0901
0104 0105 0106 0204 0205 0206 0304
0305 0306
00602EFFRIX
03
00705MATOUT
02
00806MATIN
01
00903MATRO
032400
0304 0305 0306 0204 0205 0206
0301 0302 0303
01004MADD
0602
0204 0205 0206 0304 0305 0306
01102EFFRIX
03

```

01207DEFSOL

060606

013080ARSOL

05400018

0101	0102	30
0101	0103	30
0101	0104	30
0101	0304	30
0102	0103	30
0102	0104	30
0102	0105	30
0102	0106	30
0102	0304	30
0102	0306	30
0103	0104	30
0103	0105	30
0103	0304	30
0103	0305	30
0103	0306	30
0104	0105	30
0104	0106	30
0104	0201	30
0104	0202	30
0201	0204	30
0202	0203	30
0202	0204	30
0202	0205	30
0203	0204	30
0203	0205	30
0204	0205	30
0204	0206	30
0204	0301	30
0204	0302	30
0204	0303	30
0205	0206	30
0205	0303	30
0206	0302	30
0206	0303	30
0301	0302	30
0301	0303	30
0301	0304	30
0302	0303	30
0104	0203	30
0105	0106	30
0105	0203	30
0106	0203	30
0106	0202	30
0201	0202	30
0201	0203	30
0302	0304	30
0302	0305	30
0302	0306	30
0303	0304	30
0303	0305	30
0304	0305	30
0304	0306	30
0305	0306	30
0202	0206	30

101	00
102	00
103	00

00
-3464102
-3464102

104	-60	00	00
105	-120	-3464102	-20
106	-120	-3464102	20
201	-30	5196152	00
202	-60	6928203	20
203	-60	6928203	-20
204	00	10392305	00
205	00	17320508	-20
206	00	17320508	20
301	30	5196152	00
302	60	6928203	20
303	60	6928203	-20
304	60	00	00
305	120	-3464102	-20
306	120	-3464102	20
01409CHECK			
00540018			
01510STOP			

M504, STAIR, STRIX	COMPLETED, STEP 1	379	358	378	0	21	9	2	4
M504, STAIR, EFFRIX	COMPLETED, STEP 2	9	9	18	1	0	468	12	0
		45	244	46	225	10	10	19	19
M504, STAIR, MATOUT	COMPLETED, STEP 3								
M504, STAIR, MATRO	COMPLETED, STEP 4	46	188	207	46	7	4	280	244
		244	10	19	18	9			
M504, STAIR, MADD	COMPLETED, STEP 5	9	0	9	9	5	169	171	373
		459	225	483	240	1			
M504, STAIR, EFFRIX	COMPLETED, STEP 6	9	7	18	5	27	468	12	378
		28	244	29	172	8	8	19	19
M504, STAIR, MATOUT	COMPLETED, STEP 7								
M504, STAIR, MATIN	COMPLETED, STEP 8								
M504, STAIR, MATRO	COMPLETED, STEP 9	46	188	207	46	7	4	280	244
		244	10	19	18	9			
M504, STAIR, MADD	COMPLETED, STEP 10	6	0	6	6	5	169	171	166
		225	225	240	240	1			
M504, STAIR, EFFRIX	COMPLETED, STEP 11	9	9	9	1	0	234	9	0
		45	82	46	135	10	10	10	10
M504, STAIR, DEFSQL	COMPLETED, STEP 12	0	1	243	70	28	216	45	234
		228							
M504, STAIR, BARSQL	COMPLETED, STEP 13	1	216	245	261	216	217	220	4
M504, STAIR, CHECK	COMPLETED, STEP 14	0	0	245	261	0	0	0	4
PROBLEM	200 COMPLETED								

PLEASE PRINT B2

1 1STRIX
 1 1STRIX
 2 2EFFRIX
 2 2EFFRIX
 3 5MATOLT
 3 5MATOLT
 4 3MATRC
 4 3MATRC
 5 4MADD
 5 4MADD
 6 2EFFRIX
 6 2EFFRIX
 7 5MATOLT
 7 5MATOLT
 8 6MATIN
 8 6MATIN
 9 3MATRC
 9 3MATRC
 10 4MADD
 10 4MADD
 11 2EFFRIX
 11 2EFFRIX
 12 7DEFSC

M504, STAIR PROBLEM NUMBER 202

DISPLACEMENTS FOR LOADING CONDITION 1 FOLLOW

JOINT 121	U=	-0.00063	V=	-0.00757	W=	-0.00478
JOINT 122	U=	-0.00049	V=	-0.00488	W=	-0.00005
JOINT 123	U=	0.00001	V=	-0.01028	W=	-0.00003
JOINT 124	U=	-0.00002	V=	-0.00708	W=	-0.00594
JOINT 125	U=	0.00032	V=	-0.01095	W=	-0.00008
JOINT 126	U=	-0.00227	V=	0.	W=	0.
JOINT 201	U=	-0.00067	V=	-0.00636	W=	-0.01248
JOINT 202	U=	-0.00343	V=	-0.00278	W=	-0.01968
JOINT 203	U=	0.00151	V=	-0.01105	W=	-0.01983
JOINT 204	U=	-0.00300	V=	-0.00457	W=	-0.01901
JOINT 205	U=	-0.00362	V=	-0.00731	W=	-0.02855
JOINT 206	U=	-0.00555	V=	-0.00184	W=	-0.02845
JOINT 301	U=	-0.00296	V=	-0.00499	W=	-0.01132
JOINT 302	U=	-0.00593	V=	-0.00203	W=	-0.00871
JOINT 303	U=	-0.00099	V=	-0.00552	W=	-0.00886
JOINT 304	U=	-0.00124	V=	-0.00434	W=	-0.00363
JOINT 305	U=	0.00259	V=	0.	W=	-0.00008
JOINT 306	U=	0.	V=	0.	W=	0.

DISPLACEMENTS FOR LOADING CONDITION 2 FOLLOW

JOINT 101	U=	0.04125	V=	0.07775	W=	-0.00022
JOINT 102	U=	0.03016	V=	0.04431	W=	-0.06621
JOINT 103	U=	0.02154	V=	0.11119	W=	-0.06606
JOINT 104	U=	0.03686	V=	0.08356	W=	0.02535
JOINT 105	U=	0.01594	V=	0.14910	W=	-0.00100
JOINT 106	U=	0.06059	V=	0.	W=	0.
JOINT 201	U=	0.10101	V=	0.05624	W=	0.05196
JOINT 202	U=	0.37599	V=	-0.09279	W=	0.59234
JOINT 203	U=	-0.14241	V=	0.24828	W=	0.50458
JOINT 204	U=	0.11611	V=	0.05025	W=	0.12545
JOINT 205	U=	0.04500	V=	0.07830	W=	0.27977
JOINT 206	U=	0.16601	V=	0.02153	W=	0.27510
JOINT 301	U=	0.11075	V=	0.08011	W=	0.01039
JOINT 302	U=	0.19593	V=	0.08421	W=	-0.10091
JOINT 303	U=	0.05025	V=	0.05601	W=	-0.10525
JOINT 304	U=	0.04564	V=	0.05033	W=	-0.04179

JOINT 325 U= 0.02387 V= 0. W= -0.00437
 JOINT 326 U= 0. V= 0. W= 0.

DISPLACEMENTS FOR LOADING CONDITION 3 FOLLOW

JOINT 121 U= 0.07499 V= -0.21124 W= -0.58733
 JOINT 122 U= 0.09085 V= 0.04738 W= -0.13938
 JOINT 123 U= 0.14428 V= -0.46726 W= -0.14398
 JOINT 124 U= 0.07785 V= -0.27625 W= -0.61052
 JOINT 125 U= 0.09329 V= -0.62862 W= -0.00624
 JOINT 126 U= 0.09898 V= 0. W= 0.
 JOINT 221 U= 0.05228 V= -0.24695 W= -1.38978
 JOINT 222 U= 0.08947 V= -0.01189 W= -1.53744
 JOINT 223 U= -0.02268 V= -0.54974 W= -1.53968
 JOINT 224 U= -0.03454 V= -0.18237 W= -2.12217
 JOINT 225 U= -0.16858 V= -0.49313 W= -3.23024
 JOINT 226 U= -0.23162 V= 0.15573 W= -3.23334
 JOINT 321 U= 0.04069 V= -0.13110 W= -1.33952
 JOINT 322 U= -0.22301 V= 0.05935 W= -1.10891
 JOINT 323 U= 0.25358 V= -0.24016 W= -1.11477
 JOINT 324 U= 0.07214 V= -0.10511 W= -0.51001
 JOINT 325 U= 0.28471 V= 0. W= -0.00784
 JOINT 326 U= 0. V= 0. W= 0.

DISPLACEMENTS FOR LOADING CONDITION 4 FOLLOW

JOINT 121 U= -0.02504 V= 0.03448 W= 0.11131
 JOINT 122 U= -0.02785 V= -0.01170 W= 0.02543
 JOINT 123 U= -0.01282 V= 0.07615 W= 0.02583
 JOINT 124 U= -0.00923 V= 0.02744 W= 0.07886
 JOINT 125 U= -0.00398 V= 0.05134 W= 0.00127
 JOINT 126 U= -0.01119 V= 0. W= 0.
 JOINT 221 U= -0.00358 V= 0.02267 W= 0.11771
 JOINT 222 U= -0.00529 V= 0.01051 W= 0.12555
 JOINT 223 U= 0.00221 V= 0.04171 W= 0.12564
 JOINT 224 U= 0.00604 V= 0.01570 W= 0.15656
 JOINT 225 U= 0.01374 V= 0.03230 W= 0.20789
 JOINT 226 U= 0.01927 V= -0.00131 W= 0.20753
 JOINT 321 U= 0.00401 V= 0.01655 W= 0.11244
 JOINT 322 U= 0.02742 V= 0.00824 W= 0.08451
 JOINT 323 U= -0.01478 V= 0.01766 W= 0.08521
 JOINT 324 U= -0.00324 V= 0.01437 W= 0.06831
 JOINT 325 U= -0.02757 V= 0. W= 0.00149
 JOINT 326 U= 0. V= 0. W= 0.

REACTIONS FOR LOADING CONDITION 1 FOLLOW

JOINT 126 RX= 0. RY= 0.76800 RZ= 0.00000
 JOINT 325 RX= 0. RY= 0.76800 RZ= 0.
 JOINT 326 RX= 0.00000 RY= -0.00000 RZ= 0.00000

REACTIONS FOR LOADING CONDITION 2 FOLLOW

JOINT 126 RX= 0. RY= -3.56212 RZ= -2.91667
 JOINT 325 RX= 0. RY= 0.76795 RZ= 0.
 JOINT 326 RX= -10.00000 RY= 4.33018 RZ= 2.91666

REACTIONS FOR LOADING CONDITION 3 FOLLOW

JOINT 126 RX= 0. RY= 3.16988 RZ= 4.33339
 JOINT 325 RX= 0. RY= 46.37354 RZ= 0.
 JOINT 326 RX= -9.99999 RY= -34.54341 RZ= 2.16672

REACTIONS FOR LOADING CONDITION 4 FOLLOW

JOINT 126 RX= 0. RY= -1.14434 RZ= -1.41667
 JOINT 325 RX= 0. RY= -3.59811 RZ= 0.
 JOINT 326 RX= -1.00000 RY= 2.74245 RZ= -1.58334

12 7DEFSOL

13 00ARSOL

1M504 STAIR PROBLEM NUMBER 200
 FINAL BAR STRESSES FOLLOW

TENSION +VE COMPRESSION-VE/
BAR LOADING CONDITIONS-
NUMBER 1 2 3 4 5

101	102	-0.006016	-0.006017	-0.000001	-0.615100
101	103	-0.006016	-0.006016	-0.000005	1.304900
101	104	-0.004032	0.610179	-0.396136	0.502557
101	304	-0.004032	0.610179	-0.396136	0.249224
102	103	-0.004943	-0.029604	0.941263	-0.002004
102	104	0.016900	-0.630796	-1.691678	0.593117
102	105	-0.049920	0.030211	-2.976561	0.261053
102	106	0.123057	-1.550150	-0.614582	0.232224
102	304	0.067964	0.715667	1.691678	0.515770
102	306	0.034011	-2.649021	-6.253517	0.544996
103	104	0.067964	-0.076774	2.035658	-1.347429
103	105	0.034012	0.300940	3.540526	-0.612201
103	304	0.016900	0.161647	1.433783	-1.149235
103	305	0.123058	0.161909	9.751800	-1.025523
103	306	-0.049920	0.030211	-6.019461	0.261052
104	105	-0.005331	-0.702259	-1.292071	0.655024
104	106	0.013762	1.543747	2.207724	-0.015669
104	201	0.042450	1.224300	1.743590	-0.175062
104	202	0.037149	-1.407318	-0.357147	-0.303006
201	204	0.052602	1.234527	1.743503	-0.175061
202	203	0.031149	1.617053	0.467465	-0.010700
202	204	-0.079505	-1.639735	-2.647402	0.303006
202	205	-0.066673	5.456656	1.156913	0.059130
203	204	-0.047622	-0.407104	0.504604	-0.005246
203	205	0.046463	-3.604478	-1.650697	-0.165025
204	205	0.000061	-2.719420	1.011614	0.197006
204	206	0.000060	0.719926	1.924447	-0.256017
204	301	0.052602	-2.446775	-0.942000	-0.241970
204	302	-0.047622	-1.020600	0.935206	0.161529
204	303	-0.079505	1.106029	-1.304123	0.176761
205	206	0.020045	-0.971314	-0.646400	-0.073564
205	303	-0.016700	1.492156	-0.561140	-0.109723
206	302	0.046464	-2.730502	6.094549	-0.291697
206	303	-0.066673	2.440140	-7.549440	0.457170
301	302	0.003400	0.003409	-0.000021	0.000001
301	303	0.003400	0.003409	0.000007	-0.000000
301	304	0.042450	-2.457000	-0.942070	-0.241970
302	303	0.031149	0.905009	1.220653	-0.144560
104	203	0.005106	0.364660	-0.504605	0.005247
105	106	0.017267	0.207406	1.299620	-0.264690
105	203	0.035570	0.466539	0.716713	-0.363706
106	203	-0.066672	-5.004276	-1.470204	0.059135
106	202	-0.207365	5.370732	-1.042201	0.036006
201	202	0.003400	0.003416	-0.000011	0.000001
201	203	0.003400	0.003402	0.000005	0.000001
302	304	0.005106	0.970252	-0.935214	-0.161530
302	305	-0.066673	-2.030315	-3.060149	0.457172
302	306	0.035500	-1.730923	11.312904	-0.456607
303	304	0.037149	1.776161	4.300749	-0.176760
303	305	-0.207366	2.452536	-13.359041	0.610135
304	305	0.013762	-0.054052	-1.491415	0.600045
304	306	-0.005331	0.109037	3.574659	-1.079792
305	306	0.017267	0.910230	1.634329	-0.311104
202	206	-0.016700	-0.415652	-1.067490	0.142012

13 8BARSCL
14 9CHECK

IM504 STAIR PROBLEM NUMBER 200
 SUMS OF THE INTERNAL STRESSES AT THE JOINTS FOLLOW
 LOADING CONDITION 1

JOINT	FX	FY	FZ
101	0.00000	-0.03542	-0.00000
102	0.00000	-0.10461	-0.00000
103	0.00000	-0.10461	0.00000
104	-0.00000	-0.09787	0.00000
105	-0.00000	-0.08475	-0.00000
106	0.00000	0.68325	0.00000
201	-0.00000	-0.03542	0.00000
202	0.00000	-0.10460	-0.00000
203	0.00000	-0.10460	-0.00000
204	-0.00000	-0.09787	0.00000
205	-0.00000	-0.08475	-0.00000
206	-0.00000	-0.08475	-0.00000
301	0.00000	-0.03542	0.00000
302	-0.00000	-0.10461	-0.00000
303	0.00000	-0.10461	0.00000
304	0.00000	-0.09787	-0.00000
305	-0.00000	0.68325	0.00000
306	0.00000	-0.08476	0.00000

LOADING CONDITION 2

JOINT	FX	FY	FZ
101	-0.00000	-0.03542	0.00000
102	-0.00000	-0.10460	0.00000
103	-0.00000	-0.10461	-0.00000
104	0.00000	-0.09786	-0.00001
105	0.00000	-0.08475	0.00000
106	0.00000	-3.64687	-2.91667
201	0.00000	-0.03543	-0.00002
202	4.99996	-0.10461	7.50002
203	0.00001	-0.10464	0.00002
204	0.00002	-0.09786	-0.00000
205	0.00001	-0.08474	0.00000
206	0.00000	-0.08475	0.00000
301	-0.00000	-0.03542	-0.00000
302	-0.00000	-0.10460	-0.00000
303	5.00000	-0.10460	-7.50000
304	-0.00000	-0.09786	0.00000
305	0.00000	0.68320	-0.00000
306	-10.00000	4.24542	2.91666

LOADING CONDITION 3

JOINT	FX	FY	FZ
101	-0.00000	-0.00002	-0.00001
102	-0.00000	0.00000	-0.00003
103	-0.00000	-4.99998	0.00003
104	0.00000	-0.00001	0.00000
105	-0.00000	-0.00000	-0.00000
106	-0.00001	3.16986	4.33340
201	-0.00000	0.00003	0.00002
202	5.00004	-0.00001	-0.00000
203	0.00001	0.00004	0.00002
204	0.00000	-10.00003	0.00000
205	-0.00002	-0.00002	-0.00006
206	-0.00000	0.00001	-7.50000
301	0.00003	0.00002	0.00004

302	-2.200000	-0.000000	-0.00014
303	5.000003	-0.000002	1.000006
304	0.000001	-0.000001	-0.000000
305	-0.000001	46.37355	0.000000
306	-9.999999	-34.54342	2.16672
LOADING CCNDITION 4			
JOINT	FX	FY	FZ
101	1.000000	2.000000	3.000000
102	0.000000	-0.000000	0.000000
103	0.000000	-0.000000	-0.000000
104	0.000000	0.000000	0.000000
105	0.000000	0.000000	0.000000
106	0.000000	-1.14434	-1.41667
201	0.000000	-0.000000	-0.000000
202	-0.000001	-0.000000	0.000001
203	-0.000000	-0.000000	-0.000000
204	0.000000	0.000000	0.000000
205	0.000000	0.000000	0.000000
206	0.000000	-0.000000	-0.000000
301	-0.000000	-0.000000	-0.000000
302	0.000000	0.000000	0.000001
303	-0.000000	0.000000	-0.000001
304	-0.000000	0.000000	0.000000
305	0.000000	-3.59811	-0.000000
306	-1.000000	2.74245	-1.58334

14 9CHECK

CHAPTER 3 OPERATOR'S MANUAL

A. INTRODUCTION

Aspects of the STAIR System which affect the computer operator and programmer are discussed in this and the following chapter. The principal item covered here is running a STAIR problem from data sheets prepared by the user.

The machine procedure is described for the IBM-709 and -7090. A similar procedure is used on the IBM-704 except for tape assignments.

B. INPUT-OUTPUT

Data prepared by the user is punched into two card decks (structural data for STAIR and load data for SLOP). Both decks are converted directly to BCD (Binary Coded Decimal) input tapes. The load data program binary must first be run on line with a load assembly program (SLOP) tape (B5), which assembles the loads on a binary input tape. All input to the STAIR program is thus from tape.

Off-line output from tape B2 is printed on peripheral equipment under program control* after a successful run. On-line output includes error diagnostics and tracking information, depending on the sense switch settings.

C. MACHINE OPERATION

Starting from prepared data and program tapes, a STAIR problem is run as follows:

1. Mount tapes.
 - B2 Output (any tape)
 - A6 Buffer (any tape)
 - A7 Buffer (any tape)
 - A8 Load Data (binary-output from SLOP)
 - A9 Structural Data (cards pre-stored off-line)
 - B4 STAIR Program Tape
2. Run program by loading STAIR loader on line (one-card tape loader, T/4 transfer card and several blank cards).
3. Dismount and save structural data, load data and program tapes.
4. List off-line output.

D. SENSE SWITCHES AND INDICATOR LIGHTS

Sense switch settings allow the following options during the running of a problem:

- | | | |
|------|----|--|
| SS 1 | UP | Stop at all detected errors after printing diagnostic. |
| | DN | Print diagnostic but do not stop after certain errors are detected in SLOP and subroutine STRIX (see User's Manual). |

*Experience has shown that it is more convenient to have the output printed single-spaced, but this varies with the peripheral equipment available.

- SS 2 UP No tracing print out.
- DN Print tracing information after completion of each step in ANALYSIS PROGRAM.
- SS 3 UP No off-line output from SLOP.
- DN Write assembled loads on output tape B2 for listing.
- SS 6 UP No off-line output from intermediate steps.
- DN Write all information completed in this step on output tape B2 for listing (used only for program debugging).

Under no circumstances should the I-0 check light be on at the completion of a successful run. The Divide-Check and Overflow indicators may be on.

E. PROGRAM STOPS

Most program stops will occur after an on-line diagnostic is printed. If a stop occurs without a diagnostic, the Instruction Location Counter (ILC) should be recorded and a programmer consulted.*

At the end of a successful run, all tapes on DSC A are rewound, and the program is in position to start a new problem. The next problem may be run by mounting new data tapes A8 and A9, rewinding tape B4, replacing the STAIR loader in the card reader and then clearing and loading cards.

After stops caused by tape failure, the program may be rerun, but a different tape and tape drive should be tried. If repeated stops occur, one of the error stops without a diagnostic might be suspected.

Errors in data input must be corrected in the appropriate card deck, and that deck must be rewritten on tape before the run is repeated.

F. INTERRUPTION OF A RUN

For problems requiring a large amount of machine time for their solution, it may not be possible to do all the computations without an interruption which would require getting off the machine.

To interrupt the program:

1. Mount a tape on B5 for dumping the core.
2. Put Sense Switch 4 down. This will cause the master routine to take a core dump on tape B5 after the current step is completed, then stop.
3. The partial output is on tape B2. Save tapes B5, A6, A7, A8, A9 and B4.

To continue the run at a later time:

1. Rewrite the structural data tape A9 (off-line) with the data for the remaining steps.
2. Mount tapes B5, A6, A7, A8, A9, B4 and a new B2.
3. Put Sense Switch 5 down.
4. Run program by loading STAIR loader (tape B4 card followed by a T/4 card and several blank cards).

*Also see p. 149.

CHAPTER 4 PROGRAMMER'S MANUAL

A. INTRODUCTION

The STAIR System consists of 11 routines, written on a program tape and obtained by the computer during an analysis problem, and a load assembly program used to prepare data for the analysis.

The routines on the program tape may be grouped as follows:

1. STAIR operating routines – nine routines (plus a subroutine) which perform the operations described in Chapter 1.
2. STAIR service routines – two routines which are called by several of the operating routines, and a master routine which initiates each analysis problem and calls successive operating routines.
3. Library routines – conversion, I-O and elementary function routines, as punched from the FORTRAN II compiler.
4. MISLAM – an executive routine which writes the STAIR program tape from a binary deck and controls its operation.

Except for the library routines, all these routines and a pair of load assembly programs are described on the following pages.

Since the number of instructions comprising the STAIR routines is about 30,000, there is not enough storage space in a 32,767-register computer to store all the programs and a large matrix, simultaneously. Instead, the nine operating routines are stored on tape and called into core only when needed.

Transfer of control between operations is handled by the master routine. Common storage (upper memory) is used extensively to allow continuity between routines.

B. DEFINITION OF MATRIX TERMS

A stiffness matrix K is formed from data describing the geometry of a structure and modulus of elasticity of the material E .

A matrix equation is composed of a stiffness matrix and a load matrix F which determine a displacement matrix $U = K^{-1}F$ or $F = KU$. K is a symmetrical matrix.

Where displacements are governed by external restraints, rows corresponding to those restrained displacements may be removed from the stiffness and load matrices and stored as reaction matrices (K_R, F_R). Reactions may be evaluated from:

$$R = K_R U + F_R$$

A matrix equation may be partitioned into the following form:

$$\begin{bmatrix} K_{11} & K_{12} \\ \text{---} & \text{---} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ \text{---} \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ \text{---} \\ F_2 \end{bmatrix}$$

Block elimination yields an explicit expression for U_2 :

$$K_{22}^e U_2 = F_2^e$$

$$K_{22}^e = K_{22} - K_{12}^T K_{11}^{-1} K_{12}$$

$$F_2^e = F_2 - K_{12}^T K_{11}^{-1} F_1$$

$$K_{12}^T \text{ is the transpose of } K_{12}$$

The superscript e denotes an "effective" matrix. The eliminated displacements may be evaluated from a group of matrices called "residuals":

$$U_1 = K_{11}^{-1} (F_1 - K_{12} U_2)$$

C. CORE STORAGE ALLOCATION

A map of core storage during an analysis problem is given, with approximate actual locations.

Buffer Storage	Location (octal)
	77777
	77461
Common Storage (Length Varies)	
Not Used	
STAIR Operating Routine (Length Varies)	
STAIR Service Routines	
Library Routines	
MASTER Routine	
MISLAM	11
Loader BSS	0

Common storage contains a list of parameters and arrays to be defined in the next section. All arrays are FORTRAN-stored, one-dimensional arrays. Only the upper half (above the main diagonal) of a symmetric matrix is stored, so addresses are computed for a triangular array. Stiffness matrices are stored row-wise and load matrices are stored column-wise.

D. DEFINITION OF COMMON VARIABLES

Common storage contains the following list of variables during execution of all the operating routines:

IDP	problem identification number
NSTEP	consecutive number of steps in ANALYSIS PROGRAM being executed
E	elastic modulus of bar material
NREF	number of residual and reaction matrices written on tape A6
NJREAC	number of reaction joints in structure
NUT	number of units in structure
NK	number of loading conditions
NU	number of unit whose matrix is in primary storage area
NJT	number of joints in primary row sequence
NJSEC	number of joints being eliminated from primary row sequence, or number of joints in secondary row sequence
NRJSEC	number of reaction joints among joints being eliminated
NRCSEC	number of reaction components among joints being eliminated
JREAC	list of all reactions acting on structure (four words per reaction joint giving joint number and code indicating whether or not components of reaction act in x-, y- or z-directions); (0) means no reaction, (1) means a reaction
NTAPE	tape reading error indicator (set by FIXTPE and sensed by routine calling FIXTPE)
NROW	list of joint numbers associated with matrix in primary storage area
STRIX	primary matrix storage area

The following arrays are included in common storage by all routines except STRIX (to be distinguished from array STRIX), BARSOL and CHECK.

NROSEC	secondary row sequence, associated with matrix in secondary storage area
LRSEC	list of reactions among joints being eliminated
STSEC	secondary matrix storage area

Additional parameters and arrays may be included in common storage by the use of certain operating routines, but these parameters do not affect the continuity of operation among routines.

E. TAPE STORAGE FORMATS

Storage formats for the input and buffer tapes attached to DSC A are given below. The format of the program tape B4 will be described with MISLAM.

Tape A6 - Residuals and reaction matrices in two binary-record groups.
Record 1 - IDOUT, NJT, NJSEC, NRJSEC, NRCSEC
Record 2 - NROW, LRSEC, STSEC
Tape A7 - Matrix equation storage in three binary-record groups.
Record 1 - NHED, NJT, NJSEC, NRJSEC, NRCSEC, NU
Record 2 - NROW, STRIX
Record 3 - NROSEC, LRSEC, STSEC

Tape A8 - Load data with consecutive binary records containing the load matrix for consecutive units of the structure. The first word of each record is a heading NHED.

Tape A9 - Structural data in groups of BCD records.

Following an initial group of two data records, the records for the step number and the operation number follow, each of which is followed by the data records for the step.

Records 1 and 2 - General structure data

Record 3 - Step number, operation number

Record 4...n - Data for this step

Record n+1 - Step number, operation number

NHED identifies a record group u as 32767-u.

IDOUT identifies a two-record group as a residual (0) or reaction matrix (1).

F. STAIR OPERATING ROUTINES

The nine operating routines are described on the following pages. Each description has the following format:

- (a) Operation - definition of the matrix operation performed.
- (b) Mathematical method - brief description of technique for performing the operation.
- (c) Initial condition of core - common variables which must be present to maintain continuity.
- (d) Error stops.
- (e) Output - to tape only; on-line output consists of error diagnostics and tracing information.
- (f) Final condition of core - common variables which are changed, affecting continuity.
- (g) Addressing diagrams.
- (h) Block diagrams - flow charts.

1. STRIX

Operation: Forms a stiffness matrix equation in core from input defining the geometry of a structure and its load matrix.

Mathematical Method: Terms of the matrices for single bars are computed and superimposed on the array of the final matrix. The location of the final matrix array of the bar connecting joints a and b is determined by the positions of a and b in the list of joint numbers NROW. The load matrix for the structure is read directly from tape A8, and is stored immediately after the stiffness matrix. A vector check* indicates if all the bars meeting at any joint lie in a plane; such a geometry would result in an ill-conditioned stiffness matrix.

Initial Condition of Core: Common variables E, NUT and NK must be predetermined.

*The check for a planar joint is performed in the following manner: Let L_1 = unit vector in direction of bar 1; L_2 = unit vector in direction of bar 2; L = unit vector in direction of any other bar. First, the vector product $C = L_1 \times L_2$ is executed. If each component of C is less than 0.000001, the bars are considered collinear, and the unit vector associated with the next bar is used as L_2 to form a new C . Then, the scalar product $C \cdot L$ is executed. If $C \cdot L < 0.00001$ for every remaining bar, the joint is considered planar.

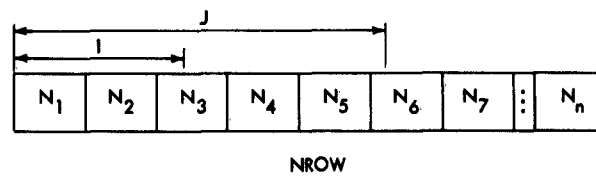
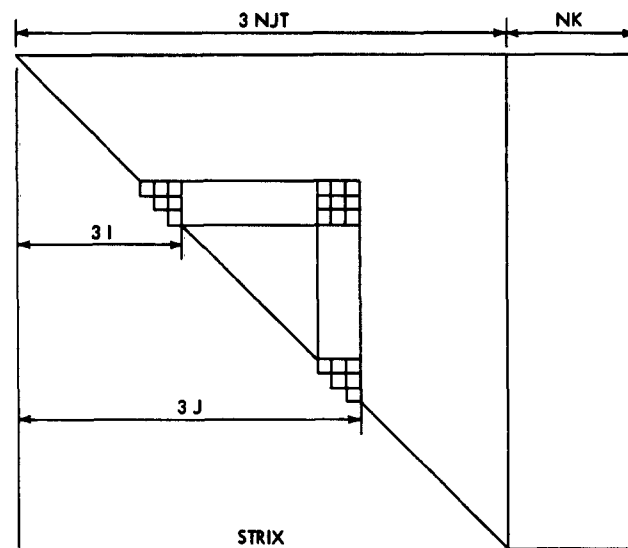
Error Stops:

- (a) Number of terms in matrix equation > 16000.
- (b) Number of bars in unit structure > 600.
- (c) Unit number > number of units.
- (d) A bar number includes the number of a joint now in NROW.
- (e) All the bars meeting at a joint lie in a plane (see Sense Switch 1 settings).
- (f) Tape failure searching for load matrix.

Output: None.

Final Condition of Core: Common variables STRIX, NROW, NJT and NU define the matrix equation which has been formed. Common variables NJSEC, NRJSEC and NRCSEC are zeroed.

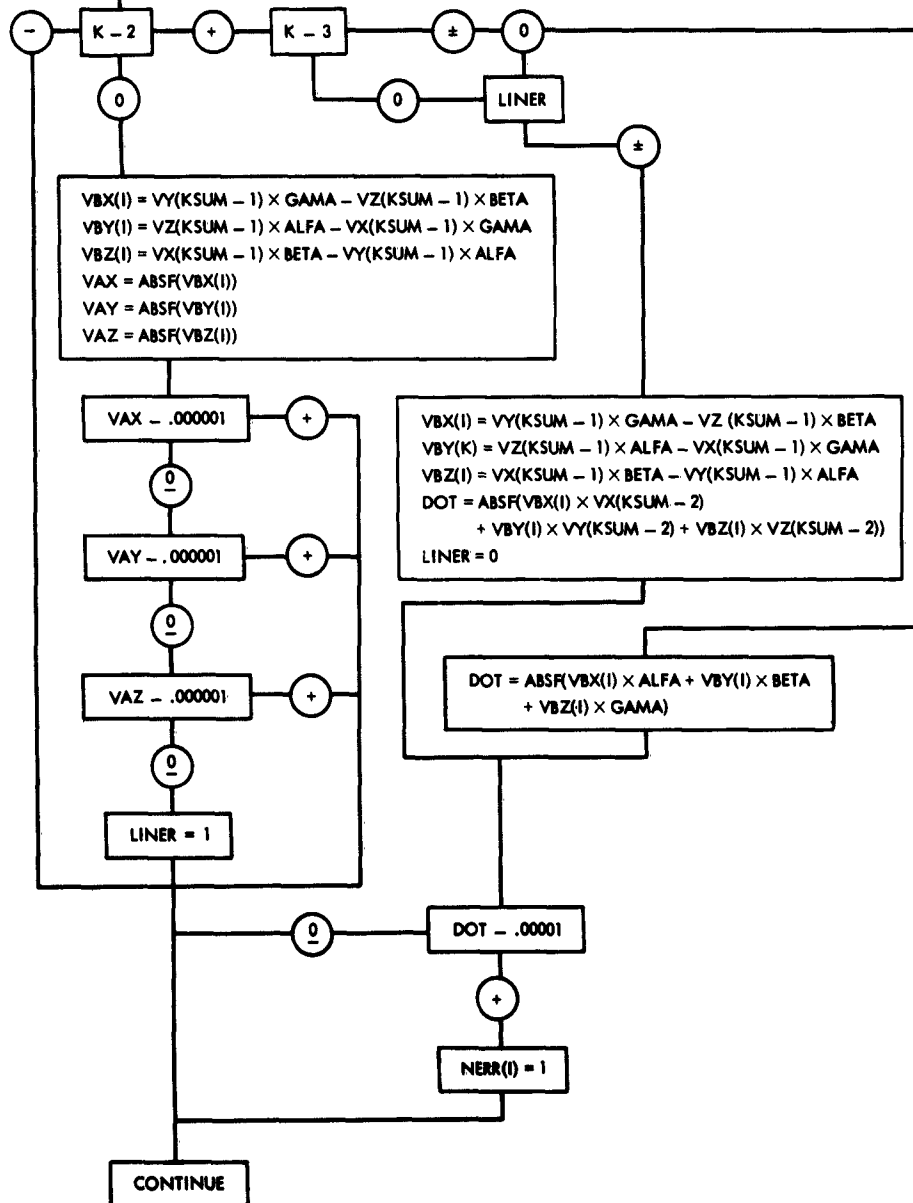
STRIX ADDRESSING DIAGRAM:

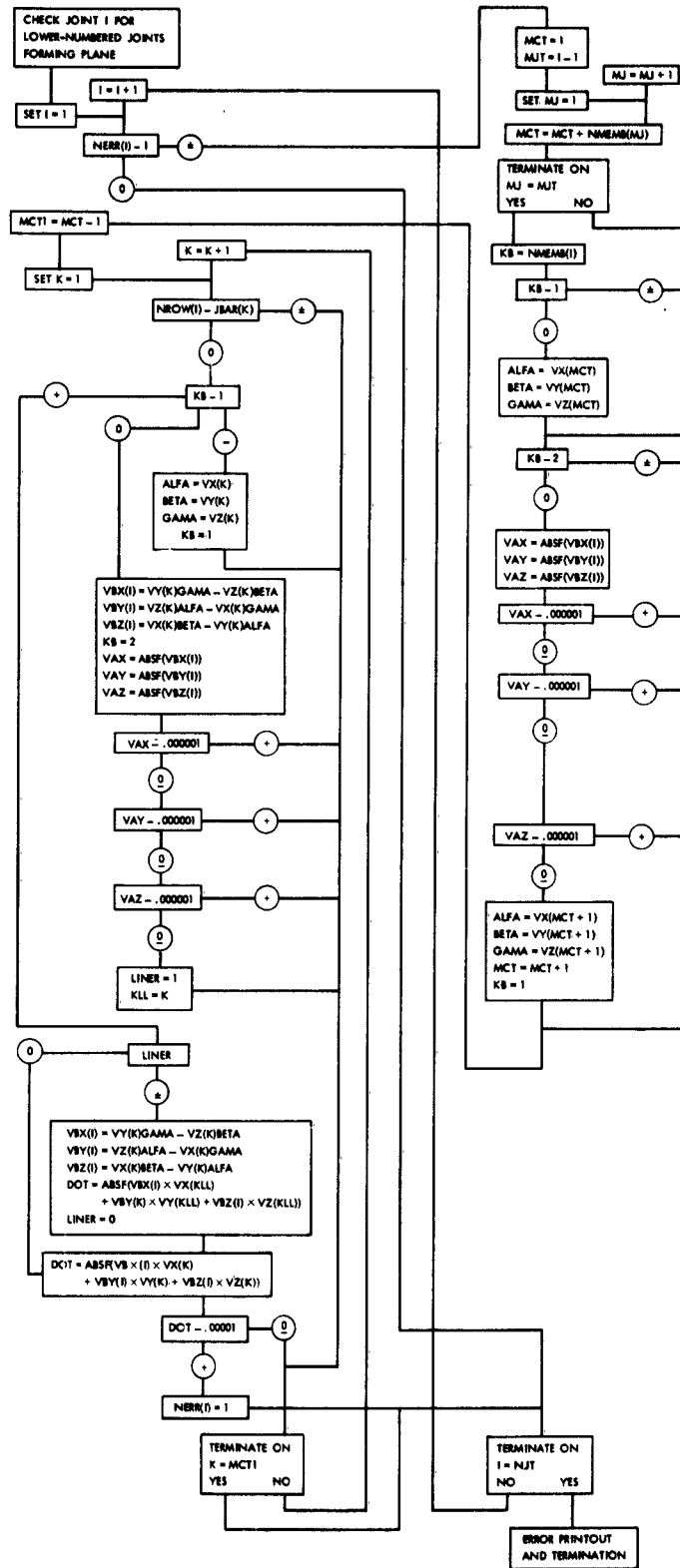


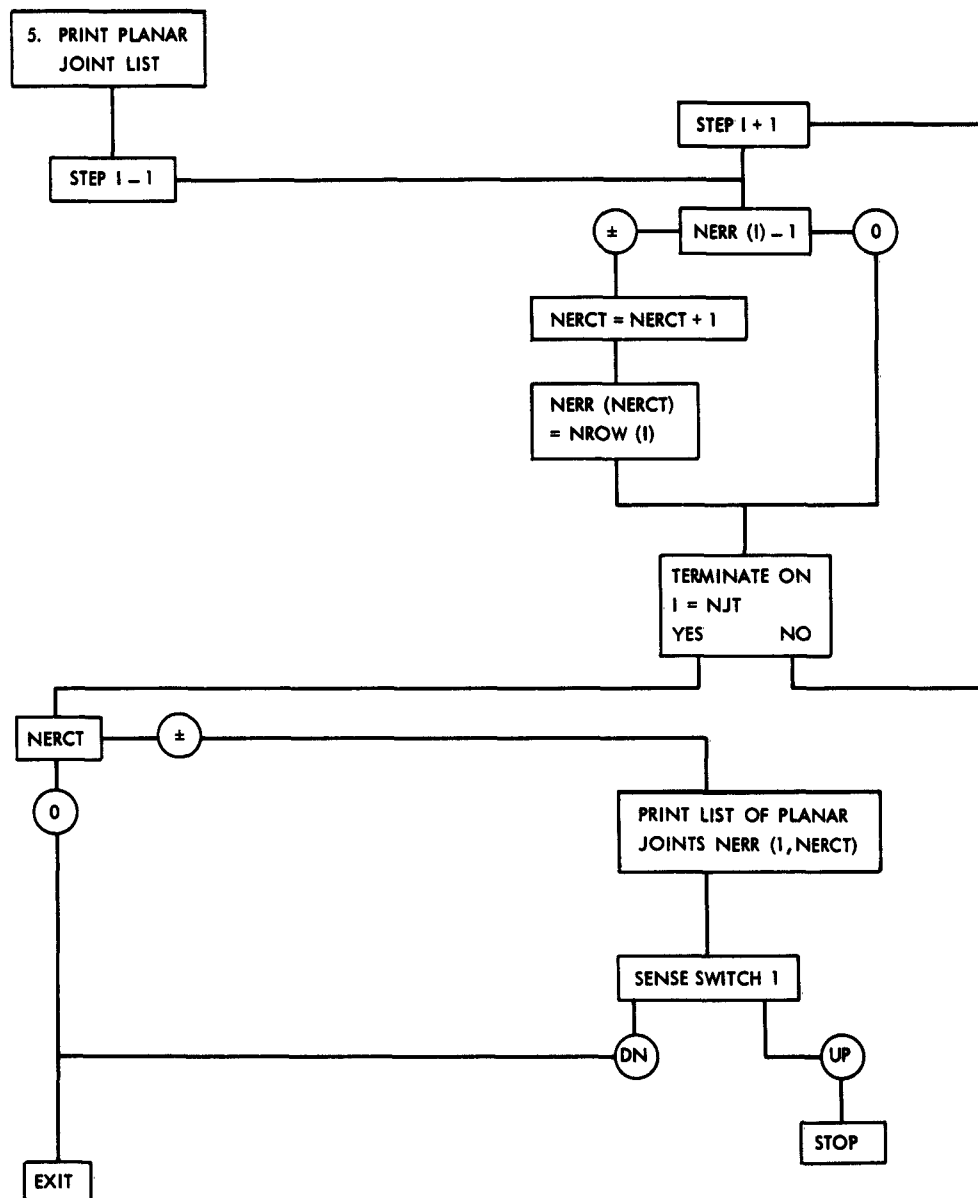
N'S JOINT NUMBERS LISTED IN ANY ORDER
MATRIX STORED FOR BAR $N_I N_J$, WHERE $N_I < N_J$ BY CONVENTION

CHECK JOINT I FOR HIGHER-NUMBERED JOINTS FORMING PLANE

$VX(KSUM) = ALFA$
 $VY(KSUM) = BETA$
 $VZ(KSUM) = GAMA$







2. EFFRIX

Operation: Eliminates a block of rows from a matrix equation in core.

Mathematical Method: The rows to be eliminated must be at the top of the original matrix equation, so that it may be subdivided as indicated:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

If any of the displacements U are fixed by external reactions, the corresponding rows from K_{11} , K_{12} and F_1 are then separated, condensed and re-stored with the reaction rows removed. K_{11} is expanded into square form in the area previously occupied by K_{11} and K_{12} and is inverted. Effective matrices are formed according to the following equations and stored at the head of the array occupied by the original matrix equation.

$$K_{22}^e = K_{22} - K_{12}^T K_{11}^{-1} K_{12}$$

$$F_2^e = F_2 - K_{12}^T K_{11}^{-1} F_1$$

Finally, the residuals K_{11}^{-1} , K_{12} and F_1 are written on tape A7.

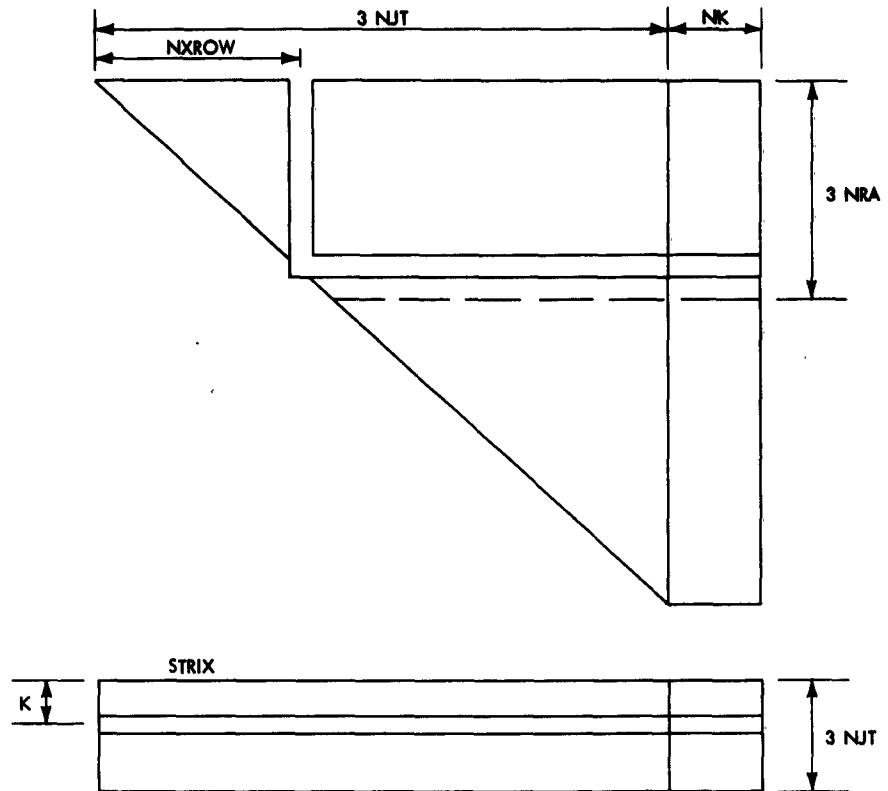
Initial Condition of Core: Common variables JREAC, NJREAC and NK must be present. A matrix equation, as formed by STRIX or MADD, must be in array STRIX, with identifying parameters in NROW and NJT.

Error Stops: Violations of size restrictions, as indicated in the addressing diagrams.

Output: A reaction matrix, if any of the eliminated displacements correspond to reactions and residuals (all on tape A6).

Final Condition of Core: Effective matrices are stored in STRIX, and residuals in STSEC. The effective matrix equation is identified by NJT and NROW, both of which are modified by the elimination of joints. NROSEC contains the original NROW; NJSEC contains the number of joints eliminated from NROW. The location and number of reactions among the eliminated displacements are indicated by NRJSEC, NRCSEC and LRSEC.

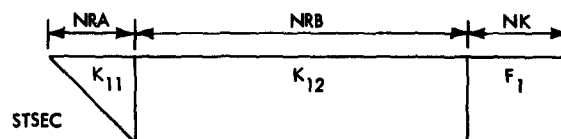
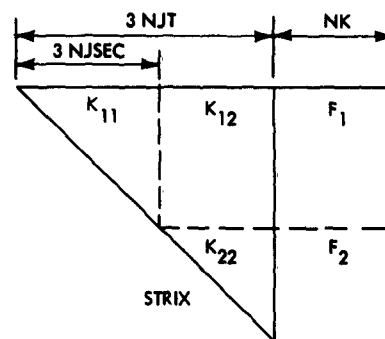
EFFRIX ADDRESSING DIAGRAMS



STSEC

SIZE RESTRICTION: $NRJSEC \leq 10$

REACTION EQUATION STORAGE BY EFFRIX



$$\text{NRA} = 3 \text{ NJSEC} - \text{NRCSEC}$$

$$\text{NRB} = 3 (\text{NJT} - \text{NJSEC})$$

SIZE RESTRICTIONS:

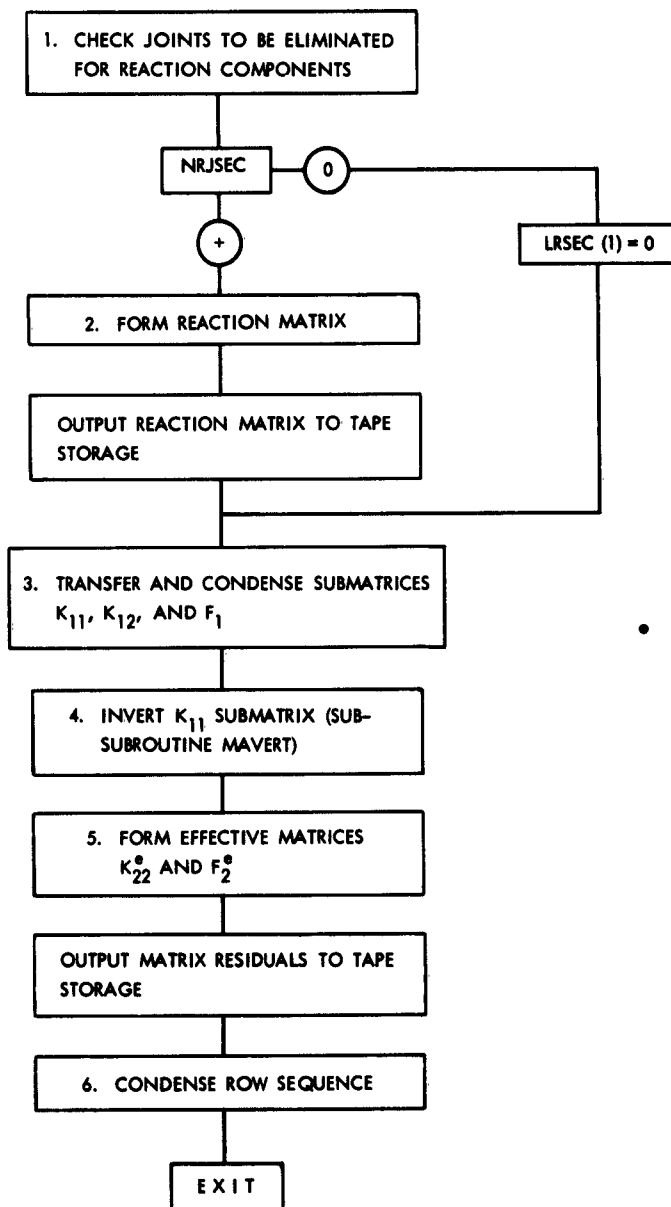
$$\text{NRA} (\text{NRA} + 1)/2 + \text{NRA} (\text{NRB} + \text{NK}) \leq 9500$$

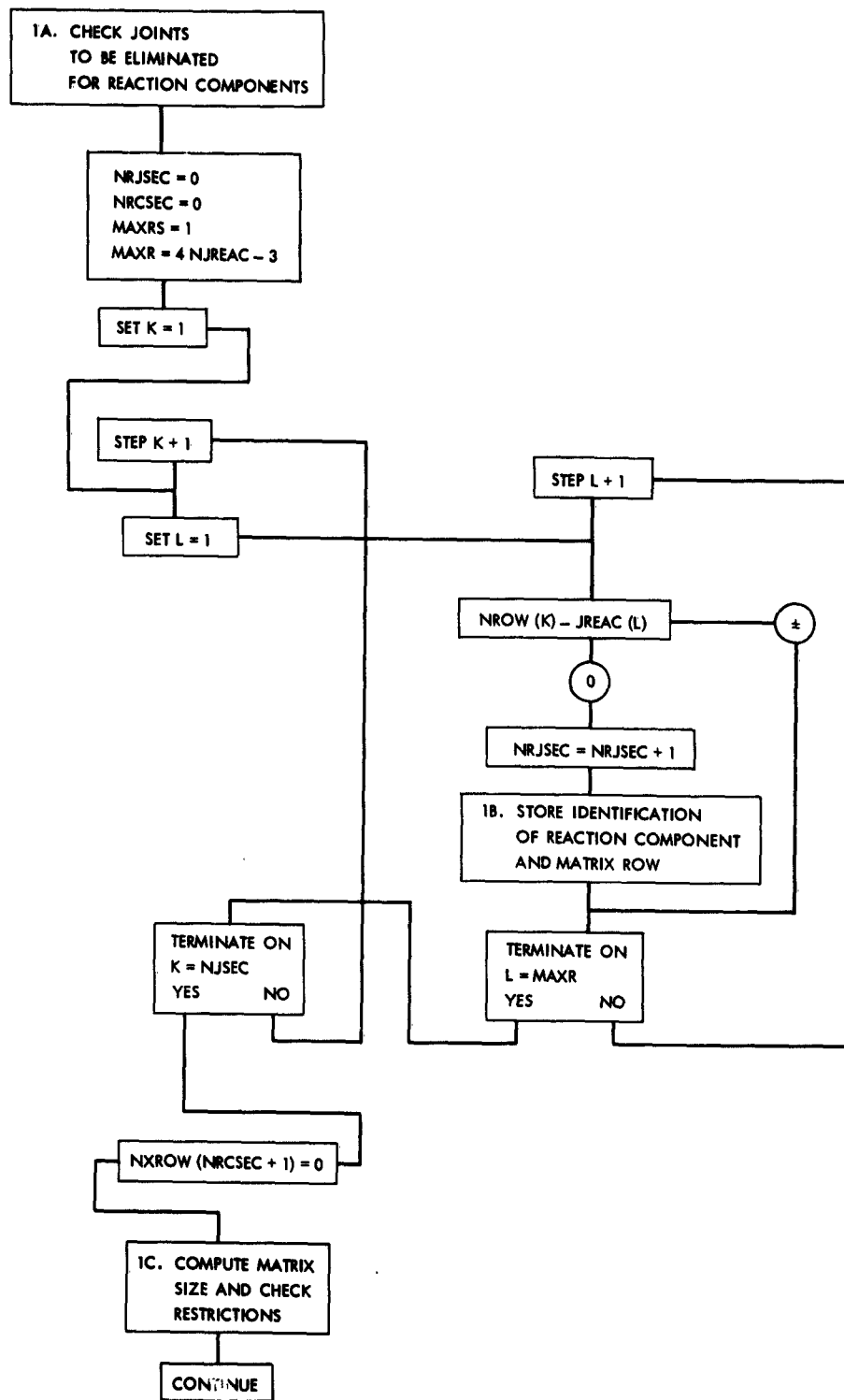
$$\text{NRA}^2 \leq 3 \text{ NJSEC} (3 \text{ NJSEC} + 1)/2 + 9 \text{ NJSEC} (\text{NJT} - \text{NJSEC})$$

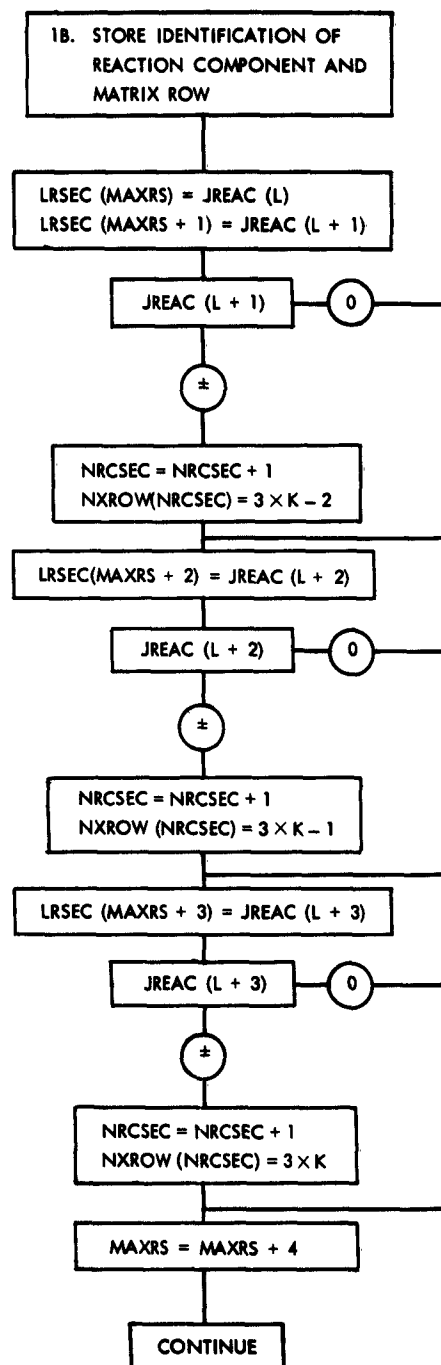
$$(3 \text{ NJT} + \text{NRB} + 1)/2 \cdot 3 \text{ NJSEC} > (\text{NRB} + 1)/2 \cdot \text{NRB}$$

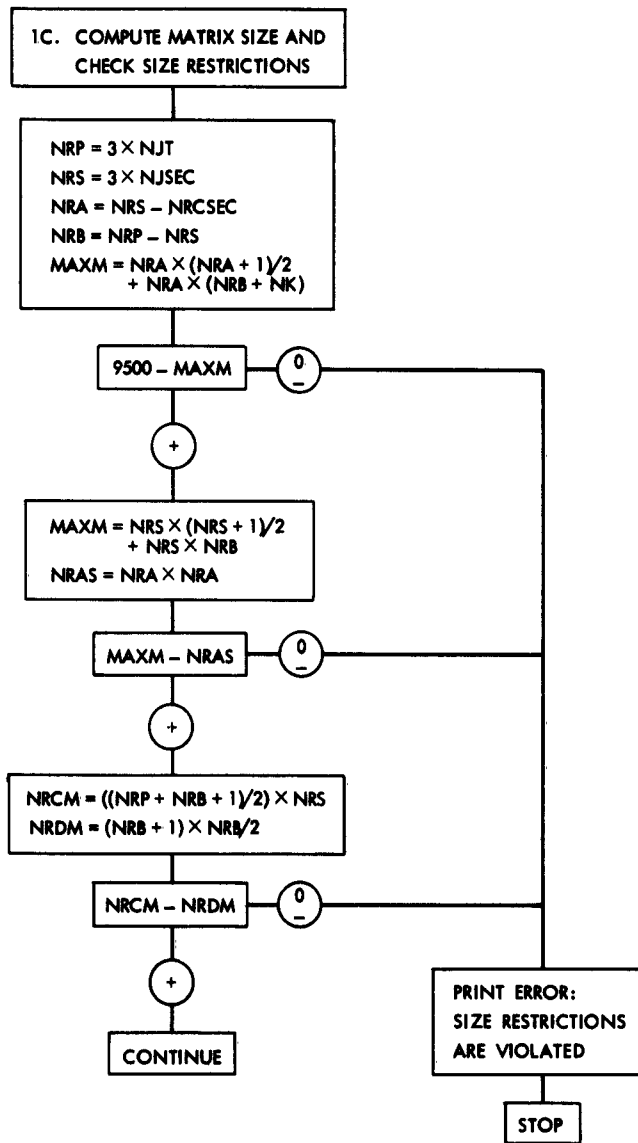
MATRIX RESIDUAL STORAGE BY EFFRIX

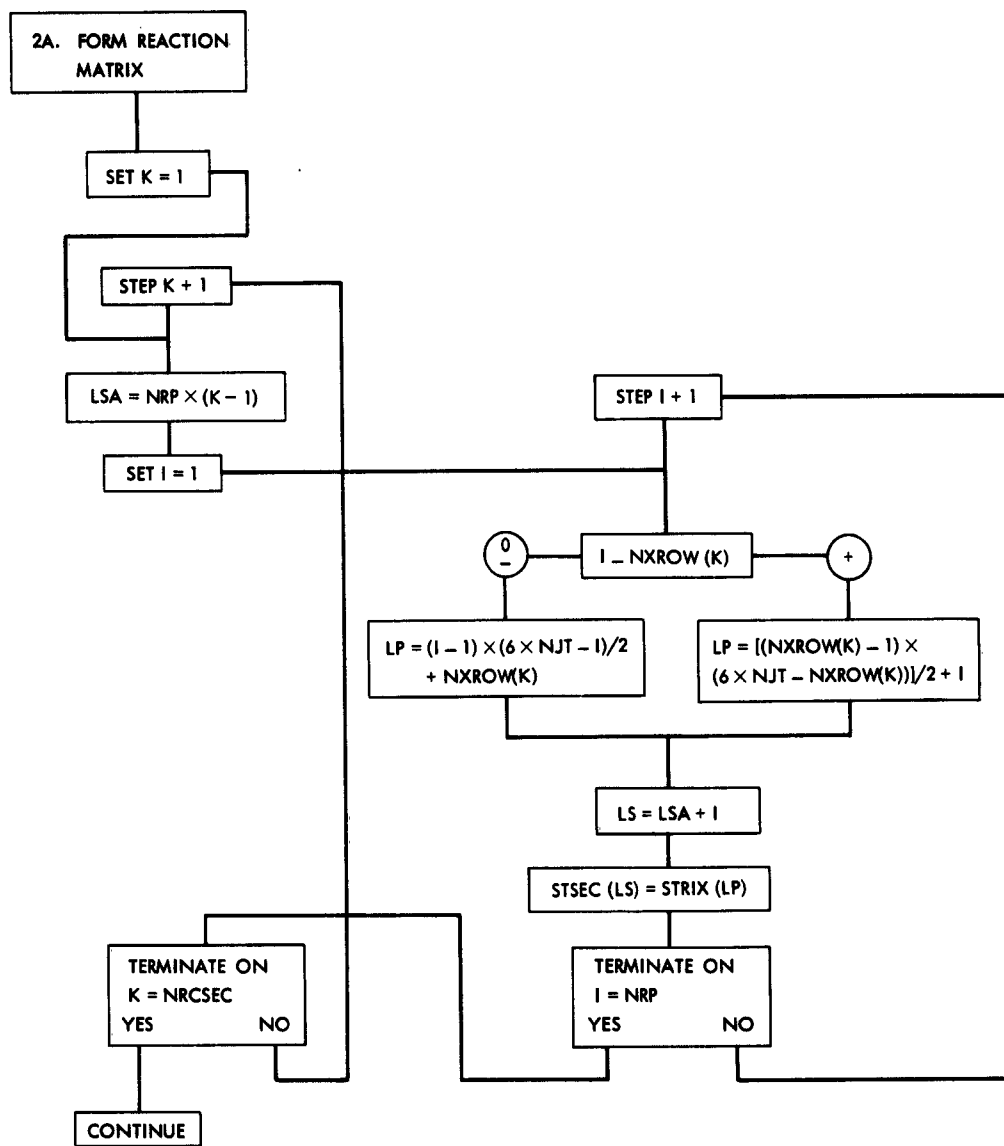
EFFRIX - GENERAL

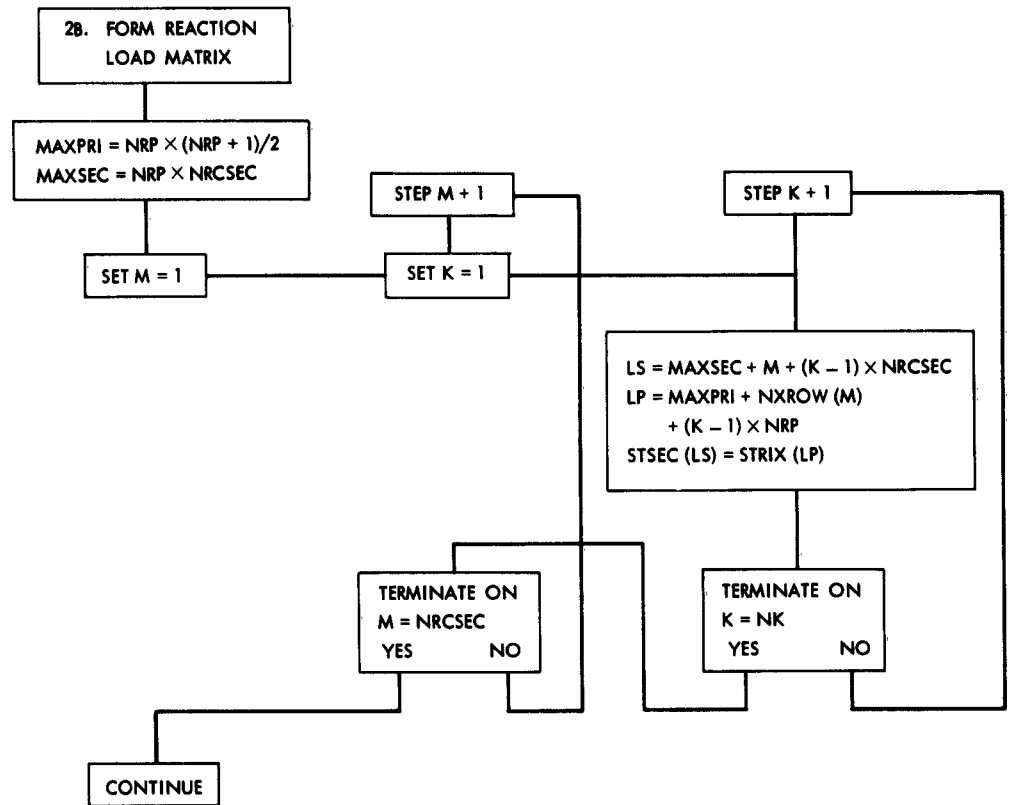










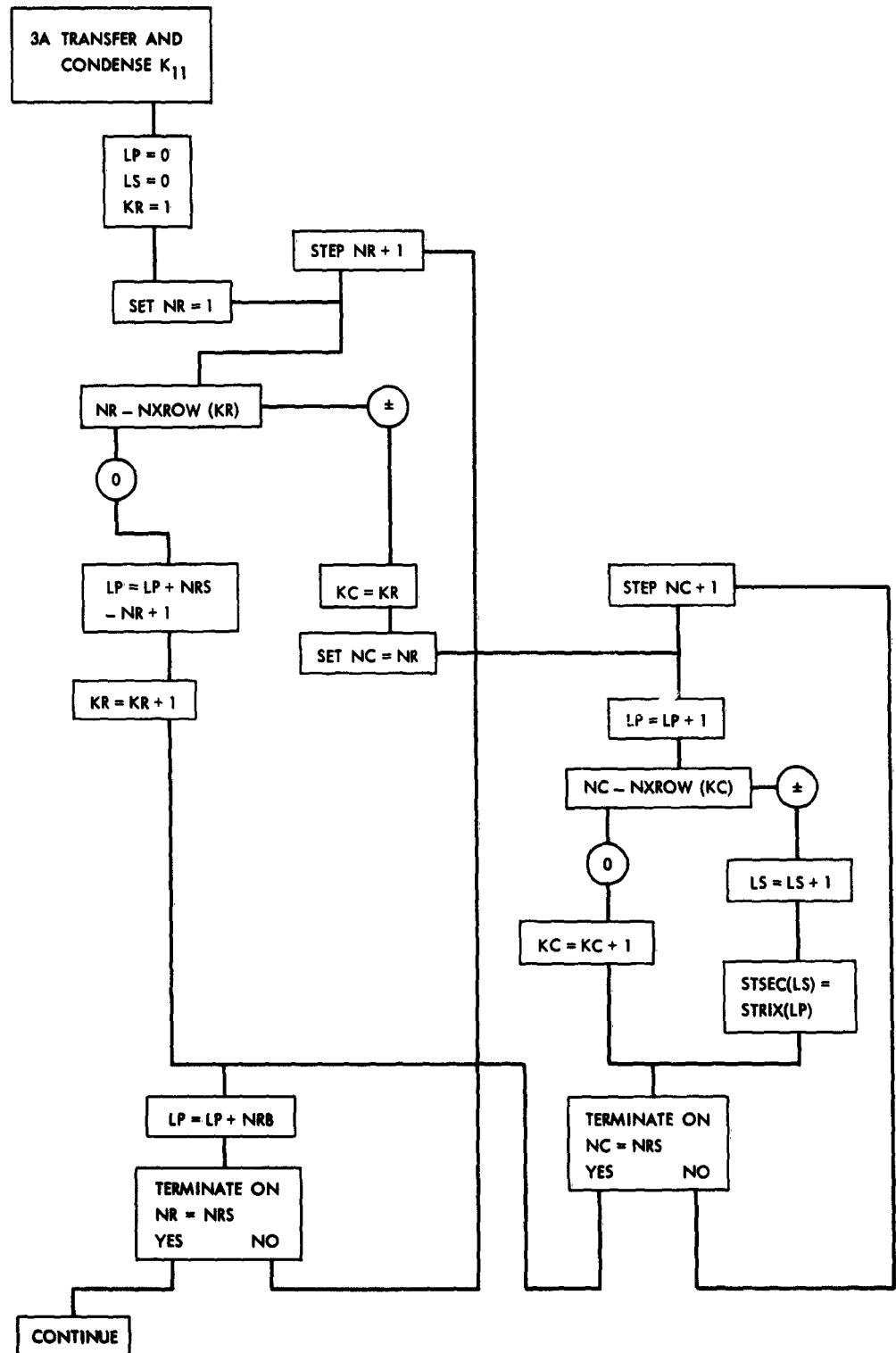


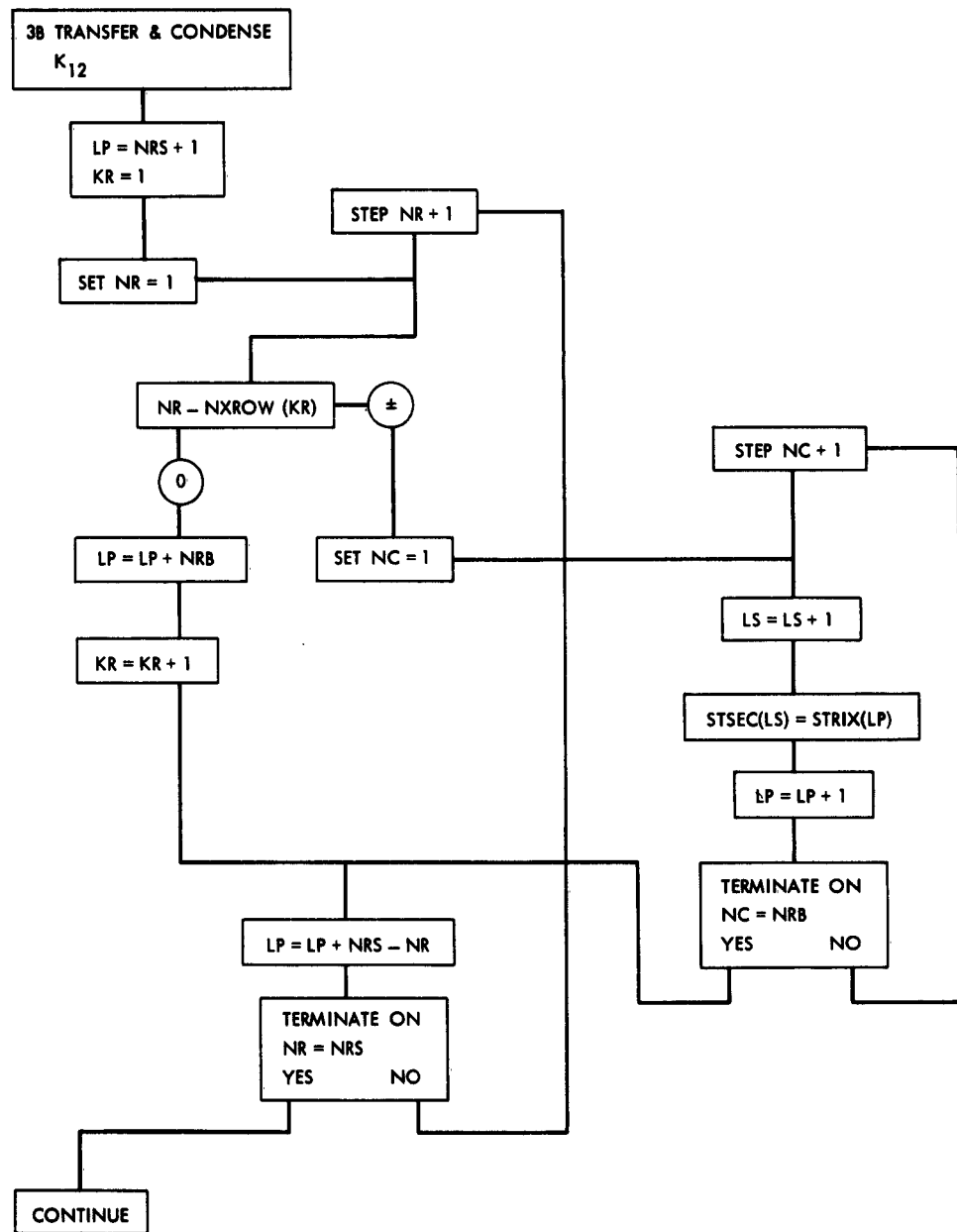
OUTPUT REACTION
MATRIX TO TAPE
STORAGE

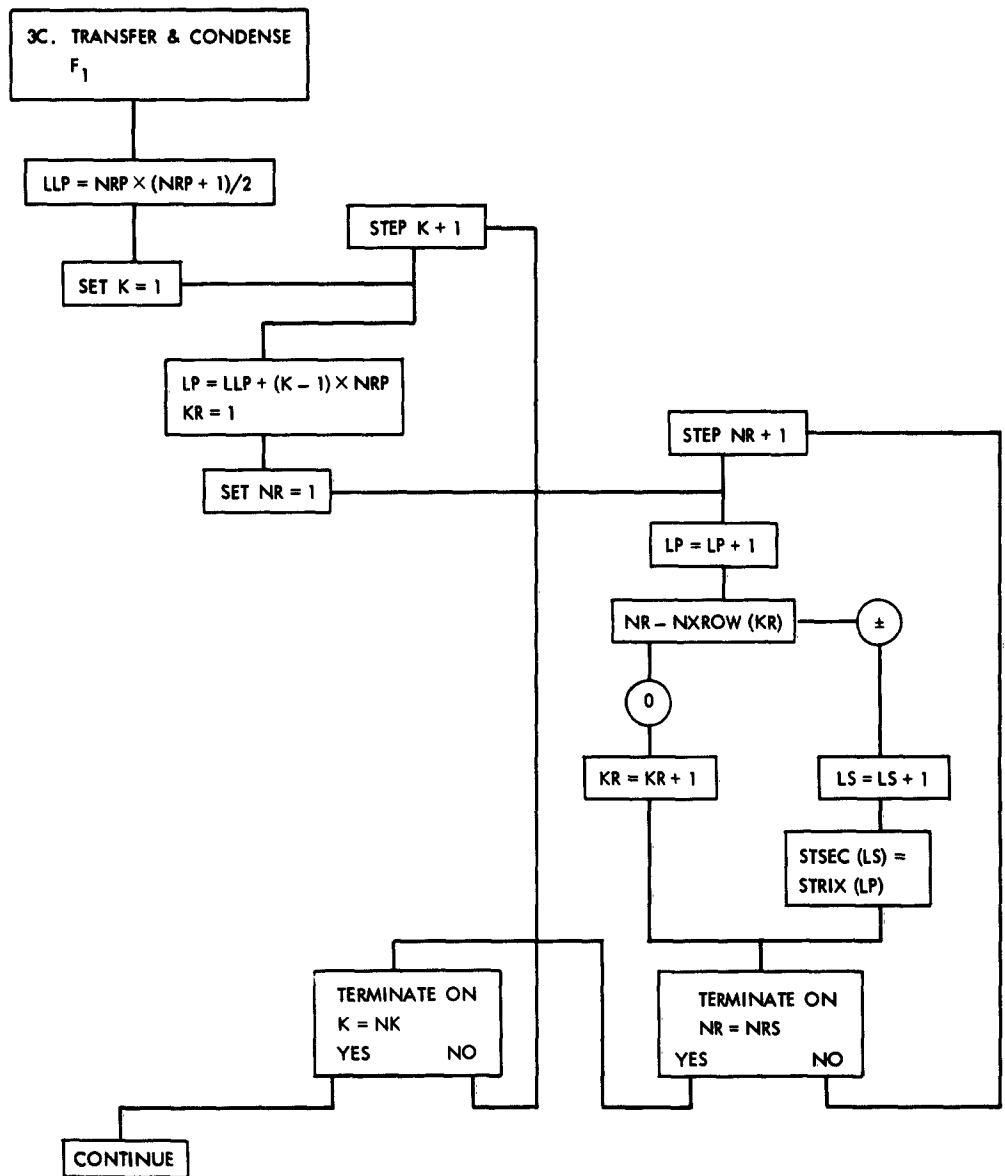
$MAXM = 4 \times NRJSEC + 1$
 $LRSEC(MAXM) = 0$
 $MAXSEC = MAXSEC + NK \times NRCSEC$
 $NREF = NREF + 1$
 $IDOUT = 1$

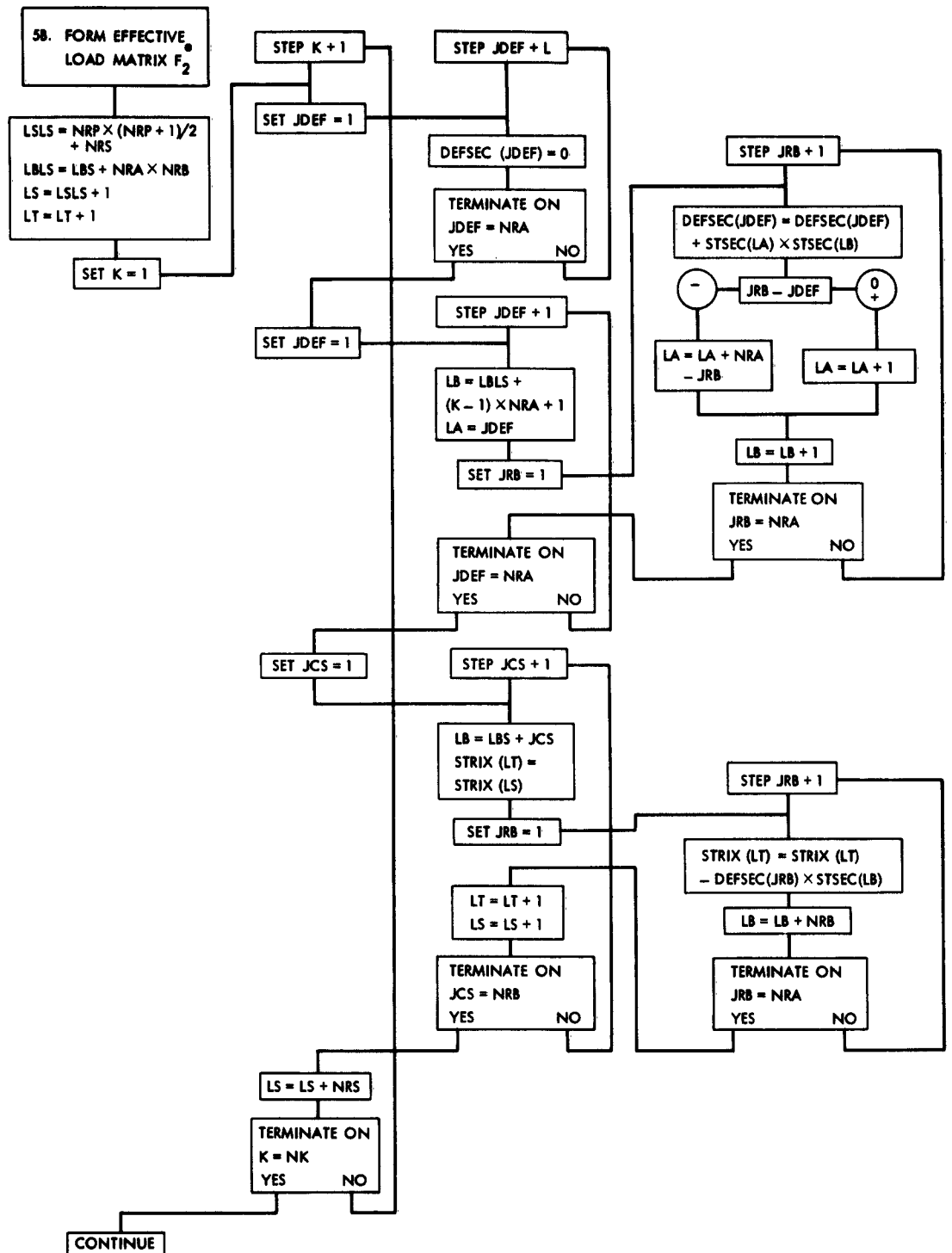
WRITE TAPE 6
 $IDOUT, NJT, NJSEC, NRJSEC, NRCSEC,$
 $NROW(I), LRSEC(I), STSEC$

CONTINUE







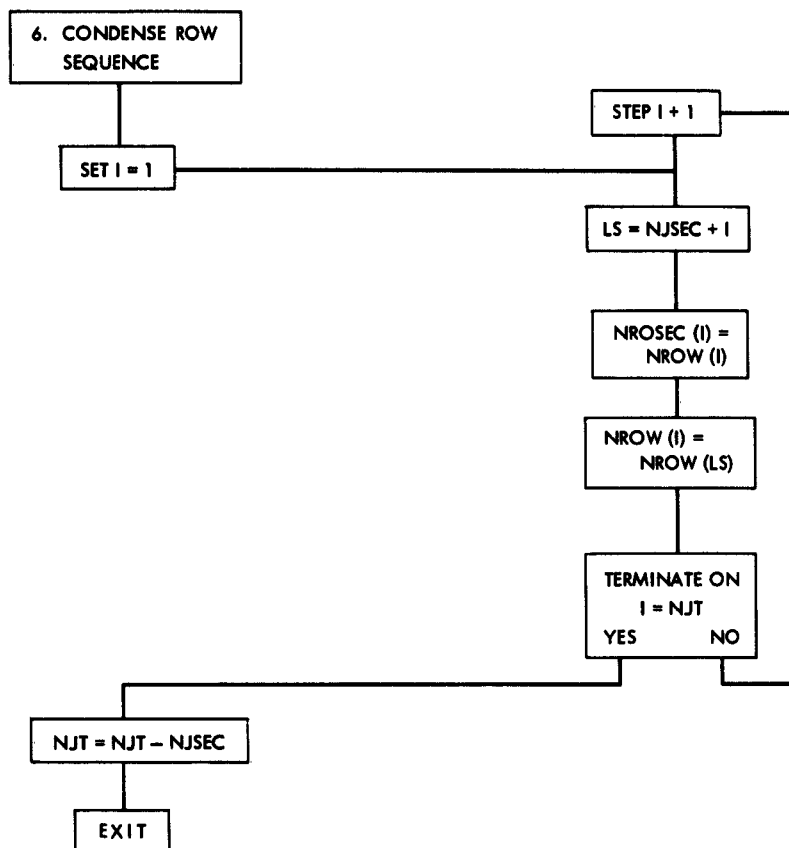


OUTPUT MATRIX RESIDUALS TO TAPE
STORAGE

$MAXM = 4 \times NRJSEC + 1$
 $LRSEC(MAXM) = 0$
 $MAXSEC = LBL5 + NRA \times NK$
 $NREF = NREF + 1$
 $IDOUT = 0$

WRITE TAPE 6
IDOUT, NJT, NJSEC, NRJSEC, NRCSEC
NROW(I), LRSEC(I), STSEC(I)

CONTINUE



3. MATRO

Operation: Forms the stiffness matrix equation of a unit from the stiffness matrix of another geometrically similar unit and the load matrix of the new unit.

Mathematical Method: Transformation formulae are applied to 3×3 blocks of terms to accomplish a rotation in the x-y coordinate plane. The load matrix of the new unit is read directly from tape A8. If an effective matrix is rotated, its residuals are also rotated and a new effective load matrix F_2^e is computed. Since the transformation formulae are applied to 3×3 blocks, they may not be applied to a residual from which rows have been removed to form a reaction matrix.

Initial Condition of Core: The matrix equation of a single unit of a structure must be stored in core, as by STRIX or EFFRIX.

Error Stops:

- (a) New unit number > quantity of units.
- (b) Reaction joints among joints in old or new effective matrices.
- (c) Tape failure searching for load matrix.

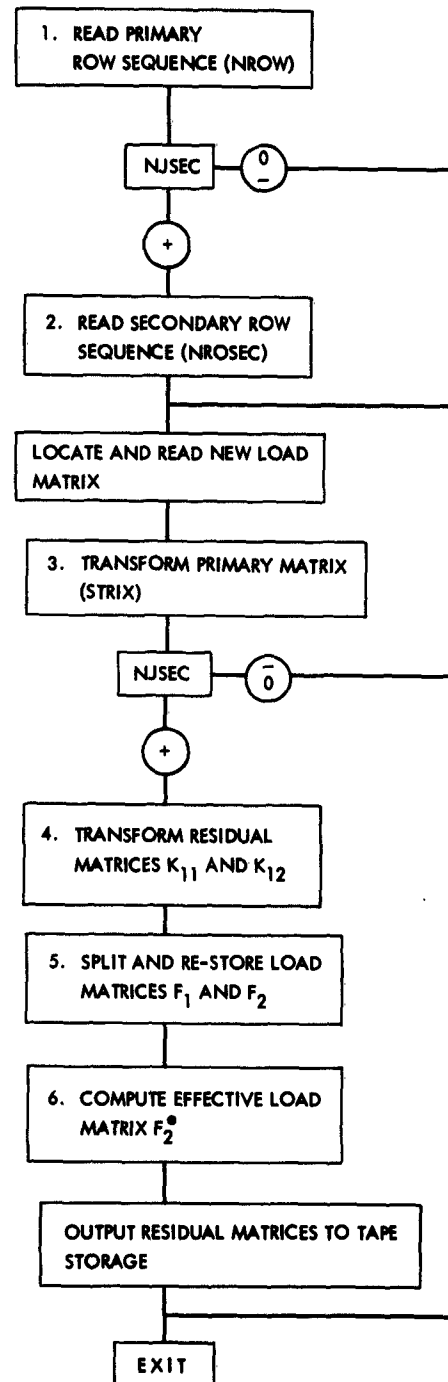
Output: Residuals for the new unit are written on tape A6 when an effective matrix is transformed.

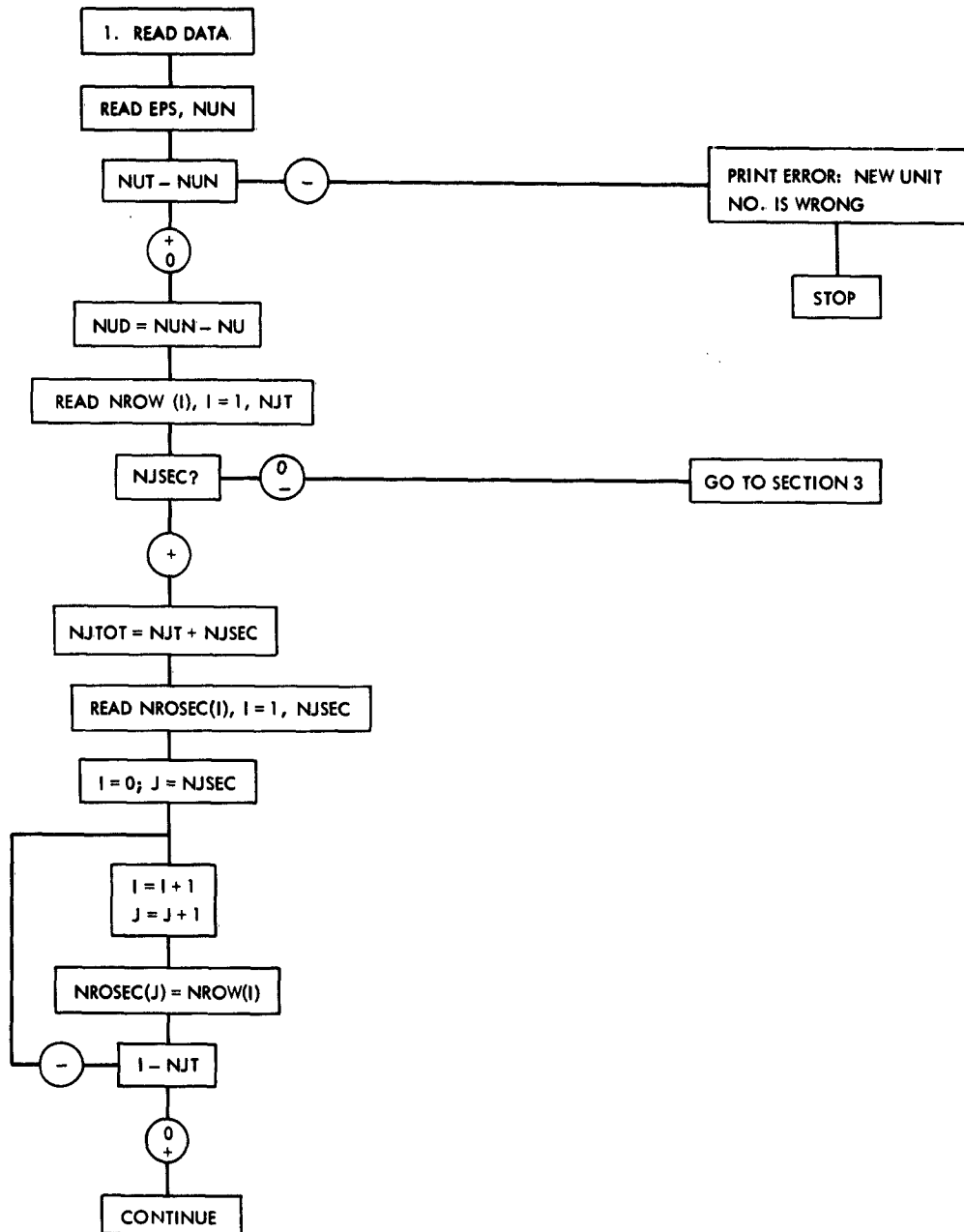
Final Condition of Core: The general map of core storage is unchanged; transformed matrices occupy STRIX and STSEC, and the new unit is identified by NU, NROW and NROSEC.

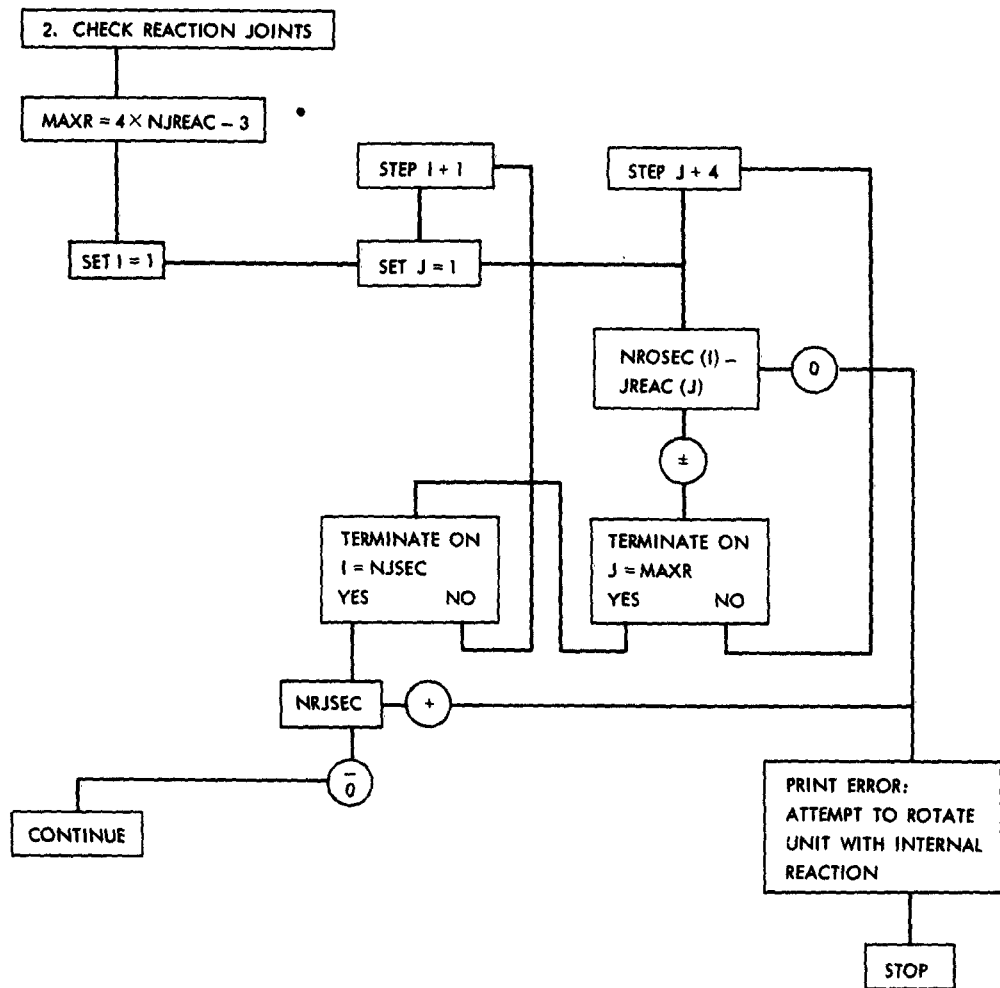
Addressing Diagram: Terms of a 3×3 block are addressed from the following location symbols computed on the diagonal of the block:

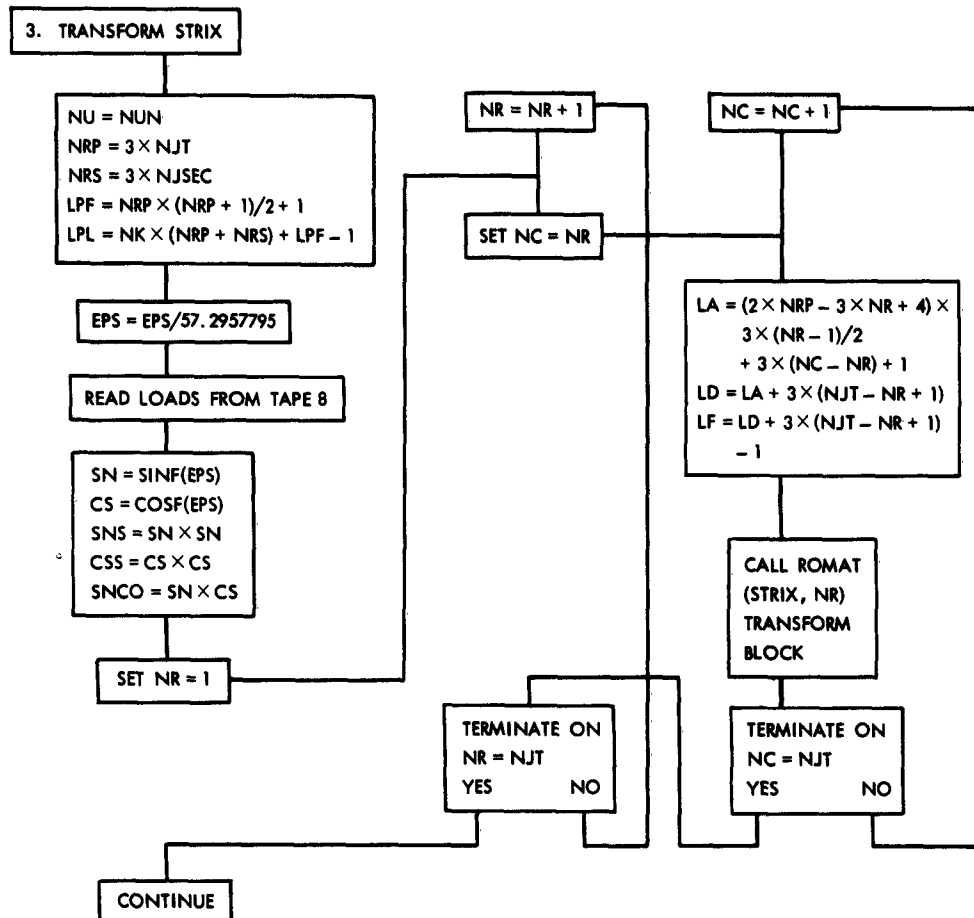
LA		
	LD	
		LF

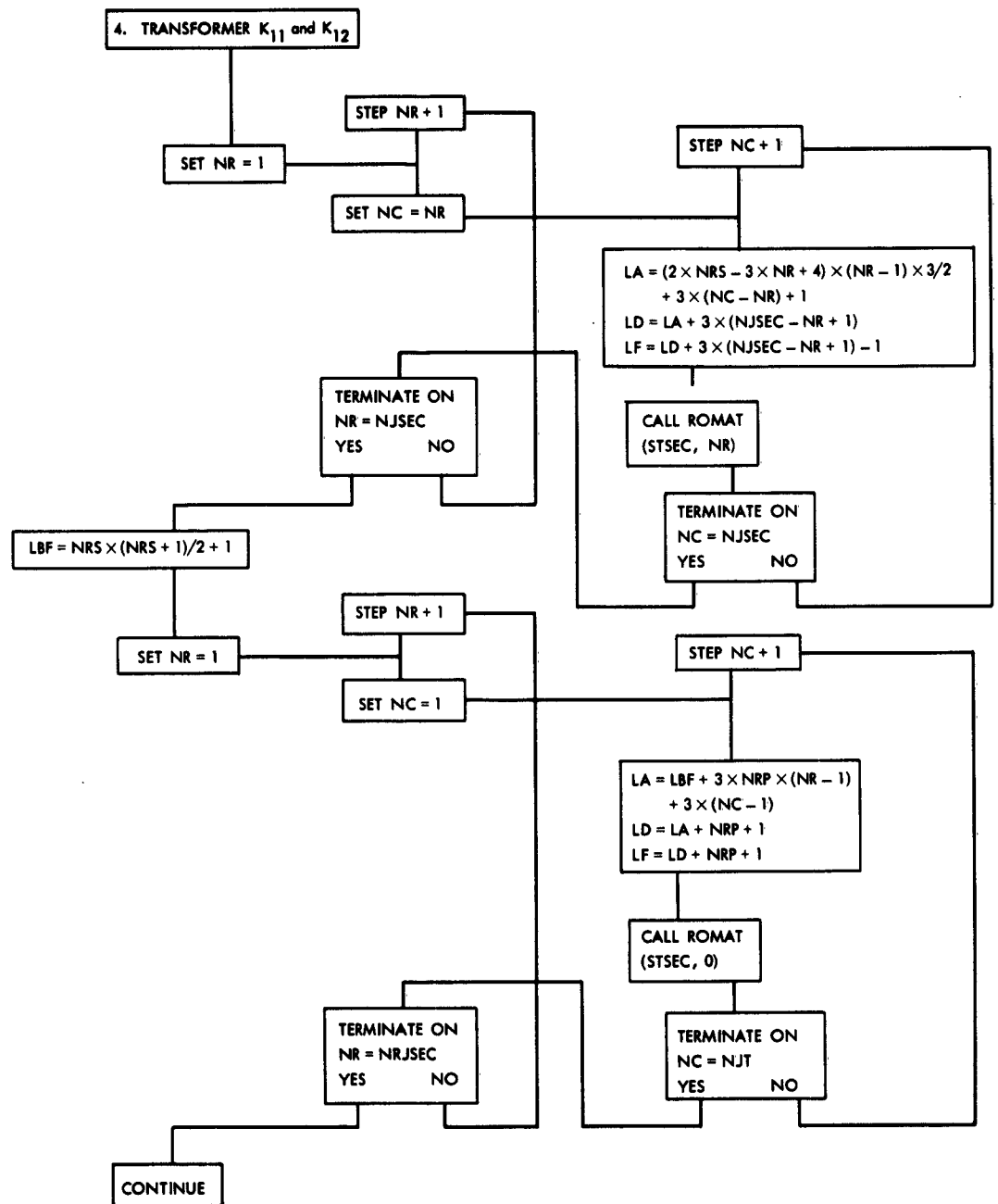
MATRO-GENERAL

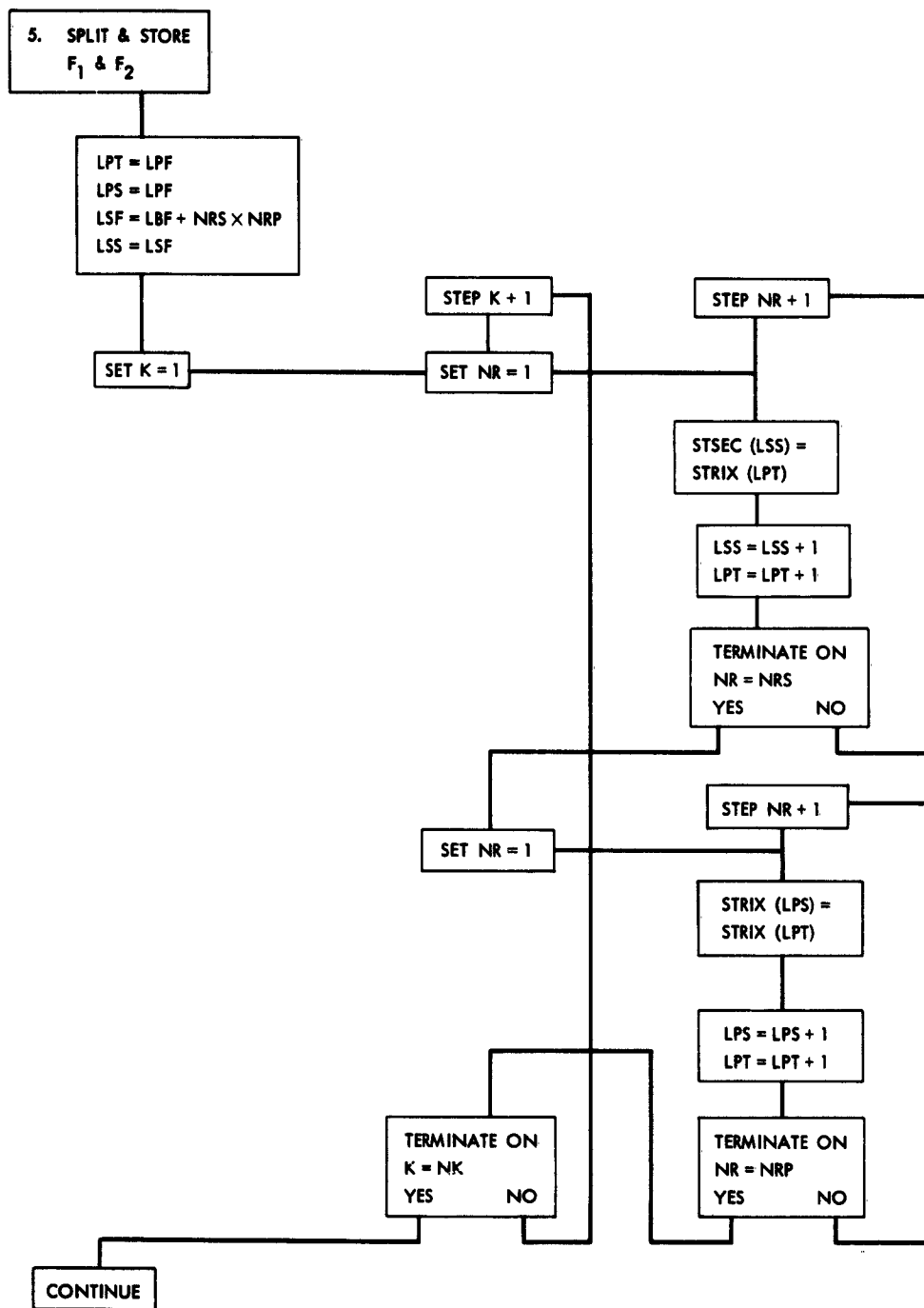


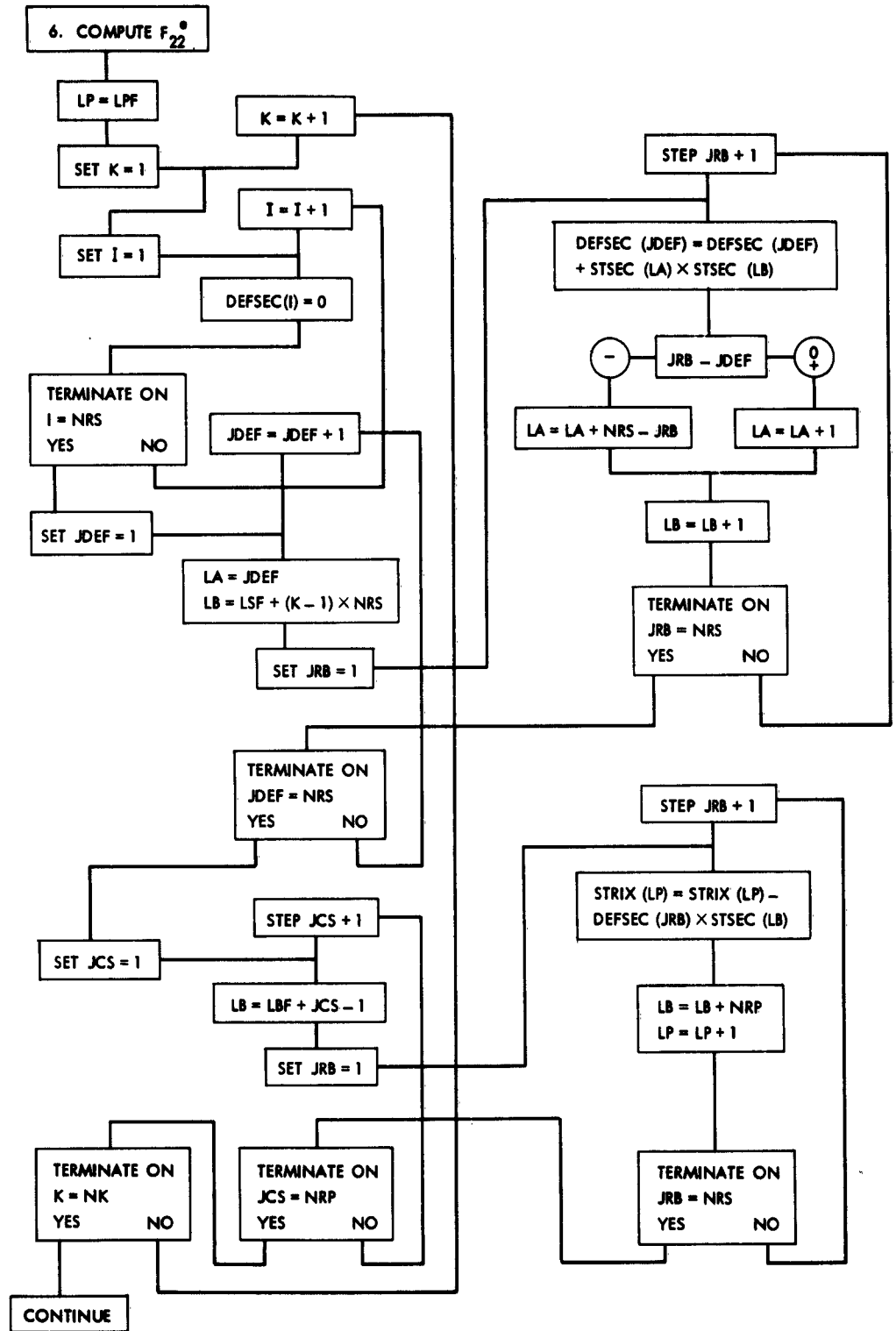










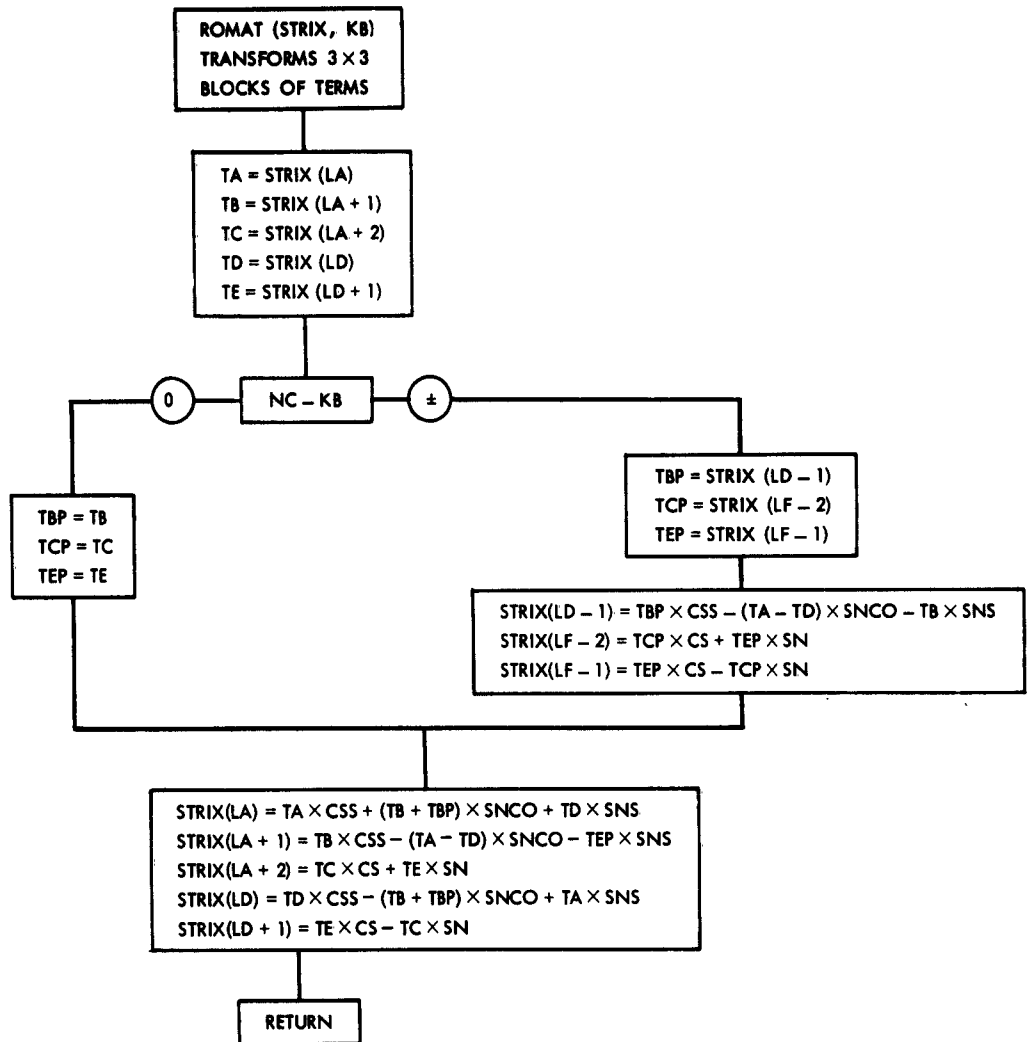


OUTPUT RESIDUAL MATRICES
TO TAPE STORAGE

$MAXM = NRS \times (NRS + 1)/2$
 $+ NRS \times (NRP + NK)$
IDOUT = 0
NREF = NREF + 1

WRITE TAPE 6
IDOUT, NJTOT, NJSEC, IDOUT, IDOUT,
NROSEC(I), IDOUT, STSEC(I)

CONTINUE



common variables: LA, LD, LF, NC, CS, SN, CSS, SNS, SNCO

subroutine arguments: K₂₂ STRIX, NR

K₁₁ STSEC, NR

K₁₂ STSEC, 0

4. MADD

Operation: Superimposes the stiffness matrix equations on two adjacent parts of a structure.

Mathematical Method: A matrix equation initially in STRIX is transferred to STSEC, and STRIX is cleared. The matrix in STSEC is identified by NJSEC and NROSEC. A new list of NJT joint numbers is read into NROW which identifies the new matrix equation to be formed in STRIX. Joint numbers in NROSEC are compared to those in NROW; the consecutive number of a joint in NROW replaces the corresponding joint number in NROSEC and 3×3 blocks of terms in STSEC are superimposed on the STRIX array, with addresses computed from the renumbered NROSEC. $NK \times 3$ blocks from the load matrix in STSEC are stored similarly. When the original in-core matrix has been restored, a second matrix is located on tape, read into STSEC, NROSEC and NJSEC, and the superposition process is repeated.

Initial Condition of Core: A matrix equation as formed by STRIX, MATRO, EFFRIX or another MADD must be in storage area STRIX; it is identified by NROW and NJT.

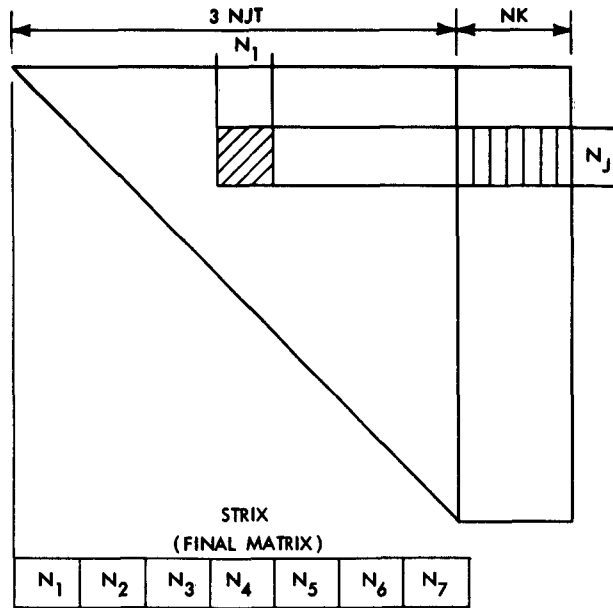
Error Stops:

- (a) Number of registers occupied by final matrix equation > 16000 .
- (b) Number of registers occupied by either equation being added > 9500 .
- (c) A joint number in NROSEC which is not in NROW.
- (d) Tape failure searching for matrix on tape, or no matrix on tape corresponding to the tape location given as input.

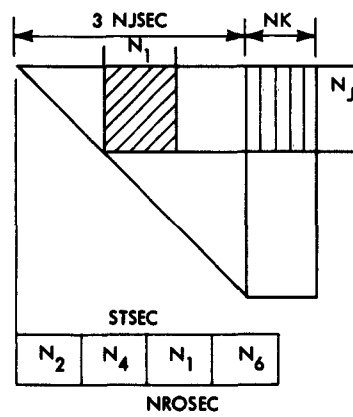
Output: None.

Final Condition of Core: A matrix equation is stored in STRIX, with identifying parameters in NROW and NJT. (Common assignments for STSEC and NROSEC differ in their routine from the others, but they do not affect continuity.) NJSEC is zeroed.

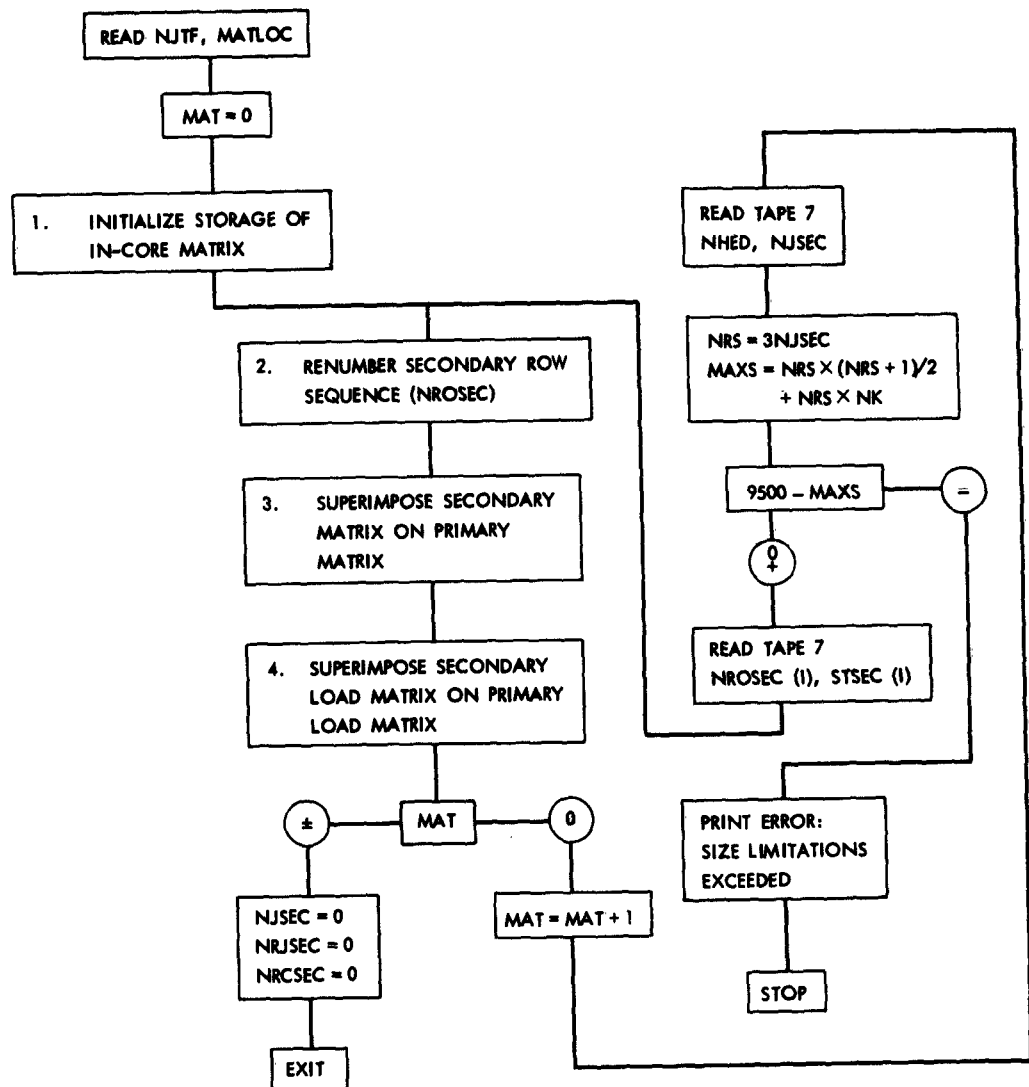
MADD ADDRESSING DIAGRAM

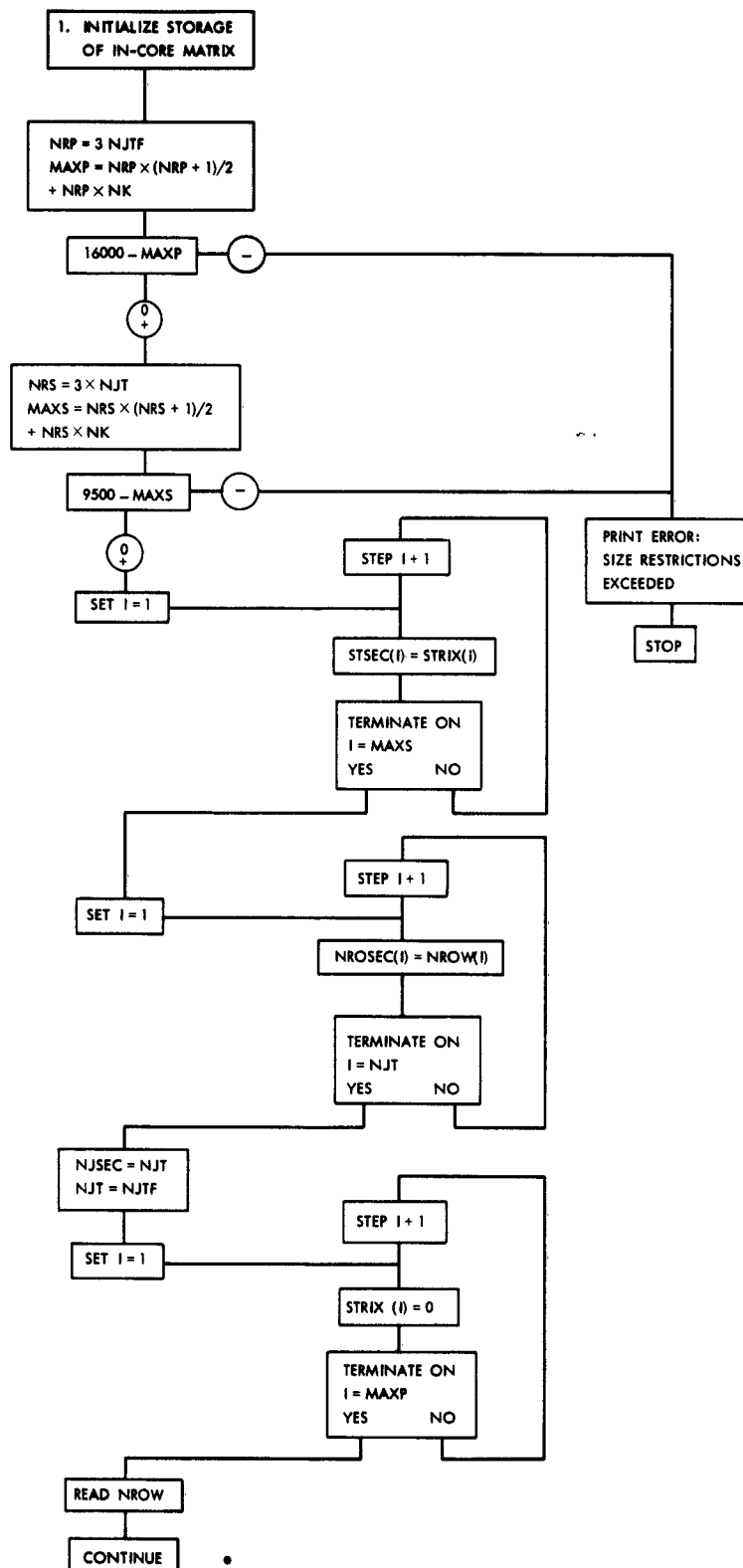


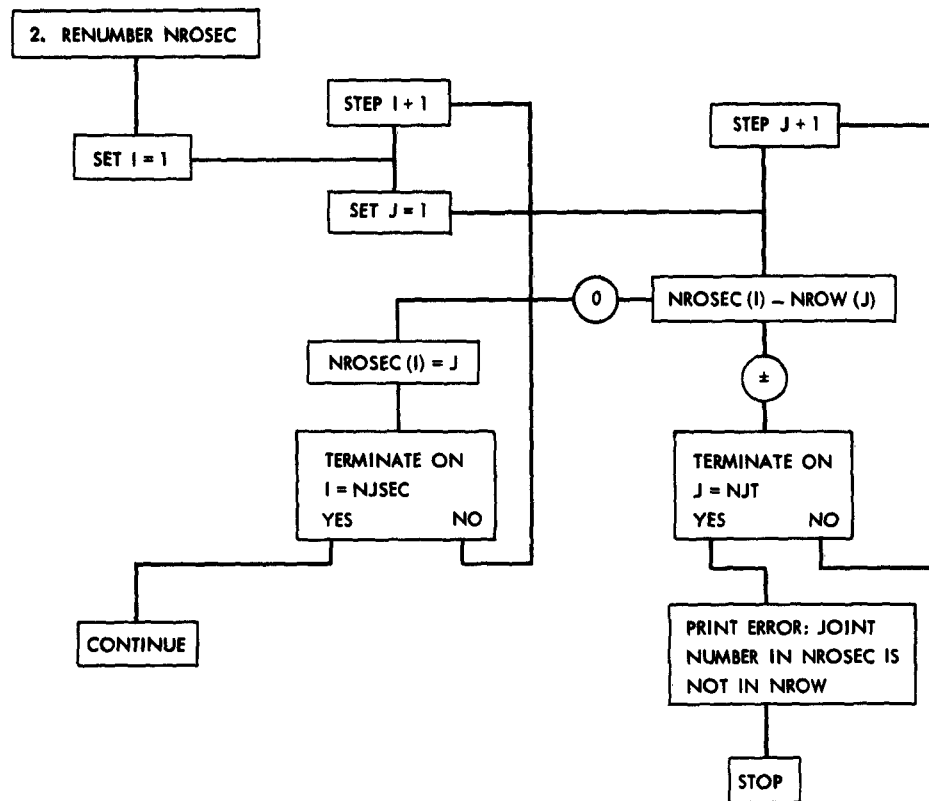
NROW

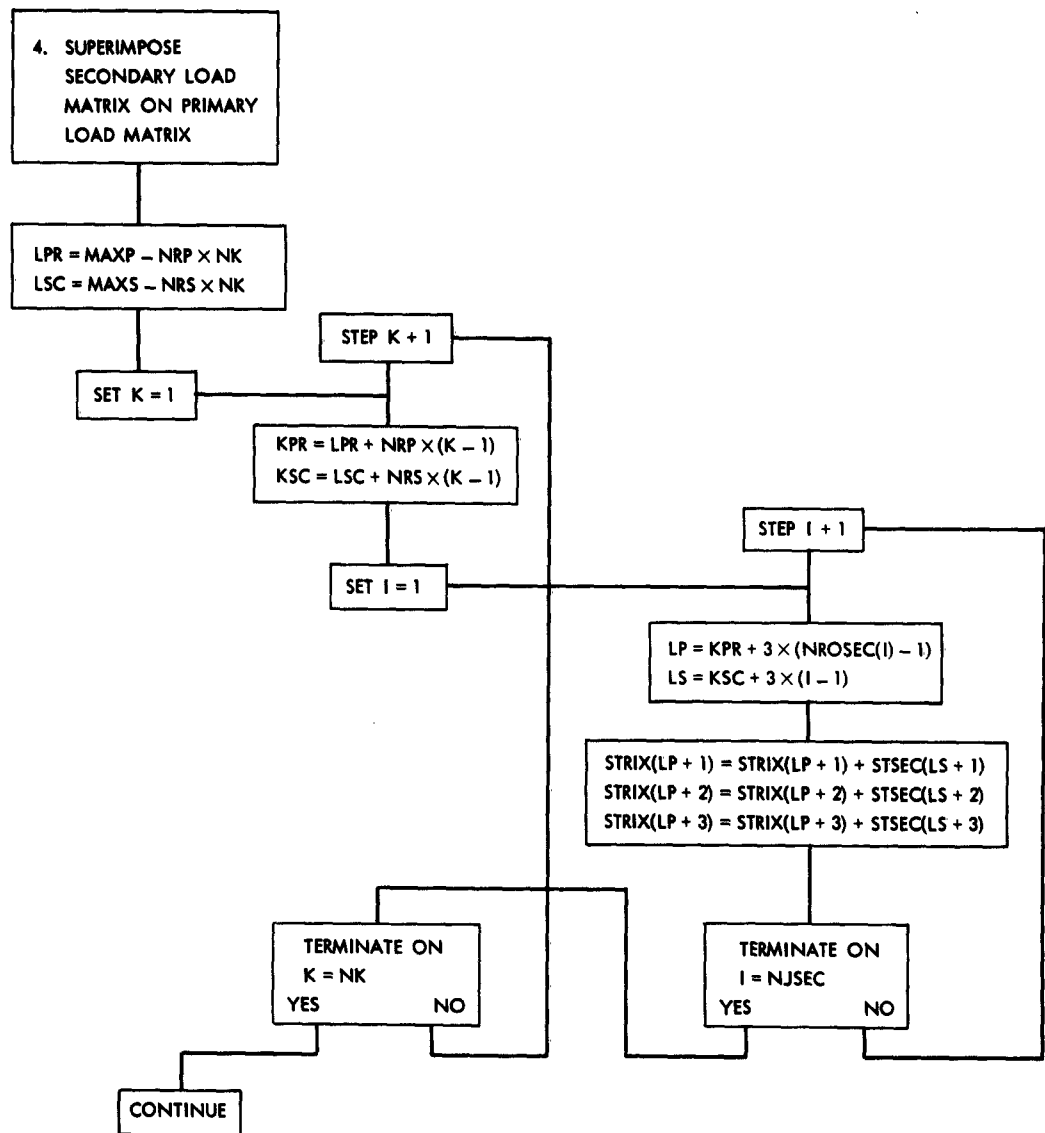


MATRIX SUPERPOSITION BY MADD









5. MATOUT

Operation: Writes a stiffness matrix on tape from core.

Mathematical Method: Matrices and identifying parameters are written as three record groups, in the format of Sec. E.

Initial Condition of Core: Matrix equations may exist in core as stored by STRIX, MATRO, EFFRIX, MADD, MATIN or another MATOUT.

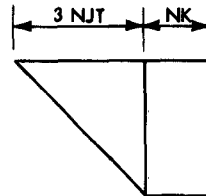
Error Stop: Tape failure searching for tape location preceding location in which MATOUT is to write, or attempt to find a tape location at which nothing has been written.

Output: None.

Final Condition of Core: No change from start of operation, except that NTAPE is nonzero if tape search has failed.

MATOUT-MATIN ARRAY DIMENSIONS:

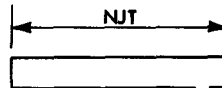
STRIX



NO. OF TERMS

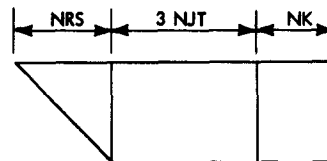
$$\text{MAXM} = 3 \text{ NJT} (3 \text{ NJT} + 1)/2 + 3 \text{ NJT} \times \text{NK}$$

NROW



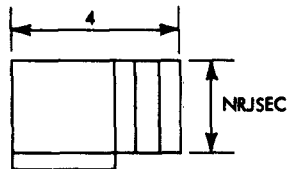
NJT

STSEC



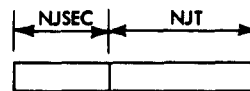
$$\begin{aligned} \text{NRS} &= 3 \text{ NJSEC} - \text{NRCSEC} \\ \text{MAXM} &= \text{NRS} (\text{NRS} + 1)/2 + \text{NRS} (3 \text{ NJT} + \text{NK}) \end{aligned}$$

LRSEC

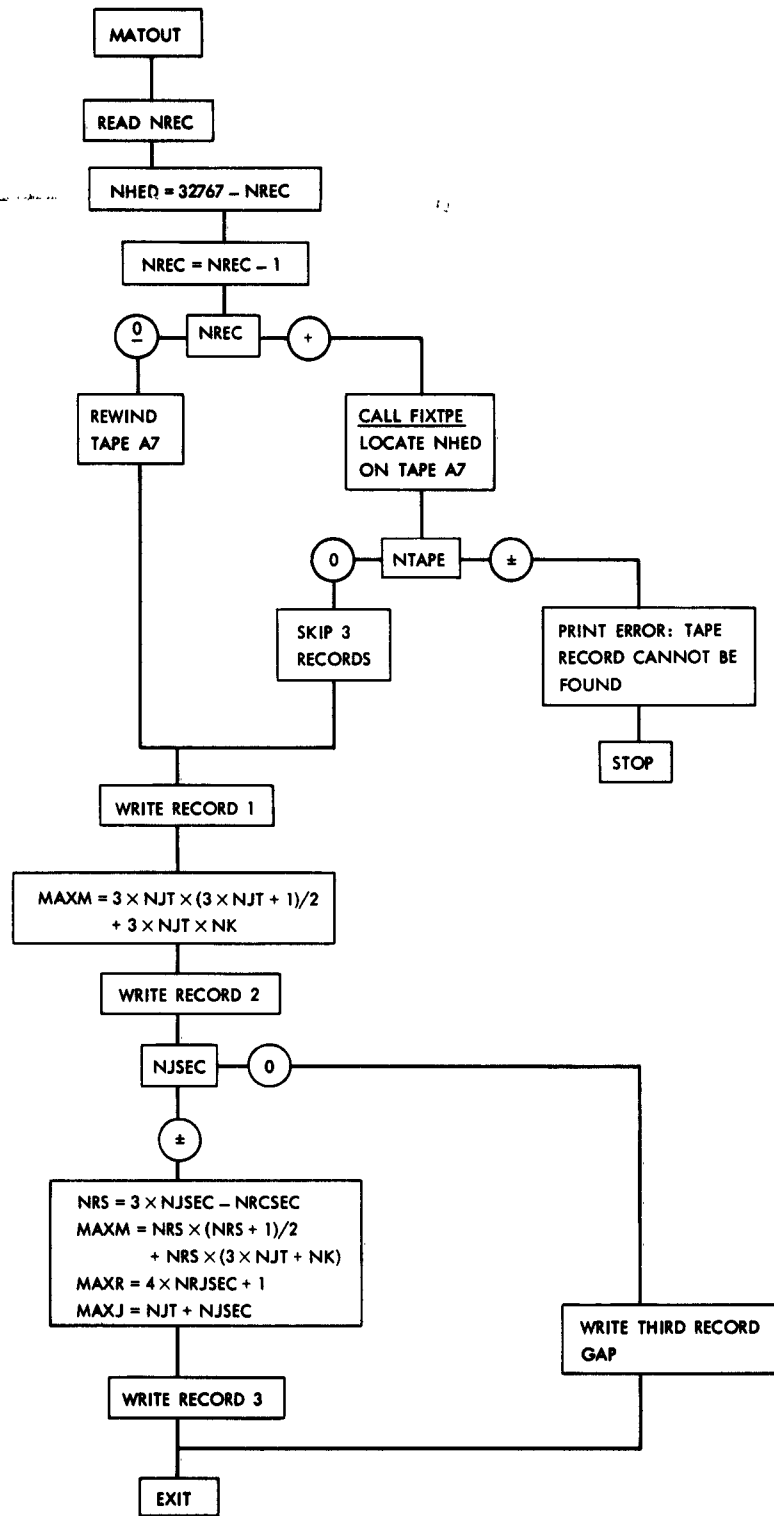


$$\text{MAXR} = 4 \text{ NRJSEC} + 1$$

NROSEC



$$\text{MAXJ} = \text{NJSEC} + \text{NJT}$$



6. MATIN

Operation: Reads a matrix into core from tape.

Mathematical Method: Matrices and identifying parameters are read into core from three record groups in the format of Sec. E.

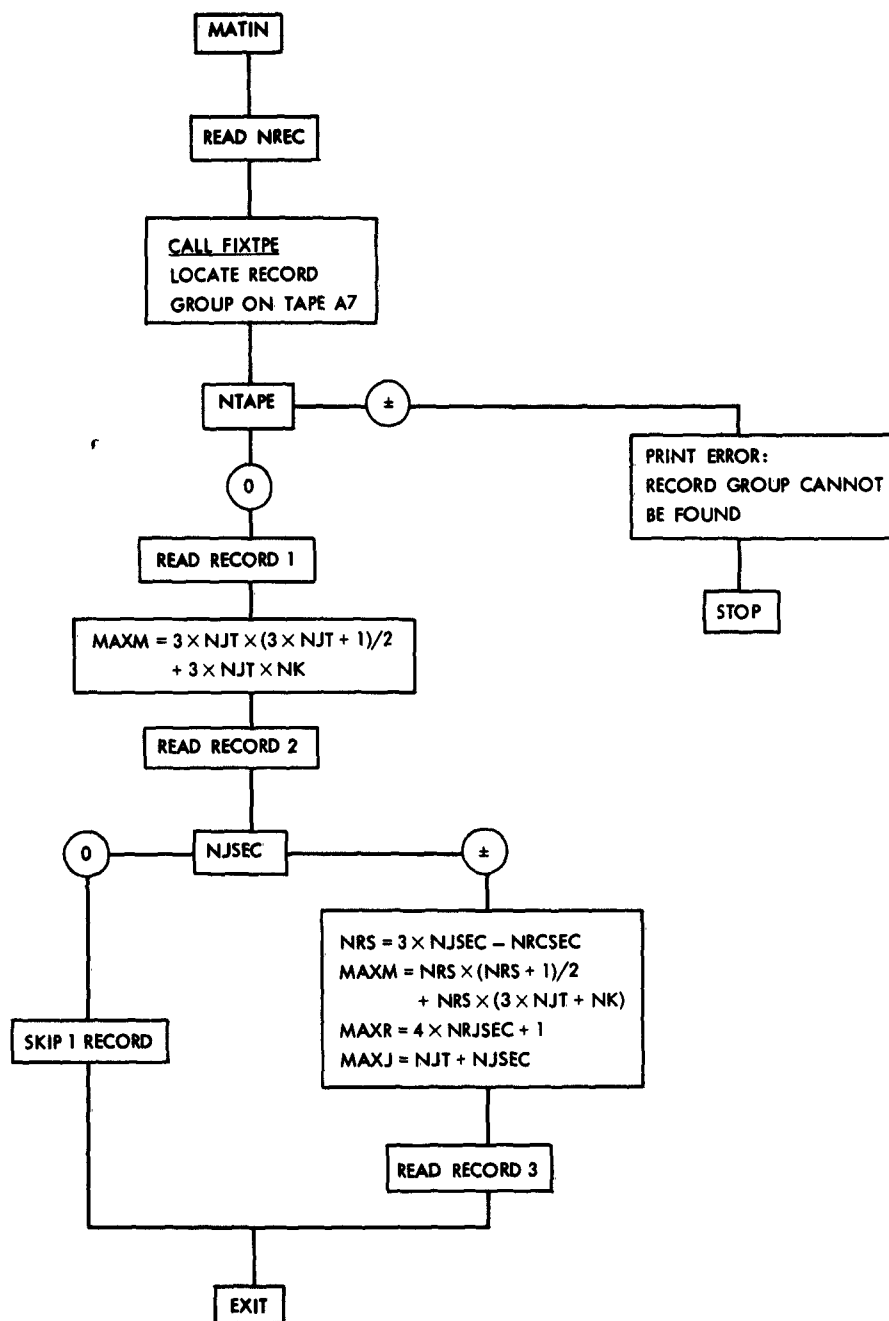
Initial Condition of Core: All matrices and identifying parameters are reset by MATIN.

Error Stop: Tape failure searching for tape location, or attempt to find record which has not been written.

Output: None.

Final Condition of Core: Same as condition when the record group was written by MATOUT (in regard to continuity of operation).

Array Dimensions: See MATOUT description.



7. DEFSOL

Operation: Evaluates joint displacements and reactions from a final stiffness matrix equation in core and a series of residuals and reaction matrices on tape.

Mathematical Method: A set of displacements is determined from the matrix equation in STRIX:

$$KU = F$$

First, if any of the displacement u is restrained by reactions, the row corresponding to the reaction component is removed from K and F and written on tape as reaction matrices (K_R, F_R). K and F are condensed and re-stored in STSEC; K is expanded into square form in STRIX and inverted. The displacement matrix U is formed by multiplication of K^{-1} and F and is stored in STSEC. An addressing system is established, and STRIX is initialized to receive displacements and reactions. After the displacement matrix U is re-stored into STRIX, successive groups of reaction matrices or residuals are read back. Reactions are computed by the equation

$$R = K_R U \text{ and } F_R$$

Displacements, previously eliminated, are evaluated with other displacements, which were eliminated at a later step:

$$U_1 = K_{11}^{-1}(F_1 - K_{12} U_2)$$

Initial Condition of Core: A matrix equation must be in STRIX as left by any of the preceding routines.

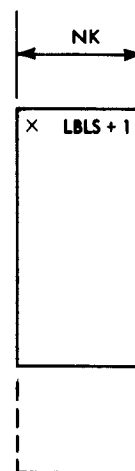
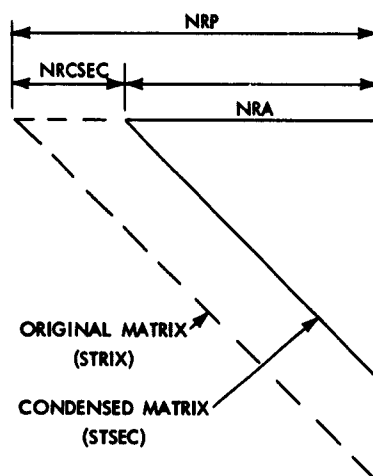
Error Stops:

- (a) Violations of size restrictions, as indicated in the storage diagrams.
- (b) Tape failure or machine failure in comparing reaction joint numbers, read back with a reaction matrix, to reaction joint numbers stored in core.

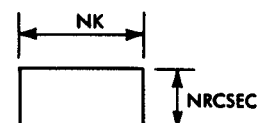
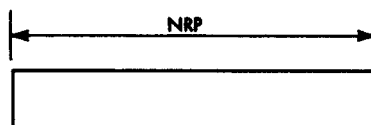
Output: Reaction matrix if reactions are located among final displacements. Displacements and reactions are written on the output tape B2 in the same format in which they are stored in core (one record per joint, giving joint number and u , v and w displacements).

Final Condition of Core: Displacements and reactions are left in STRIX as they were stored. The only operation which may logically follow DEFSOL is BARSOL.

DEFSOL MATRIX ADDRESSING DIAGRAM



REACTION EQUATIONS (STSEC)



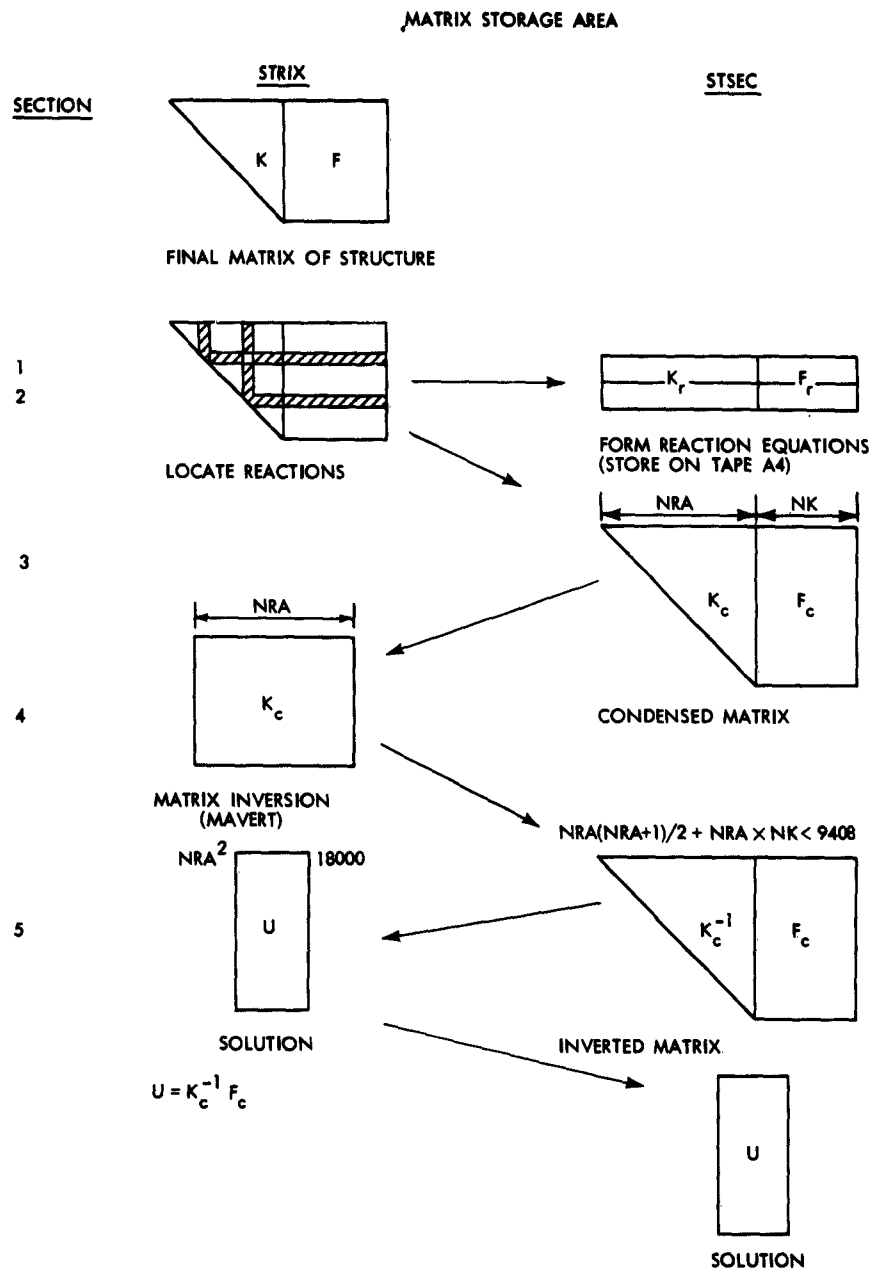
SIZE LIMITATIONS:

$$\frac{NRA(NRA + 1)}{2} + NRA \times NK \leq 9408$$

$$NRA \times NRA \leq 16000$$

$$NRCSEC \leq 10$$

DEFSOL MATRIX STORAGE DIAGRAM



Location	Joint	Number			
	Unit	Joint			
$IUADR_1 = 0$	0 1	0 1	u	v	w
.	.	.	.	.	.
$IUADR_1 + 4(M_1 - 1) + 1$	0 1	M_1	u	v	w
$IUADR_2 = 4M_1$	0 2	0 1	u	v	w
.	.	.	.	.	.
$IUADR_2 + 4(M_2 - 1) + 1$	0 2	M_2	u	v	w
.	.	.	.	.	.
$IUADR_N = 4_1^{N-1} M_1$	N	0 1	u	v	w
.	.	.	.	.	.
$IUADR_N + 4(M_N - 1) + 1$	N	M_N	u	v	w

(a) Displacement storage for one loading

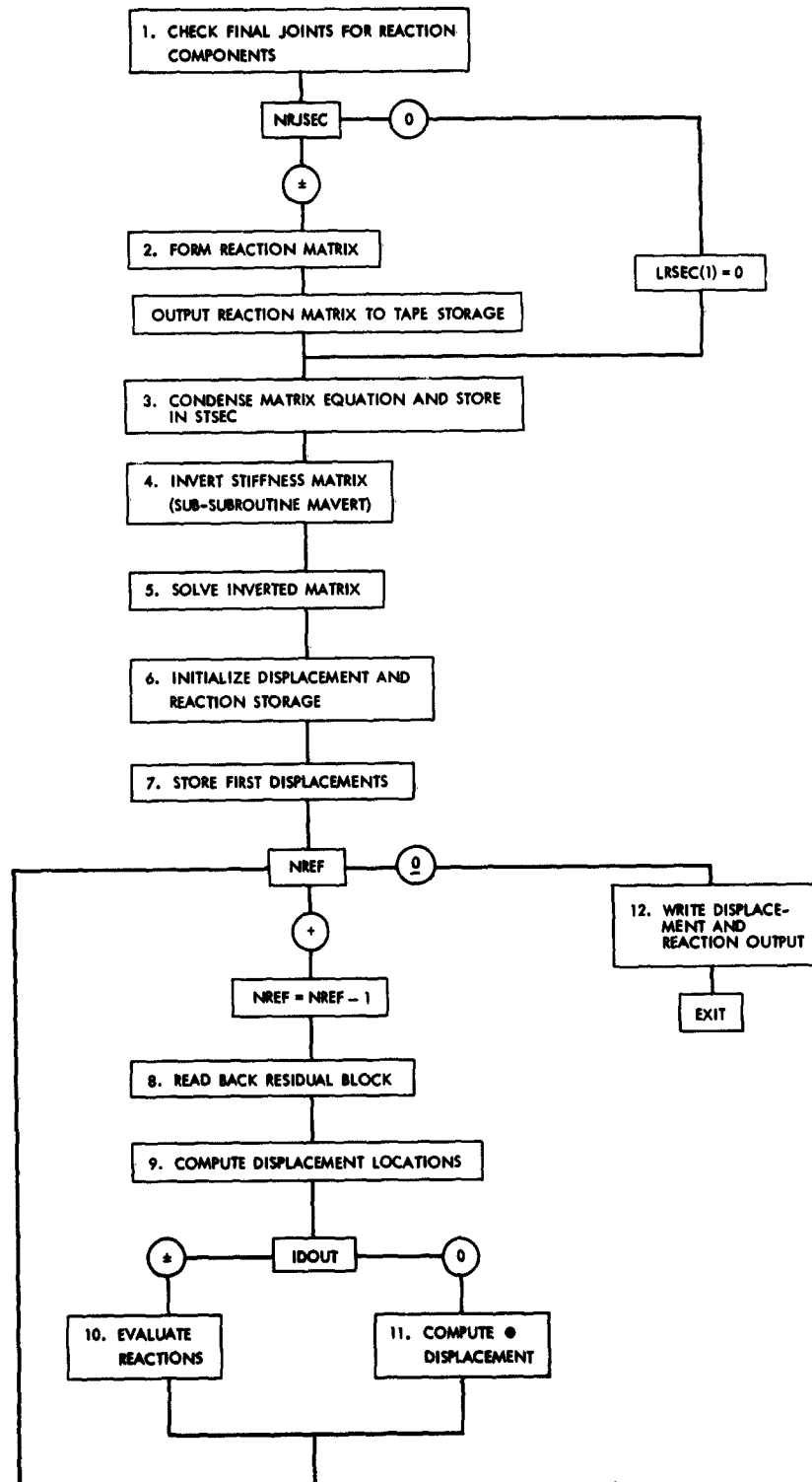
M_i = number of joint numbers prefixed by unit number i.

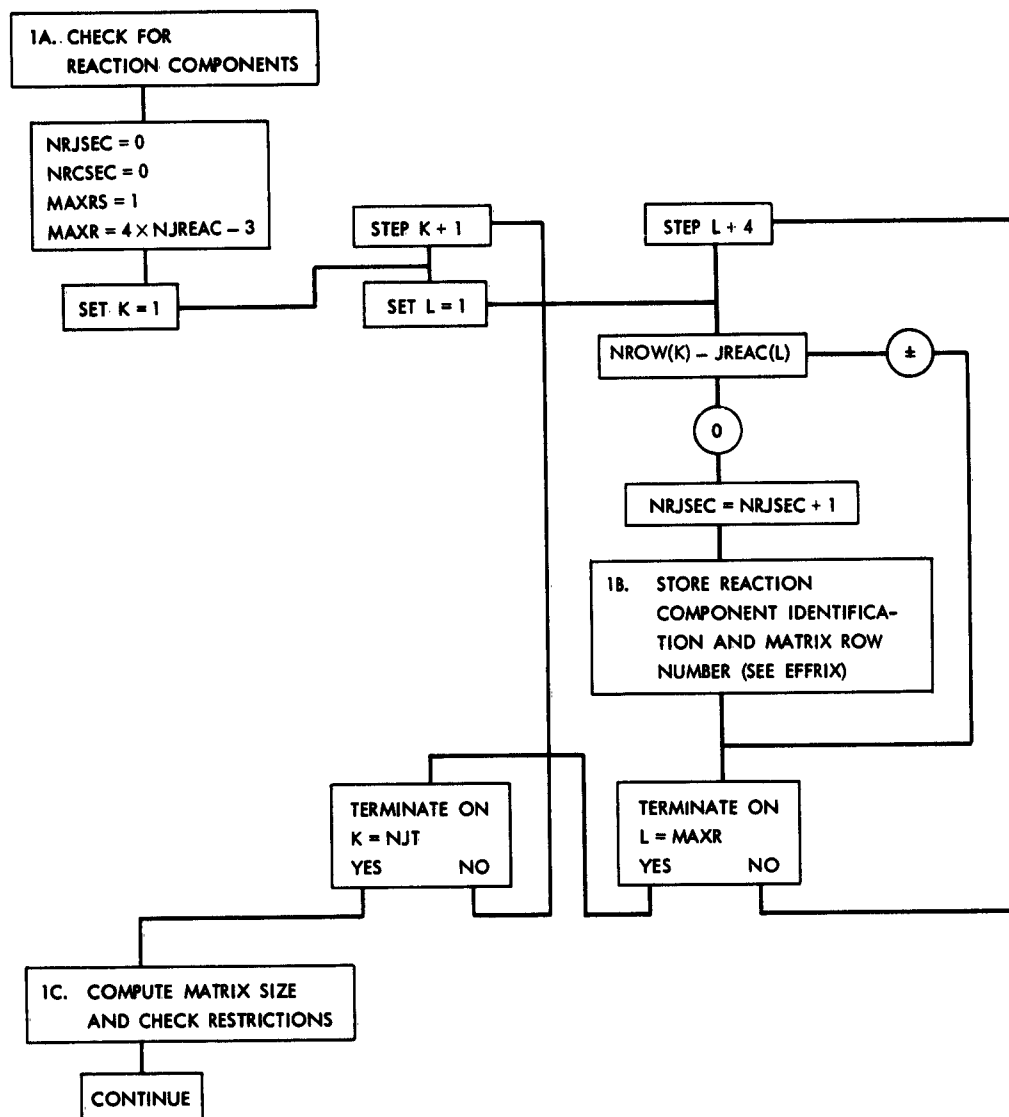
Displacements for loading 1
.
.
.
Displacements for loading NK
Reactions for loading 1
.
.
.
Reactions for loading NK

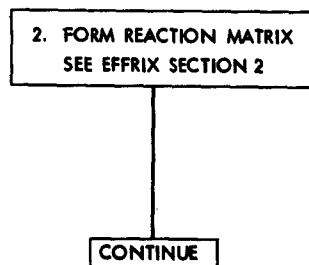
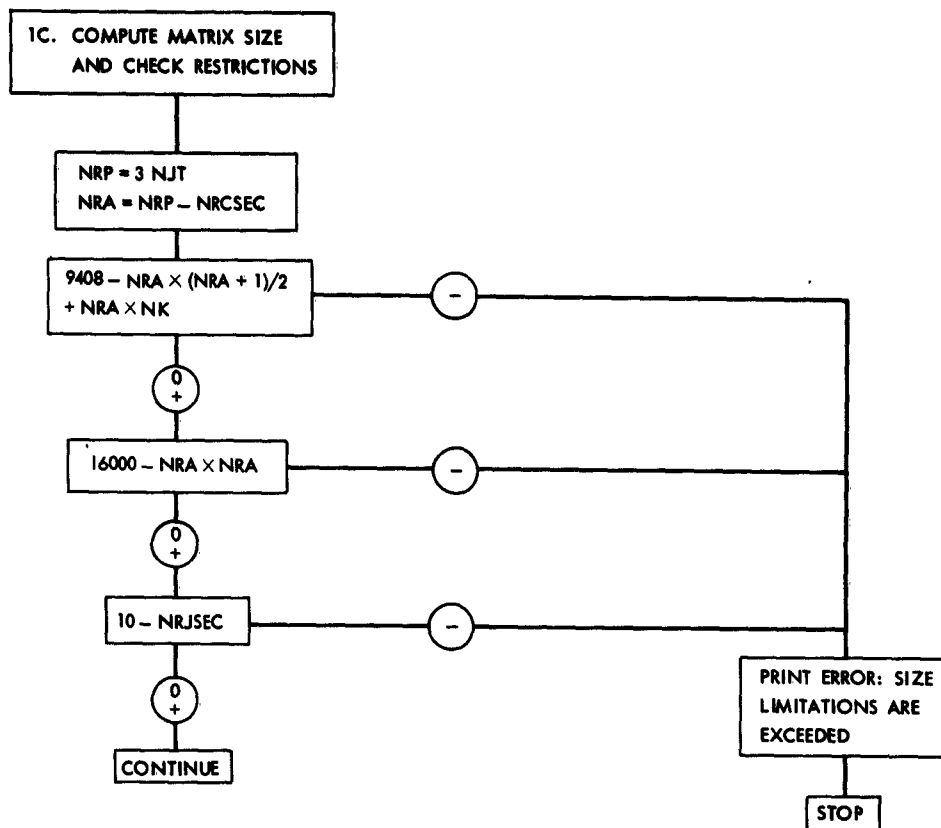
(b) STRIX map for NK loadings

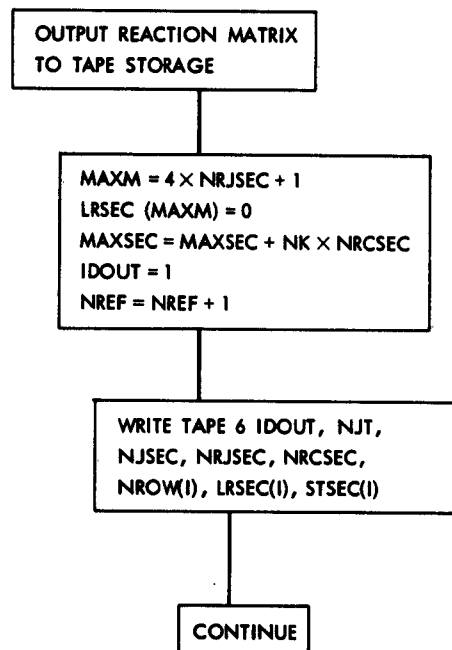
$4 \times NK \times (\text{number of joints} + \text{number of reactions}) \leq 16000$.

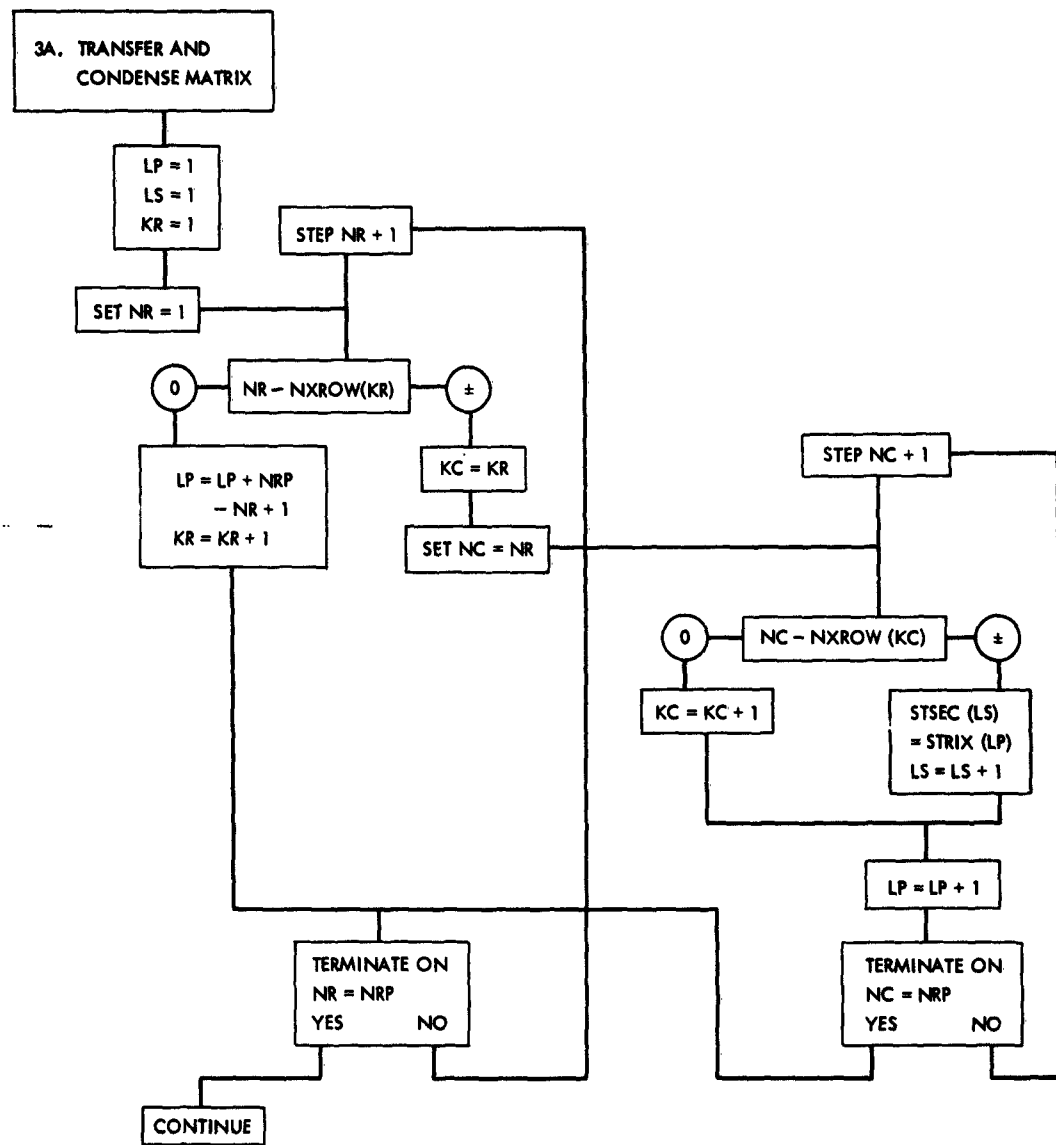
DEFSOL - BLOCK DIAGRAMS

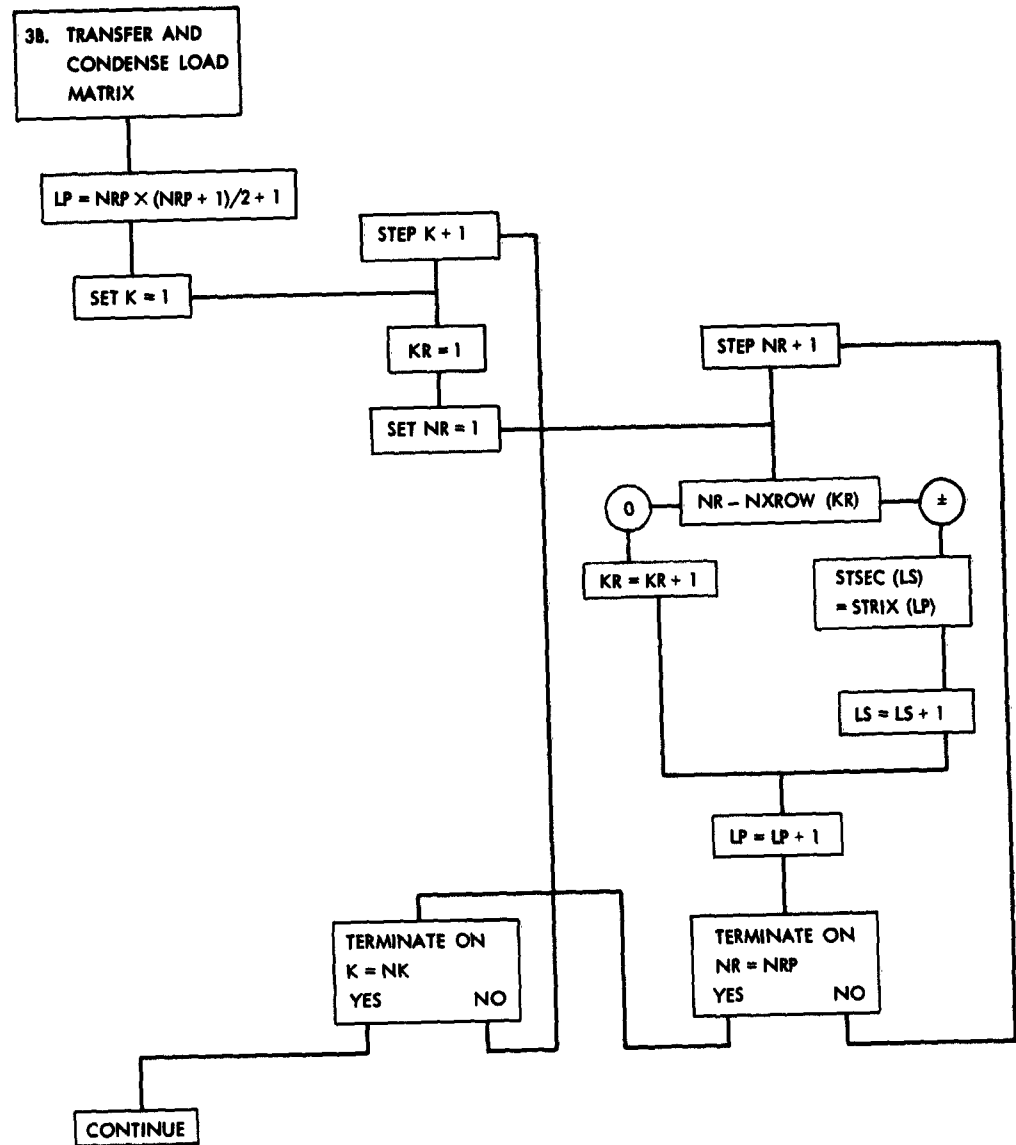


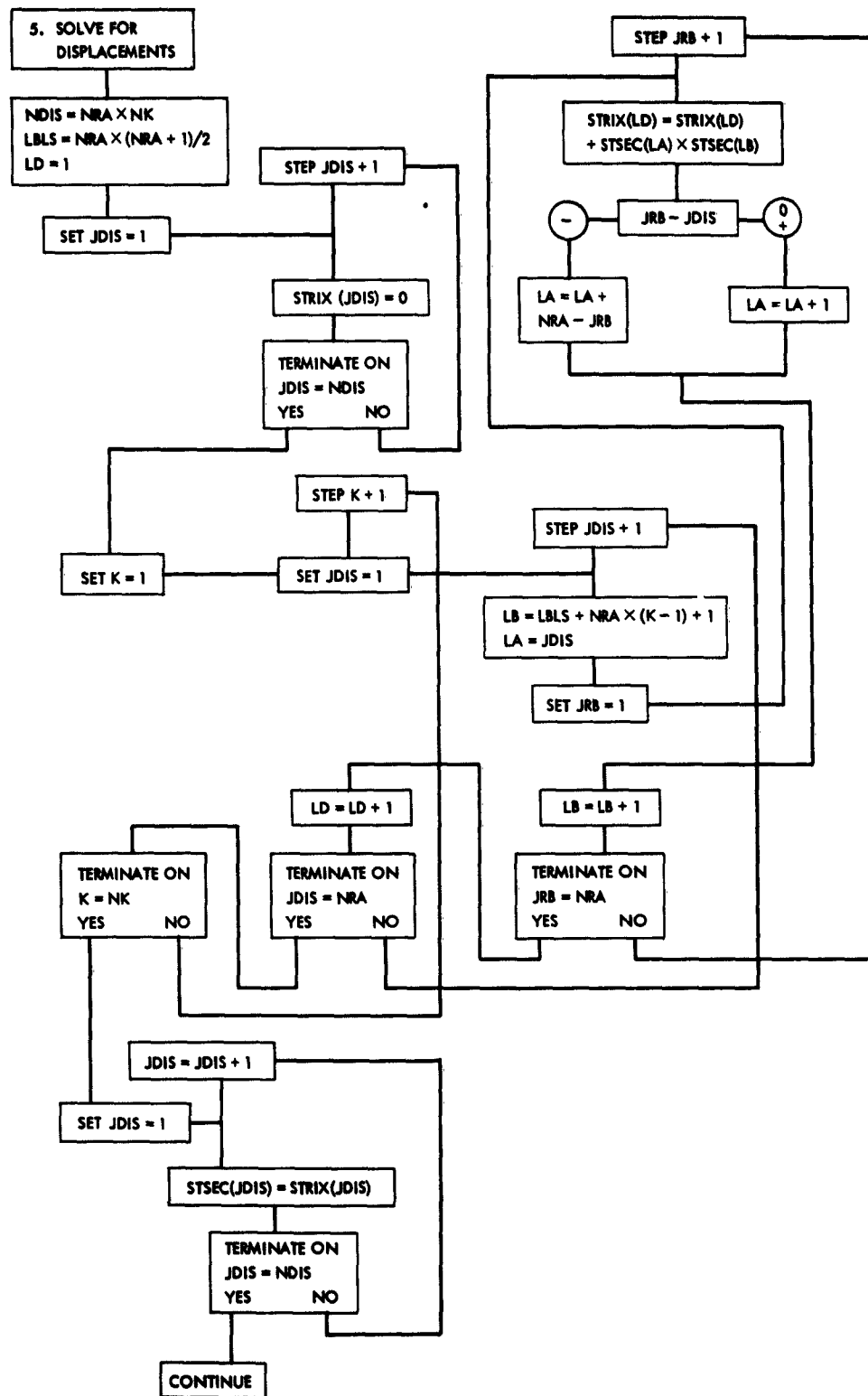


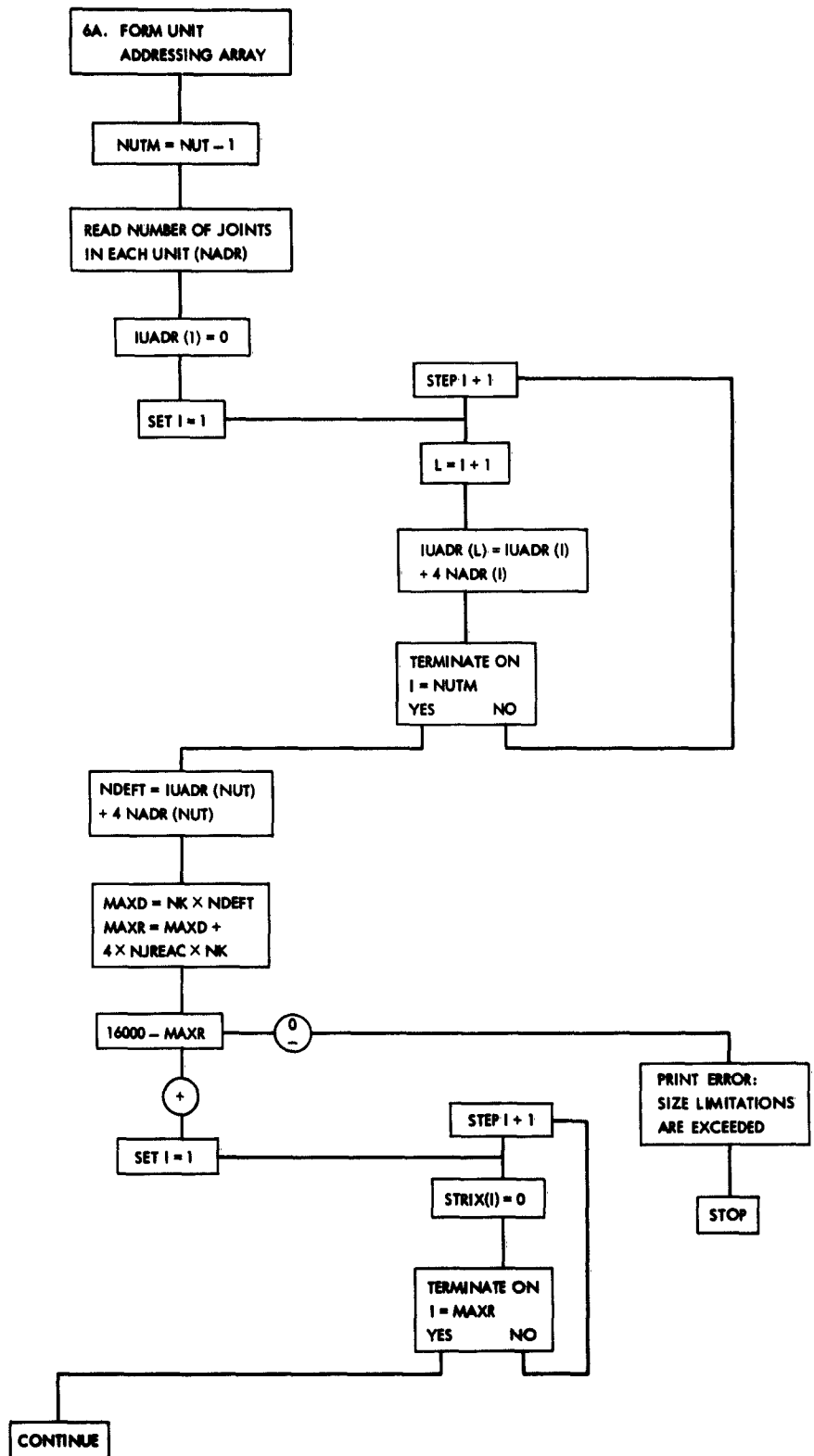


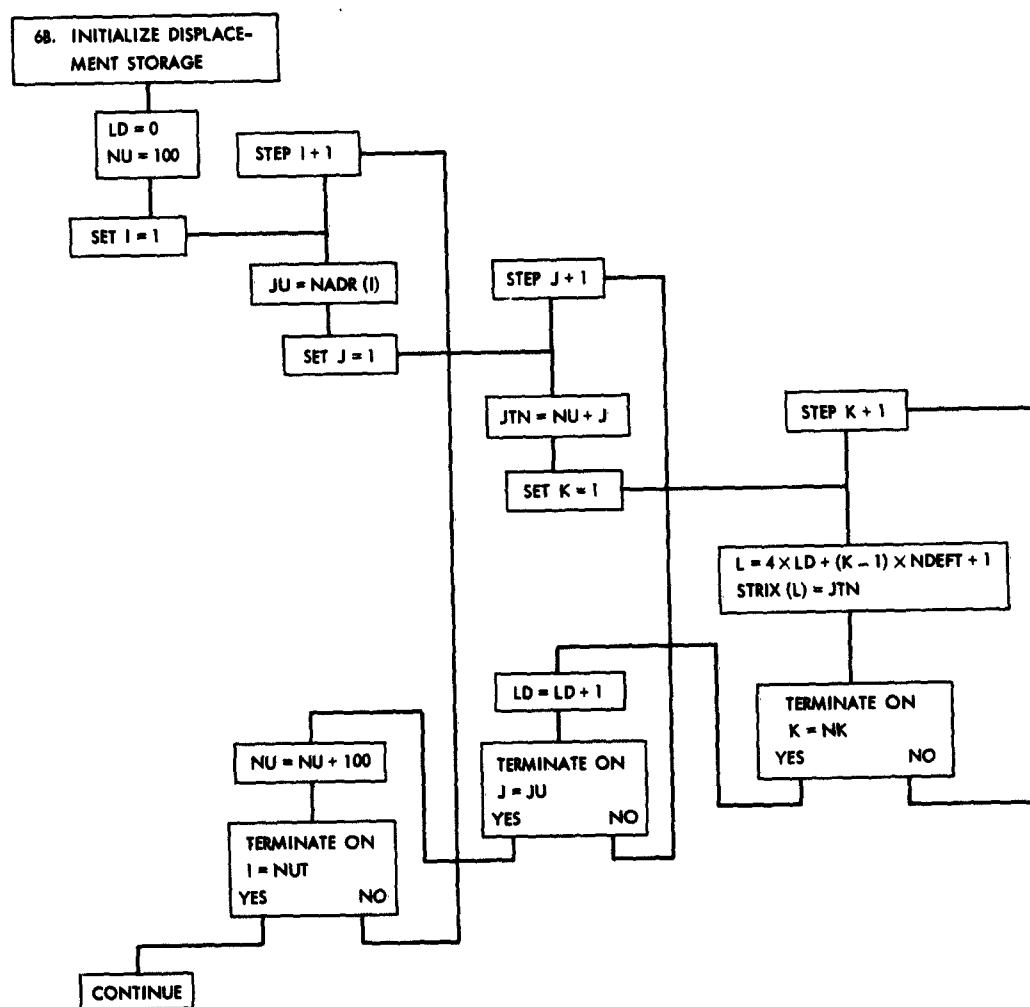


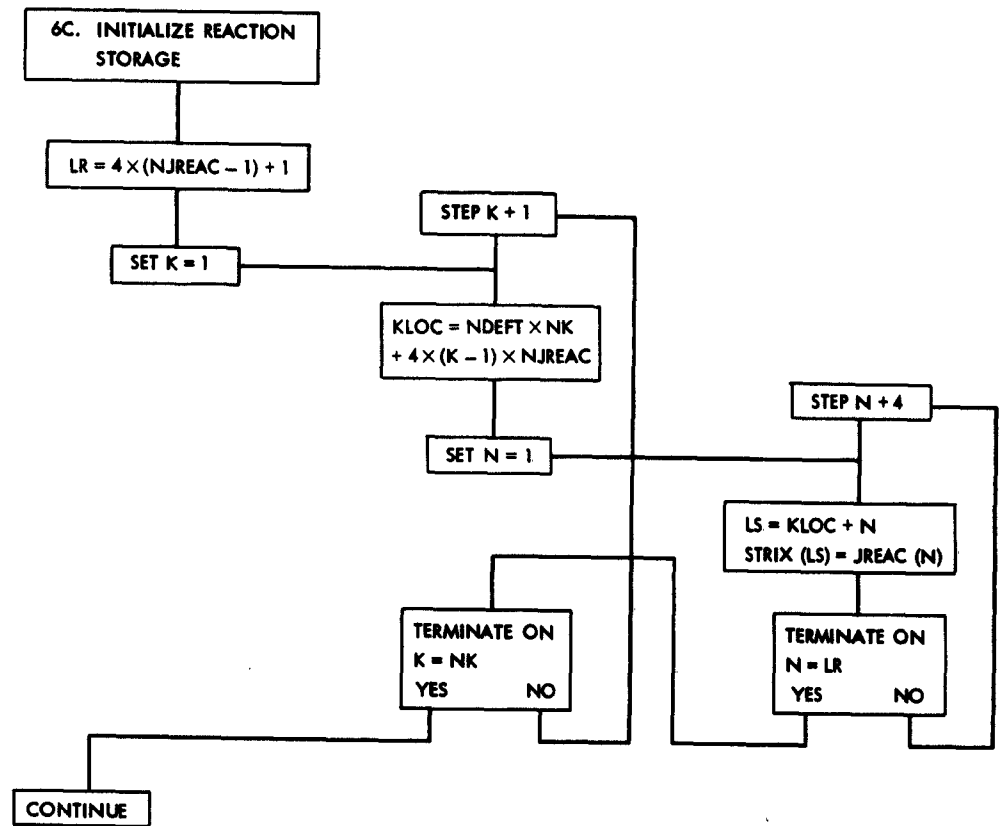


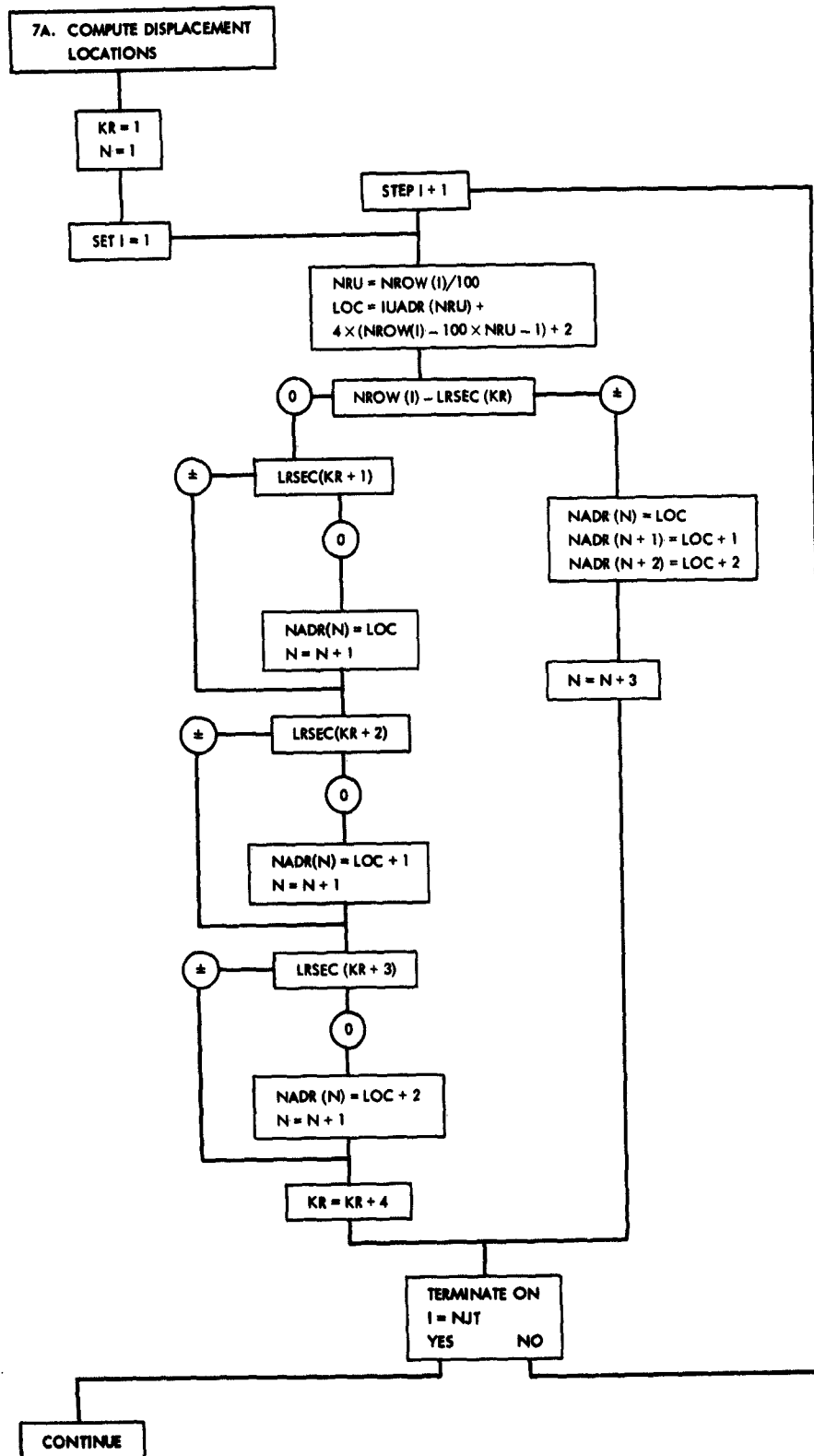


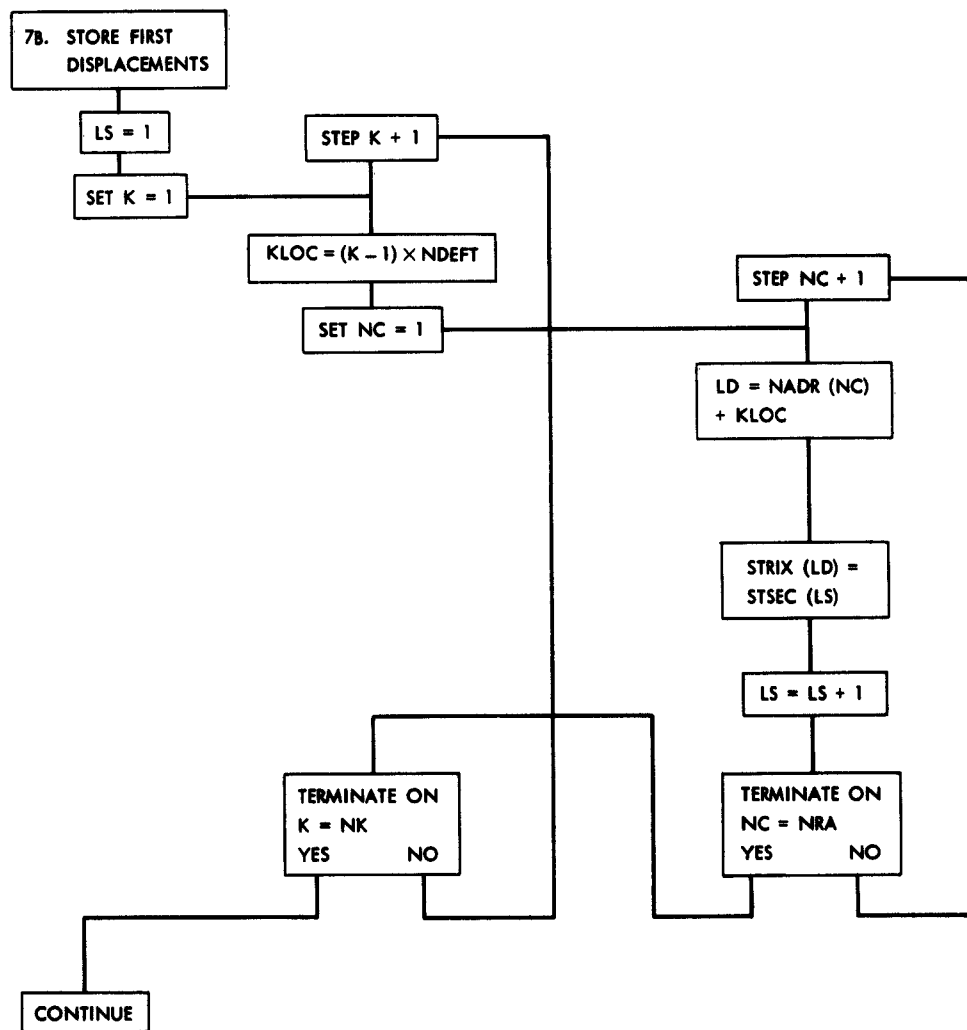


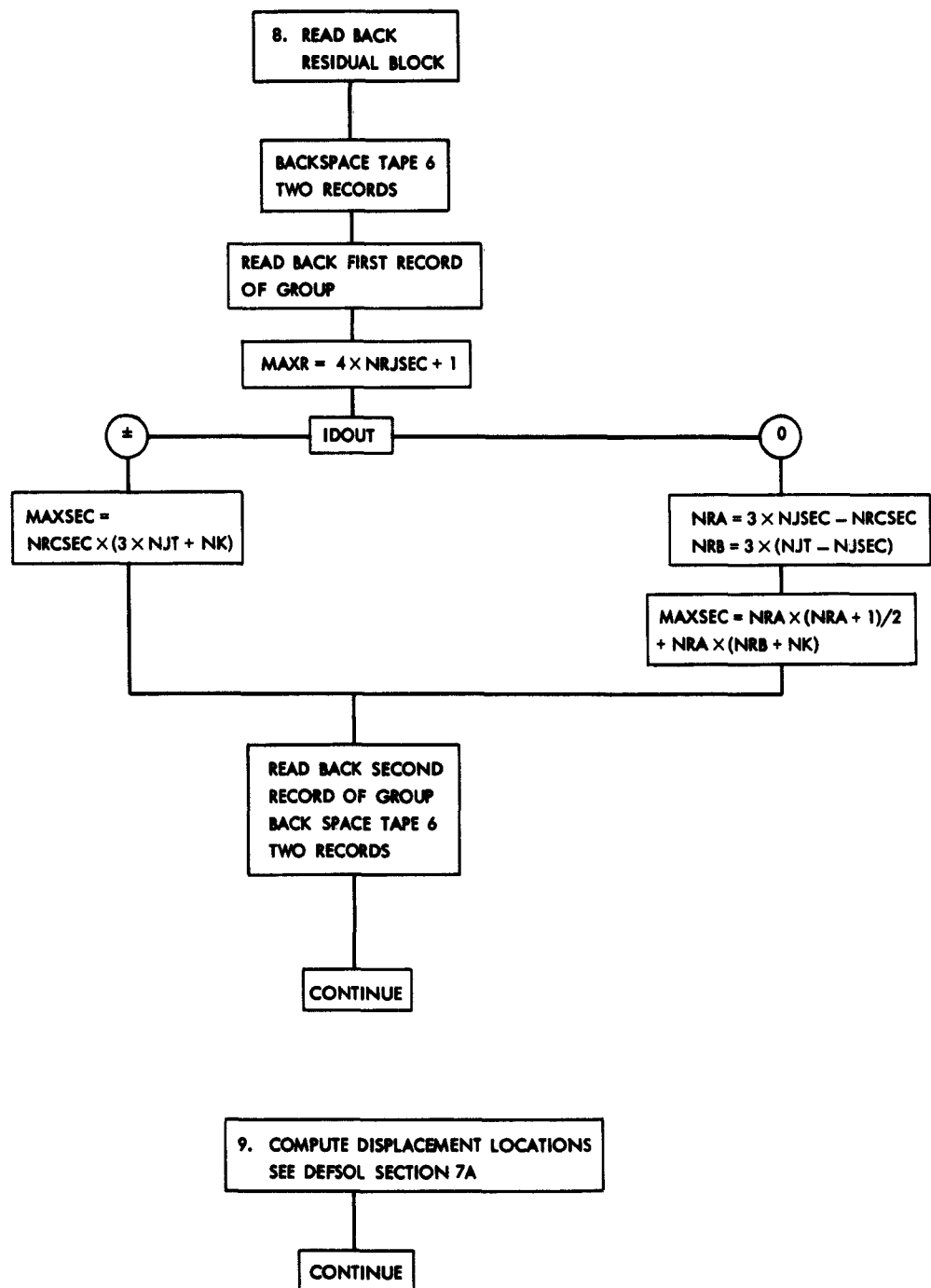


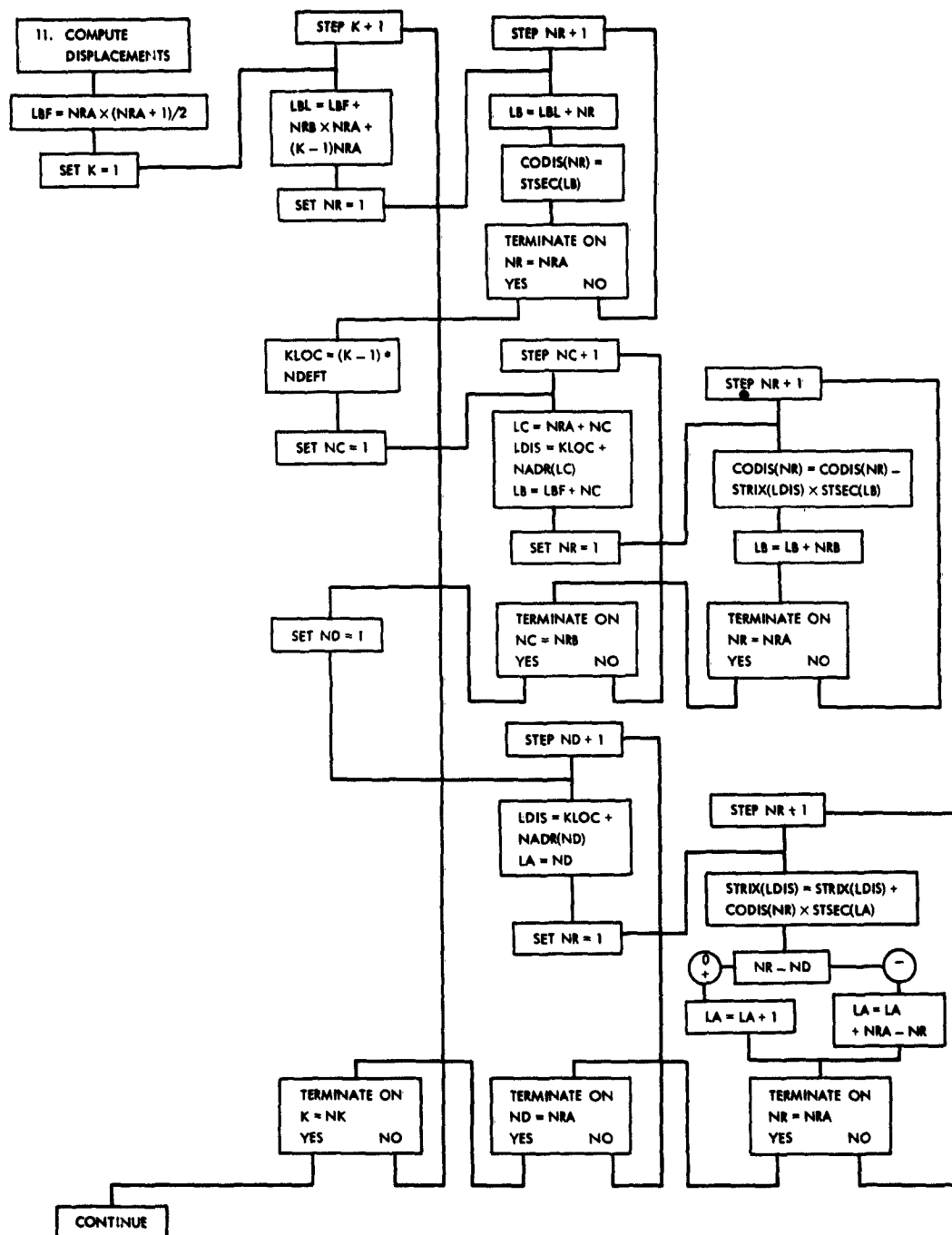


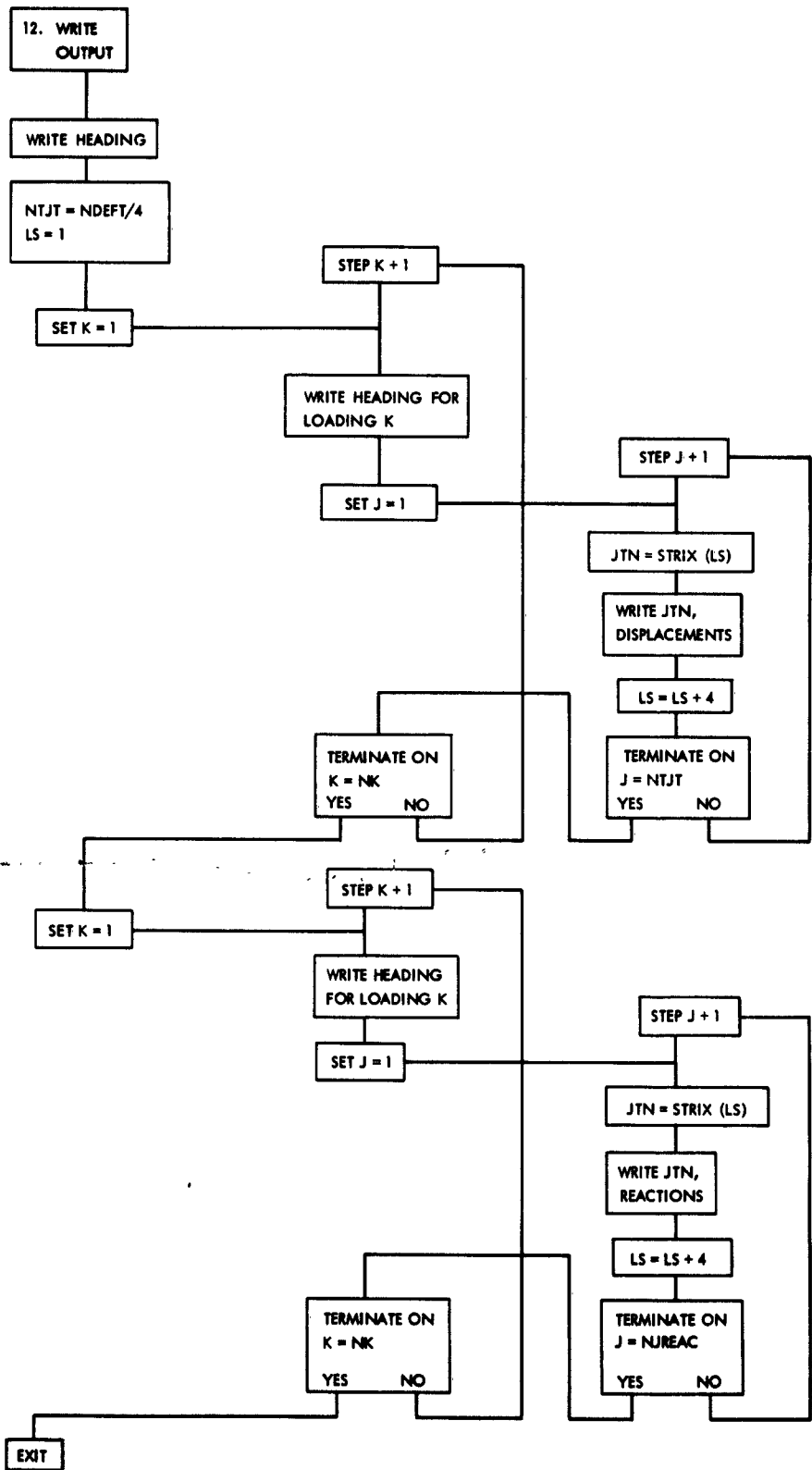












8. BARSOL

Operation: Computes the stress in any bar from the displacements and coordinates of the joints at its ends.

Mathematical Method: Unit stresses are computed and stored for all loading conditions for consecutive bars in an input list of bar numbers.

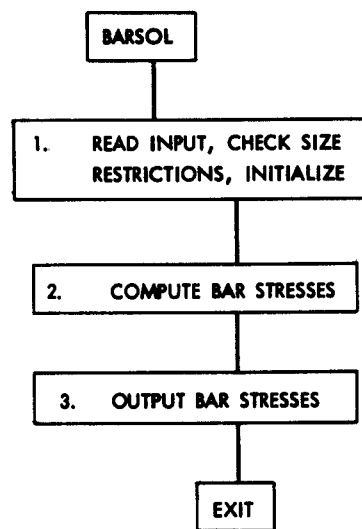
Initial Condition of Core: Common variables NK and NUT must be present. Joint displacements in STRIX and the addressing array IUADR must be as left by DEFSOL.

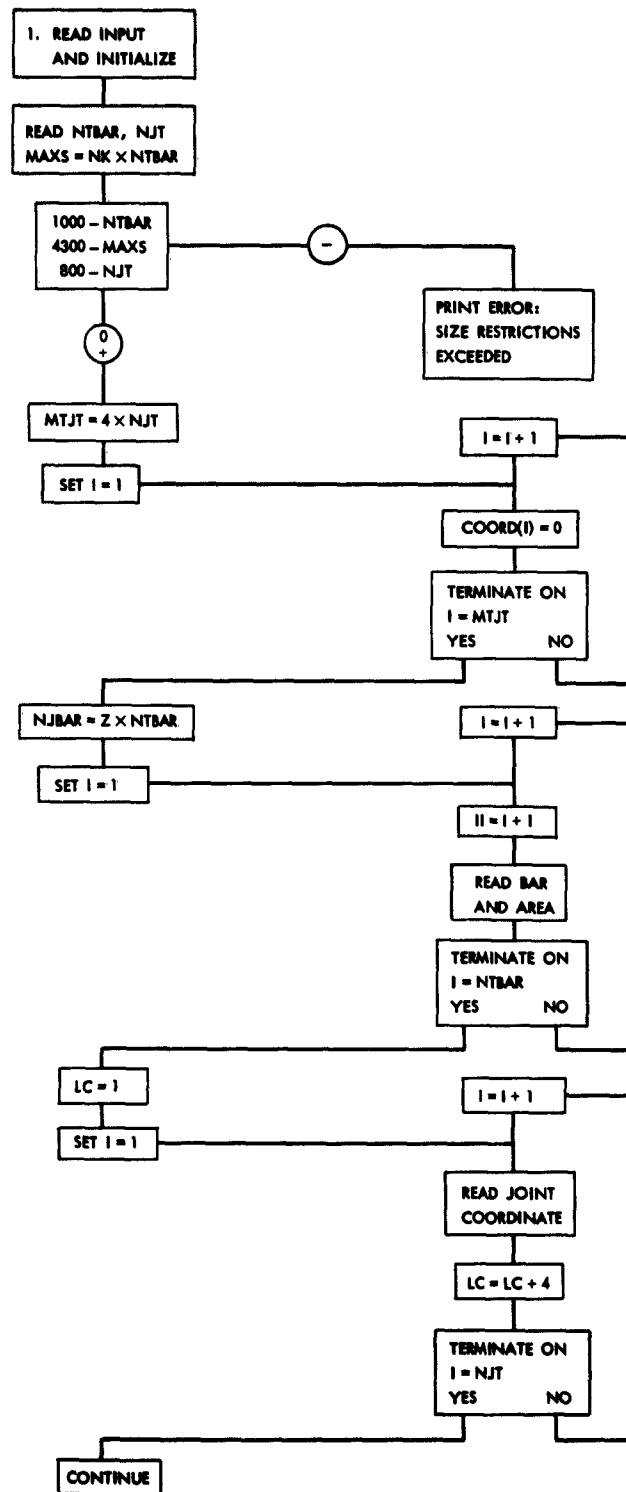
Error Stops:

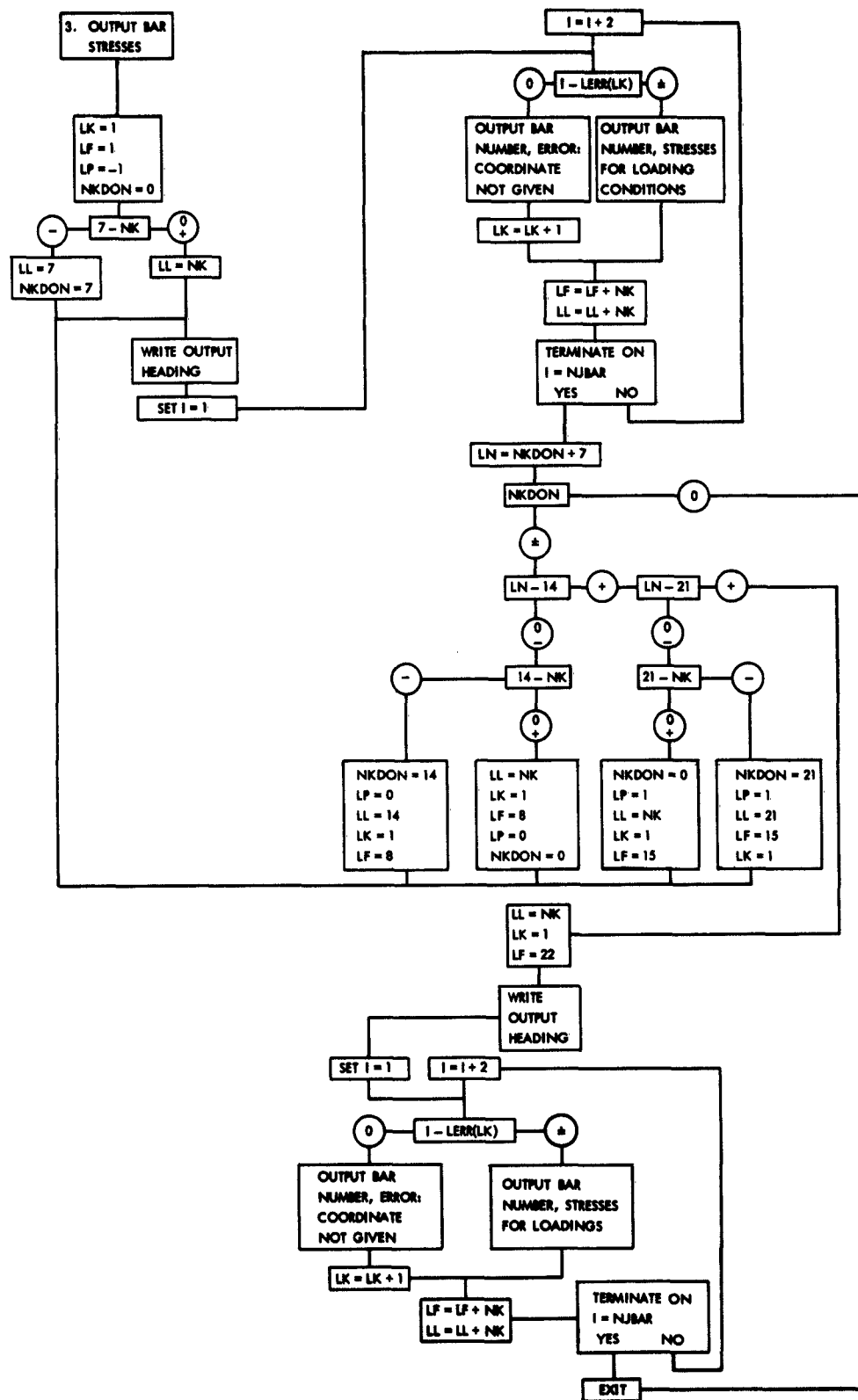
- (a) Number of bars > 1000 .
- (b) Number of bars $\times$ NK > 4300 .
- (c) Number of joints at ends of bars > 800 .
- (d) No stop occurs if an illegal bar number is given or if a needed set of coordinates is not supplied, but the error is indicated in the output.

Output: Bar stresses are written on tape B2 for listing.

Final Condition of Core: Common locations above STRIX are not changed.







9. CHECK

Operation: Sums the bar forces in the three coordinate directions at each joint for each loading condition.

Mathematical Method: The x, y and z forces in each bar are read from tape A7 and superimposed in the correct locations representing the joints to which the bar is connected.

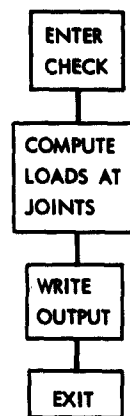
Note: Since the bar forces are computed from the actual compatible deflections of the structure, that equilibrium is satisfied at each joint becomes a necessary and sufficient condition that the final solution is unique.

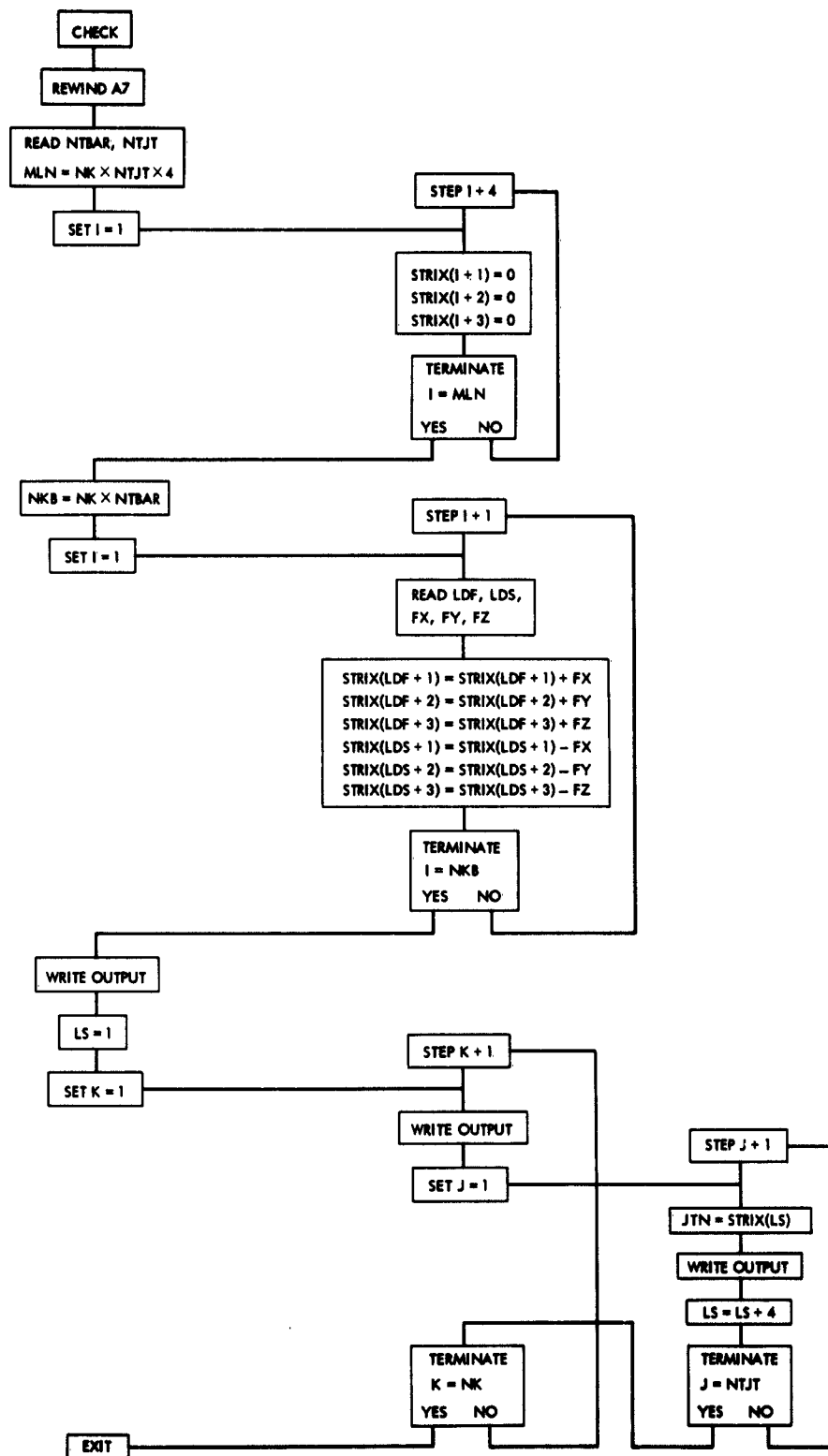
Initial Condition of Core: Common variables IDP and NK must be present.

Error Stops: None.

Output: Sum of all the bar forces at each joint are written on tape B2 for listing.

Note: CHECK is simply an internal check of consistency of operation. It indicates that a structure has been analyzed correctly; however, it may not be the desired structure if some error has been made consistently throughout the analysis, e.g., a joint coordinate that is wrong in both STRIX and BARSOL will not be shown as an error.





G. STAIR SERVICE ROUTINES

1. Master Routine

Operation:

- (a) Initializes various storage locations at the start of each problem.
- (b) Reads, checks and stores a block of initial data for each problem.
- (c) Interprets an input call for each step of an ANALYSIS PROGRAM; requests MISLAM to read the routine called into core, and transfers to the entry point of that routine.

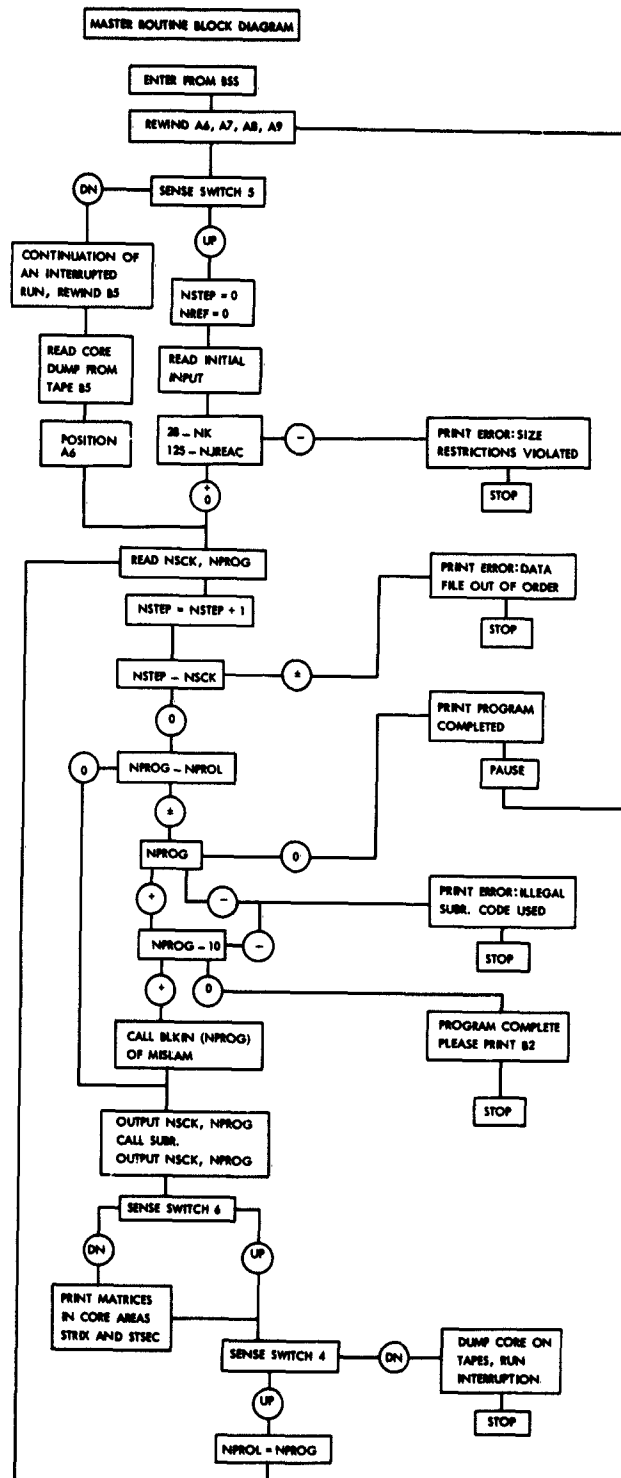
Error Stops:

- (a) $NJREAC \geq 125$.
- (b) $10 \leq NPROG < 0$ (NPROG is the code number of a STAIR operating routine on a call card).
- (c) Call cards out of consecutive order (may be caused by input error at preceding step).
- (d) $NK > 28$.

The Master Routine serves as the main program when the STAIR System is loaded by a BSS loader. Control passes directly to it when transfer vector processing is completed.

The Master Routine transfer vector contains the entry points of the nine operating routines, the MISLAM entry BLKIN which reads processed blocks from the program tape and library I-0 routines. Transfer to each of the former routines is handled by the standard FORTRAN II CALL statement, which results in a TSX to a TTR in the transfer vector.

Output: In-core matrices after any step, if Sense Switch 6 is down.



2. FIXTPE

Operation: Positions any tape on DSC A to read a record m, identified by a heading word 32767-N (77777-N octal).

Called by: STRIX, MATRO, MADD, MATIN and MATOUT.

Arguments: N, DSC A logical tape NU, number of binary data block on tape N.

Method: The first heading word encountered is compared to the desired heading word; the tape is back-spaced or forward-spaced according to the work input. If an end-of-file is encountered in forward-spacing or if the heading words are out of consecutive order, the tape is rewound and forward-spaced.

Error Return: If the tape is rewound and a second attempt to find the correct heading word is not successful, common variable NTAPE is set equal to one, and control returns to the calling program. The calling program senses the value of NTAPE and prints an on-line diagnostic.

Normal Return: When a correct heading word is read, the tape is repositioned at the start of the record containing that heading word, NTAPE is zeroed and control returns to the calling routine.

Notes: Binary records written by a FORTRAN program always begin with a zero word, followed by the list in the "WRITE" statement. No fixed-point variable written by a STAIR program may exceed 18000₄₀; heading words are thus safely distinguished from any other legal number.

3. MAVERT

Operation:

- (a) Expands a matrix stored in triangular array at the head of STSEC into square form with heading words before each row at the head of STRIX.
- (b) Inverts the matrix at the head of STRIX (using SHARE routine CLMIV-2).
- (c) Contracts the inverted matrix back into triangular array without heading words and stores it at the head of STSEC.

Called by: EFFRIX and DEFSOL.

Arguments: STRIX, STSEC, NRA (NRA is the dimension of the matrix to be inverted).

Error Stop: Underflow or overflow during a floating point operation causes transfer to a floating point trap routine. If the number can not be normalized, a stop will occur in the trap routine without an on-line diagnostic. If one or more joints among those to be EFFRIXed or any of the joints in DEFSOL are still planar, then a stop will occur in MAVERT without a diagnostic.

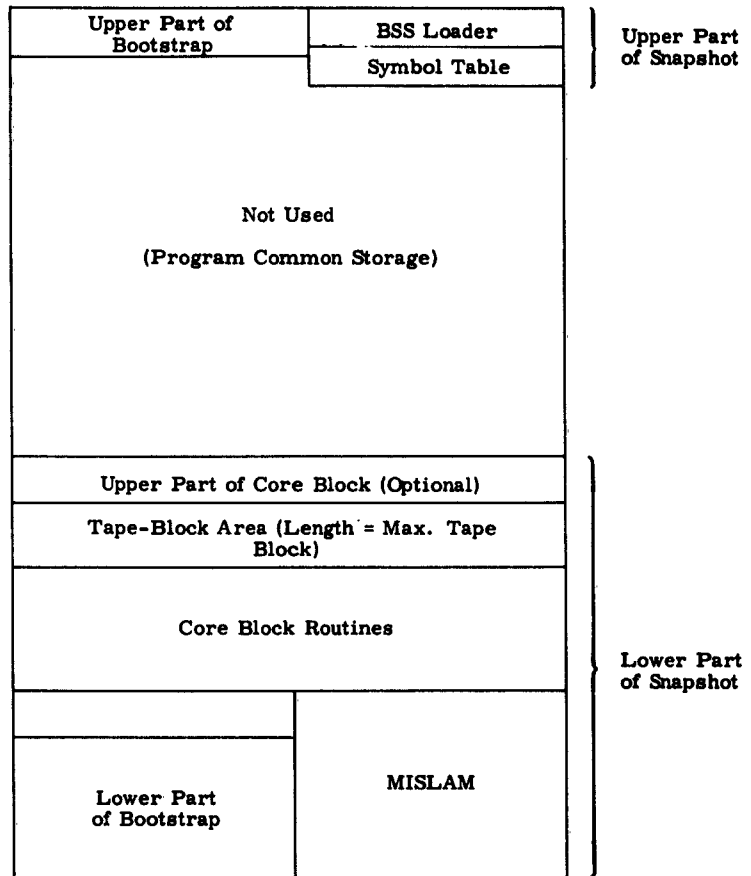
Notes: MAVERT was originally assembled for IBM-704, and, since there is no input or output, the binary deck of the 704 version is usable on IBM-709 or -7090. Several TQO instructions in the program are treated as NOP's by the 709 and 7090, which do not use the TQO instruction.

4. MISLAM for IBM-7090

Operation: Processes and writes a sequence of routines on a program tape from their relocatable binary decks. All these routines must be in the form of standard FORTRAN II sub-routines except, of course, the main program (master routine). During the running of a problem, certain of these routines may be read back and stored permanently in core. Others may be read back individually into a single core area when called by an in-core program.

Definitions:

- (a) Routines stored permanently in core make up a "core block."
- (b) Routines stored temporarily in core are called "tape blocks."
- (c) T/ cards are special transfer cards which separate various blocks in the binary program deck. T/0-T/4 cards have a 9-row punch in column 1 and 0-4 in the address field of the 9-left word.
- (d) A "snapshot" is a tape record which contains all the programs written in core at the time the snapshot is taken. The lower part of the snapshot contains the MISLAM program, a group of core-block routines, the tape block last loaded and, possibly, a second group of core-block routines. The upper part of the snapshot contains the BSS loader and program symbol table.
- (e) A "bootstrap" is a self-loading program written as the first record on the program tape. The lower part of the bootstrap locates the last snapshot written on the program tape. The upper part reads that snapshot into core and transfers to the entry point of the MISLAM program.
- (f) "Dummy records" are two-word records separating bootstrap, snapshot and tape-block records. Their presence allows some saving in time during backspacing. Two words are needed to allow a program delay during a tape search.
- (g) An "end record" is a two-word record which marks the last information written on the program tape. No reading operation may pass an end record.



Note: The bootstrap is covered by the snapshot it reads in.

Functioning of T/ Cards: When the MISLAM program or a snapshot is read into core, control passes almost at once back to the BSS* loader. The BSS reads cards from the on-line card reader. Instruction and program cards are interpreted in the normal manner with the current relocation. When a transfer card is read, control passes back to MISLAM, which branches to one of the following options:

a. T/1 Card

A T/1 card initiates a tape block.

If the preceding block was a tape block (i.e., if another T/1 card was read previously), that block is written on tape with its transfer vector unprocessed. Relocation is reset to the beginning of the tape-block area, the tape-block area is cleared and control returns to the BSS.

If the preceding block was not a tape block (it must then be the lower core block), the starting location of all tape blocks is set before control passes back to the BSS.

The length of the tape-block area is set with the length of the longest tape block read.

b. T/2 Card

A T/2 card initiates a core block.

If the preceding block was a tape block, it is written unprocessed on tape. Relocation is set to the end of the longest tape block, and control returns to the BSS.

c. T/3 card

A snapshot is written on the program tape, and control returns to the BSS.

d. T/0 card

This is the normal FORTRAN II transfer card.

If the preceding block was a tape block (i.e., if the preceding T/ card was a T/1 or a T/2 preceded by a T/1), it is written unprocessed on the program tape.

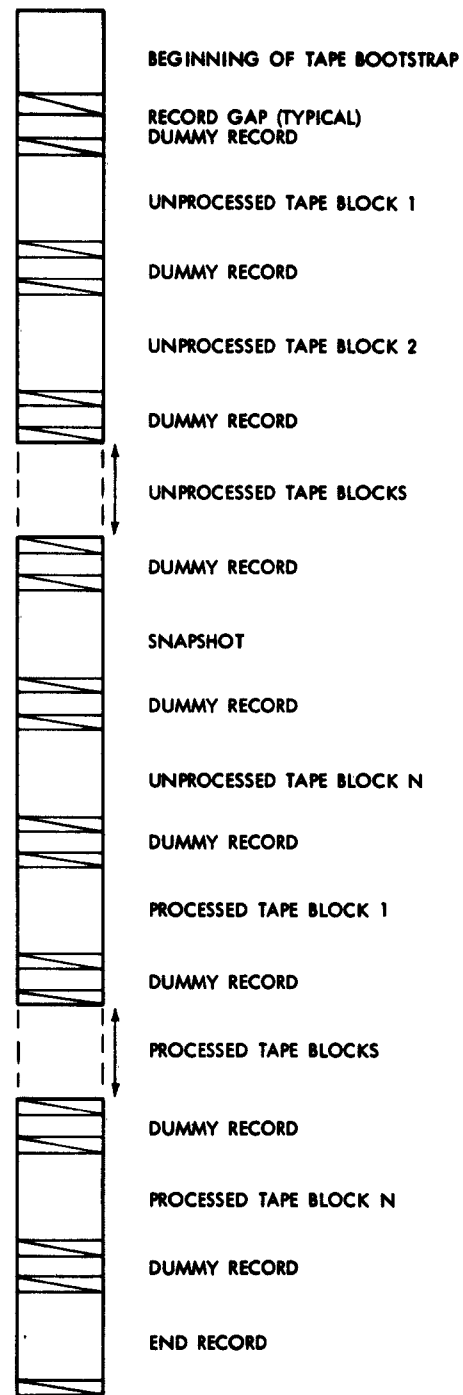
The program tape is rewound and positioned after the bootstrap record. Successive tape blocks are read from the tape, their transfer vectors are processed and they are rewritten as processed tape blocks beyond the present end record. Core block programs are similarly processed. Finally, the symbol table and BSS are cleared, and control transfers to the main program (master routine).

e. T/4 Card

If all tape blocks are correctly processed and written on tape, only the core block is processed before control passes to the main program. Usually this would be used on all normal runs instead of a T/0 card, except those which involve writing of tape blocks.

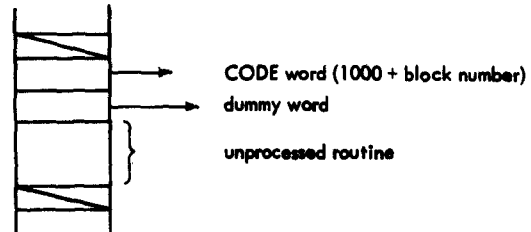
*Note: The BSS loader for the 709 or 7090 must be a 10-card loader due to timing in the instructions.

PROGRAM TAPE FORMAT:

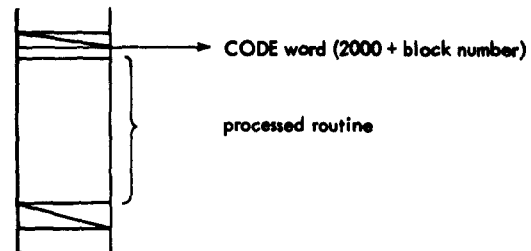


RECORD FORMATS:

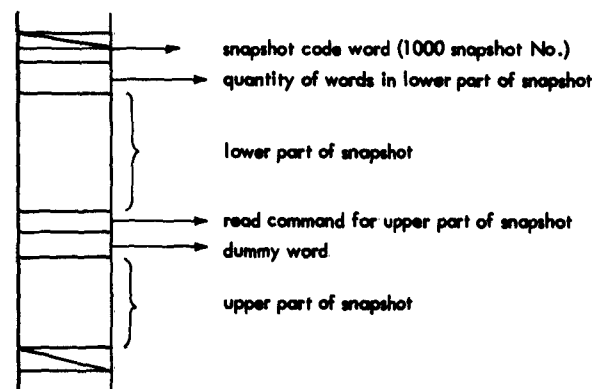
A. Unprocessed tape block



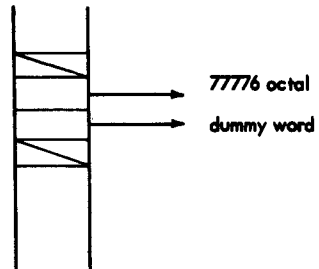
B. Processed tape block



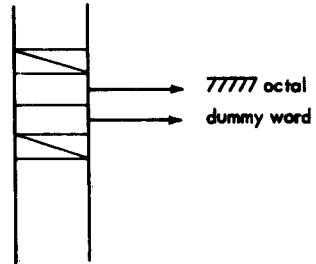
C. Snapshot



D. Dummy Record



E. End Record



Notes: Identifying code numbers are held in the decrement of the first word of each record. Dummy words are required for program delay while the preceding word is being read.

Reading Back Tape Blocks: Processed tape blocks are read into core by MISLAM at its BLKIN entry point. This entry may be made from any core-block program with a CALL BLKIN (I) statement. The tape is positioned to read tape block I, the block is read and control passes back to the calling program. The tape block is entered by a second CALL statement.

Assembly of MISLAM for IBM-709 or 7090: The SAP-coded version of MISLAM for 704 requires several modifications in the assembly program which allow various parts of MISLAM to overlap. (In particular, after the bootstrap record is written, the main part of MISLAM is stored over the bootstrap in core.) These modifications could easily be made in 9AP, so certain 709 and 7090 operation codes are defined in SAP language and the program is assembled on 704, using CORSAP as Assembler.

Error Stops:

- (a) Tape failures during reading or writing operations.
- (b) No snapshot written when writing program tape.
- (c) Proper sequence of T/ cards violated. Proper sequence when writing program tape is T/1 - T/2 - T/0. T/3 cards may be given at any time; a T/4 card may only be used alone.
- (d) Number of tape blocks > 20.
- (e) Number of snapshots > 1000.
- (f) Reference from one tape block to another (any program in the core block may refer to a tape block when it is read in).

Writing the STAIR Program Tape: The FORTRAN II and SCAT routines comprising the STAIR system are compiled into relocatable binary decks in the usual manner. These decks are combined with T/ cards into the following deck which is read on-line and written on tape B4. The order of operating routines must be followed exactly as shown. Since control passes to the main program following the T/0 card, data may be waiting for an analysis program when the program tape is written.

Binary Deck Sequence
for STAIR Program Tape

☐	<u>FORTRAN</u> <u>BSS</u> <u>LOADER</u> *
☐	<u>MISLAM</u>
☐	<u>MAIN</u>
☐	LIBRARY SERVICE ROUTINES
☐	<u>STAIR</u> ROUTINES <u>FIXTPE</u> , <u>MAVERT</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>STRIX</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>EFFRIX</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>MATRO</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>MADD</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>MATOUT</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>MATIN</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>DEFSOL</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>BARSOL</u>
☐	T/1 CARD
☐	<u>STAIR</u> ROUTINE <u>CHECK</u>
☐	T/3 CARD
☐	T/0 CARD

Subsequent Runs of Program: When a program is written on tape by MISLAM, there are several ways of initiating a run of the program.

- (a) New program is to be added to core block. (A new version of a program will replace a prior version of the program in the transfer vectors of all calling programs.)

_____ get tape B4 card (loads bootstrap)
_____ T/2 card
 new core-block program
_____ T/3 card (optional)
_____ T/0 card

- (b) No new program to be read; tape-block transfer vectors are already processed. (This is the normal procedure for initiating a run when the program is debugged.)

_____ get tape B4 card
_____ T/4 card

H. STAIR LOAD PROGRAMS

Operation: Form load matrices for each unit of a structure and write these matrices as consecutive binary records on tape, which may be read back by the STAIR program during an analysis problem.

1. SLOP 1

Method: For each of a number of loading conditions, joint loads are read into core and stored (in a format similar to displacement storage by DEFSOL). When all loads for all loadings are stored, joint lists identifying the matrices of consecutive units are read, and load matrices are assembled according to the order of joints in the joint list. Rows of each load matrix correspond to force components and columns to loading conditions. The matrices are stored columnwise. An end-of-file is written after the last load matrix. A heading word is written as the first word in each load matrix record (32767-u, where u is the number of a unit). As each term is taken from the load storage area to be stored in the matrix assembly area, its register in the load storage area is cleared; this prevents duplication of loads at joints shared by two or more units.

Error Stops:

- (a) $3 \times \text{number of loadings} \times \text{number of joints in structure} > 26000$.
- (b) $3 \times \text{number of loadings} \times \text{number of joints in any unit} > 2000$.
- (c) Joint lists out of consecutive order.

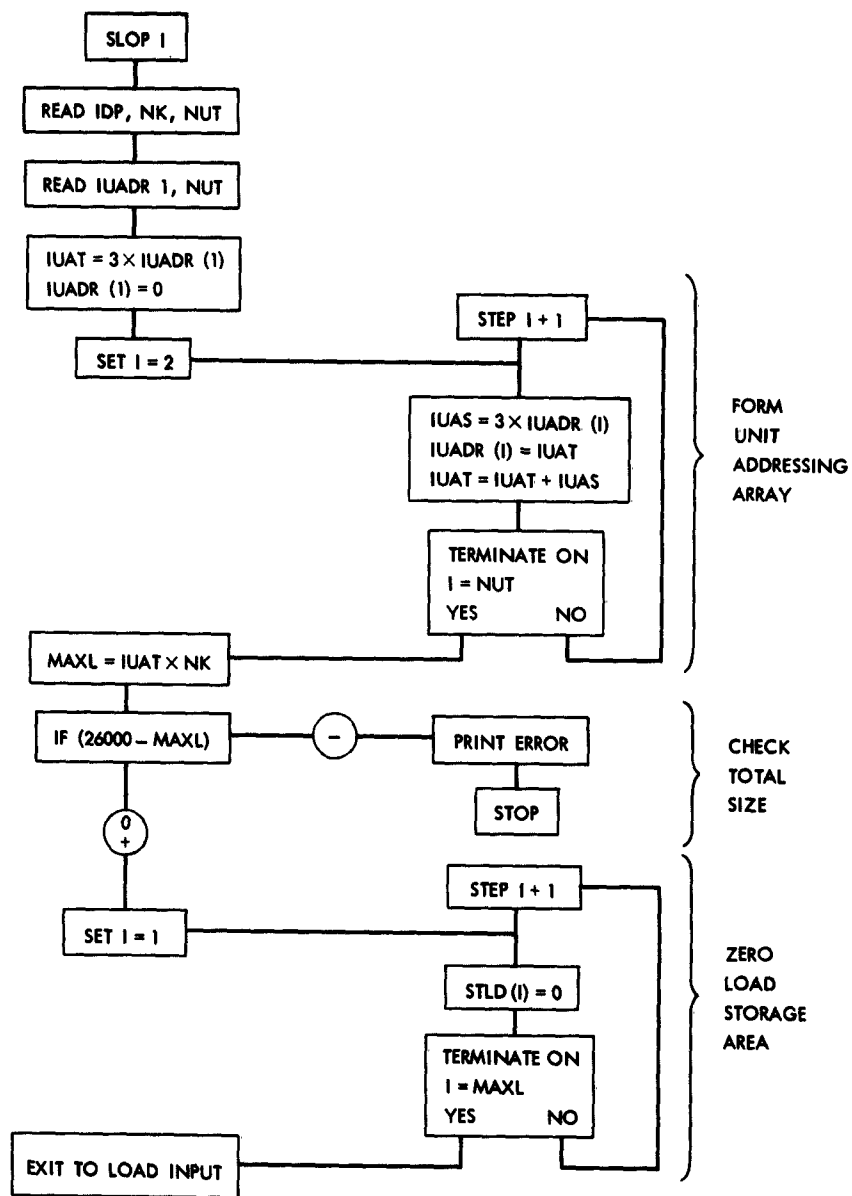
Error Diagnostic, but no Stop:

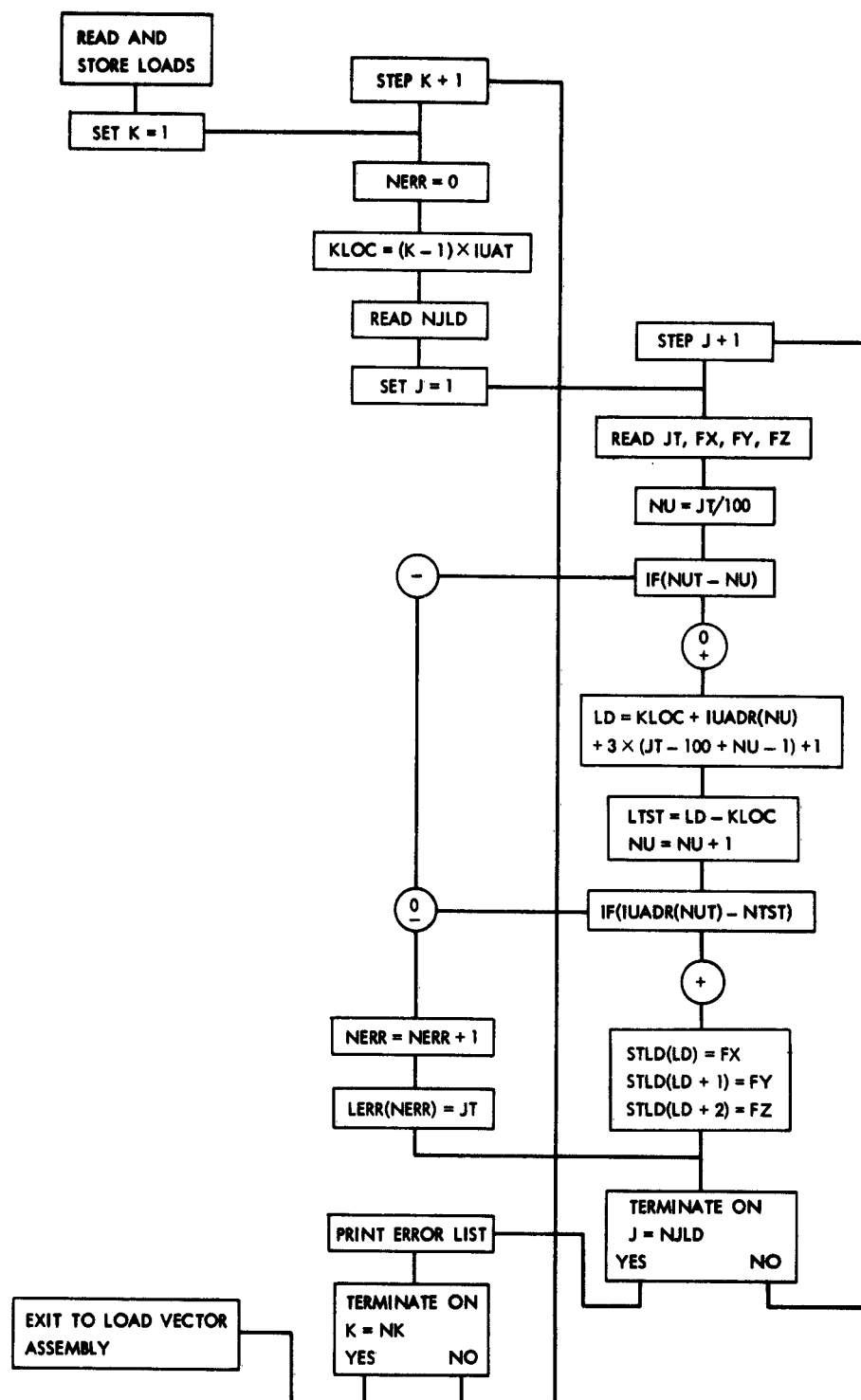
- (a) Illegal joint number with input loads.
- (b) Illegal joint number in a unit joint list.

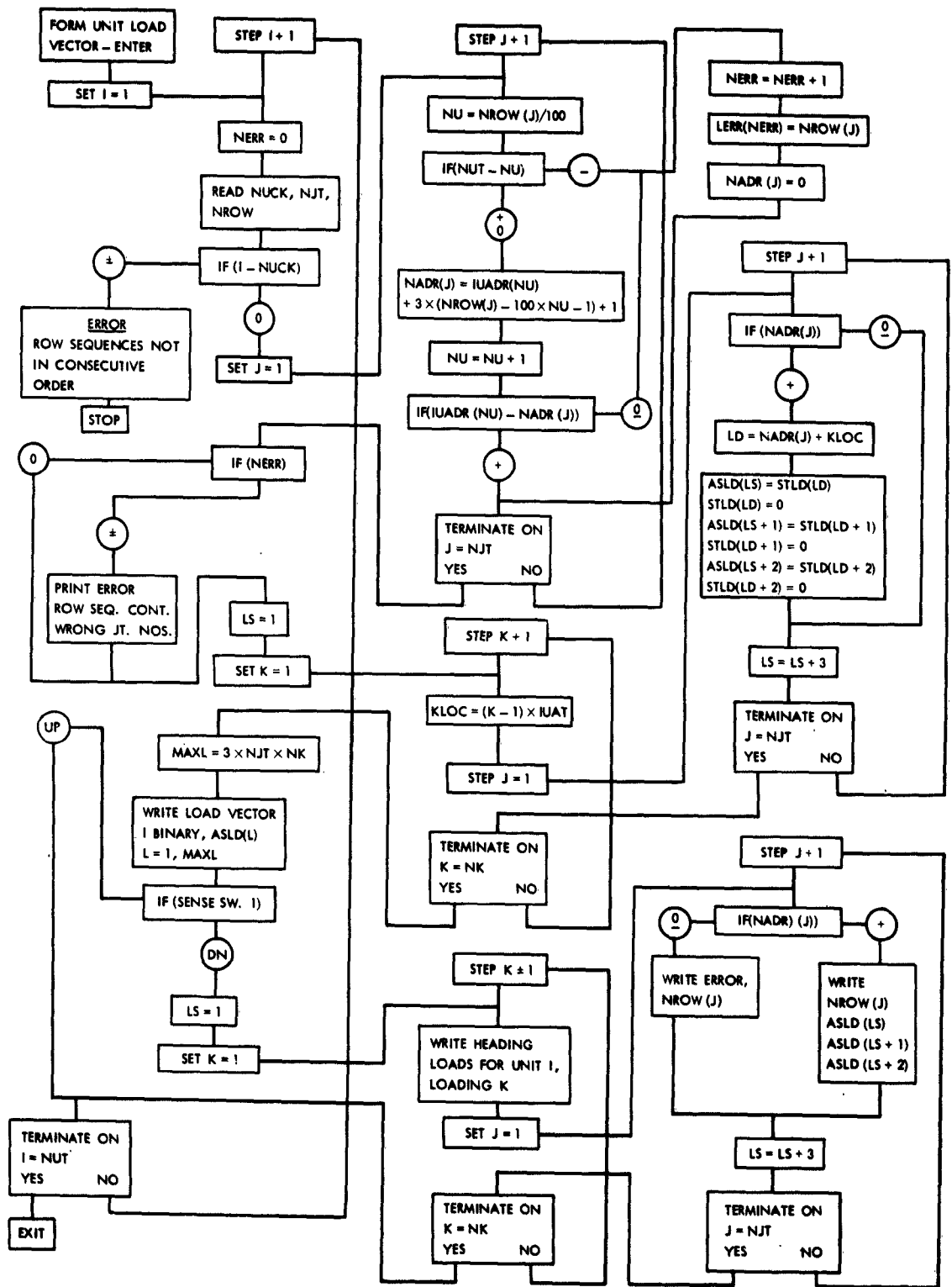
Output: Load matrices are always stored as binary records on tape A8. If Sense Switch 3 is depressed, load matrices will also be written on tape B2 in form for listing.

Storage Addressing: Joint numbers have the form $i j$, where i is a unit number and j is a joint number within that unit. Addresses in the load storage area are computed with a "unit addressing array" $IUADR$, where $IUADR_i = \sum_{k=1}^{i j} 3 u_k$ (u_k is the number of joint numbers prefixed with unit number k). The location A of the x -component of load at joint $i j$ for loading 1 is thus: $A = (i - 1) N + IUADR_i + 3(j - 1) + 1$ (N is $3 \times$ number of joints in the structure). An illegal joint number is either one of the following:

- (a) $i >$ total number of units.
- (b) $IUADR_i + 3(j - 1) + 1 > IUADR_i + 1$.







2. SLOP 2

Method: In addition to the functions performed by SLOP 1, SLOP 2 computes joint loads due to dead weight of the structure. Input to the program is divided into groups; the first word in each group selects the function to be performed (store hand-computed loads, compute dead loads or assemble load matrices).

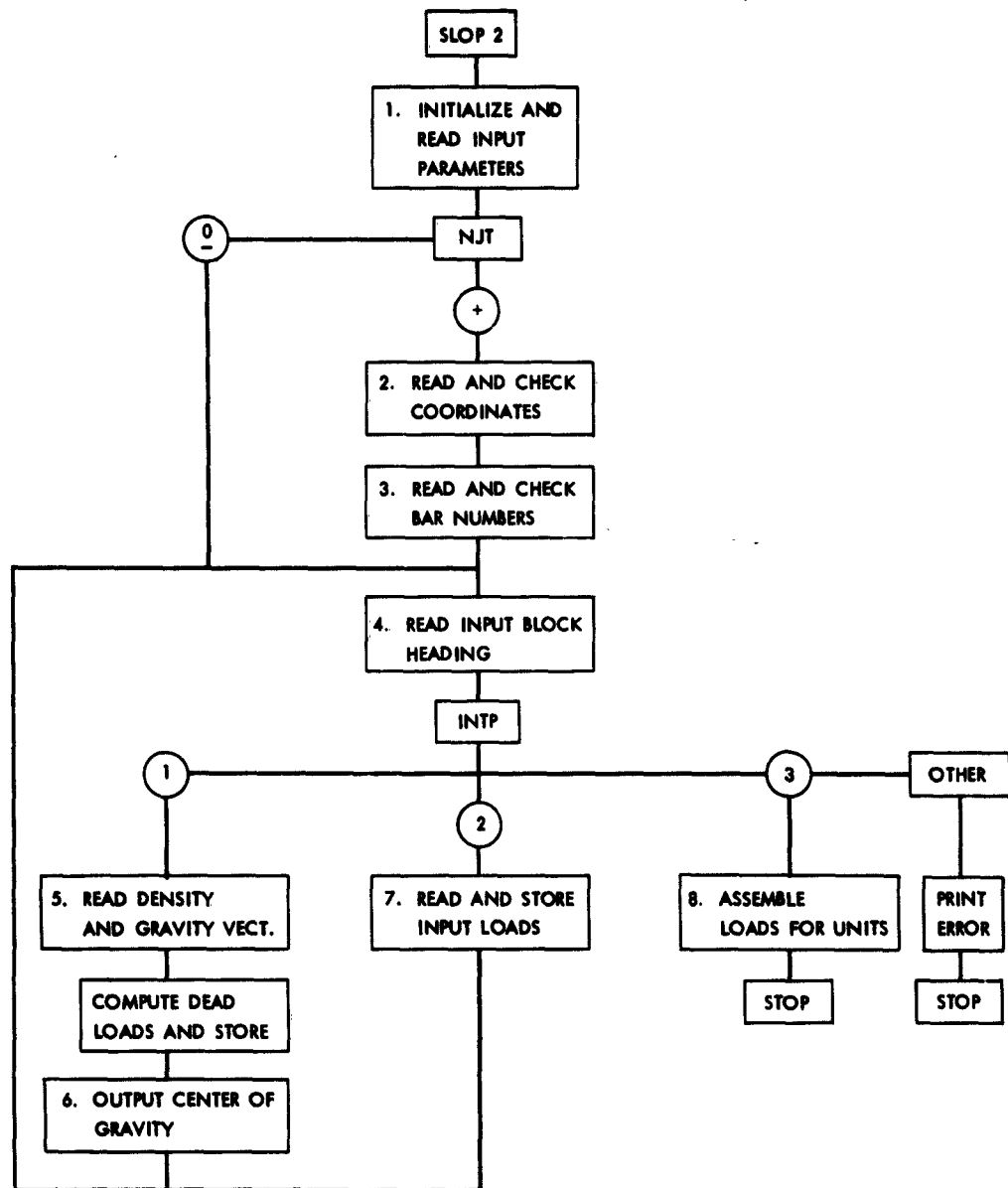
Optional Error Stops: (no stop if SS1 is down)

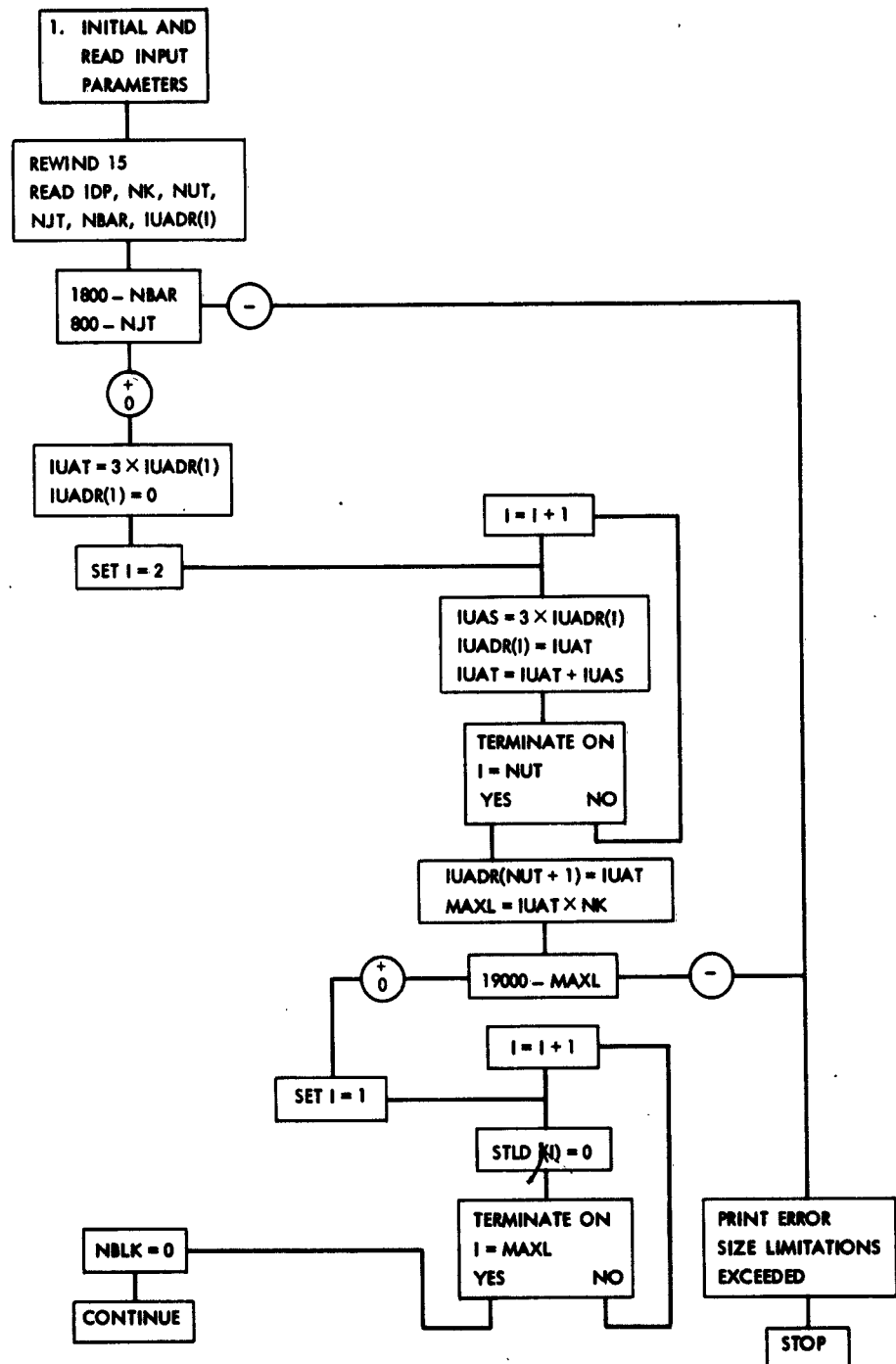
- (a) Illegal joint numbers in input list of coordinates.
- (b) Illegal bar number in input list.
- (c) Illegal joint number in input list of loads.
- (d) Illegal joint number in the joint list of a unit.

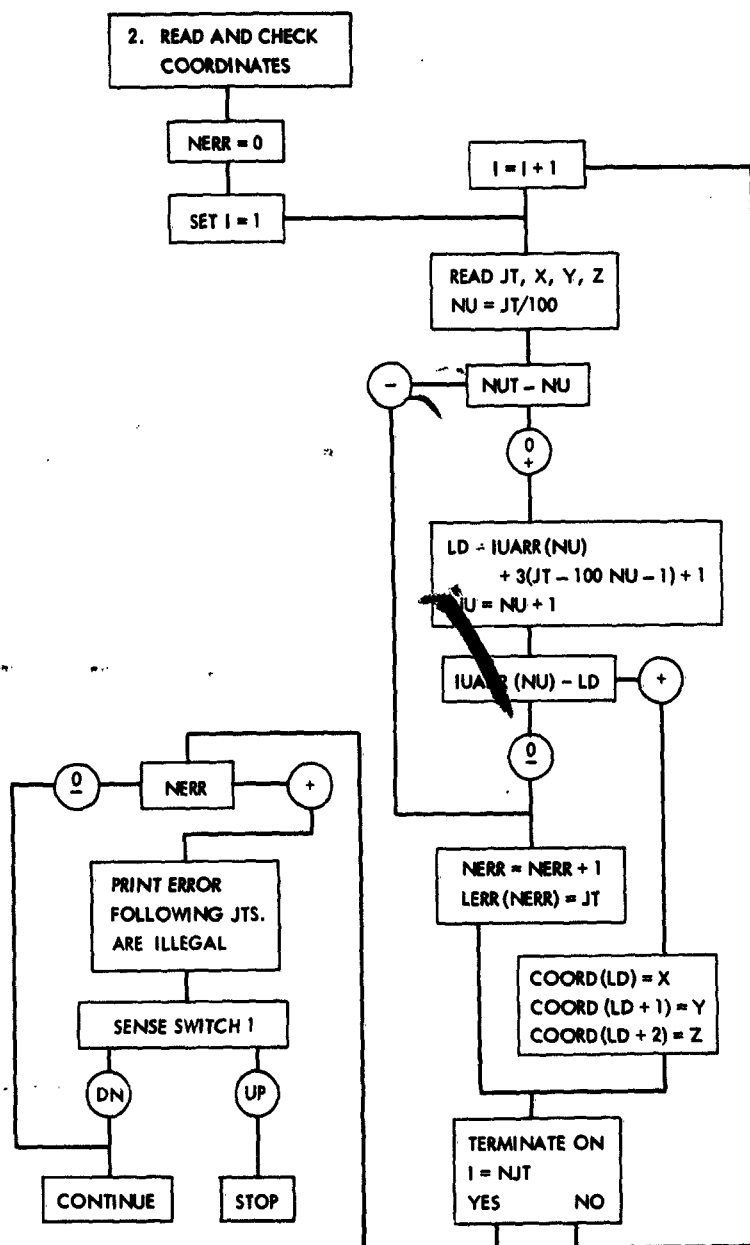
Unconditional Error Stops:

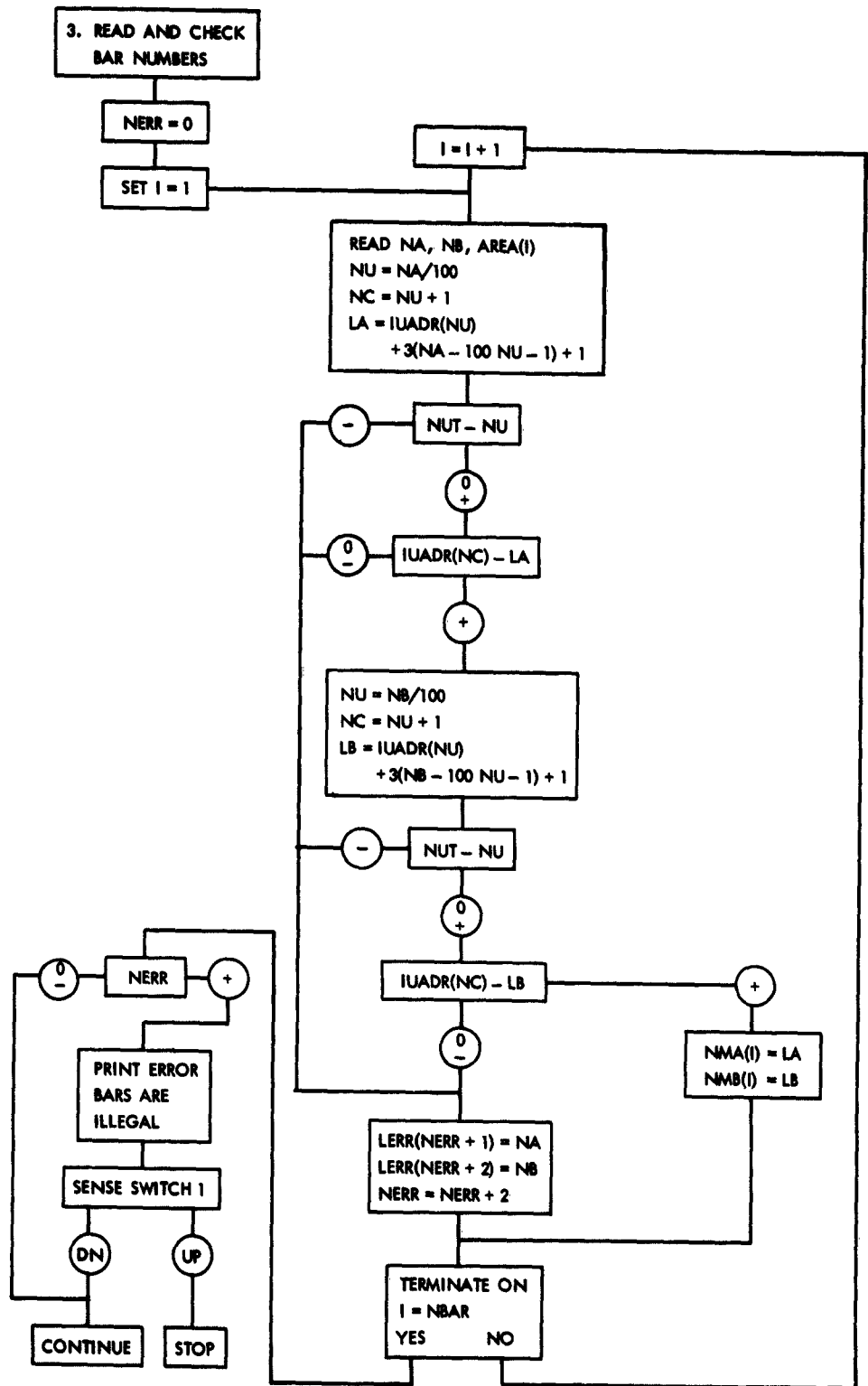
- (a) Illegal input code at head of an input group.
- (b) Illegal loading number on input group heading card.
- (c) $3 \times \text{number of loadings} \times \text{number of joints in structure} > 19000$.
- (d) Unit joint lists out of consecutive order.
- (e) $3 \times \text{number of loadings} \times \text{number of joints in any unit} > 2000$.

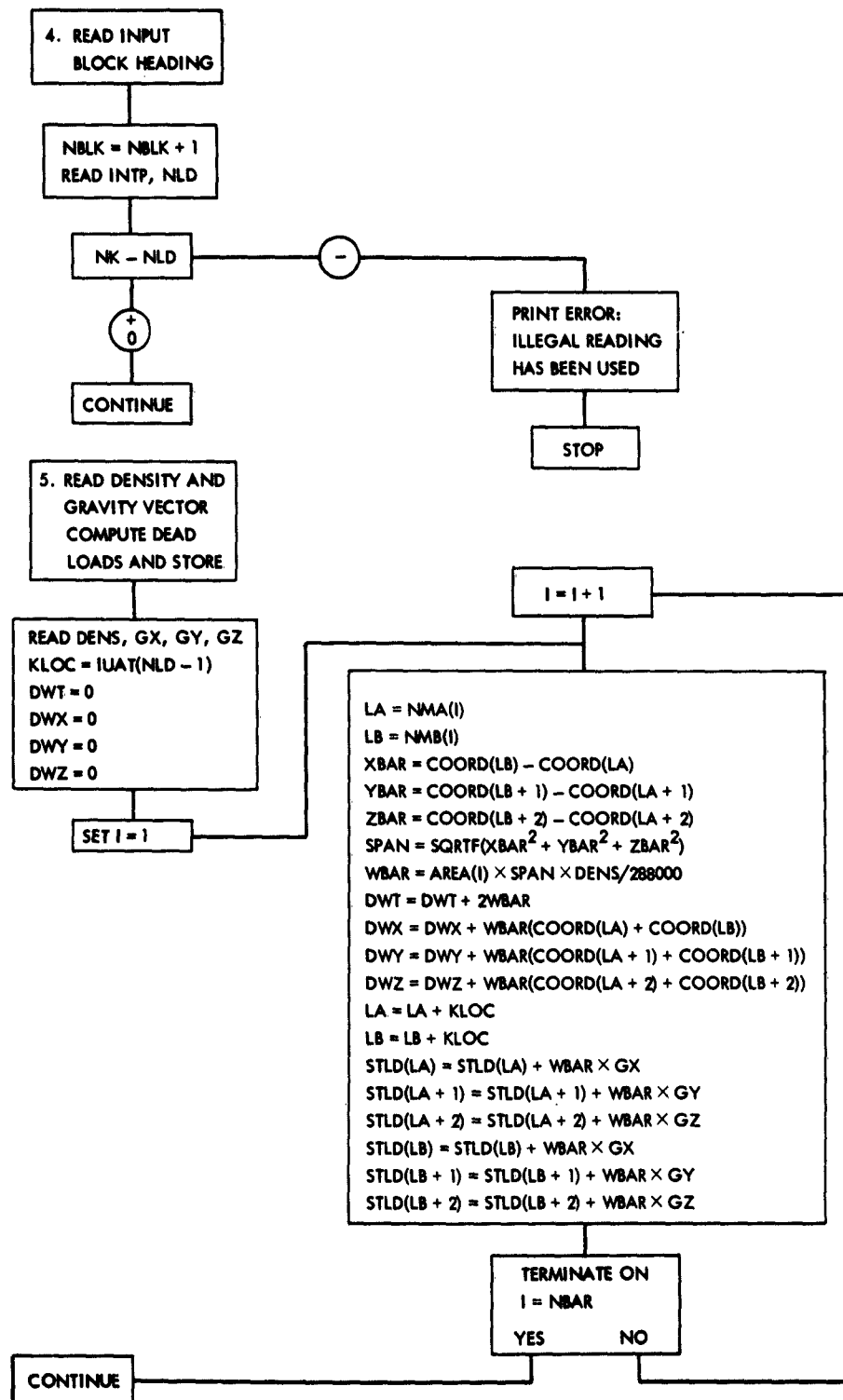
Output: In addition to the output options of SLOP 1, total weight of the structure and coordinates of the center of gravity are printed out on-line for each dead load calculation.

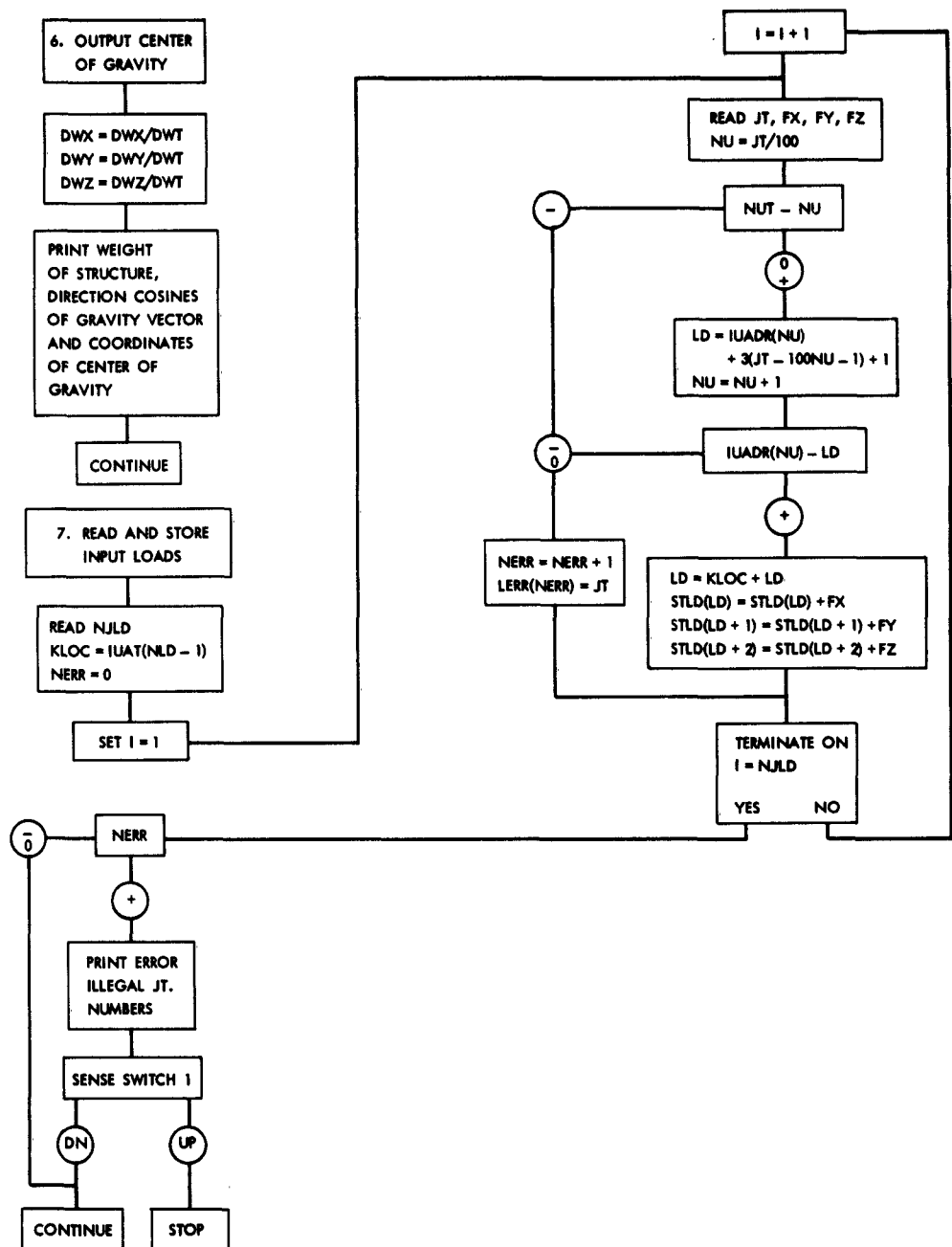


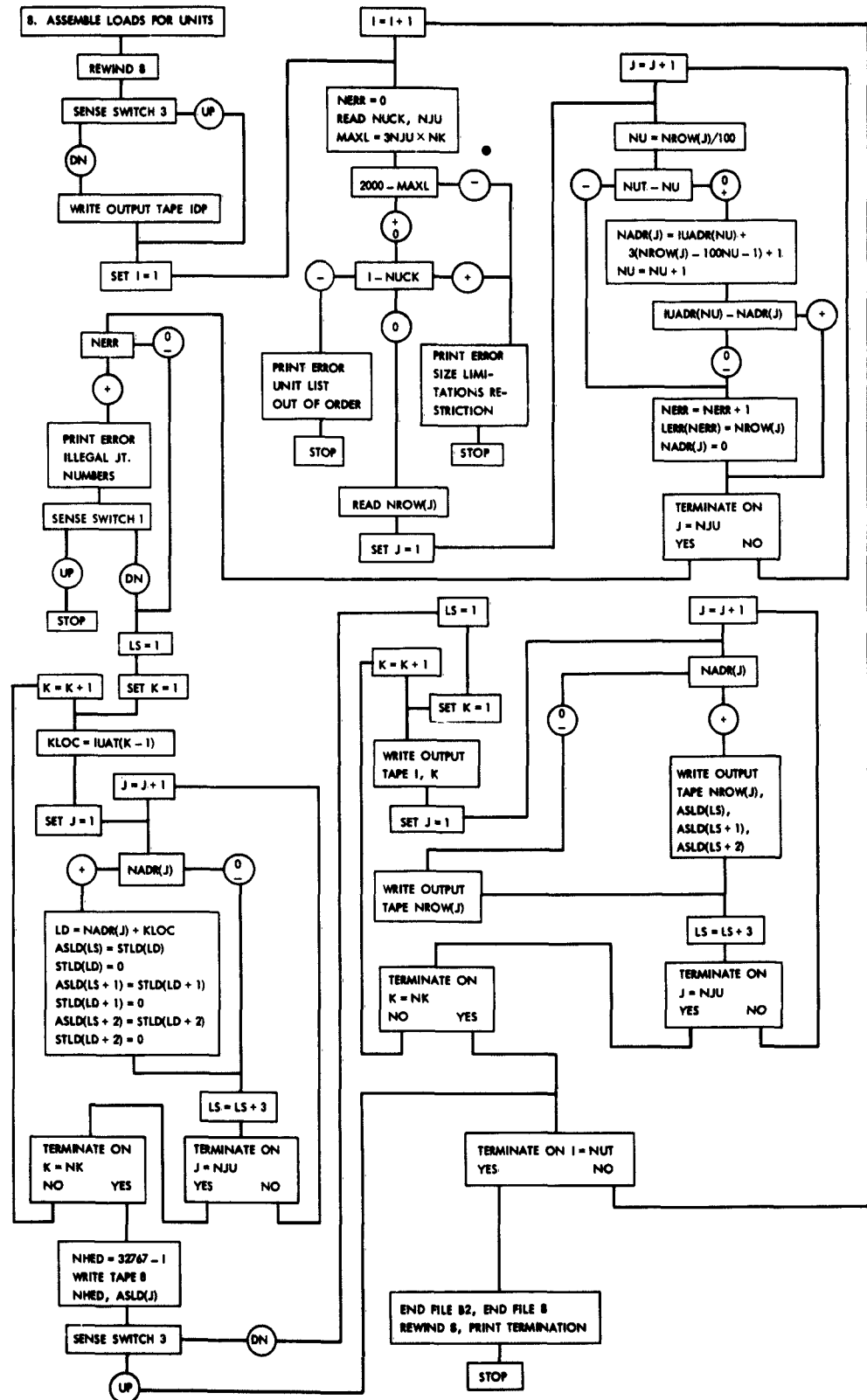












CHAPTER 5 LIMITATIONS AND EXTENSIONS

At this writing, the full capacities of the STAIR program have not been realized. Theoretically, a structure that has a total of 3875 joints (including 125 reaction joints) can be analyzed for one loading condition, but the largest structure analyzed to date has only 600 joints and three loading conditions.

For problems involving large structures (more than 400-500 joints), the limiting restrictions of the various routines can be circumvented as follows:

SLOP:	If there are no gravity loads, SLOP represents no restriction. If gravity loads are wanted, the structure can be broken into several sections and the gravity loads computed for each section separately. Then, by superimposing the loads, SLOP can be run again with the gravity loads as applied loads. Note that storage locations also could be reassigned in SLOP to accommodate a larger problem, but this measure is somewhat limited.
STRIX:	The structure can be broken into any number of units.
EFFRIX:	This routine can be called any number of times if necessary.*
MATRO:	This is limited by STRIX, and thus is not a limitation in itself.
MADD:	EFFRIX can be called any number of times before a MADD operation; thus, the matrices to be added can be reduced to the proper size.
MATOUT:	This is limited only by the limitations of the other subprograms.
MATIN:	This is limited only by MATOUT.
DEFSOL:	EFFRIX can be called any number of times before a DEFSOL operation. The only other limitation is discussed below.
BARSOL:	This routine can be called any number of times.
CHECK:	The restrictions of this routine do not affect the program.

The fourth restriction (see p. 39) of DEFSOL represents the only limitation on the STAIR program. If this restriction becomes too severe, the program could be rewritten to accommodate a special problem, but an experienced programmer should be consulted.

It should be kept in mind that the STAIR program solves only linear problems since the stiffness matrix of the structure is based on the undeformed geometry. If deflections are so large that the angles of connection of members at joints are altered, the results should be treated as only approximations of the actual structural behavior since the problem is no longer linear. Further, there is no provision in the program to handle buckling of individual members of the structure or buckling of the structure as a whole. Consequently, structural instability other than geometric instability must be analyzed by other methods, i.e., buckling of a trussed arch or column would not be indicated in the results of a STAIR ANALYSIS PROGRAM.

Computer round-off error becomes important when solving extremely large problems. It is difficult to say how much round-off error will affect any one problem, since the parameters affecting the error are a function of the problem to be solved. In general, the types of structures analyzed by STAIR are not greatly affected by round-off error, but it cannot be discounted as a possible source of inaccuracy in the results. For any given problem, it is possible to evaluate the round-off error if an experienced programmer or mathematician is consulted. Experience

*EFFRIX can be called only once for a given unit if MATRO later forms another unit from the first.

in problems with up to 200 joints has shown the results were good to five significant figures. However, this should not be used as a standard of comparison, but only as a guide.

In its present state, the program will solve for deflections for only gravity and applied loads, and the structure has to be made of members of constant density and modulus of elasticity. However, deflections caused by temperature changes or fabrication errors can be handled with slight modifications of the program. By making appropriate changes in the areas of bars in SLOP, or the actual STAIR ANALYSIS PROGRAM, members which have different densities or moduli of elasticity can be handled by the computer.

Occasionally, reactions may be in directions other than along the coordinate axes. This problem can be easily handled by using a simple one- or two-bar "tripod" with short members of large area. The bar(s) of the tripod are oriented in the direction of the desired reaction and the standard x, y and z reactions hold the ends of the bars opposite the structure.

Other changes in the input data or the program itself will extend the ANALYSIS PROGRAM to cover any type of loading or structural action to which a pin-jointed structure may be subjected. In making modifications to the input data or the program itself, a clear understanding of the mathematical method of solution by the STAIR program is necessary. Although some analogies exist between the matrix operations involved in the solution of a problem and physical operations in the actual structure, these analogies sometimes break down and will cause erroneous results. The mathematical model of the structure may not always be an exact reproduction of the physical structure, and it cannot be overemphasized that subtle differences in the behavior of the structure and its mathematical model may cause baffling and erroneous results.