FORMAT—FORTRAN MATRIX ABSTRACTION TECHNIQUE

VOLUME V—SUPPLEMENT I. ENGINEERING USER AND TECHNICAL REPORT—EXTENDED

J. PICKARD

This document has been approved for public release
and sale; its distribution is unlimited.

FOREWORD

This report was prepared by the Douglas Aircraft Company, Long Beach, California, under USAF Contract No. F33615-68-C-1633. The work was initiated under Project No. 1467 "Structural Analysis Methods", and Task No. 146705" Automatic Computer Methods of Analysis for Flight Vehicle Structures". The work was administered under the Air Force Flight Dynamics Laboratory, by Mr. J. R. Johnson, FDTR, Project Engineer.

The work reported herein was conducted during the period July 1968 through April 1970. This report was submitted by the author for publication in April 1970.

Within the Douglas Aircraft Company, Mr. P. H. Denke, Director, Scientific Computing was responsible for administration and technical progress. Mr. D. S. Warren, Manager, Advanced Design and Research, Structural Mechanics Section was principal investigator. Many other Douglas personnel contributed significantly to the project.

The general objective of the project was to update the FORMAT System documented in Volumes I through VII, as supplemented, by incorporation of additional basic capability and refinement of existing capability. The work is reported in Volume II - Supplement III, Volume V - Supplement I, Volume VI - Supplement I and Volume VII - Supplement I. A complete description of the current FORMAT System is contained in Volumes II, V, VI and VII, as supplemented (References 1 through 10). The supplements are the final reports of the investigation and conclude the work on Contract No. F33615-68-C-1633. The contractors report number is DAC-33569.

The report has been reviewed and is approved.

FRANCIS J. JANIK, JR.
Chief, Solid Mechanics Branch
Structures Division

ABSTRACT

The FORMAT System has been updated by the incorporation of additional basic capability and the refinement of existing capability. A simpler mode of updating case data and extended force method matrix generation capability has been incorporated in Phase I of the system. A refined "Structure Cutter" module, capabilities for matrix partitioning and instruction looping, and an additional eigenvalue/eigenvector extraction module have been incorporated in Phase II. Finally the limitations which existed in the matrix plotting capability in Phase III have been eliminated. Engineering user and technical information is presented in this report. Included are recommendations of improvements in implementation and utilization procedures for various computer systems.

CONTENTS

ILLUSTRATIONS

TABLES

SECTION I

INTRODUCTION

The FORMAT System has been updated to provide additional basic capability and refinements to the existing capability.

The various additions and modifications to Phase I (the case and matrix generation phase) of the system are as follows.

- Simplification of input data required in the updating of case data.
- Improved structural idealization capability in force method analyses by provision for multiple bars along a shear panel edge, and in addition, the use of a revised algorithm for generation of the matrix of weighting factors in order to represent element relative stiffnesses more effectively.
- Automatic generation of an additional matrix in force method joining analyses which enables force and moment summations of the applied loading to be obtained matrically.

Additions and modifications to Phase II (the matrix abstraction phase) of the system are as follows.

- Replacement of the "Structure Cutter" module (STRCUT) with a more sophisticated version which reduces solution time substantially and improves solution accuracy in certain cases.
- Incorporation of a capability to partition matrices columnwise via the instruction "DEJOIN".
- Incorporation of a capability to loop an instruction set via the instruction "REPEAT" and hence effect a general iterative type matrix analysis with minimal input data requirements.
- Incorporation of a QR eigenvalue/eigenvector extraction capability for real, nonsymmetric matrices of relatively small order via the instruction EIGEN3. This is intended to complement the existing eigenvalue/eigenvector extraction routines in FORMAT.

The modification to Phase III (the special output phase) of the system is as follows.

- Extensive revision of the matrix plot module in order to provide almost all the capability implicit in a hand plot of matrix data with linearly scaled axes.

The extensions to the FORMAT System are documented in supplements to four existing volumes as follows.

Volume V - Supplement I:	Engineering User and Technical Report - Extended
Volume VI - Supplement I:	Programming Documentation for Phase I - Extended
Volume II - Supplement III:	Programming Documentation for Phase II - Extended
Volume VII - Supplement I:	Programming Documentation for Phase III - Extended

Sections II, III and IV of this report present user-oriented information relative to the changes affecting Phases I, II and III of the system, respectively. Section V discusses experience garnered in the implementation and utilization of FORMAT on various computer systems and potential improvements in operating procedures are suggested. Conclusions are presented in Section VI. The Appendix contains examples of structural analysis applications which illustrate the use of the new features of FORMAT. Included is an example which illustrates the use of FORMAT for automated fully-stressed design analysis.

SECTION II

PHASE I ADDITIONS

1. CASE EDITOR UPDATE

a. General

The Update feature of the FORMAT Editor required that case data to be modified had to be identified by the use of TABLE cards in the Phase I input data. Such a card was required for each existing table to be included in the set of case data being created with optional modification. This mode of entry is not ideally suited to the common application in which all of the tables of an existing set of case data, and only those, are to be used in the creation of a modified set of case data. This corresponds to the general production case in which case modification may be necessary due to errors in joint, vector or element data input.

Provision has therefore been made for the Editor to accept a CREATE/REF card in lieu of TABLE cards as a means for identifying both the input and output case data.

b. Input

One CREATE-REF card may be used when the UPDATE option is specified on the $EDITOR card. This is immediately followed by a DATA card.

c. Output

Editor output is unchanged.

d. Application

The example shown below illustrates the Phase I input data necessary to update case "FCSE1" which is contained on case data set "CASE1, 1" and contain the updated case with the same name on case data set "CASE2, 1".

```
1     7        16
$GENERATOR
      CASE INPUT TAPE  (CASE1,1)
      CASE OUTPUT TAPE (CASE2,1)
$EDITOR        UPDATE
      CREATE FCSE1 (CASE2,1),REF FCSE1 (CASE1,1)
      DATA
        .
        .
        .
$END
```

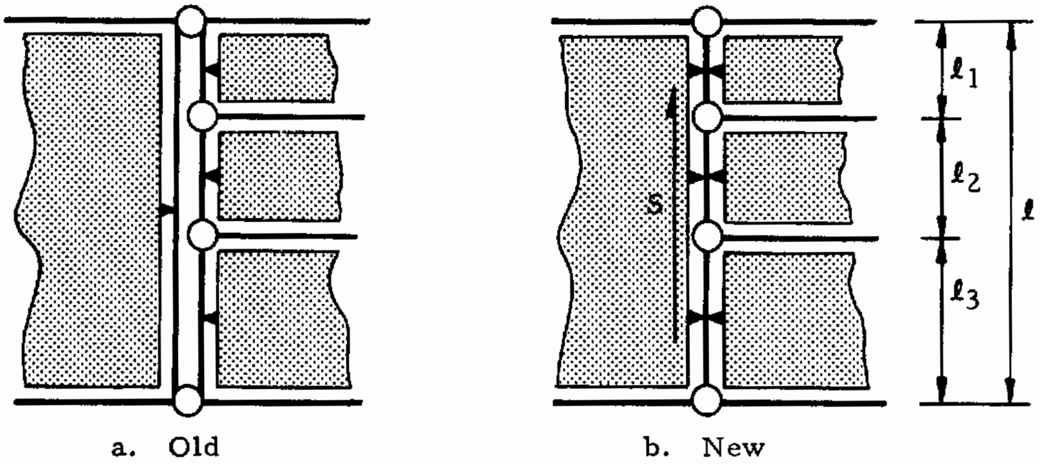
2. FORCE METHOD MATRIX GENERATION

a. Basic Static Analysis

(1) General

Structural idealizations for analysis in FORMAT by the Force Method have, in the past, been subject to the requirement that each shear panel edge be bounded by a unique bar element. Changes in "degree" of idealization across a section were effected by including coincident bars at the section appropriate to the idealization at either side as illustrated in Figure 1a. In such a case, the mode of local load transfer is somewhat erroneous and subsequent internal load interpretation cumbersome.

Consequently, the Basic Force Method Module (BFMM) has been modified to accommodate idealizations in which up to six colinear bar elements may be used to reach from one panel corner to the next as illustrated in Figure 1b where three such bars are used.



a. Old

b. New

Figure 1. Structural Idealization in the Force Method

In addition, the weighting factor matrix, W, which is generated by the BFMM for use in the "Structure Cutter" operation has been modified to more effectively represent the relative stiffnesses of all elements in any given structure. The major deficiency in the previous mode of generation of matrix W was that a rigid element (i.e., with a stiffness 100 or more times any other) would result in all other element weighting factors being set at 0.001.

Weighting factors for the reactions are set equal to unity (as before) and for the remaining elements are relative to the stiffness of the median value. The latter are normalized such that the median value is 0.1 and the values above 0.1 linearly scaled such that the maximum value is 1.0. No cutoff is used.

(2) Input

Fundamental input to the BFMM remains unchanged. The improved structural idealization capability permits up to six bar elements to be used to reach from one shear panel corner to the next. General colinearity requirements of the BFMM must be satisfied in establishing the intermediate joint coordinates. Other idealization capability is unchanged.

(3) Output

(a) Matrix Output

The BFMM generates exactly the same sequence of matrices as previously.

Only matrices P_F and T_F are affected by the added capability. The assumption of uniform shear along a panel edge results in shear transfers to edge bars which are proportional to the bar lengths, e.g., $S\ell_1/\ell$, $S\ell_2/\ell$, $S\ell_3/\ell$ in Figure 1b. These values are reflected in the P_F and T_F matrices.

The weighting factor matrix, W, is determined by initially computing the reciprocal of the main diagonal terms of the element flexibility matrix D excluding external reaction and type 1 bar element force flexibilities. If the relative stiffness of the ith element is denoted by k_i, the median value by k_{median} and the maximum value by k_{max} then the weighting factor, w_i, is obtained as follows.

For $k_i \leq k_{median}$

$$w_i = k_i/10\,k_{median}$$

For $k_i > k_{median}$

$$w_i = 0.1 + 0.9\left[\left(k_i - k_{median}\right)/\left(k_{max} - k_{median}\right)\right]$$

Weighting factors for external reactions are set equal to unity.

(b) Printed Output

Printed output is unchanged.

(c) Error Messages

The error message,

ERROR **** IN DATA FOR SHEAR PANEL ELEMENT ****

is now interpreted as follows.

Error 1 indicates the joints input for p, q, and s lie on the same line.

Error 2 indicates one of the edge lengths is zero.

Error 3 indicates the panel thickness, t, the shear modulus, G, or Young's modulus, E, or Poisson's ratio, ν, has been omitted.

Error 4 indicates one of the bars bounding the panel edges has not been input.

Error 5 indicates one or more of the joints input for p, q, r, and s has not been defined in the joint coordinate table (data code 1).

(4) Application

The mode of problem solution, i. e., a basic force method analysis is unchanged. The revised weighting factors are, however, reflected in the range of pivots which result in the "Structure Cutter" operation on the equilibrium equation in Phase II. Hence the user should be cognizant of the range of weighting in assessing these pivots (see Section III. 1).

b. Joining Analysis

(1) General

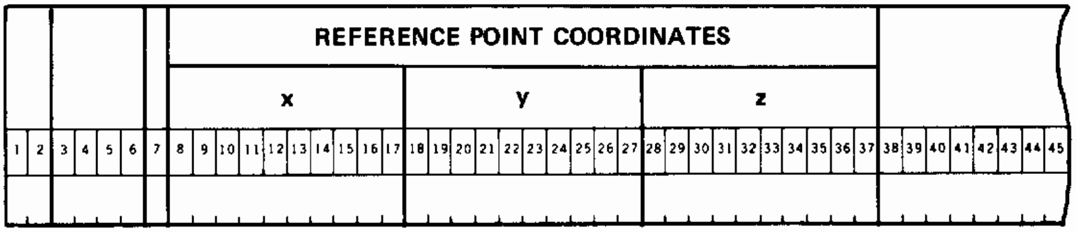
The Force Method Joining Module (FMJM) provides for the generation of a set of extractor matrices with which the user can assemble the boundary equilibrium and flexibility matrices and hence matrically join basic substructure analyses. Its capabilities have been extended to provide for the generation of one additional matrix, identified by the standard name SUM. This matrix contains the force and moment resultants of each of the unit applied joined structure external loads about axes which are parallel to the "global" x, y, z axes and with origin at a user specified reference point. Post-multiplication of the matrix SUM by a joined structure applied load matrix gives a summation check of force and moment at the reference point.

The total moment about a reference axis is composed of the resolved components of applied moments and the moments due to force components in the direction of the other two axes. The partial moment summations due to the force components are also created separately.

The SUM matrix provides for accuracy comparisons of the applied (discretized) loading with the design (usually continuous) loading, and the partial moment summations enable error sources to be readily pinpointed.

(2) Input

In order to use the FMJM the user must submit one additional data card. This card is now the first card of the special data and must immediately follow the DATA card. It contains simply the global x, y and z coordinates of the reference point about which the force and moment summations are required. The card format is the same as the three cards of substructure transformation data as shown below.



If the user wishes to assign a name to the matrix, other than the standard name of SUM, he may do so by the use of a RENAME card in the normal manner.

(3) Output

(a) Matrix Output

The matrix SUM is the first matrix output by the FMJM and is dimensioned 12 by JXL where JXL is the number of joined structure external loads. The twelve rows correspond to ΣF_x, ΣF_y, ΣF_z, ΣM_x, ΣM_y, ΣM_z, ΣM_{xfy}, ΣM_{xfz}, ΣM_{yfx}, ΣM_{yfz}, ΣM_{zfx}, and ΣM_{zfy} where the first six rows are total summations and the later six rows are partial moment summations due only to the specified components of applied forces, e.g., ΣM_{xfy} = summation of moments about the x axis due to the components of applied forces in the y direction only.

The remaining matrix output of the FMJM is unchanged.

The matrix SUM will be contained on the output data set even if the FMJM aborts because of an error related to joining vector compatibility in the generation of the remaining matrix output.

(b) Printed Output

The matrix name, row and column dimensions and description are included in the last section of printout which lists the matrices generated. Other printed output is unchanged.

(4) Application

Post-multiplication of the SUM matrix by a joined structure applied load matrix (say ϕ) via matrix abstraction in Phase II, produces a summation (say S) of forces and moments about the input reference point for each loading condition as follows.

$$S = SUM \cdot \phi$$

A summation of forces and moments about the reference point for applied loads on a selected substructure (say the bth) may be obtained by use of the following equation.

$$S = SUM \left(C_{\phi_b}^{T} \, C_{\phi_b} \right) \phi$$

where C_{ϕ_b} is the external load matrix transform for the bth substructure generated by the FMJM. Shear force and bending moment curves may thus be derived by generating SUM matrices at suitably located reference points and applying joined structure applied loads to one side of each reference point.

The SUM matrix may be generated for a single structure, providing the necessary joined structure external load (JXL) flags are included in the applied load vector case data entries. In such an application the matrix is saved on the output data set and available for use even though the FMJM eventually aborts due to the absence of compatible joining vectors on another substructure. A set of case data could therefore be created with dummy applied load vectors to represent any vector type, e.g., external load, reaction, internal element force, joining element force, etc., and the SUM matrix generated for these vectors providing they are flagged as JXL's. The FMJM also requires that at least one of the applied load vectors in the set of case data be flagged as a joining element force (JEF).

SECTION III

PHASE II ADDITIONS

1. STRUCTURE CUTTER (STRCUT)

a. General

The matrix abstraction capability of Phase II of the FORMAT System includes the "Structure Cutter" module which generates a solution of "n" linear simultaneous equations in "m" unknowns by Jordanian elimination (where $n \le m$). This module has been replaced with a more sophisticated version whose algorithm takes advantage of sparsity of the coefficient matrix and utilizes a more effective mode of pivot selection. Solution time is considerably reduced and solution accuracy improved.

The user may optionally control the pivotal acceptance levels used by the module and a list of the column numbers of the unreduced (non-pivotal) columns of the coefficient matrix is now included in the unconditional printed output for a successful execution. If execution is terminated for reason of unacceptable pivots the row numbers of the remaining (dependent) equations in which acceptable pivots cannot be found are listed.

The revised module also includes a restart capability which may be deployed should execution be terminated during the pivot selection phase for abnormal reasons, e.g., system malfunction. The four scratch data sets used during execution must be saved if a restart is to be made. This capability is fully documented in Volume II, Supplement III of this report.

b. Abstraction Instruction

"Structure-Cutter" statements are of the form:

$$c_1, \ c_2 \ = \ \pm a.\mathrm{STRCUT}. \pm b, \ (d, e, f, g, h)$$

where the solution, Y, of the system of "n" linear simultaneous equations in "m" unknowns, $\pm AY \pm B = 0$, where $n \le m$, is formed by Jordanian elimination and the two parts of the solution are named matrix $\underline{c}_1$ and matrix $\underline{c}_2$. The following auxiliary definitions apply:

a - is the transpose of the coefficient matrix, A.

b - is the transpose of the matrix of constants, B.

c_1 - is the homogeneous solution

c_2 - is the particular solution

d - is an unsigned floating point number, with or without exponent, bounding matrix element values of matrices $\underline{c}_1$ and $\underline{c}_2$ which are trivial and to be suppressed. That is the matrix element c_{ij} is suppressed if $|c_{ij}| \leq d$. If $\underline{d}$ is blank, zero valued elements are suppressed.

e - is either of the two literal constants. STOP or CONT. When $\underline{e}$ is STOP, execution is terminated if the available pivot elements do not satisfy the accuracy requirement. When $\underline{e}$ is CONT, termination of execution for reason of unacceptable pivot elements is delayed until the STRCUT instruction has been completely executed, including printing. If $\underline{e}$ is blank, the STOP option applies.

f - is the name of the matrix of weighting factors. If $\underline{f}$ is blank, a unit matrix of weighting factors automatically applies.

g - is the first pivotal acceptance level. If $\underline{g}$ is blank, 10^{-3} is used.

h - is the second pivotal acceptance level. If $\underline{h}$ is blank, 10^{-5} is used.

Matrices $\underline{c}_1$ and $\underline{c}_2$ are normal output data for this process. For the case m = n, $\underline{c}_1$ does not theoretically exist. In this case, a null matrix of order (n x 1) is generated as $\underline{c}_1$.

This subroutine unconditionally prints both a list of pivot element values with the corresponding column numbers of matrix A and a list of the column numbers of the unreduced (nonpivotal) columns of matrix A as special output data.

If execution is terminated for reason of unacceptable pivot elements the best remaining pivot is printed together with the row number of matrix A from which it emanates. Row numbers for all remaining rows which contain unacceptable pivots are also listed.

c. Error Messages

Error messages emanating from the Structure Cutter are listed below in alphabetical order on the first word.

CANNOT LOCATE MATRIX FOR STRUCTURE CUTTER.

ERROR IN STRUCTURE CUTTER INPUT - IMAX = **** AND
JMAX = ****

ERROR IN STRUCTURE CUTTER INPUT - NULL COLUMNS
MATRIX ****** NULL COLUMN = **** (etc.)

ERROR IN STRUCTURE CUTTER INPUT - NULL ROWS
NULL ROW = **** (etc.)

```
INSUFFICIENT STORAGE FOR STRUCTURE CUTTER

INSUFFICIENT TAPES FOR STRUCTURE CUTTER

MATRIX IS SINGULAR.  BEST UNACCEPTABLE PIVOT = ±0.XXXXXXE-XX
EQUATION ****
FOLLOWING EQUATIONS CONTAIN UNACCEPTABLE PIVOTS
                ****  (etc.)
```

d. Application

The original "Structure Cutter" module solves a matrix equation of the form

$$A\,Y = B$$

accepting matrices A and B directly as input. The revised "Structure Cutter" module solves a matrix equation

$$A\,Y + B = 0$$

accepting matrices A and B in transposed form as input.

The most common application of the module is to solve the joint equilibrium equation

$$P_F\,F + P_\phi = 0$$

in the basic analysis of a structure by the Force Method. The Basic Force Method Module outputs the matrices P_F and P_ϕ signed according to this equation and in transposed form, i. e., $P_F{}^T$ and $P_\phi{}^T$. The revised "Structure Cutter" therefore provides for direct input of the generated matrices without prior transposition as was previously the case.

In a basic Force Method analysis, therefore, the first three statements in the Phase II abstraction sequence, i. e.,

```
7
PF       =  PFT .TRANSP.
PO       =  POT .TRANSP.
FXR,FOR  =  PF  .STRCUT. -PO ,(1.0E-8,STOP,W)
```

will be replaced by the single statement

```
FXR,FOR  =  PFT .STRCUT. POT ,(1.0E-8,STOP,W,,)
```

This is further illustrated in Example 1 of the Appendix.

2. MATRIX PARTITIONING (DEJOIN)

a. General

The base capability for matrix abstraction has been extended by the incorporation of the matrix operation DEJOIN (opposite of ADJOIN). This provides for column partitioning of a matrix into two user-named submatrices, a capability hitherto effected by post multiplication of the subject matrix by two card input column extractor matrices.

b. Abstraction Instruction

Matrix Dejoin statements are of the form:

$$c_1, c_2 \quad = \quad a.\text{DEJOIN.}\ (j, 0)$$

where the matrix a is column partitioned immediately before its jth column and the resulting dejoined matrices are named $\underline{c}_1$ and $\underline{c}_2$ (i. e., $[\underline{c}_1 \vdots \underline{c}_2] = a$). Matrices $\underline{c}_1$ and $\underline{c}_2$ are of order $(m \times \underline{j-1})$ and $(m \times \underline{n-j+1})$ respectively with matrix $\underline{a}$ of order $(m \times n)$. Note that $1 \leq j \leq n$.

The "0" argument in the statement indicates column dejoining of matrix $\underline{a}$. Row dejoining may be effected by initially transposing matrix $\underline{a}$. Provisions have been made to accept a "1" in place of the "0" to indicate row dejoin; currently the module will branch to a nonexistent subroutine.

c. Error Messages

MATRIX COLUMN DIMENSION IS TOO SMALL IN . DEJOIN.

This error results when the column number, j, is greater than the matrix column dimension, n.

d. Application

Two example applications are given below.

(1) X, Y = Z . DEJOIN. (40, 0)

If Z is of order 300 x 100 then
X will be of order 300 x 39 and
Y will be of order 300 x 61

(2) G, P = H . DEJOIN. (1, 0)

G will be a null column with
the row dimension of H, and
P will be a copy of H.

3. INSTRUCTION LOOPING (REPEAT)

a. General

The operation REPEAT has been added to the matrix abstraction capability of the system. The new operation provides a looping capability analagous to the FORTRAN "DO" statement where a certain sequence of instructions is to be repeated a specified number of times.

The sequence of instructions to be repeated is expanded into the range of the REPEAT loop during preprocessing, and unique matrix names attained by appending subscripts which are automatically incremented each time the sequence is repeated. Manipulation of subscripted matrices in the instruction sequence prior and subsequent to the REPEAT loop is entirely general and provision is made for the card input of such matrices. In the absence of a REPEAT statement in the sequence of instructions, use of subscripts on matrix names is optional.

Potential applications include synthesis of fully-stressed structural designs, analysis of structural nonlinearity due to large deflections, creep, short-time plasticity and combinations thereof, 3-dimensional matrix algebra and solutions of systems of nonlinear equations. REPEAT provides for a more expedient mode of abstraction instruction input in such applications.

b. Abstraction Instruction

"Repeat" statements are of the form

REPEAT (n, m)

where the arguments are

n - the number of abstraction instructions in the sequence immediately following the REPEAT statement which are to be repeated

m - the number of times the sequence of n instructions is to be repeated

The instruction can be literally interpreted as "repeat the following series of n instructions m times".

The sequence of instructions is expanded into the range of the REPEAT loop during preprocessing. Matrix names which are initially subscripted automatically have their subscripts incremented by one each time the sequence is repeated; unsubscripted matrix names remain the same.

c. Subscripted Matrix Names*

Subscripted matrix names are specified in an abstraction instruction as one to six alphameric characters, the first of which must be alphabetic (as previously). The subscript of the matrix name, if any, must be a decimal integer between 1 and 9999 enclosed in slashes. If a matrix name is not subscripted, integer one is assumed. Negative or zero subscripts are not allowed.

The matrix name has the form:

NAMEA / k / or NAMEA

where k is a one to four digit decimal integer.

Subscripted matrix names are specified in card input matrix data by the entry of the matrix name in card columns 67 through 72 (as previously) and the subscript in card columns 73 through 76 (right justified). A modified version of the card input matrix data standard form is shown in Table I.

d. Restrictions

The use of subscripted matrix names is restricted to Phase II of the FORMAT System. Restrictions on the use of the REPEAT loop are as follows:

- A statement number may not appear on a statement which lies in the range of a REPEAT loop. This implies there can be no transfer into or within the range of the loop.
- Nesting of REPEAT loops is not permitted.
- The total number of statements generated by the REPEAT loop is restricted by the amount of working storage (NWORK) available for the instruction analyzing module and the allocation module. (Typically with NWORK = 10000, approximately 100 instructions are permitted.)
- Matrix names on the left side of an equals sign must be subscripted.

---

* The matrix name is stored in memory as one character per word. The seventh word of the matrix name contains a positive or negative integer. The absolute value of this integer is the subscript of the matrix name. The sign of this integer is the sign of the matrix name.

TABLE I. SUBSCRIPTED MATRIX FORM

Header:

H	I_{MAX}	J_{MAX}
1	2 3 4 5	6 7 8 9
H		

MATRIX NAME	SUB-SCRIPT	SEQ. NO.
67 68 69 70 71 72	73 74 75 76	77 78 79 80
		0

Punch in all cards:

MATRIX NAME	SUB-SCRIPT	SEQ. NO.
67 68 69 70 71 72	73 74 75 76	77 78 79 80
		1

I	J	VALUE	EXP	I	J	VALUE	EXP	I	J	VALUE	EXP
2 3 4 5	6 7 8 9	11 12 13 14 15 16 17 18 19	20 21 22	24 25 26 27	28 29 30 31	33 34 35 36 37 38 39 40 41	42 43 44	46 47 48 49	50 51 52 53	55 56 57 58 59 60 61 62 63	64 65 66

Note: Matrix data for a case must be followed by an End of Matrix Data card (E in c.c.1)

e. Error Messages

Additional control error messages which pertain to the REPEAT module and which emanate from the instruction processor module are listed below.

INST10	STATEMENT NUMBER SPECIFIED WITHIN RANGE OF LOOP. STATEMENT NUMBER IGNORED
INST11	SYNTAX ERROR IN -REPEAT- INSTRUCTION
INST12	MATRIX NAME LEFT OF EQUALS SIGN NOT SUBSCRIPTED WITHIN RANGE OF LOOP
INST13	INVALID NESTED LOOPS
INST14	SYNTAX ERROR IN SUBSCRIPTED MATRIX NAME
INST15	INSUFFICIENT CORE STORAGE FOR PROCESSING LOOP
INST16	RANGE OF REPEAT LOOP IS UNSATISFIED
INST**	THIS INSTRUCTION NOT AVAILABLE

Other control error messages which include matrix names as additional descriptive information have been modified to accommodate subscripted matrix names.

f. Application

As an example of the use of REPEAT consider a nonlinear matrix equation of the form

$$A_0 + A_1 x + A_2 x^2 = 0$$

which may be processed iteratively to approximate x as follows.

$$x_{i+1} = - A_1^{-1} (A_0 + A_2 x_i^2)$$

The appropriate abstraction instruction sequence using REPEAT is as follows.

```
1      7
$INSTRUCTION
       A1INV    = -AI        .INVERS.
       X   /1/  =  A1INV    .MULT.    AO
       PRINT(,,,)X/1/
       REPEAT (6, 7)
       XR /1/   =  X   /1/  .RENAME.
       X2 /1/   =  XR  /1/  .EMULT.   X   /1/
       AX2/1/   =  A2       .MULT.    X2  /1/
       AAX/1/   =  A0       .ADD.     AX2/1/
       X   /2/  =  A1INV    .MULT.    AAX/1/
       PRINT(,,,)X/2/
```

where matrices A0, A1, and A2 are either card input or are available on an input matrix data set.

The effective expanded instruction sequence which would result is as follows.

```
$INSTRUCTION
       A1INV    = -A1        .INVERS.
       X   /1/  =  A1INV    .MULT.    AO
       PRINT(,,,)X/1/
       XR /1/   =  X   /1/  .RENAME.
       X2 /1/   =  XR  /1/  .EMULT.   X   /1/
       AX2/1/   =  A2       .MULT.    X2  /1/
       AAX/1/   =  A0       .ADD.     AX2/1/
       X   /2/  =  A1INV    .MULT.    AAX/1/
       PRINT(,,,)X/2/
       XR /2/   =  X   /2/  .RENAME.
       X2 /2/   =  XR  /2/  .EMULT.   X   /2/
       AX2/2/   =  A2       .MULT.    X2  /2/
       AAX/2/   =  A0       .ADD.     AX2/2/
       X   /3/  =  A1INV    .MULT.    AAX/2/
       PRINT(,,,)X/3/
                .
                .
                .
                .
       XR /7/   =  X   /7/  .RENAME.
       X2 /7/   =  XR  /7/  .EMULT.   X   /7/
       AX2/7/   =  A2       .MULT.    X2  /7/
       AAX/7/   =  A0       .ADD.     AX2/7/
       X   /8/  =  A1INV    .MULT.    AAX/7/
       PRINT(,,,)X/8/
```

A further illustration of the use of REPEAT in the synthesis of a fully stressed truss structure is given in Example 1 of the Appendix.

4. EIGENVALUE-EIGENVECTOR EXTRACTION (EIGEN3*)

a. General

A third eigenvalue-eigenvector extraction capability, EIGEN3, has been incorporated in Phase II of FORMAT to complement the existing routines of EIGEN and EIGEN2. EIGEN is an in-core routine for real, symmetric matrices (typically, with NWORK = 10000 the matrix may be as large as 130 x 130). EIGEN2 is an out-of-core routine for real, large order matrices (up to 2000) which may be nonsymmetric, and is specifically designed for analyses wherein the prime objectives are the largest eigenvalue and the corresponding eigenvector, e.g., elastic instability analyses. Subsequent eigenvalues and eigenvectors can be determined but rate of convergence may not be ideal in all cases.

EIGEN3 is an in-core routine based on the QR method for real, smaller order matrices which may be nonsymmetric. It extends the range of eigensolution capability in FORMAT by providing economic extraction of all eigenvalues and eigenvectors of a real, nonsymmetric matrix. In order to minimize the amount of calculations necessary, the matrix is first reduced to an upper Hessenberg form before the QR transform is applied. Acceleration of convergence is achieved by origin shifting and eigenvectors are obtained by inverse iteration. Internal array dimensioning and computation in double precision is necessary to satisfy accuracy requirements on a 36 bit word machine. Input and output remain in the standard FORMAT single precision form however with the necessary interface provided internally in EIGEN3. The double precision requirement necessarily restricts the size of eigenmatrix that can be accommodated by the routine.

b. Abstraction Instruction

Eigenvalue-Eigenvector Extraction Statements are of the form

$$c_1, c_2, c_3, c_4 \quad = \quad a\,.\mathrm{EIGEN3.}\ (d)$$

where $\underline{d}$ eigenvalues and the corresponding eigenvectors are extracted from the matrix $\underline{a}$, the real and imaginary parts of the eigenvalues are named matrix $\underline{c}_1$ and $\underline{c}_2$ and the real and imaginary parts of the eigenvectors are

---

* Originally made available by the University of Wisconsin as subroutine subprogram UNSEIG.

named matrix $\underline{c}_3$ and $\underline{c}_4$, respectively. The following auxiliary definitions apply with matrix $\underline{a}$ of order n x n .

c_1 - is the matrix of real eigenvalues ($\underline{d}$ x 1)

c_2 - is the matrix of imaginary eigenvalues ($\underline{d}$ x 1)

c_3 - is the matrix of real eigenvectors (n x $\underline{d}$)

c_4 - is the matrix of imaginary eigenvectors (n x $\underline{d}$)

a - is the name of the input eigenmatrix (n x n)

d - is the number of eigenvalues requested (an unsigned integer $\leq$ n)

The sequence of eigenvalues in matrices $\underline{c}_1$ and $\underline{c}_2$ is in decending order of the value of the summed-squares of the real and imaginary components. Rows of a matrix of eigenvalues correspond to columns of the associated matrix of eigenvectors. Each eigenvector is normalized on the element of largest size in that vector.

The size of input matrix that can be accommodated by EIGEN3 is dependent on the number of words of working core storage (NWORK) available. If N is the order of the input matrix then

$$\text{NWORK} \geq 8N^2 + 16N$$

Typically with NWORK = 10000, N may be as large as 34.

c. Error Messages

Error messages which emanate from EIGEN3 are listed below.

EIGEN301	MATRIX ****** COULD NOT BE FOUND
EIGEN302	THE SIZE OF WORKING STORAGE ****** MUST BE INCREASED TO HANDLE THE INPUT MATRIX ******
EIGEN303	THE INPUT MATRIX ****** IS DIMENSIONED ****, **** THE INPUT MATRIX MUST BE SQUARE
EIGEN304	THE NUMBER OF EIGENVALUES **** IS GREATER THAN THE ORDER OF THE INPUT MATRIX.

SECTION IV

PHASE III ADDITIONS

1. MATRIX PLOT

a. General

The graphic display capabilities in Phase III of the FORMAT system include a matrix plot feature which provides for the display of matrix data in the form of a plot of consecutive values from specified columns or partial columns of matrices as a function of row number. The values are plotted as ordinates with abscissas at equal intervals.

The limitations of such a plot have been eliminated by the incorporation of the following features.

- Linear distance scale for the axis of abscissas with the capability to plot values in user-specified rows of a matrix column as ordinates with user-specified abscissas.
- Optional specification of extreme values and grid intervals for both the axes of abscissas and the axis of ordinates.
- Multiple combinations of columns, matrices and absicass on the same frame.
- Extended labeling and identification capability.

In addition, a considerable reduction in execution time has been effected by modifying the mode of retrieval of matrix data.

Matrix plot thus provides almost all the capability implicit in a linear plot which is effected manually.

Basic input consists of the matrices to be processed and special data to describe the graphs desired. In any one execution of matrix plot up to 10 frames may be processed involving up to 100 abscissas. Up to 10 different sets of abscissas may be used. Tabular data identifying abscissa values and corresponding row numbers of the matrix values to be plotted as ordinates constitute part of the special input data. All of the matrix columns involved in the plot may be initially identified in order to reduce access time of the input data set. In such a case the columns will be automatically copied onto a scratch data set and retrieved from this data set during processing. This mode of operation is recommended as standard for all applications. For each frame plotted, special input data describing format and contents is required. Extreme values for the left and right ends of the axis of abscissas and the top and bottom ends of the axis of ordinates may be optionally set as may the size of the reference grid which is plotted. This provides for common axes on separate frames and hence easier subsequent comparison where required. A frame may contain up to 10 curves with optional line connectedness of the plotted points. Title lines (72 characters) may be appended to both axes and the legend relating plotted point symbols and matrix columns which appears at the top of the frame may have additional descriptive information appended at the user's option.

b. Input

(1) Control Data

Execution of matrix plot procedure is initiated by the FORMAT System by means of control cards in the Phase III input data as follows.

```
1     7        16
$PLOT          MATRIX
      DATA
```

This is exactly the same as previously. Special data following the DATA card is however completely revised and is described in detail below.

(2) Special Data

The format and data entries of the matrix plot special data are illustrated in Figure 2.

Card 1 contains an integer value NCASE which is the total number of frames desired.

Card 2 contains the integer values NABS and NSETS which are the total number of abscissas and the total number of sets of abscissas input. (See card 5)

Card 3 contains the literal constant COPY or NOCOPY left justified in card column 1 and the integer value NMAT. When COPY is entered, all of the matrix columns involved in the plot will be automatically extracted from the input data set(s) and copies onto a scratch data set. NMAT is then the total number of matrices from which the columns are to be extracted. When NOCOPY is entered no copy will be made and NMAT is left blank.

Card 4 is entered for each matrix from which columns are to be copies when COPY is entered on card 3. It contains the alphameric matrix name, NAME, the total number of columns to be copied from the

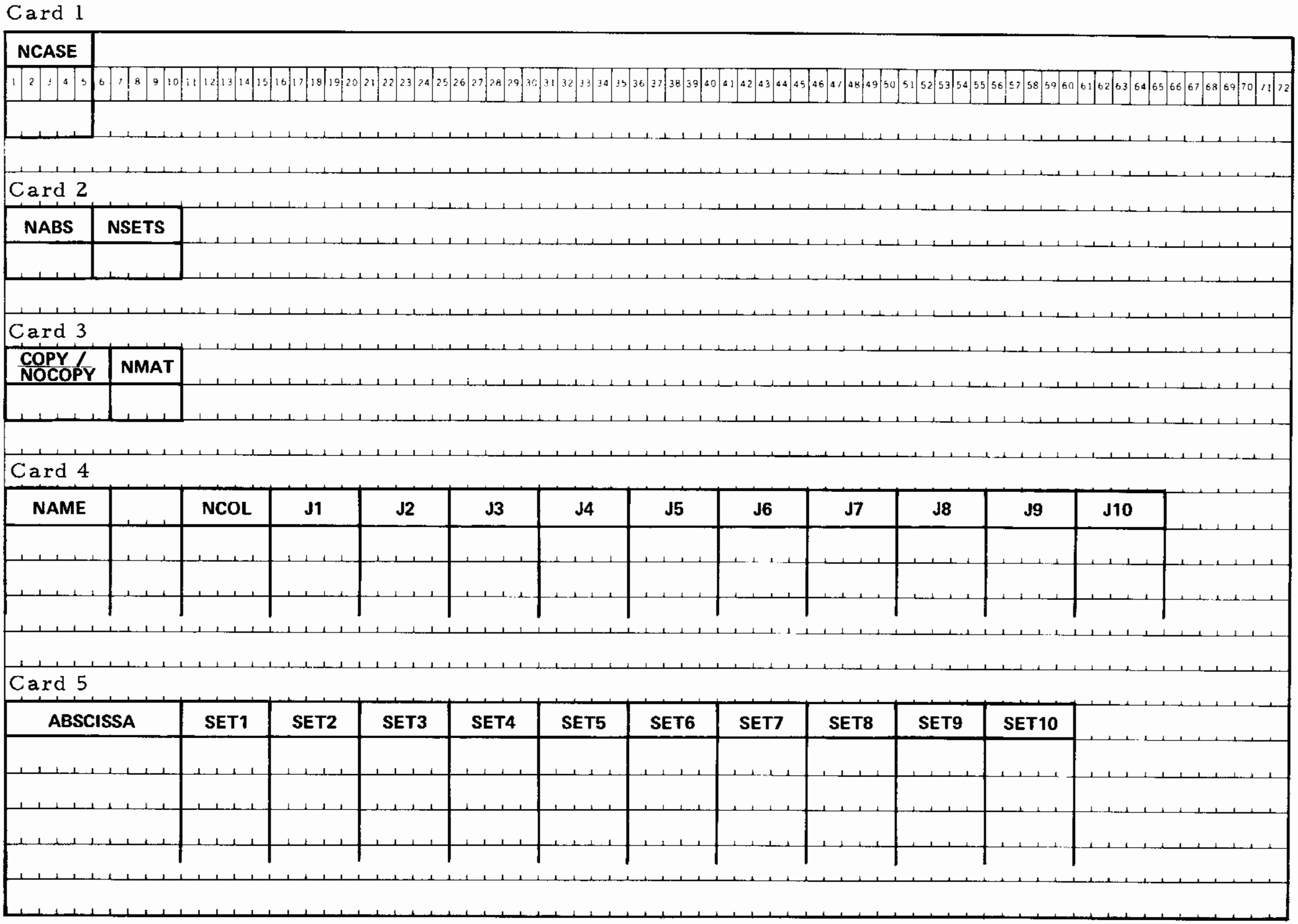


Figure 2. Special Data - Matrix Plot

Card 6

JCOL	XL	XR	DX	YB	YT	DY	
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 41 42 43 44 45	46 47 48 49 50 51 52 53 54 55	56 57 58 59 60 61 62 63 64 65	66 67 68 69 70 71 72

Card 7

VERTICAL TITLE

Card 8

HORIZONTAL TITLE

Card 9

NAME		COL	SET	LINE	LEGEND

Figure 2. Special Data - Matrix Plot (Continued)

matrix, NCOL, and the corresponding column numbers, J1, J2, . . J10. If more than 10 columns are involved sequential cards bearing only the additional column numbers may be entered as required (i. e., NAME and NCOL are omitted on such cards).

Card 5 is entered for each abscissa used in the plot. It contains the abscissa value, ABSCISSA, and the row numbers corresponding to the matrix elements to be plotted at this abscissa located in the appropriate abscissa set fields, SET1, SET2, ... SET10. The total number of such cards, NABS, and the total number of abscissa sets used, NSETS, are the values entered on card 2. The card 5 set provides a table relating abscissa value and ordinate row location in a matrix for subsequent use by simply referencing the set number, i. e., 1 through 10. Plotted point connectedness is defined by the sequence in which these cards are entered and if any row number entries are omitted from an abscissa set then connectedness will be from the entry before to the entry after the omitted. Discontinuous (stepped) plots may be obtained by repeating the abscissa value from one card to the next.

The complete set of cards 1 through 5 is entered once per plot. The remaining set of cards 6 through 9 is entered once per frame.

Card 6 contains the total number of columns (curves) plotted on the frame, integer JCOL, which must always be entered. The remaining entries are optional and are, in turn, the left and right extreme values and grid increment required for the axis of abscissas, XL, XR and DX, and the bottom and top extreme values and grid increment required for the axis of ordinates, YB, YT and DY. If the pair of extremes are omitted the values will automatically be set such that the axis reflects the range of values plotted and fills the standard frame size.

Card 7 contains the line of vertical title data (72 characters) to be incorporated on the left side of the frame.

Card 8 contains the line of horizontal title data (72 characters) to be incorporated at the bottom of the frame.

Card 9 is entered for each column (curve) plotted on the frame. It includes the name of the matrix containing the column to be plotted, NAME, and the appropriate column number, COL. The next entry is the abscissa set number, SET, which by reference to the card 5 entries defines which of the elements of the matrix column are to be plotted as ordinates and what their associated abscissas are. Line connectedness of plotted points, LINE, is obtained by the entry of integer 1 in this field (0 for no lines). The last entry, LEGEND, contains descriptive data (47 characters) to be appended to the symbol legend which unconditionally appears at the top of the frame. Finally, the total number of such cards must correspond to the value of JCOL entered on card 6.

c. Output

(1) Printed Data

A complete listing of the special data input is unconditionally printed. In addition, for each column plotted, the extracted values of abscissa and ordinate are tabulated.

(2) Plotted Data

Plotted points corresponding to a common curve are indicated on the plotted output by a common symbol as defined by the legend appearing at the top of the frame. The scales appended to both axes reflect the range of values plotted (or the user-specified extremes) and are adjusted to the standard seven-inch frame size with appropriate allowance for the symbol legend.

(3) Error Messages

Error messages emanating from the Matrix Plot procedure are listed below in alphabetical order on the first word.

COLUMN **** DOES NOT EXIST IN THIS MATRIX - ****** -
IT WILL BE IGNORED

COPY ERROR - THE ****TH MATRIX COULD NOT BE FOUND

INCORRECT COPY CARD OR CARD NONEXISTENT *****

NCASE = ****. DUE TO SIZE LIMITS ONLY FIRST 10 CASES
WILL BE USED

NCOL = ****. WARNING - THIS COL DOES NOT EXIST IN
THE ****TH MATRIX

NORDS = ****. DUE TO SIZE LIMITS ONLY FIRST 100 VALUES
WILL BE USED.

NSETS = ****. DUE TO SIZE LIMITS ONLY FIRST 10 VALUES
WILL BE USED

NUMCOL = **** DUE TO SIZE LIMITS ONLY FIRST 10 COLS
WILL BE USED

PLOTXY ERROR IN X-AXIS SCALING

PLOTXY ERROR IN Y-AXIS SCALING

PLXY03 MATRIX ****** COULD NOT BE FOUND

```
PLXY04 MATRIX ****** IS NOT AMONG ONE OF THE AVAILABLE
INPUT MATRICES

PLXY05 ALL ELEMENTS IN COLUMN **** OF MATRIX ******
ARE ZERO OR THIS COLUMN NUMBER EXCEEDS THE
MAXIMUM COLUMN NUMBER OF THE MATRIX

TABLE SIZE LIMITS HAVE BEEN EXCEEDED

THE MAXIMUM NO. OF COLS HAS BEEN EXCEEDED -
ICOL = ****  JMAX = ****

THE MAXIMUM NO. OF ROWS HAS BEEN EXCEEDED -
IROW = ****  IMAX = ****

THE MAX NUMBER OF 100 COLS TO BE COPIED HAVE
BEEN EXCEEDED IN THE ABOVE LINE
```

d. Application

A practical illustration of the use of the revised matrix plot capability is given in Example 2 of the Appendix.

SECTION V

OPERATIONAL CONSIDERATIONS

1. REVIEW OF IMPLEMENTATIONS

The current version of the FORMAT System has been implemented and is operational on a number of different computers. Known implementations at several installations are shown in Table II.

TABLE II. FORMAT IMPLEMENTATIONS

Computer	Operating System
IBM 7094 IBM 7094/7044 DCS IC 6000	IBSYS IBSYS IBSYS
IBM 360 Model 50 IBM 360 Model 65 IBM 360 Model 75 IBM 360 Model 85	OS 360/MVT OS 360/ASP OS 360/ASP OS 360/ASP
GE 635	GECOS II GECOS III
UNIVAC 1108	EXEC VIII
CDC 6400 CDC 6600	SCOPE 3.0 SCOPE 3.1

Optimal implementation of the FORMAT System in the multi-task environment of third generation machinery requires in-depth consideration of all operating system characteristics at the installation. Salient aspects common to the various operating systems and general recommendations aimed at improved operating procedures are presented in the following section.

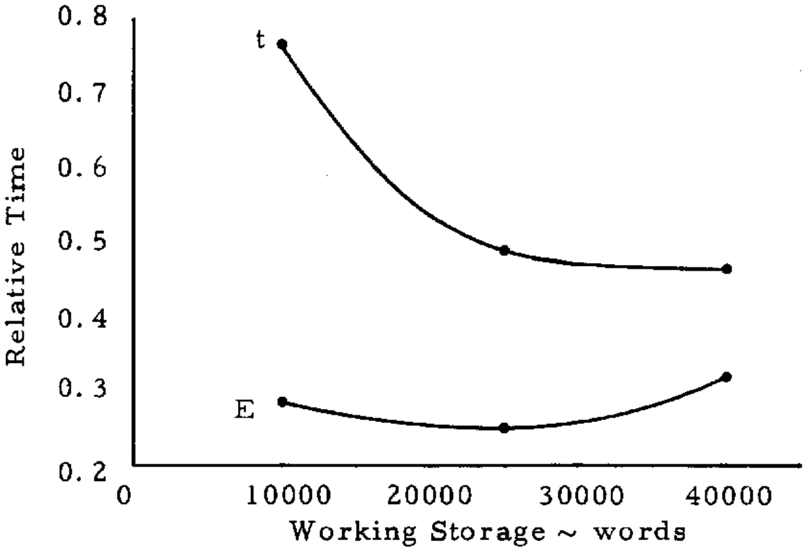
2. RECOMMENDATIONS

a. General Considerations

The experience gained in implementing the FORMAT System on the various operating systems described above has provided additional information regarding the interaction between the strategy used by FORMAT and third generation operating system environments. This information emphasizes that the cost effectiveness of FORMAT is highly dependent on the degree of compatibility achieved between it and the operating system on which it is implemented. In addition to those considerations previously discussed in Section IV of Reference 1 regarding program implementation, effective implementation of the FORMAT System requires consideration of all operating system characteristics, and, in particular, data set buffering, record blocking, basic cycle times for computation and input/output, FORTRAN compiler code optimization, and the algorithm used by the operating system to determine individual run costs.

The basic strategy of FORMAT Phase II is to store all intermediate matrices as well as all master input and output matrices on peripheral data sets which requires a considerable amount of input/output processing. Regardless of the size of working storage used by FORMAT, the same amount of input/output processing will be done for storing and retrieving this matrix data. While this input/output takes place, the working storage area is effectively idle. A second type of input/output processing is required in individual matrix operations in Phase II such as an out of core matrix multiply where multiple passes must be made over the premultiplier matrix. In this case, enlarging the size of working storage will reduce the amount of this type of input/output processing by reducing the number of passes required. However, the cost of all input/output processing can be reduced on most operating systems by increasing the size and number of buffers allocated to each data set. Therefore, some tradeoffs are necessary between core requirements for FORMAT working storage and for data set buffers.

Since in third generation multi-task environments, job cost is determined not only by execution time which is comprised of both computation time and input/output processing time, but also by core requirements and other machine resources used, overall cost effectiveness can therefore be achieved only by minimizing both run time and machine resources according to the job cost account system being used. To illustrate, the graph below is a plot of relative time versus the working storage used by FORMAT for a given large order (out of core) matrix multiplication.



Peripheral data set parameters, e.g., size and number of buffers, record blocking, were optimized for the operating system being used and were held constant for all runs. The run time, t, decreases as the size of working storage is increased since this reduces the number of passes over the premultiplier matrix. However, at approximately 25,000 words for working storage, the effective time, E, which corresponds directly to run cost, is at a minimum. On this operating system, the run cost is given by $E = tRK$ where t is the run time, R is the total core requirement, and K is a constant used by the job cost accounting system.

Beyond 25,000 words of working storage, the effective time increases as the total core requirement increases, the implication being that the cost of additional core required to reduce the amount of input/output processing (passes over the premultiplier matrix) is greater than the savings due to decreased running time. Consider the limiting case where working storage is half that required to do a matrix multiplication in core in which case only one pass is necessary over the premultiplier matrix. It becomes clear that the cost of doubling the size of working storage would be far greater than the savings obtained by eliminating the input/output necessary for one pass.

b. Implementation

Effective FORMAT implementation requires three basic steps. First, timing and cost test runs should be made to optimize all input/output processing. Optimum data set parameters can then be determined and specified in subsequent runs using the appropriate operating system control cards. Second, timing and cost test runs should be made to optimize the size of working storage used in Phase II of FORMAT. These runs should use the optimum data set parameters determined in the first step. Preferably, these runs would be single matrix operations using nominal size matrices. For example, A = B .MULT. C where B is 1000 x 500, C is 500 x 500, the maximum matrix size, KONST, is 2000, and the minimum working storage, NWORK, is 9000 or 4.5 KONST. Having the results of these two steps, step three is the selection of an optimum total core requirement for storing the FORMAT program, work space, and buffer space. This value is primarily a function of the operating system environment (total available core and job cost accounting system) and the nominal size of production runs. After determining total core requirements and allotting space for storing the FORMAT program, the remaining core space should be allotted to FORMAT working storage and data set buffers in an optimum manner consistent with the findings of steps 1 and 2.

Following is a list of specific considerations and recommendations for optimal implementation of the system.

- FORMAT + WORK AREA + BUFFERS = CORE REQUIRED

  Minimum NWORK = 4.5 x KONST
  Maximum NWORK is not necessarily optimum

- Use at least two buffers per data set (20% savings in MULT on 7094 going from single to double buffers)

- Determine buffer size for optimized input/output processing cost

- Optimize NWORK by running test cases on typical MULT operation (matrix of order 1/2 KONST)

- Use variable blocked record format (80% savings in MULT on 360 going from variable to variable blocked records)

- Use separate access mechanisms for each data set

- Consider an implementation with near minimum NWORK to be used with small order problems

Recommendations for optimal use of the system are as follows.

- Save matrices for restart purposes on a different master output data set from those saved for future use

- Save matrices for future use in optimum groups for future operations

- Take advantage of the efficiency afforded in the MULT and the TMULT operation when the premultiplier fits in core.

- Use the STRCUT operation for the solution of simultaneous equations or inversion with sparse coefficient matrices.

SECTION VI

CONCLUSIONS

The FORMAT System has been updated by the incorporation of additional basic capability and the refinement of existing capability.

A simpler mode of updating case data and extended force method matrix generation capability has been incorporated in Phase I of the system. A refined "Structure Cutter" module, capabilities for matrix partitioning and instruction looping, and an additional eigenvalue/eigenvector extraction module have been incorporated in Phase II. Finally the limitations which existed in the matrix plotting capability in Phase III have been eliminated.

Recommendations of improvements in implementation and utilization procedures based on experience gained on various computer systems are also presented. Potential cost savings afforded by efficient implementation of the system are shown to be considerable.

The example chosen to illustrate the use of the instruction looping capability also demonstrates the feasibility of fully stressed design optimization in FORMAT. In the practical case a significant amount of hand generated, card input data is currently required in such an application. The provision for automatic generation of this data is a logical extension of the basic force method matrix generator, and would enable production analyses of this type to be routinely performed.

APPENDIX

ILLUSTRATIVE EXAMPLES

This appendix illustrates the use of the new features of FORMAT in typical structural analyses.

The first of these examples demonstrates the use of the REPEAT capability in the synthesis of a fully-stressed redundant structure. The structure is limited in size and complexity in order to illustrate all aspects of input and output as simply as possible, but the mode of operation is equally applicable to a large complex system.

The second example demonstrates the use of the MATRIX PLOT capability in a production type structural application. Internal loads and deflections for a structural component of a major substructure of a current aircraft type are plotted for selected load conditions. The mode of retrieval of the plotted data from the linear analysis matrix data is illustrated in detail.

1. SYNTHESIS OF A FULLY STRESSED TRUSS STRUCTURE

a. Description

The simple four bar truss shown in Figure 3 is used to demonstrate the use of the instruction looping capability (REPEAT) in the synthesis of a fully stressed structure. A simple structure, limited in size and complexity, is chosen in order to illustrate all aspects of input and output as simply as possible. The method of solution can however be extended to a large complex system and is therefore discussed in detail. The basic linear analysis of the truss is by the force method and therefore serves to illustrate the use of the revised "Structure Cutter" module in an abstraction instruction sequence. Complete idealization data for the truss is given in Figure 3.

b. Derivation of Method

In the linear analysis of statically inderterminate structures by the force method, the element forces F are given by the matrix equation

$$F = f_x X + f_\phi \phi \qquad (1)$$

where f_x and f_ϕ are matrices of the element forces in the structure statically equivalent to unit values of the redundants X and the applied loads ϕ, respectively. The redundants are given by

$$X = - (f_x^T D f_x)^{-1} f_x^T D f_\phi \phi \qquad (2)$$

where D is a matrix of the flexibilities of the unassembled elements of the structure and the superscript T indicates transposition.

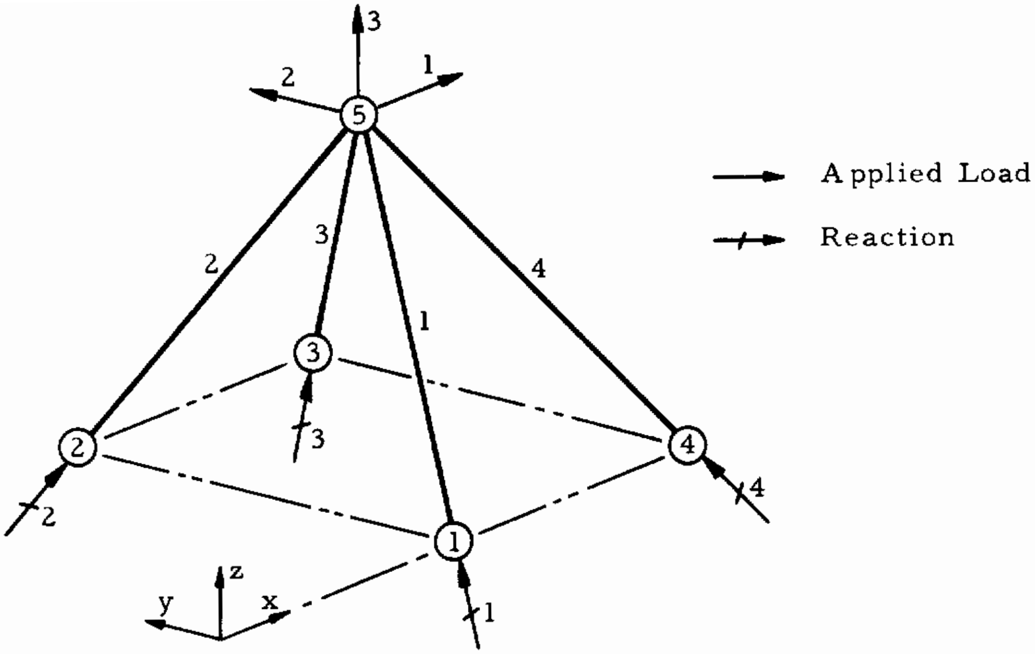
Define the following matrices to conform with the row format of the matrix of elements forces F.

σ_T – diagonal matrix of allowable tensile stresses (all positive)

σ_C – diagonal matrix of allowable compressive stresses (all negative)

A_M – column matrix of lower limit bar areas

A – column matrix of original bar areas



Joint No.	Coordinates (in.)		
	x	y	z
1	400	0	0
2	400	192	0
3	604	192	0
4	604	0	0
5	460	120	96

Vector No.	Applied Loads (lb.)		
	Cond. 1	Cond. 2	Cond. 3
1	10000	20000	-20000
2	10000	20000	20000
3	-30000		

Bar No.	$\ell/2$ (in.)
1	82.486
2	67.082
3	93.723
4	105.300

Lower limit on all bar areas = 0.1 in.^{2}

Material - $E = 10^7$ lb. in.$^{-2}$

$\rho = 0.1$ lb. in.$^{-3}$

Stress limits + 30000 lb.in.$^{-2}$

- 25000 lb.in.$^{-2}$

Figure 3. Truss Idealization

Determine matrices A_T and A_C from the following equations

$$A_T = \sigma_T^{-1} F = \sigma_T^{-1} f_\phi \phi + \sigma_T^{-1} f_x X \tag{3}$$

$$A_C = \sigma_C^{-1} F = \sigma_C^{-1} f_\phi \phi + \sigma_C^{-1} f_x X \tag{4}$$

Then a first approximation to the optimum bar areas necessary to satisfy allowable stress and minimum gage requirements is given by the maximum positive elements in the rows of matrices A_T, A_C and A_M adjoined. Let this adjoined matrix be denoted by $\overline{A}$, which can be expressed matrically as

$$\overline{A} = [A_T \vdots A_C \vdots A_M] \tag{5}$$

A column matrix of first approximation optimum bar areas A_N is then given by

$$A_N = [\,\overline{A}\,]_{MAX} \tag{6}$$

and the matrix of element flexibilities D_N associated with these areas may be obtained from the equation

$$D_N = \lceil A \rfloor \lceil A_N \rfloor^{-1} D \tag{7}$$

where the $\lceil\;\rfloor$ brackets indicate a diagonal matrix.

Equations (2) through (7) may be solved iteratively using the matrices A_N and D_N from Equations (6) and (7) which result from one cycle, as the starting values for matrices A and D in the next cycle. The sequence described is repeated until the required convergence of bar areas is obtained.

Equation (5) may also be written

$$\overline{A} = [\,(A_\phi + A_x) \vdots A_M\,] \tag{8}$$

where
$$A_\phi = (\sigma_T^{-1} f_\phi \phi \vdots \sigma_C^{-1} f_\phi \phi) \tag{9}$$

and $$A_x = (\sigma_T^{-1} f_x X \vdots \sigma_C^{-1} f_x X) \qquad (10)$$

Matrices A_ϕ and A_M remain constant during the iteration and hence the creation of matrix A in this form is more economical than by the use of Equation (5).

c. Solution Procedure

The solution procedure in FORMAT is as follows.

(1) Execute Phase I of FORMAT to generate the matrices necessary for a linear analysis of the structure by the force method. Use unit values for the bar areas. The generated matrices are identified as follows.

$$D, W, e_T^T, I_F, I_E, P_F^T, P_\phi^T, T_F, T_\phi \; G_R, G_B^T, G_{PF}, G_{PS}$$

(2) Execute Phase II of FORMAT to determine the following constant matrices.

f_{x_r}, f_{ϕ_r} from "Structure Cutter" on P_F^T, P_ϕ^T and W.

$$f_x = T_F \cdot f_{xr}$$

$$f_\phi = T_F \cdot f_{\phi r}$$

$$f_\phi \Phi$$

$$A_\phi = (\sigma_T^{-1} f_\phi \Phi \vdots \sigma_C^{-1} f_\phi \Phi)$$

$$A_{x\phi T} = \sigma_T^{-1} f_x$$

$$A_{x\phi C} = \sigma_C^{-1} f_x$$

(3) Continue execution of Phase II by repeated determination of the following matrices until the required convergence of bar areas

is obtained. Matrices which change from one cycle to the next are indicated by numerical subscripts which are appropriate to the first cycle

$$X_1 = (f_x^T D_1 f_x)^{-1} f_x^T D_1 f_\phi \phi$$

$$A_{x1} = (\sigma_T^{-1} f_x X_1 \vdots \sigma_C^{-1} f_x X_1)$$

$$\overline{A}_1 = [(A_\phi + A_{x1}) \vdots A_M]$$

$$A_2 = [\overline{A}_1]_{MAX}$$

$$D_2 = \lceil A_1 \rfloor \lceil A_2 \rfloor^{-1} D_1$$

Four iterations should be sufficient to effect the necessary convergence on bar areas in most practical applications. In such a case the matrices generated in the fourth cycle would by X_4, A_{x4}, $\overline{A}_4$, A_5 and D_5.

(4) Complete execution of Phase II by evaluation of internal forces, stresses, deflections, etc., appropriate to the final areas, A_4.

$$F_4 = f_\phi \phi + f_x X_4$$

$$\sigma_4 = \lceil A_4 \rfloor^{-1} F_4$$

Since matrix D_5 would be generated but not used in such an application, it must either be printed or saved to effect a successful execution of Phase II. In the normal application the matrices of bar areas A_1, A_2, . . etc., would be printed for convergence comparison, and matrices of total structure weight, W_1, W_2, . . . etc., would be generated and printed also.

d. Input

Complete input for both Phases I and II is listed in Table III.

TABLE III. INPUT DATA FOR EXAMPLE 1

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        10        20        30        40        50        60        70        80
$GENERATOR
      CASE OUTPUT TAPE   ( DOCAS, 021670 )
      MATRIX OUTPUT TAPE ( DOGEN, 021670 )
$EDITOR        NEW
      CREATE  TRUSS ( DOCAS, 021670 )
      DATA
 1   1   400.0       0.0       0.0
 1   2   400.0     192.0       0.0
 1   3   604.0     192.0       0.0
 1   4   604.0       0.0       0.0
 1   5   460.0     120.0      96.0
 3   1                                  1   5                1
 3   2                                  2   5                1
 3   3                                  3   5                1
 3   4                                  4   5                1
 4   1    1.0                           5                    1
 4   2              1.0                 5                    1
 4   3                        1.0       5                    1
19
20   1    1   5          1.0
20   2    2   5          1.0
20   3    3   5          1.0
20   4    4   5          1.0
20   11                 10.0+6
20   21                 10.0+6
20   31                 10.0+6
20   41                 10.0+6
99
$MATGEN        FORCE
      REF  TRUSS ( DOCAS, 021670 )
      OPTION ( BASIC, ALL )
$END
```

TABLE III. INPUT DATA FOR EXAMPLE 1 (CONTINUED)

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         10        20        30        40        50        60        70        80
$FORMAT
$RUN           GO, LOGIC
      ANALYSIS    ( FORMAT IV - ILLUSTRATIVE EXAMPLE NO. 1)
      PROBLEM     ( EXAMPLE NO. 1 - FULLY STRESSED TRUSS   )
      INPUT TAPE  ( DOGEN, 021670 )
      OUTPUT TAPE ( DOMAT, 021670 )
$INSTRUCTION
      FXR, FOR  =  PFT    .STRCUT.  POT , (1.0E-6,,W,,)
      FX        =  TF     .MULT.    FXR
      FO        =  TF     .MULT.    FOR
      FOO       =  FO     .MULT.    O
      STREC     =  ST     .POWER.   CONST(1,1)
      SCREC     =  SC     .POWER.   CONST(1,1)
      AOT       =  STREC  .MULT.    FOO
      AOC       =  SCREC  .MULT.    FOO
      AO        =  AOT    .ADJOIN.  AOC
      AXOT      =  STREC  .MULT.    FX
      AXOC      =  SCREC  .MULT.    FX
C
      REPEAT (16,4)
      WT  /1/   =  RHOL   .MULT.    A   /1/
      DFX /1/   =  D   /1/.MULT.    FX
      DXX /1/   =  DFX /1/.TMULT.   FX
      DXO /1/   =  DFX /1/.TMULT.   FOO
      X   /1/   = -DXX /1/.SEQEL.   DXO /1/
      AXT /1/   =  AXOT   .MULT.    X   /1/
      AXC /1/   =  AXOC   .MULT.    X   /1/
      AX  /1/   =  AXT /1/.ADJOIN.  AXC /1/
      ATC /1/   =  AO     .ADD.     AX  /1/
      ABAR/1/   =  ATC /1/.ADJOIN.  AM
      AENV/1/   =  ABAR/1/.ENVROW.
      A   /2/   =  AENV/1/.MULT.    C
      AREC/2/   =  A   /2/.POWER.   CONST(1,1)
      ARAT/2/   =  A   /1/.EMULT.   AREC/2/
      ARD /2/   =  ARAT/2/.DIAGON.
      D   /2/   =  ARD /2/.EMULT.   D   /1/
```

TABLE III. INPUT DATA FOR EXAMPLE 1 (CONTINUED)

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         10        20        30        40        50        60        70        80
      PRINT(,,,)WT/1/,WT/2/,WT/3/,WT/4/,D/5/
      FXX /4/   =  FX      .MULT.    X   /4/
      F   /4/   =  FOO     .ADD.     FXX /4/
      AINV/4/   =  AREC/4/.DIAGON.
      S   /4/   =  AINV/4/.MULT.     F   /4/
      PRINT(,,,) A/2/,A/3/,A/4/,A/5/,F/4/,S/4/
$MATRIX        LIST
H   3   3                                                         O
    1   1  10000.         2   1  20000.         3   1 -20000.     O
    1   2  10000.         2   2  20000.         3   2  20000.     O
    1   3 -30000.                                                 O
H  12  12                                                         ST
    5   5  30000.         6   6  30000.         7   7  30000.     ST
    8   8  30000.         9   9  30000.        10  10  30000.     ST
   11  11  30000.        12  12  30000.                           ST
H  12  12                                                         SC
    5   5 -25000.         6   6 -25000.         7   7 -25000.     SC
    8   8 -25000.         9   9 -25000.        10  10 -25000.     SC
   11  11 -25000.        12  12 -25000.                           SC
H  10   1                                                         CONST
    1   1   -1.0                                                  CONST
H   1  12                                                         RHOL
    1   5  82.486  -01   1   6  67.082  -01   1   7  93.723  -01RHOL
    1   8 105.300  -01   1   9  82.486  -01   1  10  67.082  -01RHOL
    1  11  93.723  -01   1  12 105.300  -01                       RHOL
H  12   1                                                         AM
    5   1    0.1         6   1    0.1         7   1    0.1        AM
    8   1    0.1         9   1    0.1        10   1    0.1        AM
   11   1    0.1        12   1    0.1                             AM
H   2   1                                                         C
    1   1    1.0                                                  C
H  12   1                                                         A
    5   1    1.0         6   1    1.0         7   1    1.0        A
    8   1    1.0         9   1    1.0        10   1    1.0        A
   11   1    1.0        12   1    1.0                             A
E
$END
```

In the abstraction instruction sequence of Phase II, the matrix names assigned are capitalized versions of the matrix notation used in the preceding theory where possible. Card input matrices σ_T, σ_C, A, A_M and $\rho \ell$ (with matrix names ST, SC, A, AM and RHOL) have a row format which corresponds to that of the matrix of internal element forces. This row format is listed in the force method generator output but can also be readily assembled à priori by the user. In this example the format is as follows.

Rows 1, 2, 3, 4 → Reactions 1, 2, 3, 4
Rows 5, 6, 7, 8 → Axial load at P end of bars 1, 2, 3, 4
Rows 9, 10, 11, 12 → Axial load at Q end of bars 1, 2, 3, 4

The above card input matrices therefore contain zeros in rows 1 through 4 and entries in rows 5 through 12 as follows

σ_T, a diagonal matrix, containing the constant 30000.0
σ_C, a diagonal matrix, containing the constant -25000.0
A , a column matrix, containing the constant 1.0
A_M, a column matrix, containing the constant 0.1
$\rho \ell$, a column matrix, containing the product of the half bar lengths and weight densities

The matrix $\rho \ell$ is used to generate structure weight by use of the equation

$$W_T = (\rho \ell)^T A \tag{11}$$

Three other matrices are also card input with matrix names CONST, C and Φ. Matrix CONST contains the single element -1.0 which is used as exponent in POWER operations to invert matrix elements (and hence to invert diagonal matrices containing some zero diagonal elements) Matrix C is a postmultiplier which extracts the first column of an n x 2 matrix; in this case the matrix of maximum and minimum values of bar areas which is obtained by the ENVROW operation. Matrix Φ is a matrix of the three applied load conditions.

e. Output

Matrix output for the problem is presented in Table IV. The matrices of structure weight, WT/*/ are appropriate to the start of the four iterations and the matrices of bar areas, A/*/, are for the end of each iteration. Matrices of internal loads and stress, F/4/ and S/4/, at the end of the fourth iteration are included. Bar stresses at or near the limiting values are underlined.

TABLE IV. OUTPUT DATA FOR EXAMPLE 1

```
                  EXAMPLE NO. 1 - FULLY STRESSED TRUSS
                                    MATRIX  WT    /   1/                              PAGE    1
                 CUTOFF  =   0.000000E-38                  SIZE     1  BY     1
            ROW                 ROW                 ROW                 ROW
COL        1   1    0.697182E 02
                                    MATRIX  WT    /   2/                              PAGE    1
                 CUTOFF  =   0.000000E-38                  SIZE     1  BY     1
            ROW                 ROW                 ROW                 ROW
COL        1   1    0.451729E 02
                                    MATRIX  WT    /   3/                              PAGE    1
                 CUTOFF  =   0.000000E-38                  SIZE     1  BY     1
            ROW                 ROW                 ROW                 ROW
COL        1   1    0.443474E 02
                                    MATRIX  WT    /   4/                              PAGE    1
                 CUTOFF  =   0.000000E-38                  SIZE     1  BY     1
            ROW                 ROW                 ROW                 ROW
COL        1   1    0.439889E 02
```

TABLE IV. OUTPUT DATA FOR EXAMPLE 1 (CONTINUED)

```
                  EXAMPLE NO. 1 - FULLY STRESSED TRUSS

                                   MATRIX  A     /   2/                              PAGE    1

                 CUTOFF =    0.000000E-38                  SIZE    12  BY     1

             ROW                ROW                ROW                ROW

 COL       1   5   0.793884E 00   6   0.738715E 00   7   0.725218E 00   8   0.406993E 00
               9   0.793884E 00  10   0.738715E 00  11   0.725218E 00  12   0.406993E 00

                                   MATRIX  A     /   3/                              PAGE    1

                 CUTOFF =    0.000000E-38                  SIZE    12  BY     1

             ROW                ROW                ROW                ROW

 COL       1   5   0.855121E 00   6   0.724685E 00   7   0.744820E 00   8   0.311312E 00
               9   0.855121E 00  10   0.724685E 00  11   0.744820E 00  12   0.311312E 00

                                   MATRIX  A     /   4/                              PAGE    1

                 CUTOFF =    0.000000E-38                  SIZE    12  BY     1

             ROW                ROW                ROW                ROW

 COL       1   5   0.884227E 00   6   0.713983E 00   7   0.759773E 00   8   0.265002E 00
               9   0.884227E 00  10   0.713983E 00  11   0.759773E 00  12   0.265002E 00

                                   MATRIX  A     /   5/                              PAGE    1

                 CUTOFF =    0.000000E-38                  SIZE    12  BY     1

             ROW                ROW                ROW                ROW

 COL       1   5   0.898995E 00   6   0.707300E 00   7   0.769109E 00   8   0.237529E 00
               9   0.898995E 00  10   0.707300E 00  11   0.769109E 00  12   0.237529E 00
```

TABLE IV. OUTPUT DATA FOR EXAMPLE 1 (CONTINUED)

```
          EXAMPLE NO. 1 - FULLY STRESSED TRUSS

                         MATRIX  F     /   4/                                     PAGE    1

          CUTOFF =    0.000000E-38                 SIZE     12  BY      3

        ROW                  ROW                  ROW                  ROW

COL     1  1   -0.556921E 04   2   0.176825E 05   3   0.192277E 05   4   0.162513E 04
           5    0.556921E 04   6  -0.176825E 05   7  -0.192277E 05   8  -0.162513E 04
           9    0.556921E 04  10  -0.176825E 05  11  -0.192277E 05  12  -0.162513E 04

COL     2  1   -0.269699E 05   2  -0.437344E 04   3   0.122890E 04   4  -0.396157E 04
           5    0.269699E 05   6   0.437344E 04   7  -0.122890E 04   8   0.396157E 04
           9    0.269699E 05  10   0.437344E 04  11  -0.122890E 04  12   0.396157E 04

COL     3  1    0.558204E 04   2   0.151904E 05   3  -0.212231E 05   4  -0.712587E 04
           5   -0.558204E 04   6  -0.151904E 05   7   0.212231E 05   8   0.712587E 04
           9   -0.558204E 04  10  -0.151904E 05  11   0.212231E 05  12   0.712587E 04

          EXAMPLE NO. 1 - FULLY STRESSED TRUSS

                         MATRIX  S     /   4/                                     PAGE    1

          CUTOFF =    0.000000E-38                 SIZE     12  BY      3

        ROW                  ROW                  ROW                  ROW

COL     1  5    0.629839E 04   6  -0.247660E 05   7  -0.253072E 05   8  -0.613251E 04
           9    0.629839E 04  10  -0.247660E 05  11  -0.253072E 05  12  -0.613251E 04

COL     2  5    0.305011E 05   6   0.612542E 04   7  -0.161746E 04   8   0.149492E 05
           9    0.305011E 05  10   0.612542E 04  11  -0.161746E 04  12   0.149492E 05

COL     3  5   -0.631291E 04   6  -0.212756E 05   7   0.279335E 05   8   0.268899E 05
           9   -0.631291E 04  10  -0.212756E 05  11   0.279335E 05  12   0.268899E 05
```

The variation of bar area and structural weight with iteration is also illustrated in Figure 4. Asymptotes are for bar stresses within ½% of the limiting value and were obtained by continued iteration of the analysis. The most significant aspect of this plot is that the structure weight is within 5% of the minimum value after the first iteration.

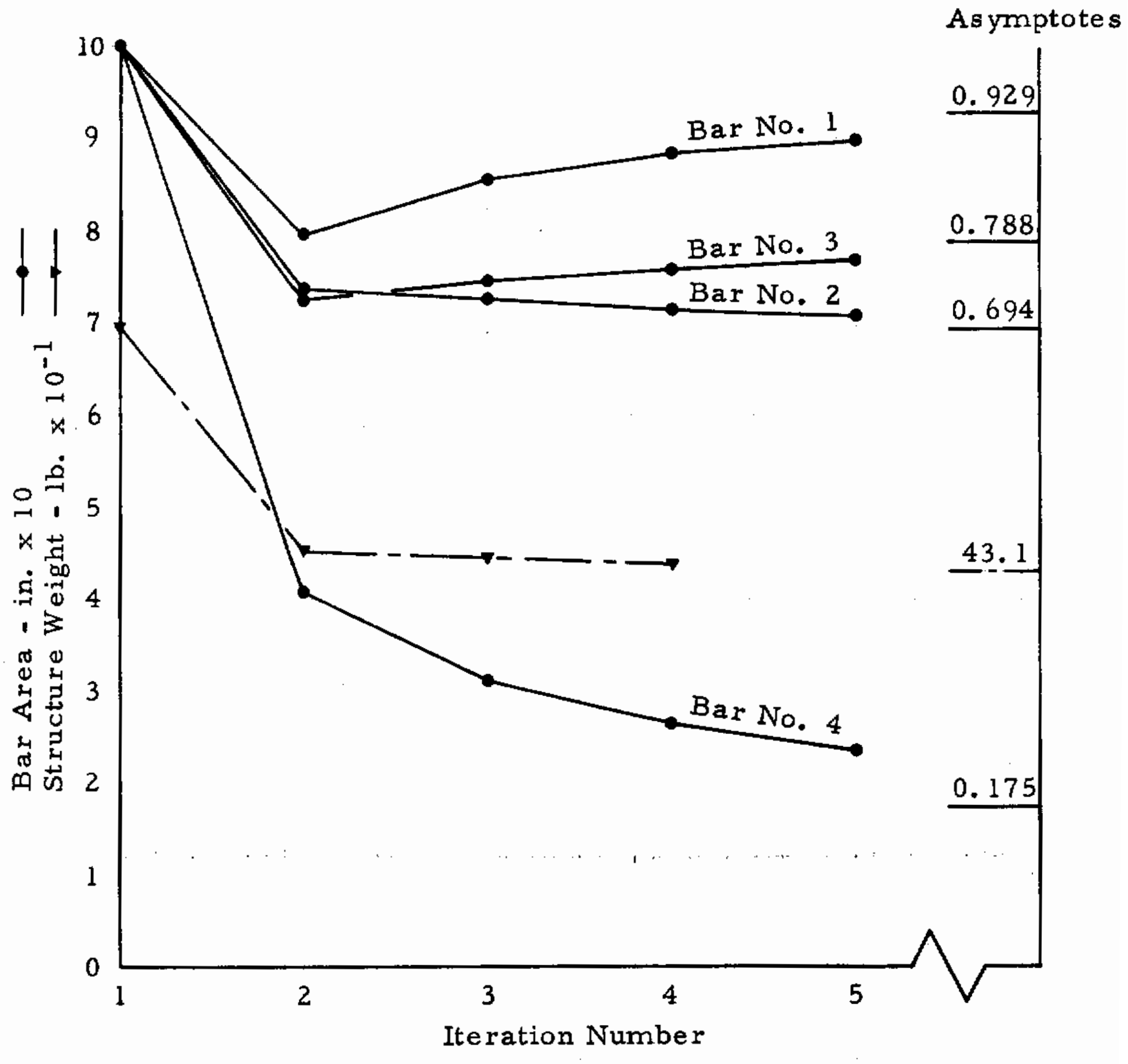


Figure 4. Iterated Values of Truss Bar Areas and Total Weight

2. ILLUSTRATION OF MATRIX PLOT

a. Description

The use of the matrix plot capability is illustrated in a typical production structural application. Internal loads and deflections for a structural component of a major substructure of a current aircraft type are plotted for selected external loading conditions. This specific example is chosen since it represents data that emanates from a strength analysis which would normally be hand plotted for comparison purposes during the development of a design and also for inclusion in a final strength report. It therefore serves to illustrate the mode of retrieval of the plotted data from the linear analysis matrix data.

The idealized form of the structural component and its location in the analyzed substructure is shown in Figure 5. It consists of the front spar of a lower vertical tail assembly, the latter being one of 16 substructures which were analytically joined to give a solution for the complete rear fuselage/empennage assembly. The substructure is geometrically symmetric about the aircraft buttock plane and consequently only one half was modelled and analyzed by the symmetric/antisymmetric reaction disconnect process in FORMAT. Internal forces and structural deflections are therefore available for both left and right sides of the substructure due to nonsymmetric loading.

The desired plot is as follows.

(i) Bar axial loads for the inner and outer caps of the spar "ring" as a function of angular distance from aircraft centerline for both left and right sides of the spar separately.

(ii) Shear forces along the inner edge of the ring web as a function of angular distance from aircraft centerline for both left and right sides.

(iii) Lateral deflection of the left edge of the spar as a function of vertical distance above the fuselage reference plane.

In each case the plot is for three loading conditions, numbered 4, 12 and 20 in the sequence analysed.

Matrices of bar element forces named BARL and BARR, and shear panel element forces named PANQL and PANQR, for the left and right sides respectively are contained on the matrix data set named "VSS021, 121891". The matrix of structural deflections named DTJ021 is contained on a second matrix data set named "JOIN02, 120391".

Partial listings of the joint and applied load vector data and of the row format of the matrices of resultant bar loads and shear panel forces are given in Table V. These are automatically printed when the Basic Force Method Module is executed and are used here to identify abscissa

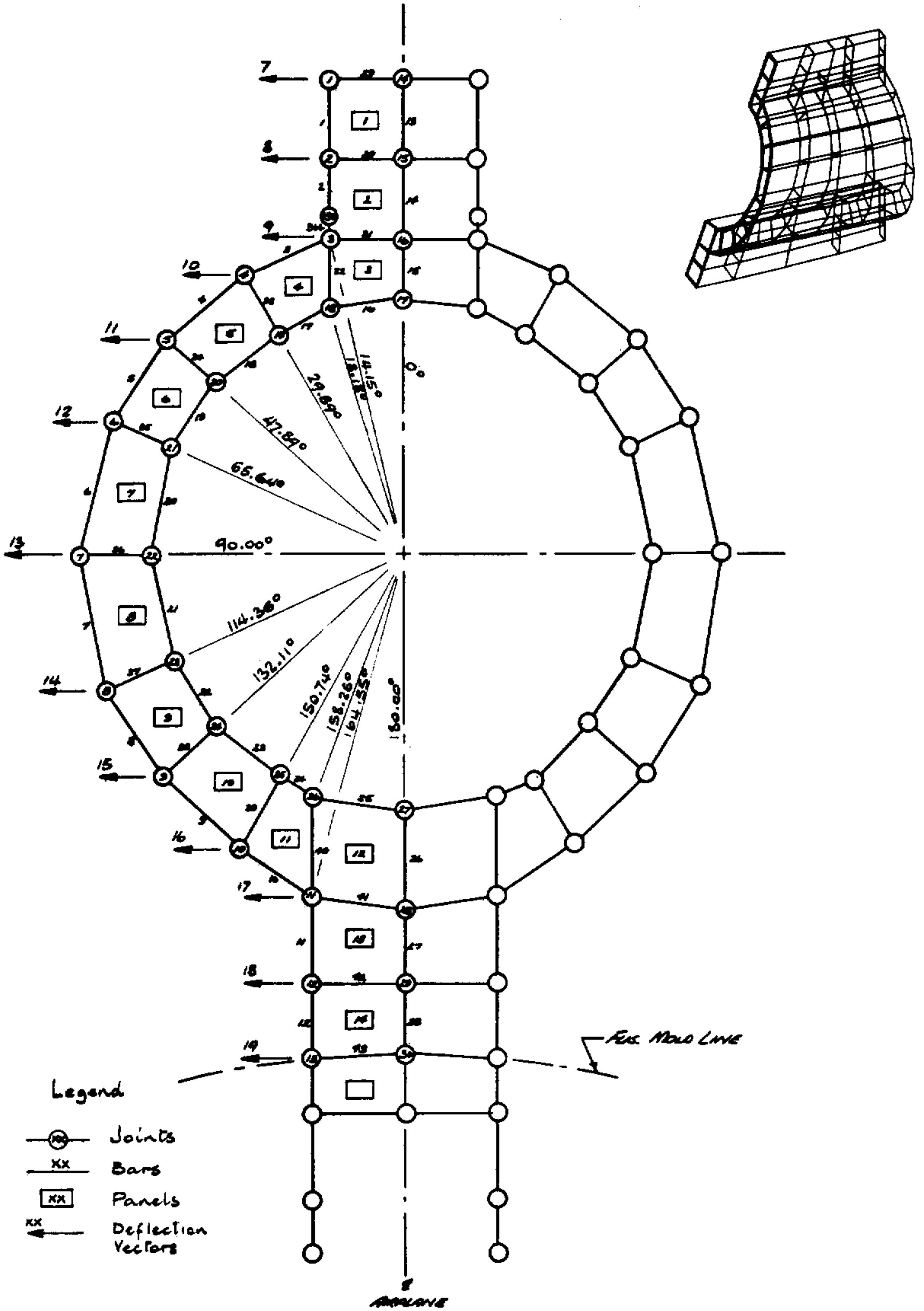


Figure 5. Spar Idealization (Example 2)

TABLE V. BFMM PRINTED DATA FOR EXAMPLE 1

```
                         JOINT DATA, MASTER CASE SJW021

 DATA      JOINT                         COORDINATES
 CODE       NO.          X                    Y                   Z

   1         1       12.117600          2082.713806          258.408298 •
   1         2       12.661000          2077.285706          244.327299 •
   1         3       13.204400          2071.857605          230.246300 •
   1         4       27.029600          2069.802094          224.914200 •
   1         5       40.539900          2065.795105          214.519600 •
   1         6       50.120200          2060.418701          200.572800 •
   1         7       55.250000          2051.669891          177.877600 •
   1         8       51.486100          2042.682602          154.363900 •
   1         9       42.386200          2036.565704          138.695999 •
   1        10       28.822600          2031.839600          126.436100 •
   1        11       15.690900          2029.756897          121.033300 •
   1        12       15.745500          2024.831802          108.257000 •
   1        13       15.800000          2019.906601           95.480700 •
   1        14        0.                2082.713806          258.408298
   1        15        0.                2077.285706          244.327299
   1        16        0.                2071.857605          230.246300
   1        17        0.                2067.944397          220.095209
   1        18       13.204400          2067.167419          218.079620
   1        19       21.158900          2065.863892          214.698099
   1        20       31.734900          2062.727203          206.561199
   1        21       39.234400          2058.518494          195.643499
   1        22       43.250000          2051.669891          177.
   1        23       39.234400          2044.82
   1        24       31.576900
   1        25
   1
```

TABLE V. BFMM PRINTED DATA FOR EXAMPLE 1 (CONTINUED)

```
                              APPLIED LOAD DATA, MASTER CASE SJW021

DATA COND  VECT  TYPE  JT M  JT N                   VECTOR COMPONENTS                       J  J  J  J  J  J  S  S  M
CODE  NO    NO   CODE   NO    NO          X                  Y                  Z           E  P  X  D  R  P  S  A  S
                                                                                            F  R  L  V  O  S  M  M  T
  4    1     1    3     23     8    0.899654E 00  -0.157047E 00  -0.407380E 00              1  0  0  0  0  0  0  0  0
  4    2     2    3     21     6    0.899655E 00   0.157042E 00   0.407381E 00              1  0  0  0  0  0  0  0  0
  4    3     3    3     11    28   -0.991091E 00  -0.479085E-01  -0.124269E 00              1  0  0  0  0  0  0  0  0
  4    4     4    3     10    11   -0.914997E 00  -0.145119E 00  -0.376459E 00              1  0  0  0  0  0  0  0  0
  4    5     5    3      9    10   -0.730711E 00  -0.245557E 00  -0.636995E 00              1  0  0  0  0  0  0  0  0
  4    6     6    3      8     9   -0.451275E 00  -0.320981E 00  -0.832560E 00              1  0  0  0  0  0  0  0  0
  4    7  ▲  7    1      1     0    0.100000E 01   0.             0.                        1  0  1  1  0  0  0  0  0
  4    8  ▲  8    1      2     0    0.100000E 01   0.             0.                        1  0  0  0  0  0  0  0  0
  4    9  ▲  9    1      3     0    0.100000E 01   0.             0.                        1  0  1  0  0  0  0  0  0
  4   10  ▲ 10    1      4     0    0.100000E 01   0.             0.                        0  0  0  0  0  0  0  0  0
  4   11  ▲ 11    1      5     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   12  ▲ 12    1      6     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   13  ▲ 13    1      7     0    0.100000E 01   0.             0.                        0  0  1  1  0  0  0  0  0
  4   14  ▲ 14    1      8     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   15  ▲ 15    1      9     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   16  ▲ 16    1     10     0    0.100000E 01   0.             0.                        0  0  0  0  0  0  0  0  0
  4   17  ▲ 17    1     11     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   18  ▲ 18    1     12     0    0.100000E 01   0.             0.                        0  0  0  0  0  0  0  0  0
  4   19  ▲ 19    1     13     0    0.100000E 01   0.             0.                        0  0  1  0  0  0  0  0  0
  4   20    20    1     31     0    0.100000E 01   0.             0.                        1  0  1  1  0  0
  4   21    21    1     32     0    0.100000E 01   0.             0.                        0
  4   22    22    1     33     0    0.100000E 01   0.             0.
  4   23    23    1     34     0    0.100000E 01   0.
  4   24    24    1     35     0    0.100000E 0
  4   25    25    1     36     0
  4   26    26    1
  4   27
```

TABLE V. BFMM PRINTED DATA FOR EXAMPLE 1 (CONTINUED)

ROW FORMAT OF RESULTANT BAR LOADS, MASTER CASE SJW021

ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION
1	20000101	2	20000102	3	20000201	4	20000202	5	20000203
6	20000205	○ 7	20000301	○ 8	20000302	○ 9	20000401	○ 10	20000402
○ 11	20000501	○ 12	20000502	○ 13	20000601	○ 14	20000602	○ 15	20000701
○ 16	20000702	○ 17	20000801	○ 18	20000802	○ 19	20000901	○ 20	20000902
○ 21	20001001	○ 22	20001002	23	20001101	24	20001102	25	20001201
26	20001202	27	20001301	28	20001302	29	20001401	30	20001402
31	20001501	32	20001502	□ 33	20001601	□ 34	20001602	□ 35	20001701
□ 36	20001702	□ 37	20001801	□ 38	20001802	□ 39	20001901	□ 40	20001902
□ 41	20002001	□ 42	20002002	□ 43	20002101	□ 44	20002102	□ 45	20002201
□ 46	20002202	□ 47	20002301	□ 48	20002302	□ 49	20002401	□ 50	20002402
□ 51	20002501	□ 52	20002502	53	20002601	54	20002602	55	20002701
56	20002702	57	20002801	58	20002802	59	20002901	60	20002902
61	20003001	62	20003002	○ 63	20003101	○ 64	20003102	65	20003201
66	20003202	67	20003301	68	20003302	69	20003401	70	20003402
71	20003501	72	20003502	73	20003601	74	20003602	75	20003701
76	20003702	77	20003801	78	20003802	79	20003901	80	20003902
81	20004001	82	20004002	○ 83	20004101	○ 84	20004102	85	20004201
86	20004202	87	20004301	88	20004302	89	20004401	90	20004402
91	20004501	92	20004502	93	20004503	94	20004505	95	20004601
96	20004602	97	20004701	98	20004702	99	20004801	100	20004802
101	20004901	102	20004902	103	20005001	104	20005002	105	20005101
106	20005102	107	20005201	108	20005202	109	20005301	110	[illegible]
111	20005401	112	20005402	113	20005501	114	20005502	[illegible]	
116	20005602	117	20005701	118	20005702	119			
121	20005901	122	20005902	123	[illegible]				
126	20006102	127	20006201	128					
131	20006401	132							
136	20006602								
141									

Resultant bar force number
Bar number
Bar data code number

TABLE V. BFMM PRINTED DATA FOR EXAMPLE 1 (CONTINUED)

ROW FORMAT OF RESULTANT SHEAR PANEL AND MEMBRANE FORCES, MASTER CASE SJW021

ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION	ROW	DEFINITION
1	30000101	2	30000102	3	30000103	4	30000104	5	30000201
6	30000202	7	30000203	8	30000204	9	30000301	△ 10	30000302
11	30000303	12	30000304	13	30000401	△ 14	30000402	15	30000403
16	30000404	17	30000501	△ 18	30000502	19	30000503	20	30000504
21	30000601	△ 22	30000602	23	30000603	24	30000604	25	30000701
△ 26	30000702	27	30000703	28	30000704	29	30000801	△ 30	30000802
31	30000803	32	30000804	33	30000901	△ 34	30000902	35	30000903
36	30000904	37	30001001	△ 38	30001002	39	30001003	40	30001004
41	30001101	△ 42	30001102	43	30001103	44	30001104	45	30001201
△ 46	30001202	47	30001203	48	30001204	49	30001301	50	30001302
51	30001303	52	30001304	53	30001401	54	30001402	55	30001403
56	30001404	57	30001501	58	30001502	59	30001503	60	30001504
61	30001601	62	30001602	63	30001603	64	30001604	65	30001701
66	30001702	67	30001703	68	30001704	69	30001801	70	30001802
71	30001603	72	30001804	73	30001901	74	30001902	75	30001903
76	30001904	77	30002001	78	30002002	79	30002003	80	30002004
81	30002101	82	30002102	83	30002103	84	30002104	85	30002201
86	30002202	87	30002203	88	30002204	89	30002301	90	30002302
91	30002303	92	30002304	93	30002401	94	30002402	95	30002403
96	30002404	97	30002501	98	30002502	99	30002503	100	30002504
101	30002601	102	30002602	103	30002603	104	30002604	105	30002701
106	30002702	107	30002703	108	30002704	109	30002801	110	[illegible]
111	30002603	112	30002804	113	30002901	114	30002902	[illegible]	
116	30002904	117	30003001	118	30003002	119	[illegible]		
121	30003101	122	30003102	123	30003103				
126	30003202	127	30003203	128					
131	30003303	132							
136	30003404								
141									

Resultant shear panel or membrane force number
Element number
Element data code number

values and row locations of the matrix elements involved in the plot. All of the data relevant to the subject spar is included in these partial lists.

b. Input Data

Complete input for Phase III of FORMAT to effect the matrix plot is shown in Table VI.

The input matrix data sets are initially identified as "VSS021, 121891" (containing matrices BARL, BARR, PANQL and PANQR) and "JOIN02, 120391" (containing matrix DTJD21).

The first card of the special data (card type 1) implies that four frames are output.

The second card (card type 2) defines the total number of abscissas as 37 and the number of different abscissa sets as four, i.e. for axial loads on the ring outer cap, for axial loads on the ring inner cap, for shear forces along the inner edge of the ring, and for lateral deflections of the spar.

The third card (card type 3) implies that a copy of the matrix columns used in the plot will be made on a scratch data set and the number of matrices involved as five. The following card set (card type 4) identifies each matrix by name, the number of columns to be copied and the individual column numbers.

The next card set (card type 5) specifies the 37 abscissa values used and the row numbers corresponding to the matrix elements to be plotted at these abscissa located in the appropriate set field. In this case the sequence of abscissa entries is angular distance from aircraft centerline to "ring" joints followed by vertical distance above fuselage reference plane of the lateral deflection vectors. The following table illustrates the development of the first part of this sequence and shows the connectivity required in the plot (see also Figure 5).

Joint No.	0°	Abscissa Set			
		Set 1	Set 2	Set 3	Set 4
16, 17	0	63	33	10	
3	14.15	64			
3	14.15	7			
18	18.18		34	10	
18	18.18		35	14	
4, 19	29.89	8	36	14	
4, 19	29.89	9	37	18	
etc.					

Set 1 implies the following type plot

Abscissa	Ordinate
0	Axial load at p end of bar 31 (joint 16)
14.15	Axial load at q end of bar 31 (joint 3)
14.15	Axial load at p end of bar 3 (joint 3)
29.89	Axial load at q end of bar 3 (joint 4)
29.89	Axial load at p end of bar 4 (joint 4)
etc.	

where connectivity of the plotted points is in the above sequence. The manner of identification of the appropriate matrix element row numbers for entry in the various abscissa sets is indicated in Table V.

For each of the four frames plotted the special data defines the number of curves plotted, and extremes and grid increments for both axes (card type 6), labelling data for the two axes (cards type 7 and 8), and the data identifying each curve plotted (card type 9) including the matrix name, column number, abscissa set number, line connectedness option and identifying legend. In this example extremes and grid increments for the axis of ordinates are not known à priori and are therefore omitted.

c. Output

A complete listing of the special data input is unconditionally printed by the matrix plot module and for each column plotted, the extracted values of abscissa and ordinate are tabulated. These listings are not reproduced here.

Plotted output which results from this example is reproduced in Figure 6.

TABLE VI. INPUT DATA FOR EXAMPLE 2

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         10        20        30        40        50        60        70        80
$OUTPUT
      MATRIX INPUT TAPE ( VSS021, 121891 )
      MATRIX INPUT TAPE ( JOIN02, 120391 )
$PLOT          MATRIX
      DATA
    4
   37    4
COPY     5
BARL          3    4   12   20
BARR          3    4   12   20
PANQL         3    4   12   20
PANQR         3    4   12   20
DTJ021        3    4   12   20
     0.0     63   33   10
    14.15    64
    14.15     7
    18.18          34   10
    18.18          35   14
    29.89     8   36   14
    29.89     9   37   18
    47.89    10   38   18
    47.89    11   39   22
    65.64    12   40   22
    65.64    13   41   26
    90.0     14   42   26
    90.0     15   43   30
   114.36    16   44   30
   114.36    17   45   34
   132.11    18   46   34
   132.11    19   47   38
   150.74    20   48   38
   150.74    21   49   42
   158.26         50   42
   158.26         51   46
```

TABLE VI. INPUT DATA FOR EXAMPLE 2 (CONTINUED)

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         10        20        30        40        50        60        70        80
   164.55    22
   164.55    84
   180.0     83   52   46
  258.408                    7
  244.327                    8
  230.246                    9
  224.914                   10
  214.519                   11
  200.572                   12
  177.877                   13
  154.563                   14
  138.695                   15
  126.436                   16
  121.033                   17
  108.257                   18
   95.480                   19
    6   0.0      180.0      10.0
                   AXIAL LOAD   - LB -
      ANGULAR DISTANCE FROM CENTER-LINE OF SPAR NO. 1   - DEGREES -
BARL          4    1    1  OUTER CAP, LEFT SIDE, COND.  4
BARL          4    2    1  INNER CAP, LEFT SIDE, COND.  4
BARL         12    1    1  OUTER CAP, LEFT SIDE, COND. 12
BARL         12    2    1  INNER CAP, LEFT SIDE, COND. 12
BARL         20    1    1  OUTER CAP, LEFT SIDE, COND. 20
BARL         20    2    1  INNER CAP, LEFT SIDE, COND. 20
    6   0.0      180.0      10.0
                   AXIAL LOAD   - LB -
      ANGULAR DISTANCE FROM CENTER-LINE OF SPAR NO. 1   - DEGREES -
BARR          4    1    1  OUTER CAP, RIGHT SIDE, COND.  4
BARR          4    2    1  INNER CAP, RIGHT SIDE, COND.  4
BARR         12    1    1  OUTER CAP, RIGHT SIDE, COND. 12
BARR         12    2    1  INNER CAP, RIGHT SIDE, COND. 12
BARR         20    1    1  OUTER CAP, RIGHT SIDE, COND. 20
BARR         20    2    1  INNER CAP, RIGHT SIDE, COND. 20
```

TABLE VI. INPUT DATA FOR EXAMPLE 2 (CONTINUED)

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         10        20        30        40        50        60        70        80
    6   0.0     180.0      10.0
                   SHEAR FLOW   - LB PER IN -
      ANGULAR DISTANCE FROM CENTER-LINE OF SPAR NO. 1  - DEGREES -
PANQL         4    3    1  INSIDE EDGE, LEFT SIDE , COND.  4
PANQL        12    3    1  INSIDE EDGE, LEFT SIDE , COND. 12
PANQL        20    3    1  INSIDE EDGE, LEFT SIDE , COND. 20
PANQR         4    3    1  INSIDE EDGE, RIGHT SIDE, COND.  4
PANQR        12    3    1  INSIDE EDGE, RIGHT SIDE, COND. 12
PANQR        20    3    1  INSIDE EDGE, RIGHT SIDE, COND. 20
    3   0.0     300.0      10.0
                   LATERAL DEFLECTION   - IN -
      VERTICAL DISTANCE FROM F.R.P.   - IN -
DTJ021        4    4    1  SPAR NO. 1, COND.  4
DTJ021       12    4    1  SPAR NO. 1, COND. 12
DTJ021       20    4    1  SPAR NO. 1, COND. 20
$END
```

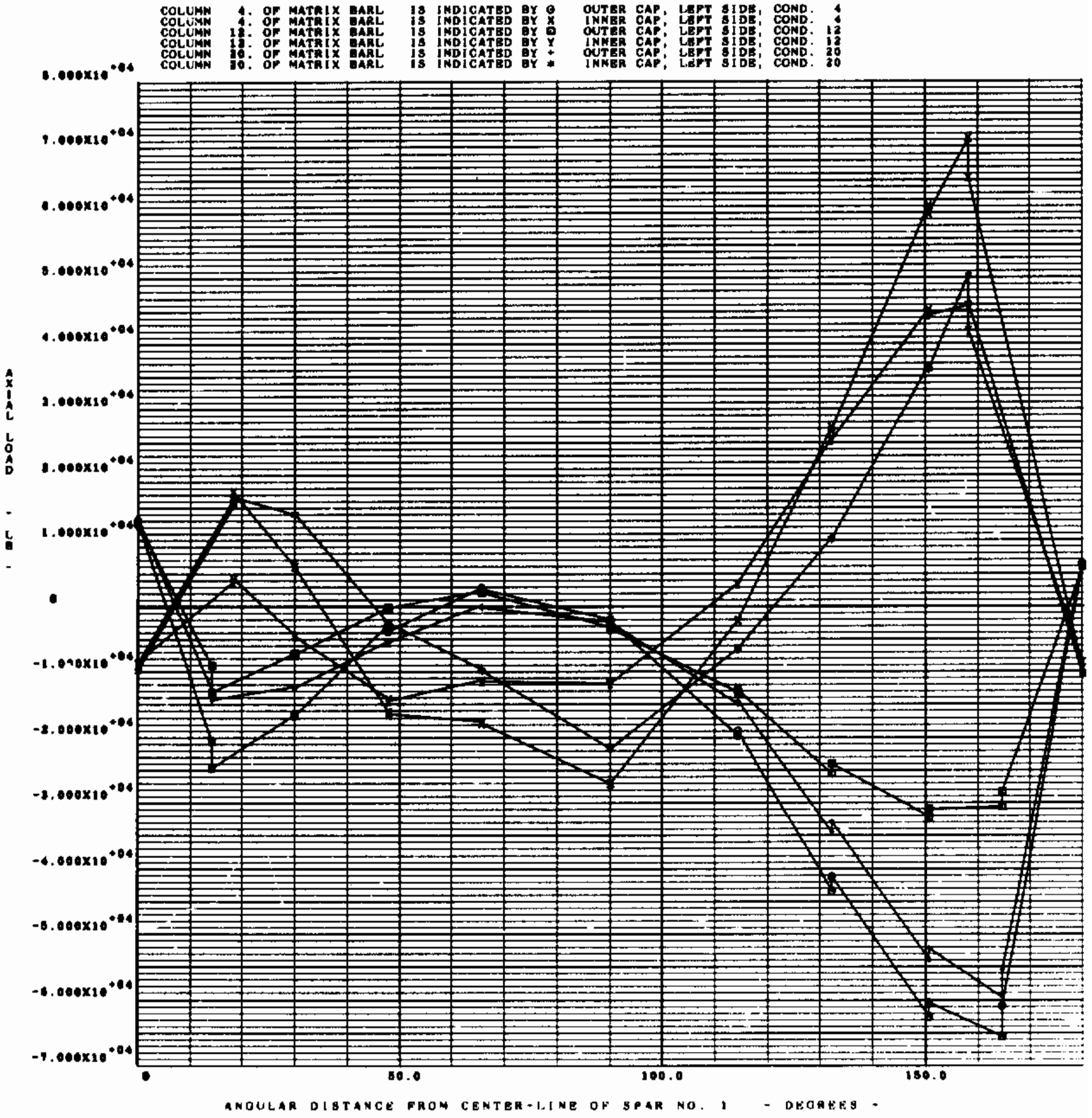


Figure 6. Plotted Output for Example 2

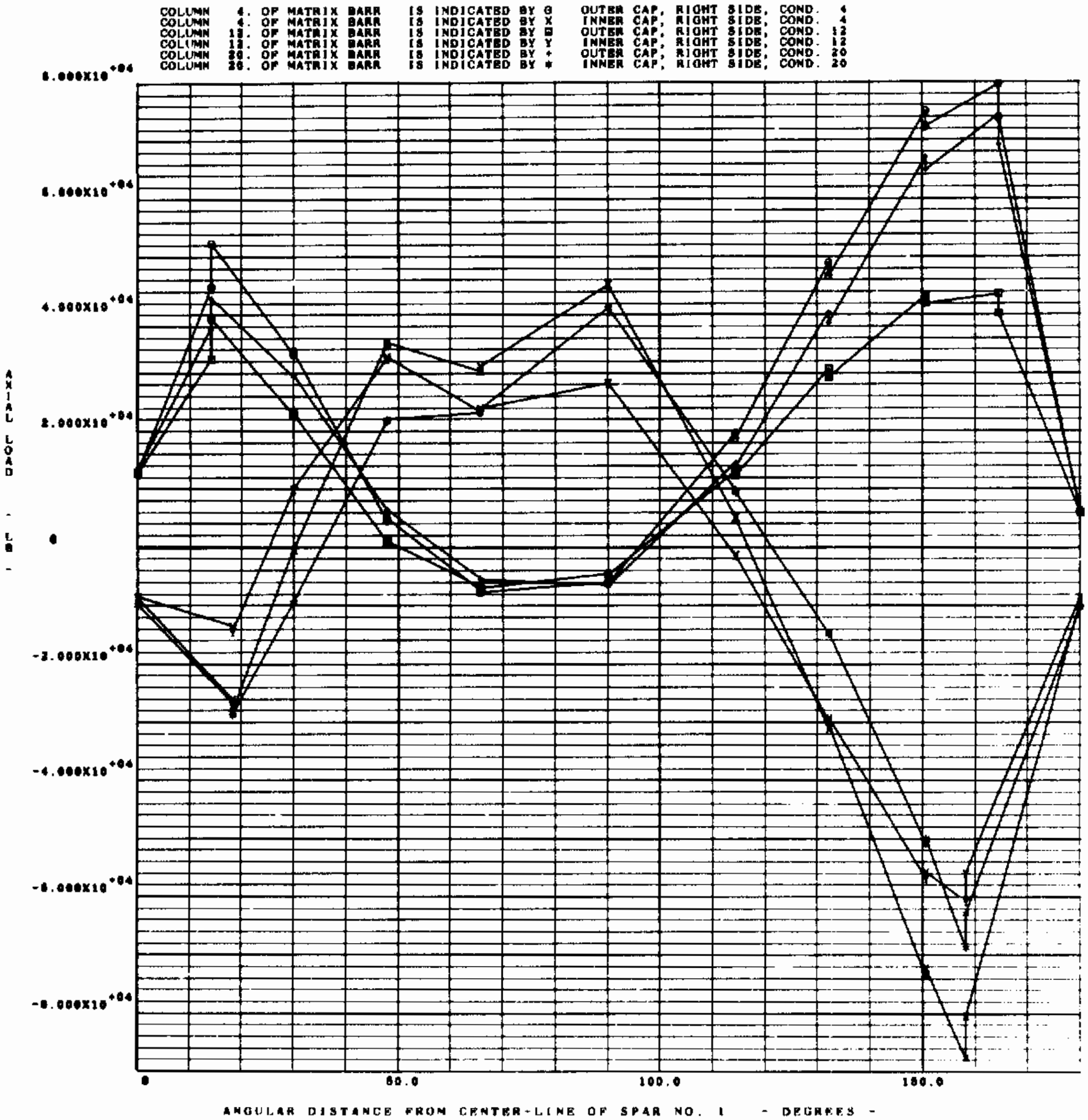


Figure 6. Plotted Output for Example 2 (Continued)

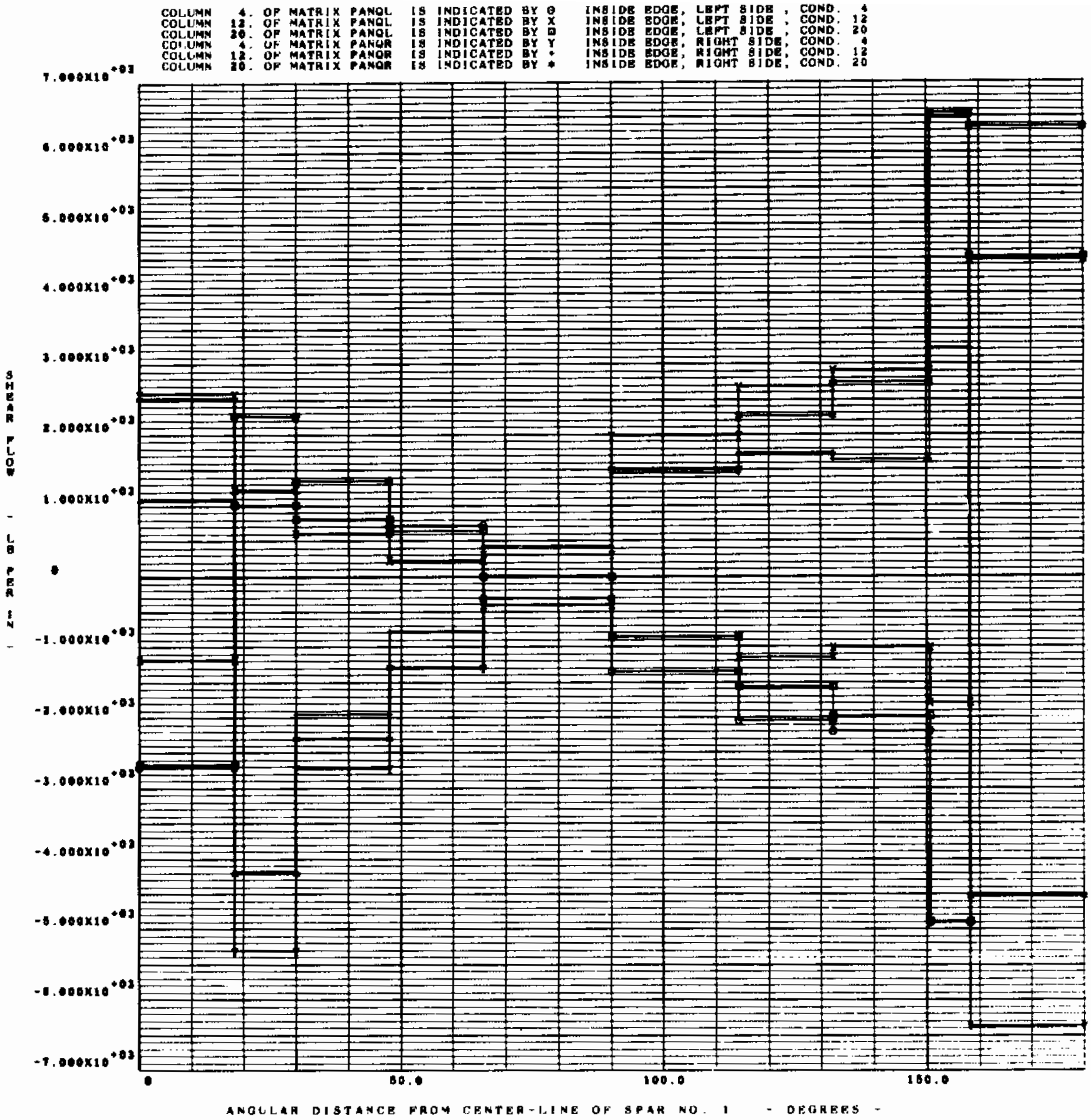


Figure 6. Plotted Output for Example 2 (Continued)

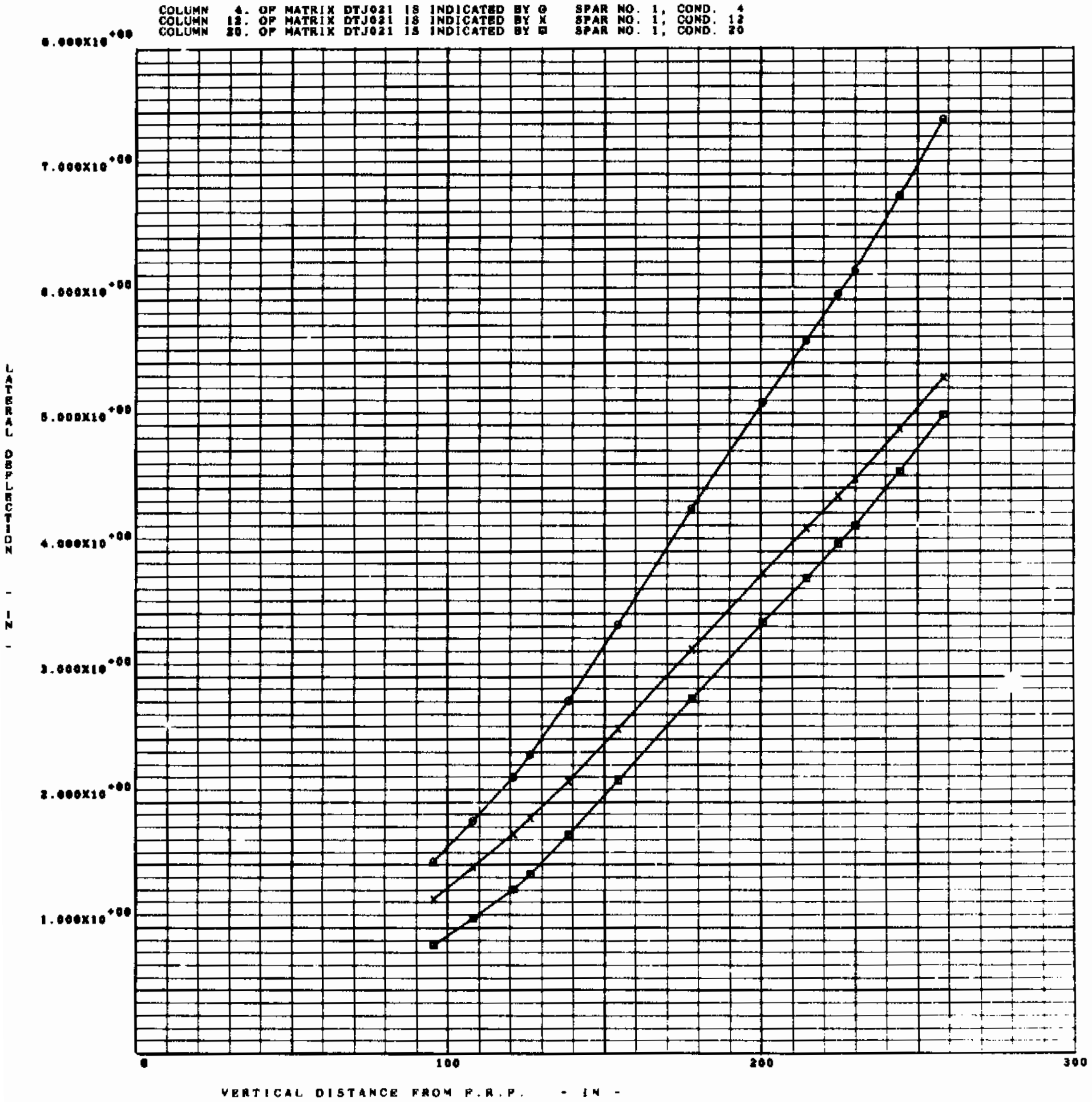


Figure 6. Plotted Output for Example 2 (Continued)

REFERENCES

1. J. P. Cogan, Jr., FORMAT II - Second Version of Fortran Matrix Abstraction Technique; Volume II. Description of Digital Computer Program, AFFDL-TR-66-207, Volume II, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, March 1967.

2. W. J. Lackey, R. E. Wild, FORMAT II - Second Version of Fortran Matrix Abstraction Technique; Volume II, Supplement I. Description of Digital Computer Program, AFFDL-TR-66-207, Volume II, Supplement I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, November 1968.

3. C. G. Hooks, FORMAT II - Second Version of Fortran Matrix Abstraction Technique; Volume II, Supplement II. Description of Digital Computer Program System/360. AFFDL-TR-66-207, Volume II, Supplement II, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio,February 1969.

4. W. J. Lackey, S. H. Miyawaki, FORMAT - Fortran Matrix Abstraction Technique; Volume II, Supplement III. Description of Digital Computer Program - Extended. AFFDL-TR-66-207, Volume II, Supplement III, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1970.

5. J. Pickard, FORMAT - Fortran Matrix Abstraction Technique; Volume V. Engineering User and Technical Report. AFFDL-TR-66-207, Volume V, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, October 1968.

6. J. Pickard, FORMAT -Fortran Matrix Abstraction Technique; Volume V, Supplement I. Engineering User and Technical Report - Extended. AFFDL-TR-66-207, Volume V, Supplement I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1970.

7. J. P. Cogan, Jr., R. C. Morris, and J. R. Wells, FORMAT - Fortran Matrix Abstraction Technique; Volume VI. Description of Digital Computer Program - Phase I. AFFDL-TR-66-207, Volume VI, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, September 1968.

8. R. C. Morris, FORMAT - Fortran Matrix Abstraction Technique; Volume VI, Supplement I. Description of Digital Computer Program - Phase I-Extended. AFFDL-TR-66-207, Volume VI, Supplement I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1970.

9. R. C. Morris, J. R. Wells, and P. S. Yoon, FORMAT - Fortran Matrix Abstraction Technique; Volume VII. Description of Digital Computer Program - Phase III. AFFDL-TR-66-207, Volume VII, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, September 1968.

10. J. A. Frank, FORMAT - Fortran Matrix Abstraction Technique; Volume VII, Supplement I. Description of Digital Computer Program - Phase III - Extended. AFFDL-TR-66-207, Volume VII, Supplement I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1970.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA – R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Douglas Aircraft Company

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
N/A

3. REPORT TITLE
FORMAT - FORTRAN MATRIX ABSTRACTION TECHNIQUE
Volume V-Supplement I. Engineering User and Technical Report-Extended

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Report 7/1/68 - 6/30/70

5. AUTHOR(S) *(First name, middle initial, last name)*
J. Pickard

6. REPORT DATE
August 1970

7a. TOTAL NO. OF PAGES
69

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.
F33615-68-C-1633

b. PROJECT NO.
1467

c. Task No. 146705

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
DAC 33569

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
AFFDL-TR-66-207 Vol. V, Supplement I

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Air Force Flight Dynamics Laboratory
Wright-Patterson Air Force Base, Ohio

13. ABSTRACT

The FORMAT System has been updated by the incorporation of additional basic capability and the refinement of existing capability. A simpler mode of updating case data and extended force method matrix generation capability has been incorporated in Phase I of the system. A refined "Structure Cutter" module, capabilities for matrix partitioning and instruction looping, and an additional eigenvalue/eigenvector extraction module have been incorporated in Phase II. Finally the limitations which existed in the matrix plotting capability in Phase III have been eliminated. Engineering user and technical information is presented in this report. Included are recommendations of improvements in implementation and utilization procedures for various computer systems.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
1. Structural Analysis 2. Matrix methods 3. Matrix abstraction 4. Digital computer methods						