TRACK TRAIN DYNAMICS

MATHEMATICAL MODEL

USER's MANUAL: SPAR 10 STRUCTURAL ANALYSIS, FINITE ELEMENT PROGRAM, FOR USE ON A DEC-2050 COMPUTER SYSTEM

REPORT NO. R-468

TRACK TRAIN DYNAMICS

AAR

NATIONAL GOVERNMENT-INDUSTRY RESEARCH PROGRAM ON TRACK-TRAIN DYNAMICS

06 – Signals, Control and Communications

Association of American Railroads

Research and Test Department

USER's MANUAL: SPAR 10 STRUCTURAL ANALYSIS, FINITE ELEMENT PROGRAM, FOR USE ON A DEC-2050 COMPUTER SYSTEM

REPORT NO. R-468

B. Prasad

Som P. Singh

May, 1981

AAR Technical Center

Chicago, Illinois

1. REPORT NO.	**2. REPORT DATE**	**3. PERIOD COVERED**
R-468	May, 1981	

4. TITLE AND SUBTITLE
User's Manual: SPAR 10 Structural Analysis, Finite Element Program, For Use on a DEC-2050 Computer System

5. AUTHOR(S)

B. Prasad and Som P. Singh

6. PERFORMING ORGANIZATION NAME AND ADDRESS
Association of American Railroads
Track Train Dynamics - Phase III
3140 S. Federal Street
Chicago, Ill 60616

7. TYPE OF REPORT
Analytical

8. CONTRACT OR GRANT NO.

DOT-FR 64228

9. SPONSORING AGENCY NAME AND ADDRESS
Department of Transportation
Federal Railroad Administration
400 Seventh Street, S. W.
Washington, D. C. 20590

10. NO. OF PAGES
127

11. NO. OF REFERENCES
7

12. SUPPLEMENTARY NOTES

13. ABSTRACT

The SPAR program, originally developed by W. D. Whetstone, is an analytical system for calculating static deflections and stresses, natural vibration modes and frequencies, and buckling loads and mode shapes associated with linear, finite element structural simulations.

This Report extends the available documentation for SPAR and concerns its implementation on the Association of American Railroads DEC-2050 Computer System. Most of the processors, although of the same name as in the original version, have been modified to be operative on a DEC-2050 System.

The United States Government and the Association of American Railroads assume no liability concerning the SPAR program and its usage

14. SUBJECT TERMS
Finite Element Stress Analysis
Structural Analysis
Structural Simulations

15. AVAILABILITY STATEMENT
J. G. Britton
Executive Director-Technical Center
Association of American Railroads
3140 S. Federal Street
Chicago, Illinois 60616

Steering Committee for the Track Train Dynamics Program

Chairman
J. G. German
Vice President
Engineering
Missouri Pacific Co.

Vice Chairman
W. J. Harris, Jr.
Vice President
Research and Test Department
Association of American Railroads

W. S. Autrey
Chief Engineer
Atchison, Topeka & Santa Fe Railway Co.

W. Bertaux
General Manager
Locomotive Operations
General Electric Co.

A. D. DeMoss
Senior Vice President
Southern Pacific Transportation Co.

Peter Detmold
Chairman
Canadian Railway Advisory Committee

H. F. Dick
Executive Vice President
General Manager
Advanced Products Group
Pullman Standard

M. Ephraim
Chief Engineer
Electro-Motive Division
General Motors Corporation

W. S. Hansen
President
A. Stucki Co.

S. B. Hobbs
Deputy Director
Systems Development
Department of Transportation
Transportation Systems Center

E. F. Lind
Plant Manager
Los Angeles Locomotive Running Maintenance Plant
Southern Pacific Transportation Co

E. R. Mathews
Director
Transportation Test Center
Federal Railroad Administration

R. A. Matthews
Vice President
Railway Progress Institute

W. McLaren
Chief Current Technology Division
Transport Canada-
Research & Development Centre

R. W. Radford
Chief Mechanical & Electrical Engineer
CN Rail

R. G. Rayburn
Vice President
Transportation
Chessie System

G. E. Reed
Assistant General Manager
AMCAR Division
ACF Industries

J. F. Roseman
Assistant Chief Mechanical Officer
National Railway Passenger Corp

D. V. Sartore
Chief Engineer Design
Burlington Northern, Inc

H. L. Scott, Jr.
Vice President - Operations
Norfolk and Western Railway Co.

P. S. Settle
Vice-President Commercial
Transportation Product Group
Portec, Inc

W. W. Simpson
Vice President
Engineering
Southern Railway Company

W. K. Smith
Vice President and Director of Transportation
General Mills

D. L. Spanton
Director
Office of Rail Safety Research
Federal Railroad Administration

D. R. Sutliff
Chief Engineer
Engineering Analysis
ACF Industries AMCAR Division

G. A. Thelen
Director
Mechanical Engineering
Conrail

H. C. Woodman
Vice President - Operations
Railcar Division
Portec, Inc

PAST CHAIRMEN

R. D. Spence (Chairman)
Executive Vice President
Family Lines System

L. S. Crane (Chairman)
Chairman and Chief Executive Officer
Conrail

J. L. Cann (Chairman)
Vice President
Operations
CN Rail

BACKGROUND INFORMATION

ON THE

TRACK-TRAIN DYNAMICS PROGRAM

The Track-Train Dynamics Program encompasses studies of the dynamic interaction of a train consist with track as affected by operating practices, terrain, and climatic conditions.

Trains cannot move without these dynamic interactions. Such interactions, however, frequently manifest themselves in ways climaxing in undesirable and costly results. While often differing and sometimes necessarily so, previous efforts to reasonably control these dynamic interactions have been reflected in the operating practices of each railroad and in the design and maintenance specifications for track and equipment.

Although the matter of track-train dynamics is by no means a new phenomenon, the increase in train lengths, car sizes, and loadings has emphasized the need to reduce wherever possible excessive dynamic train action. This, in turn, requires a greater effort to achieve more control over the stability of the train as speeds have increased and railroad operations become more systematized.

The Track-Train Dynamics Program is representative of many new programs in which the railroad industry is pooling its resources for joint study and action.

A major planning effort on track-train dynamics was initiated in July 1971 by the Southern Pacific Transportation Company under contract to the AAR and carried out with AAR staff support. Completed in early 1972, this plan clearly indicated that no individual railroad had both the resources and the incentive to undertake the entire program. Therefore, AAR was authorized by its Board to proceed with the Track-Train Dynamics Program.

In the same general period, the FRA signaled its interest in vehicle dynamics by development of plans for a major test facility. The design of a track loop for train dynamic testing and the support of related research programs were also pursued by FRA.

In organizing the effort, it was recognized that a substantial body of information and competence on this program resided in the railroad supply industry and that significant technical and financial resources were available in government.

Through the Railroad Progress Institute, the supply industry coordinated its support for this program and has made available men, equipment, data from earlier proprietary studies, and monetary contributions.

Through the FRA, contractor personnel and direct financial resources have been made available.

Through the Transport Canada Research and Development Centre (TDC), the Canadian Government has made a major commitment to work on this problem and to coordinate that work with the United States' effort.

Through the Office de Recherces et D'Essais, the research arm of the Union Internationale des Chemins de Fer, the basis for a full exchange of information with European groups active in this field has been arranged.

The Track-Train Dynamics Program is managed by the Research and Test Department of the Association of American Railroads under the direction of an industry-government steering committee. Railroad members are designated by elected members of the AAR's Operation-Transportation General Committee, supply industry members by the Railway Progress Institute, U. S. Government members by the Federal Railroad Administration, and Canadian Government members by the Transport Development Centre. Appropriate task forces and advisory groups are established by the Steering Committee on an ad hoc basis as necessary to pursue and resolve elements of the program.

The staff of the program comprises AAR employees, personnel contributed on a full- or part-time basis by railroads or members of the supply industry, and personnel under contract to the Federal Railroad Administration or the Transportation Development Agency.

The program plan as presented in 1972 comprises:

1) Phase I -- 1972-1974

   Analysis of an interim action regarding the present dynamic aspects of track, equipment, and operations to reduce excessive train action.

2) Phase II -- 1974-1977

   Development of improved track and equipment specifications and operating practices to increase dynamic stability.

3) Phase III -- 1977-1982

   Application of more advanced scientific principles to railroad track, equipment, and operations to improve dynamic stability.

Phase I officially ended in December of 1974. The major technical elements of Phase I included:

a) The establishment of the dynamic characteristics of track and equipment.

b) The development and validation of mathematical models to permit the rapid analysis of the effects on dynamic stability of modifications in design, maintenance, and use of equipment and track structures.

c) The development of interim guidelines for train handling, makeup, track structures, and engineer training to reduce excessive train action.

The major technical elements of Phase II include:

a) The adaptation of Phase I analytical models to allow for conducting parameter investigations in the area of track, trucks, draft gear and cushion units, and vehicle behavior.

b) The development of fatigue analysis guidelines.

c) The development of a comprehensive program for identifying the loads to which track, vehicles, and vehicle components are subjected.

Reports on all elements of Phase I and Phase II activities have been essentially completed, and are available through the AAR. A list of the Track Train Dynamics publications is available upon request.

The Phase III program, now actively underway, includes:

a) The development of performance guidelines for the design, development, and fabrication of a dynamically stable bulk commodity car.

b) The application of high speed data processing systems as on-board aid to train operations.

c) The demonstration of advanced coupling, suspension, braking, and draft systems to enable thorough evaluation of potential benefits.

d) The development methods for measuring and monitoring the physical strength of track structures.

EXECUTIVE SUMMARY

SPAR is a system of computer programs, used primarily to perform stress, buckling and vibrational analyses of generalized linear finite element systems. The various computer programs comprising SPAR are called processors and are run in a logical sequence in order to achieve the intended analytical goal. The processors are run independently of each other by communicating automatically and directly with a body of information known as the Data Complex. The sections of the Data Complex are generated by individual processors and utilized by those processors that are next in sequence.

The principal advantages of the SPAR program: effective interactive operation, efficient data management, and efficient use of execution time, central memory storage and secondary data storage.

The SPAR program was originally developed by W. D. Whetstone, under a contract from the National Aeronautics and Space Administration (NASA). The Association of American Railroads (AAR) obtained the program from the Illinois Institute of Technology, Chicago, Illinois and it has been implemented on the AAR's DEC-2050 computer system. In this document, the SPAR system, along with it's member elements (Processors) is described in detail for the benefit of users that are not familiar with the program. The procedures for creating the Data Complex, manipulating the data libraries, conducting static, dynamic and buckling analyses, and the interactive and batch mode execution of the program are explained. The outputs of the processor(s) and the error messages that can be obtained during execution are also interpreted and explained.

ACKNOWLEDGEMENTS

This report resulted from one of the Advanced Analytical Technique Projects from Phase III of the Government-Industry Track Train Dynamics Research Program. This Project supports the adaptation, implementation, demonstration and use of computerized design evaluation techniques by the railroad industry.

The implementation and evaluation of the SPAR structural analysis finite element program on the Association of American Railroad's DEC-2050 computer system was made possible by technical support from both N.A.S.A. and the Illinois Institute of Technology, Chicago, Illinois. The work was done by Dr. B. Prasad, and made possible by financial support from the Federal Railroad Administration of the U.S. Department of Transportation.

Special thanks are due to Dr. Raphael T. Haftka of the Illinois Institute of Technology, who made the original UNIVAC code available to the A.A.R., and provided suggestions for its implementation on the DEC-2050 system.

Dr. Storaasli of N.A.S.A. contributed substantially to the successful SPAR conversion and implementation on the DEC-2050 system through his willingness to answer questions and to provide technical advice.

Thanks are also due to Dr. V.K. Garg for suggesting this project and providing the necessary technical support.

Finally, we have made generous use of the program documentation shown in References 1 through 5 in the preparation of this report and in the overall implementation of this program.

Thanks are due to Dr. Robert F. Breese, for the technical editing and his many valuable suggestions to make the contents of this document more comprehensible, and to Ms. Marva Stevens and Mrs. Patricia A. Collier, who did a fine job in preparing such a complicated manuscript.

TABLE OF CONTENTS

PAGE

1.0 INTRODUCTION 1

1.1 DEC 2050 Computer System 3

2.0 SPAR STRUCTURAL ANALYSIS PROGRAM 3

2.1 Technical Capabilities 3

2.2 Organization of SPAR Processors 4

2.3 Processor Efficiency and Memory Requirements . 7

2.4 Data Base Complex 8

2.4.1 SPAR Data Sets 10

2.4.2 Table of Contents 11

2.5 SPAR Element Library 13

2.6 General Input Rules 16

2.7 Interactive and Batch Operations 17

2.8 Typical Execution Sequence 19

2.8.1 Static Analysis 20

2.8.2 Dynamic Analysis 20

2.8.3 Buckling Analysis 20

2.9 SPAR Output Data Sets 21

2.10 Error Messages 25

3.0 STRUCTURAL DEFINITION PROCESSORS 27

3.1 TAB: Basic Table Inputs 27

3.1.0 TAB Subprocessors 27

3.1.1 Title Printout-Subprocessor TITLE . . . 31

PAGE

3.1.2 Text Printout - Subprocessor TEXT. 31

3.1.3 Material Constants - Subprocessor MATC . . 31

3.1.4 Non-Structural Weights - Subprocessor NSW 32

3.1.5 Alternate Reference Frame-Subprocessor ALTREF 32

3.1.6 Joint Locations - Subprocessor JLOC . . . 33

3.1.6.1 One Dimensional Generation . . 33

3.1.6.2 Two Dimensional Generation . . 35

3.1.7 Joint Reference Frames - Subprocessor JREF 35

3.1.8 Beam Orientation - Subprocessor MREF . . . 36

3.1.9 Beam Rigid Links - Subprocessor BRL . . . 37

3.1.10 Section Properties of Type E21 General Beam Elements - Subprocessor BA 37

3.1.11 Section Properties of Type E22 and E25 Beam Elements - Subprocessor BB 40

3.1.12 Section Properties of Type E23 Bar Elements with Axial Stiffness - Subprocessor BC 41

3.1.13 Section Properties of Type E24 Planar Beam Elements - Subprocessor BD 42

3.1.14 Section Properties of Type E31, E32, E33, E41, E42 and E43 Shell Elements - Subprocessor SA 42

3.1.15 Section Properties of Type E44 Shear Panel Elements - Subprocessor SB 44

3.1.16 Constraint Case n - Subprocessor CON . . . 45

3.1.17 Joint Elimination Sequence - Subprocessor JSEQ 46

3.1.18 Rigid Mass Definition - Subprocessor RMASS 46

PAGE

3.2 ELD: Element Definition Processor 47

3.2.1 Input Data Cards 48

3.2.2 Mesh Generation Commands 50

3.2.2.1 Two Node Element Mesh Generation . . 50

(a) Option 1 (NETOPT=1): Rectilinear Network 50

(b) Option 2 (NETOPT=2): Closed Polynomial Network 50

(c) Option 3 (NETOPT=3): Open Intersecting Network 53

3.2.2.2 Three Node Element Mesh Generation . 53

(a) Option 1 (NETOPT=1): Triangular Network 53

(b) Option 2 (NETOPT=2): Open Triad Polynomial Network 53

(c) Option 3 (NETOPT=3): Closed Triad Polynomial Network 57

3.2.2.3 Four Node Element Mesh Generation . . 57

(a) Option 1 (NETOPT=1): Rectangular Network 57

(b) Option 2 (NETOPT=2): Cylindrical Network 57

3.3 E-E State Definition 61

3.4 EKS: Element Intrinsic Stiffness and Stress Matrix Generator 61

4.0 SYSTEM MATRIX PROCESSORS 61

4.1 TOPO: Element Topology Analyzer 63

4.2 K: System Stiffness Matrix Assembler 63

4.3 INV: SPAR Matrix Decomposition Processor . . . 63

5.0 UTILITY PROCESSORS 64

5.1 AUS: Arithmetic Utility System 64

PAGE

5.1.1 Subprocessor ALPHA 67

5.1.2 Subprocessor TABLE 67

(a) Example No. 1 67

(b) Example NO. 2 68

5.1.3 Subprocessor ELDATA 69

5.1.4 Subprocessor SYSVEC 70

5.2 DCU: Data Complex Utility 71

5.2.1 DCU Commands 71

5.2.2 TOC Information Summary 72

5.3 VPRT: Vector Printer 73

6.0 STATIC SOLUTIONS 75

6.1 Point Forces and Moments Acting on Joints . . 75

6.2 Specified Joint Motions 77

6.3 SSOL: Displacement Data Generator 78

6.4 GSF: Stress Data Generator 78

6.4.1 RESET Controls for GSF 79

6.5 PSF: Stress Table Printer 80

6.5.1 RESET Controls for PSF 81

6.5.2 PSF Output for a Type E21 General Beam Element and DISPLAY=1 (Default Value) 82

6.5.3 PSF Output for a Type E21 General Beam Element and DISPLAY=2 82

6.5.4 PSF Output for a Type E21 General Beam Element and DISPLAY=3 82

6.5.5 PSF Output for a Type E23 Bar Element with Axial Stiffness 82

6.5.6 PSF Output for a Type E24 Planar Beam Element 82

PAGE

6.5.7 PSF Output for a Type E41 Shell Element and DISPLAY=1 (Default Value) 82

6.5.8 PSF Output for a Type E43 Shell Element and DISPLAY=1 (Default Value) 82

6.5.9 PSF Output for a Type E44 Shear Panel Element and DISPLAY=1 (Default Value) 82

7.0 REFERENCES 91

8.0 APPENDICES 92

8.1 Appendix A
Summary List of SPAR Processor Commands . . . 93

A.1 TAB 94

A.2 TAB Sub-processors 94

A.3 CON n: (constraint case n). 97

A.4 REPEAT n^1, inc^1, $jmod^1$. 97

A.5 RMASS 97

A.6 ELD 97

A.7 AUS/TABLE 100

A.8 AUS/SYSVEC. 100

A.9 AUS/ELDATA. 100

A.10 AUS/ALPHA 101

8.2 Appendix B
Shell Section Properties 102

B.1 Formats 103

B.2 ISOTROPIC Sections. 106

B.3 LAMINATE Sections 107

B.4 Structural Weight Per Unit Area 108

B.5 INVM and INVB Commands (Affecting only MEMBRANE,PLATE, UNCOUPLED and COUPLED). . . 108

B.6 Error Codes 109

B.7 Output Data Set Contents 109

LIST OF TABLES

TABLE		PAGE
1.	SPAR Processors	5
2.	Sample SPAR Data Library Table of Contents.	12
3.	SPAR Element Library.	15
4.	Relationships between Processors and Associated Data Set Names	22
5.	Summary of Data Sets Involved in Static Solutions . .	24
6.	TAB Subprocessors for the Definition of Geometric and Material Properties	29
7.	Types of Beam Sections Associated with Subprocessor BA	38
8.	Data Inputs for Two, Three and Four Node Elements . .	49
9.	AUS Subprocessors Used for General Arithmetical Operations.	66

LIST OF FIGURES

FIGURE PAGE

1. Organization of the SPAR Analytical Program on a DEC 2050 Computer System 9

2. SPAR Global Reference Frame or Axes (Default). . . . 28

3. Sample Axis Rotation 34

4. Dimensional Input Parameters for Structural Elements Associated with Subprocessor BA 39

5. Diagram for Subprocessor BD 43

6. Element Reference Frame Assignments 51

7. Example of a Two Node Element Mesh Generation, Using Option 1 to Generate a Rectilinear Network . . 52

8. Example of a Two Node Element Mesh Generation, Using Option 2 to Generate a Closed Polynomial Network 54

9. Example of a Two Node Element Mesh Generation, Using Option 3 to Generate an Open Intersecting Network 55

10. Example of a Three Node Element Mesh Generation, Using Option 1 to Generate a Triangular Network . . 56

11. Example of a Three Node Element Mesh Generation, Using Option 2 to Generate an Open Triad Polynomial Network 58

12. Example of a Three Node Element Mesh Generation, Using Option 3 to Generate a Closed Triad Polynomial Network. 59

13. Example of a Four Node Element Mesh Generation, Using Option 1 to Generate a Rectangular Network . . 60

14. Example of a Four Node Element Mesh Generation, Using Option 2 to Generate a Cylindrical Network . . 60

FIGURE		PAGE

15. Data Flow Diagram for the SPAR System Matrix Processors 62

16. Data Flow Diagram for the Calculation of Static Solutions 76

17. Sample PSF Output for a Type E21 General Beam Element and DISPLAY=1 (Default Value) 83

18. Sample PSF Output for a Type E21 General Beam Element and DISPLAY=2 84

19. Sample PSF Output for a Type E21 General Beam Element and DISPLAY=3 85

20. Sample PSF Output for a Type E23 Bar Element with Axial Stiffness 86

21. Sample PSF Output for a Type E24 Planar Beam Element 87

22. Sample PSF Output for a Type E41 Shell Element and DISPLAY=1 (Default Value) 88

23. Sample PSF Output for a Type E43 Shell Element and DISPLAY=1 (Default Value) 89

24. Sample PSF Output for a Type E44 Shear Panel Element and DISPLAY=1 (Default Value) 90

1.0 INTRODUCTION

The SPAR finite-element structural analysis program [1]* is a collection of processors that efficiently perform particular steps in a structural analysis procedure through the use of a common data base complex. The data base complex is a standardized organization of data, formatted and controlled by a self-contained set of data handling utility processors, which are also a part of the SPAR system. These data handling utilities were developed by W. D. Whetstone [1] and provide the essential capabilities which contribute to the effectiveness of the SPAR analytical system.

The purpose of the present report was to extend the available documentation for the SPAR system and providing the basic information needed to run the program. It should be noted that, although most of the processors retain the same names that were used in the original version, they have been modified sufficiently to become operational on a DEC-2050 system. Although the changes, in terms of the user's input, have been minimized to the extent possible, some differences are now present.

This manual has been organized such that new users should have little or no difficulty in following it. It is suggested that the user practice using SPAR on a sample problem as he follows the different instructions. In this report, Section 1.0 is the Introduction and describes the DEC-2050 system. Section 2.0 deals with the SPAR program, its structure, capabilities,

---

*The numbers in square brackets [ ] refer to the References, listed in Section 7.0 of this Report.

organization and the users' interaction in different modes of operation.

Sections 3.0, 4.0, 5.0 and 6.0 describe the individual SPAR processors and their input data requirements. Thirteen SPAR processors have been grouped into four categories, according to their functional characteristics. TAB, ELD, E and EKS have been grouped under Structure Definition Processors, Section 3.0. These processors are required to form the element stiffness and stress matrices and to generate the element information packets for each element type. Section 4.0 deals with the System Matrix Processors: the processors required to assemble and factorize the system stiffness matrix. Section 5.0 outlines the various Utility Processors which are generally of two types. The first is useful for performing general utility operations, such as entering data created by other programs into the SPAR data complex, copying or printing libraries. The second type is useful for applying various structural loads, such as concentrated loads or moments, dislocations, inertial loads, pressures, etc. Section 6.0 describes the processors which are used to obtain the static solutions, i.e., the stresses and displacements in the structure.

1.1 DEC-2050 Computer System

The SPAR program is now operational on a DEC-2050 computer system, located at the Association of American Railroad's Technical Center in Chicago, Illinois.

This system consists of the following:

- One DEC-2050 Main Frame, with a 512K word (1 word = 36 bits) core memory
- Three DEC-RP06 Disk Drives (200 megabyte capacity)
- Two DEC-RP04 Disk Drives (100 megabyte capacity)
- Three 800/1600 BPI Tape Drive Units
- One Card Reader
- One Versatec Electrostatic Dot Matrix Plotter
- One DN-20 Data Communication Front End
- Two Tektronix 4010-1 Graphic Terminals
- Forty alpha-numeric CRT Terminals (Digital VT52 and Lear-Seigler ADM3A)
- Two LA36 DEC Writer II Terminals

The system is presently operating under the TOPS-20 Operating System, Version 4.

2.0 SPAR STRUCTURAL ANALYSIS PROGRAM

2.1 Technical Capabilities

SPAR is an analytical program capable of calculating static deflections and stresses; natural vibration frequencies and modes; and buckling loads and mode shapes of linear finite-element structural simulations. The structural simulations are composed of finite elements connected at specified joints which can have three translational and three rotational components. Those finite elements which are currently available for simulating the stiffness characteristics of a structure include axial bars, beams of general cross section, triangular and quadrilateral plates, with an option to specify coupled or uncoupled membrane and bending stiffnesses, and quadrilateral shear panels. The properties of the plates may be specified as layers in a laminate of composite materials, and there is also a provision for warping of the quadrilateral plate element. The mass properties of a structure are represented by both structural and nonstructural masses, associated with the stiffness elements, and by concentrated masses at the joints. The loading data can include any or all of the following categories: point forces or moments acting at the joints, specified joint motions, inertial loads, thermal or pressure loads and initial strains in individual elements. The technical capability contained in the SPAR system is presently available for operation on DEC-2050 computer systems, and the version of SPAR denoted as Level 11 is currently operational.

2.2 Organization of SPAR Processors

The SPAR finite-element structural analysis procedure is divided into a sequence of steps or functions which must be performed. The computer code required for each of these steps is referred to herein as a "processor." Processors are separate portions of the SPAR system, which are selectively executed in a logical sequence in order to perform a desired analysis. Each processor is designed to perform a limited, yet distinct and complete, function. The functions of each of the processors in the version of SPAR denoted as Level 11 are given in Table 1. Processors TAB through KG read the user's

Table 1 SPAR Processors

Processor Name	Function
TAB	Creates the data sets containing tables of joint locations, section properties, material constants, etc.
ELD	Defines the finite elements making up the model.
E	Generates the sets of information for each element, including connected joint numbers, geometrical data, and material and section property data.
EKS	Adds the stiffness and stress matrices for each element to the set of information produced by the E processor.
TOPO	Analyzes the element interconnection topology and creates data sets used to assemble and factor the system mass and stiffness matrices.
K	Assembles the unconstrained system stiffness matrix in a sparse format.
M	Assembles the unconstrained system mass matrix in a sparse format.
KG	Assembles the unconstrained system initial-stress (geometric) stiffness matrix in a sparse format.
INV	Factors the assembled system matrices.
EQNF	Computes the equivalent joint loading, associated with thermal, dislocational and pressure loading inputs.
SSOL	Computes the displacements and reactions resulting from the applied loading at the joints.

Table 1 - Continued

Processor Name	Function
GSF	Generates the element stresses and internal loads.
PSF	Prints the information generated by the GSF processor.
EIG	Solves the linear vibration and bifurcation buckling eigenproblems.
DR	Performs a dynamic response analysis.
SYN	Produces the mass and stiffness matrices for systems comprised of interconnected substructures.
STRP	Computes the eigenvalues and eigenvectors of the substructured systems.
AUS	Performs an array of matrix arithmetic functions, and is used in the construction, editing and modification of data sets.
DCU	Performs an array of data management functions, including a display of the table of contents, data transfer between libraries, changing data set names, printing data sets, and transferring data between libraries and sequential files.
VPRT	Performs the editing and printing of data sets which are in the form of vectors in the data libraries.

input, form element matrices, and assemble the element matrices into system matrices, which represent the overall stiffness and mass of the structure. Solutions of the system matrix equations are performed by processors INV through PSF. The EIG and DR processors are used for eigensolutions and dynamic response analyses, respectively. Calculations involving substructuring are performed by SYN and STRP. The remaining processors listed in Table 1 provide three types of functions which are general and not limited to structural analysis: (1) the arithmetic utility system, AUS, provides a general matrix input and arithmetic capability, (2) the data complex utility, DCU, manages and prints data, and (3) the VPRT processor performs the editing and printing of data sets in the form of vectors, such as applied loads and moments.

2.3 Processor Efficiency and Memory Requirements

A general characteristic of the SPAR processors is their efficiency with respect to both computer memory and processing time requirements. The method for handling the large, sparse matrices encountered in finite-element structural analysis, in a manner to achieve this efficiency, is described in Reference 2. All of the processors make extensive use of auxillary disk storage and automatically or dynamically allocate the main computer memory so that large structural simulations can be analyzed. The method used for eigenvalue and eigenvector calculations [3] performs a vibration analysis without first reducing the number of degrees of freedom being considered. This capability allows a single simulation to be used for both static and dynamic calculations.

The SPAR system is designed for effective interactive operation via teletype and/or graphic terminals. All input is in a free-field format, and executive control commands call selected processors for execution. The related input data for a given processor is typed-in sequentially at a user's console keyboard after successive prompts from the operating system. The computational sequence is continued by using another executive control command to call the next desired processor. This flexibility for selective execution of various processors gives the user considerable versatility in performing structural analyses.

2.4 Data Base Complex

A set of data handling utilities (the same for each processor) is used to transfer data between the processor's working storage area in the central memory of the computer and a group of files located on an auxiliary storage device, as shown in Figure 1. This group of files is referred to as the data base complex.

The data base complex is composed of data files resident on auxiliary disk or drum storage, as shown at the bottom of Figure 1. A maximum of 26 of these data files, each referred to as a "library," are available for use. Users can elect to store the entire data complex in a single library file. These files are recognized by the AAR's DEC-2050 system as having names: SPARA., SPARB., SPARC., ...SPARZ. Library Unit Numbers 5 and 6 have been reserved for input and output devices, therefore the library names, SPARE and SAPRF, are nonexistent and should not be accessed. The files are referred to by the

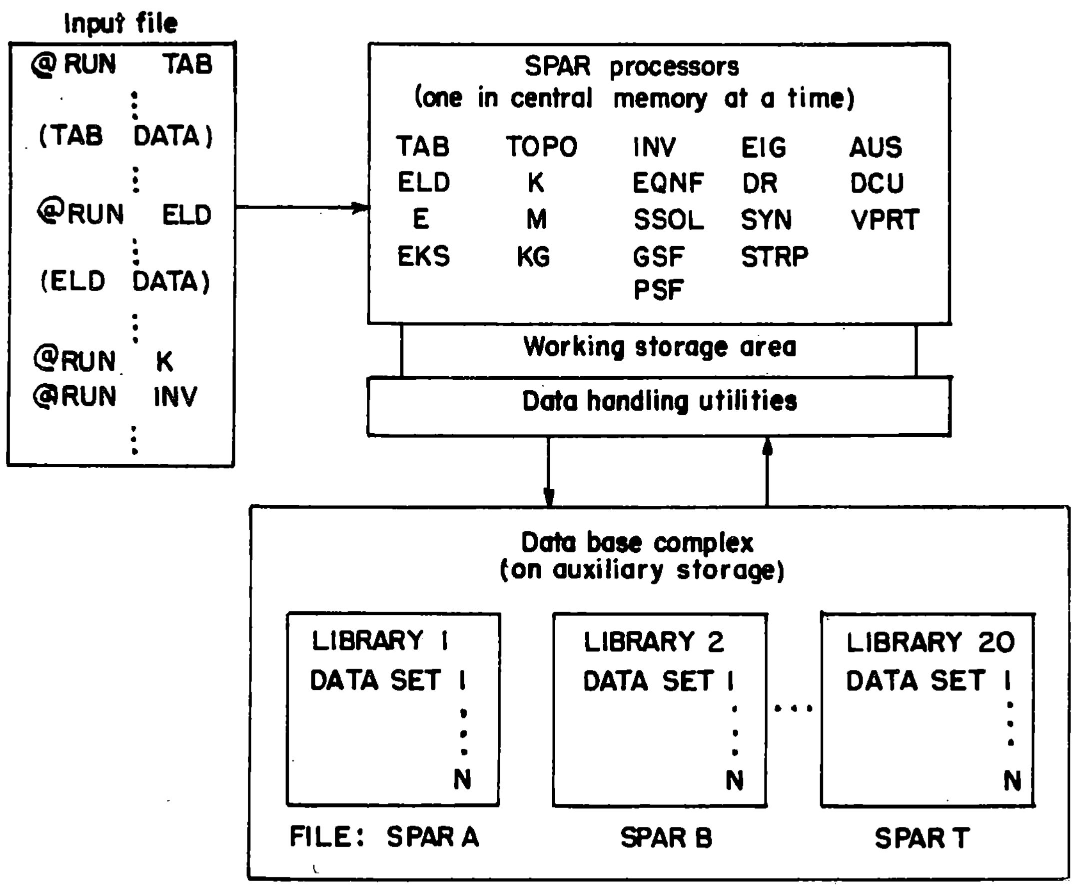


Figure 1. Organization of the SPAR Analytical Program on a DEC 2050 Computer System.

SPAR user as Libraries 1 to 26, with Libraries 1 to 20, with the exception of 5 and 6, available for general use and Libraries 21 to 26 reserved for temporary intermediate storage.

The organization of the data base complex provides significant benefits to the user, such as the ability to reference data by alphanumeric names, automatic "bookkeeping" procedures for the data, expedited communication of data with programs external to SPAR and simplified retention of data between separate computer runs.

2.4.1 SPAR Data Sets

A SPAR data set is a grouping of one or more words that are stored in a library and which can be referred to or accessed as a single entity. All data that are communicated among SPAR processors are handled in terms of data sets.

The data sets are identified by a four-word name: NAME1, NAME2, NAME3, and NAME4. The words NAME1 and NAME2 contain up to four alphanumeric characters, and NAME3 and NAME4 are integers. The following are examples of valid names:

JLOC	BTAB	2	5
K	SPAR	36	0
VIBR	MODE	1	1

The names may be used to indicate the contents of the data sets; for example, the first example above is used to refer to joint locations; the second, to the assembled structural stiffness matrix; and the third, to natural vibrational mode shapes.

A standard format is used for all data sets contained in the libraries. Each data set is composed of a number of blocks, and each block may be interpreted as a two-dimensional matrix. These matrices are dimensioned (NI, NJ), and each block length is always NI times NJ. The information within each block is ordered by column when stored on the disk.

2.4.2 Table of Contents

A table of contents (TOC) is used to relate the names, disk addresses and characteristics (size, type, etc.) of all data sets resident in the data base complex. A TOC exists for each library that is used. A listing of a typical TOC is shown in Table 2. Such a listing is produced by a SPAR utility processor and provides a concise summary of the data produced by a given computer run.

A TOC listing contains one line or row per data set, with each line containing 12 entries, as shown in Table 2. The first column, denoted SEQ, provides a sequence number for each of the data sets, and is not actually stored as one of the 12 entries. The four-word data set name (shown as N1, N2, N3, and N4 in Table 2) is given as the last four entries and is used to identify a particular data set. A description of the first eight entries for a data set are given in the following Table (from Reference [1]):

TOC Item	Description
RR	Disk address pointer to first word of data set at beginning of a new sector; a preceeding minus sign means that the data set has been "disabled" (still resident on disk but cannot be accessed).
DATE	Date of insertion
TIME	Time of entry into the processor which inserted the data set into the library

Table 2

Sample SPAR Data Library Table of Contents.

SEQ	RR	DATE	TIME	ER	WORDS	NJ	NI*NJ	TY	DATA SET NAME N1	N2	N3	N4
1	7	761022	081237	0	18	1	18	0	JDF1	BTAB	1	8
2	-8	761022	081237	0	250	250	250	0	JREF	BTAB	2	6
3	-12	761022	081237	0	12	1	12	1	ALTR	BTAB	2	4
4	13	761022	081237	0	30	2	30	4	TEXT	BTAB	2	1
5	14	761022	081237	0	10	1	10	1	MATC	BTAB	2	2
6	15	761022	081237	0	48	4	48	1	ALTR	BTAB	2	4
7	16	761022	081237	0	750	250	750	1	JLOC	BTAB	2	5
8	28	761022	081237	0	250	250	250	0	JREF	BTAB	2	6
9	32	761022	081237	0	25	1	25	1	SA	BTAB	2	13
10	33	761022	081237	0	250	250	250	0	CON		1	0
11	37	761022	081237	0	2250	250	2250	1	QJJT	BTAB	2	19
12	73	761022	081241	0	3136	196	896	0	DEF	E43	11	4
13	129	761022	081241	0	2	1	2	0	GD	E43	11	4
14	130	761022	081241	0	15	1	15	0	GTIT	E43	11	4
15	131	761022	081241	0	12	12	12	0	NELZ	BTAB	1	11
16	132	761022	081241	0	5	1	5	0	KE		0	0
17	133	761022	081241	0	7	1	7	0	NS		0	0
18	134	761022	081241	0	1	1	1	3	ELTS	NAME	0	0
19	135	761022	081241	0	1	1	1	0	ELTS	LTYP	0	0
20	136	761022	081241	0	1	1	1	0	ELTS	NNOD	0	0
21	137	761022	081241	0	1	1	1	0	ELTS	ISCT	0	0
22	138	761022	081241	0	1	1	1	0	ELTS	NELS	0	0
23	139	761022	081241	0	1	1	1	0	ELTS	LE3	0	0
24	140	761022	081244	0	1500	250	1500	-1	APPL	FORC	1	1
25	164	761022	081244	0	1500	250	1500	-1	APPL	FORC	2	1
26	188	761022	081244	0	1500	250	1500	-1	UNIT	VEC	1	1
27	212	761022	081244	0	1500	250	1500	-1	APPL	MOTI	5	1
28	236	761022	081244	0	1	1	1	-1	TOT	FORC	5	1
29	237	761022	081244	0	1	1	1	-1	CONV	AUS	1	1
30	238	761022	081244	0	1500	250	1500	-1	DISP	INC	1	1
31	262	761022	081245	0	6272	250	896	0	KMAP		1087	17
32	360	761022	081245	0	8960	250	1792	0	AMAP		1675	28
33	506	761022	081254	0	62720	196	320	4	E43	EFIL	11	4
34	1486	761022	081248	0	20	20	20	0	DIR	E43	11	4
35	1487	761022	081248	0	1500	250	1500	-1	DEM	DIAG	0	0
36	1511	761022	081302	0	42560	250	2240	1	K	SPAR	36	0
37	2176	761022	081323	0	50176	250	3584	1	INV	K	1	0
38	2960	761022	081333	0	1500	250	1500	-1	STAT	DISP	100	1
39	2984	761022	081333	0	1500	250	1500	-1	STAT	REAC	2	1
40	3008	761022	081345	0	10780	196	5555	-1	STRS	E43	100	1

TOC Item	Description
ER	Error Code: 0 no error detected during generation of the data set 1 minor error 2 fatal error -1 incomplete data set
WORDS	Total number of words in the data set; data sets are generally comprised of a sequence of physical records, or "blocks"; each block is a two-dimensional matrix, dimensioned (NI, NJ), i.e., NI rows, NJ columns; the block length is always NI*NJ
NJ	See above
NI*NJ	See above
TY	Type Code: 0 integer -1 real -2 double precision 4 alphanumeric

An understanding of the information contained in a TOC is necessary, in order to use the data handling routines in a user program to access the data sets.

2.5 SPAR Element Library

As shown in Table 1, there are over twenty processors that make up SPAR, although some are optional. Each processor is run individually, and the order in which each is run is important, since the output from one may be an input for another.

Each processor's output is automatically given a data set name and sent to a library file. This is usually Library #1, and may be accessed by the user at any time (see the command:DCU).

All input is free field, and SPAR does not recognize any symbols to the right of the dollar sign ($). This makes it convenient to include comments by the user, if desired.

The SPAR element library consists of twelve distinct element types, of which five are beam elements, three are triangular planar elements, (membrane, bending and coupled), three are quadrilateral elements with warping (membrane, bending and coupled) and the remaining one is a shear panel element. Each specific element type is identified by three digits, the first of which is a letter 'E' indicating the word element; the second digit refers to the number of nodes forming the element; the last digit is used to indicate the different kinds of elements (e.g., membrane, bending, etc.) available in each nodal class (2 node, 3 node, etc.). A list of the element types along with short descriptions are summarized in Table 3. This list includes the corresponding section property types which need to be defined in processor TAB.

Table 3 SPAR Element Library

Abbreviated Element Name (ELD)	Associated Section Properties (TAB)	Description
E21	BA	General beam elements, such as angles, wide flanges, tees, zees, tubes, etc.
E22	BB	Beams of finite length, for which the 6x6 intrinsic stiffness matrix is directly specified.
E23	BC	Bar elements having only axial stiffness.
E24	BD	Plane beam.
E25	BB	Zero-length element used to elastically connect two coincident joints.
E31	SA	Triangular membrane; flat; aeolotropic, using TM constant-stress formulation.
E32	SA	Triangular bending element; flat; aeolotropic.
E33	SA	Triangular membrane plus bending element; flat; aeolotropic, using TM and TPB7 formulations.
E41	SA	Quadrilateral membrane; aeolotropic, using QMB5 hybrid formulation.
E42	SA	Quadrilateral bending element; aeolotropic, using hybrid formulation.
E43	SA	Quadrilateral membrane plus bending element; aeolotropic, using hybrid formulation.
E44	SB	Quadrilateral shear panel, using hybrid formulation.

2.6 General Input Rules

On a DEC-2050 system, SPAR consists of an array of independently executable programs (called Processors), resident on a read-only public cataloged directory (SPAR). SPAR permits a very complex input system, and once mastered is extremely efficient. For further information, the user is referred to the SPAR Reference Manual [1].

Since each processor is run separately, the data for each processor should be read in separately. This excludes any data set that SPAR reads from the data base complex, that is resident in the specified catalogued library. The type of data required as input will be explained for each processor individually. In general, the SPAR processors are built on the following two rigid input formats.

The first format is as follows:

Card Image	Meaning
@RUN PROGX	$ Begin execution of program PROGX;

(RESET CARDS)	$ Default parameters (usually optional);
	$ RESET statements allow the
	$ analyst to alter certain parameters
	$ which control processor operation,
	$ e.g., the library unit number, flags
	$ for error and display modes, and
	$ other miscellaneous quantities.

STOPS	Data card terminator.

The second format is as follows:

Card Image	Meaning
@XQT PROGB	$ Begin execution of PROGB;
(RESET CARDS)	$ RESET cards (see previous description)

Card Image	Meaning

Data Cards	\$ These represent the data cards
	\$ which need to be supplied in a
	\$ specific order, as
	\$ demanded by an individual
	\$ processor.

STOP\$	Record or data card terminator

The '\$' sign represents the record terminator, and all characters to the right of '\$' are ignored.

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

5000. = .5+4 = .05+5 = 50.00+2, etc.

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

Each card begins an input record. A record is terminated by an end-of-card, with "STOP" being the last record on that card.

2.7 Interactive and Batch Operations

SPAR can be run either interactively or in a batch mode of operation, and examples of each are given in Appendix C.

In an interactive mode, the user supplies the sequence of input cards for running a particular processor by means of the terminal.

In a batch mode of operation, the user submits a .CTL (Control) file, which contains the input data for running a specific problem on SPAR. A control file therefore allows the computer to read all controls and TOPS 20 commands (e.g., DEFINE, RUN, COPY, etc.) from a file, instead of from the user's terminal. The advantage of this set-up is that only one command (@SUBMIT) is needed to run the program, and all of the necessary commands and data are on one file, in their relative order of execution.

The following format is frequently used for SUBMIT:

@SUBMIT JØBNAME (.CTL default)/SWITCHES

JOBNAME	is the name (six alphanumeric characters, maximum) to be assigned to the batch job. The system used a file type of .CTL, unless specified otherwise.

EXAMPLE

@SUBMIT	JØB1.CTL/TIME:00:30:10

This command places the entries into the batch input queue, for processing the controls contained in the filespec: JØB1.CTL. The maximum time set from beginning to end of the job is 0 hours, 30 minutes and 10 seconds. The computer then takes over and obtains all of the required commands from the .CTL file. When the program has finished execution, the output will go into a file designated as: JOB1.LOG.

By use of the following command:

@DEFINE 6:LPT:

the output from each processor, such as TAB, ELD, TOPO, etc., will go directly to the line printer. The JOB1.LOG file then contains the input and corresponding output response prompts, execution time and other necessary batch job processing parameters.

2.8 Typical Execution Sequence

As stated previously, SPAR is a system of programs, in which processors are selectively called and executed in a logical sequence, in order to perform a desired analytical task. This flexibility in selecting the execution of various processors gives the user considerable versatility in performing structural analyses.

Some of the most commonly used sequences are as follows:

Input	Function
@RUN TAB RESET Cards Data Cards STOP$	Creates the data sets containing the tables of joint locations, section properties, material constants, etc.
@RUN ELD RESET Cards Data Cards STOP$	Defines the basic mesh of the finite elements.
@RUN E RESET Cards STOP$ @RUN EKS RESET Cards STOP$	Forms a complete, detailed, finite element model (e.g., element geometry, intrinsic stiffness and stress matrices, etc.) of the structure. The resulting data sets are referred to collectively as the E-State.
@RUN TOPO RESET Cards STOP$	Analyzes the element interconnectivities.
@RUN K RESET Cards STOP$	Forms the system stiffness matrix, K.
@RUN M RESET Cards STOP$	Forms the mass matrix, M, for the system, if required (optional).
@RUN INV RESET Cards STOP$	Factors the system K (or other designated system matrix, e.g., K + Kg or K - cM, with the use of the RESET control cards).

The above sequence of processors is common to most analytical tasks.

2.8.1 Static Analysis:

Input	Function
@RUN AUS RESET Cards Data Cards STOP$	Defines the applied loadings, temperatures, etc.
@RUN SSOL RESET Cards STOP$	Calculates the static displacement solutions.
@RUN VPRT RESET Cards Data Cards STOP$	Displays the static displacements.
@RUN GSF RESET Cards STOP$	Calculates the stresses and internal loads.
@RUN PSF RESET Cards STOP$	Edits and displays the calculated stresses.

2.8.2 Dynamic Analysis:

Input	Function
@RUN EIG RESET Cards STOP$	Calculates the eigenvalues and corresponding vibrational mode shapes.
@RUN AUS RESET Cards Data Cards STOP$	Defines the applied loadings, etc.
@RUN DR RESET Cards STOP$	Calculates the dynamic response.

2.8.3 Buckling Analysis:

Input	Function
@RUN KG RESET Cards STOP$	Forms the geometric stiffness matrix, K_g, based upon a designated internal load state.
@RUN EIG RESET Cards STOP$	Calculates the buckling modes.

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

2.9 SPAR Output Data Sets

One of the important features of SPAR is its ability to easily and frequently read and write from the common data base complex. Each processor in SPAR references only a few of the data sets that can be accessed. The TOC segments of the specified library are automatically searched until a match with the required data set name and the existing data set on the data base is found. If the data set does not exist or completely match, an error message is displayed. Similarly, each processor or sub-processor produces a collection of data sets (with pre-defined data set names) that are written on the data base complex in the same sequence that the processors or sub-processors are called. Table 4 lists the relationships between specific processors or sub-processors and the associated data sets names that are generated. These are important to the user, because only the data set names are listed in the Table of Contents (See DCU). It is,therefore, possible to determine which sub-processors have been run successfully and those that have failed. The associated list of error messages are given in Section 2.10.

Table 5 summarizes the data sets used in Static Problem Solutions.

Table 4.

Relationships Between Processors and Associated Data Set Names.

Processor	Sub-Processor	Data Sets	Description	Report Section Number
TAB	START	JDF1 BTAB		
	JREF	JREF BTAB	Joint Reference Frame	3.1.7
	ALTREF	ALTR BTAB	Alternate Reference Frame	3.1.5
	TITLE	NDAL		
	TEXT	TEXT BTAB	Alphanumeric Text	3.1.2
	MATC	MATC BTAB	Material Constants	3.1.3
	JLOC	JLOC BTAB	Joint Locations	3.1.6
	MREF	MREF BTAB	Beam Orientations	3.1.8
	BA,BB,etc.	BX BTAB	Beam Section Properties	3.1.10,11
	CON	CON	Constraint Definitions	3.1.16
	SA,SB	AR BTAB	Shell or Panel Section Properties	3.1.14,15
		QJJT BTAB		
	BRL	BRL BTAB	Beam Rigid Links	3.1.9
	JSEQ	JSEQ BTAB	Joint Elimination Sequence	3.1.17
	RMASS	RMAS BTAB	Rigid Masses	3.1.18
ELD	EXY	DEF EXY	Basic Element Definitions	
		GD EXY	Group Directory	
		GTIT EXY	Group Titles	
		KE	Element Packet Pointers	
		NELZ BTAB		
		NS	Element Packet Entries	

Table 4 : (Cont'd)

Processor	Sub-Processor	Output Data Sets	Description	Report Section Number
		ELTS NAME	Element Names	
		ELTS LTYPE	Element Types	
		ELTS ISCT	Element Sections	
		ELTS NELS	Element Numbers	
		ELTS LE3	Element Directory	
TOPO		AMAP	Topological Array	
		KMAP		
E		EXY EFIL	Element Information File (EIP)	
		DIR EXY	Directory of EIP	
		DEM DIAG	Lumped Mass Matrix	
EKS		None		
K		K SPAR	System Stiffness Matrix (Unconstrained)	
M		CEM SPAR	Consistent Mass Matrix	
INV		INV K	System Stiffness Matrix (Factored)	
AUS		User Specified		See Table 5
		APPL		See Table 5
		NODA		See Table 5
SSOL		STAT DISP		See Table 5
		STAT REAC		See Table 5
GSF		STRS EXY		See Table 5
PSF		None		See Table 5
EQNF		EQNF FORC		See Table 5
		IS EXY		See Table 5

Table 5.

Summary of Data Sets Involved in Static Solutions

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

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

2.10 Error Messages

Various error messages may originate during the use of the SPAR processors and they are usually self-explanatory. Messages indicating insufficient core space refer only to the data space requirements, and not to the total program space requirements. At the beginning of execution of each processor, a message: DATA SPACE = XXXXX, indicates the amount of currently available working core space. This information enables the user to estimate the total field length needed in the rerun.

Another common message is: ERROR IN CONTROL STATEMENT SYNTAX. This indicates that an illegal RESET name was given, or that the wrong type of data (e.g., integer floating point, typeless, etc.) was inputted.

Another common error message is: LIB READ ERROR. NU.L. KORE.IERR=XX XX XX XX NAME = N1 N2 n3 n4.

Symbols in this message are defined as follows:

- NU = SPAR logical file number.
- L = Data set block length (if the data set exists).
- KORE = Core space available for loading the data set.
- IERR = Error code (see Section 2.4.2). If IERR = 2, core space is not sufficient to load a single block of the data set.
- NAME = Data set name. The last two words of the data set name are always printed as integers. Accordingly, if they are MASKed, they will not be readable (e.g., on DEC systems, they will be displayed as ***).

Many fatal error messages appear in the following form:

FATAL ERROR. NERR, NIND, XXXX, xxxx

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

NERR, NIND	Meaning
XLIB, n	An attempt has been made to refer to the non-existent SPAR logical file n.
ASG, n	An attempt has been made to refer to the non-existent SPAR logical file, n.
NIND, 1	The user has supplied a file that is not in the SPAR direct-access library format to be used as a library. The parameter 1 is used in diagnosis.
RIND, n	The maximum number of data sets allowed for library n has been reached. The normal limit for SPAR Level 9 is 2048 data sets per library.

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

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

INDEX=---

These parameters have the following meanings:

NU = SPAR logical file number (1 to 26).

IWR = 1, if writing on mass data storage; 2, if reading.

IOP = Addressing mode indicator.

KSHFT = Addressing parameter.

L = Number of words in the requested transmission.

ISTAT = I/O status code.

INDEX = Parameters used in diagnosis.

3.0 STRUCTURAL DEFINITION PROCESSORS

3.1 TAB: Basic Table Inputs

The most important input for processor TAB is the START card. This card gives the number of joints in the structure and the joint motion components (if any), which are identically zero for all joints. The format is:

START J, M1 M2 ...

where J = number of joints in the structure;

M1 M2 . . . are the constrained joint motion components.

As an example:

START 33, 4 5 6

is a START card that defines a structure with 33 joints, and involves the following constrained motions: (a) no rotation about the 1 axis: (4); no rotation about the 2 axis (5); and no rotation about the 3 axis (6) at any joint. It does, however, allow for translations in the 1, 2 and 3 directions.

START 33, 1 2 3 4 5 6 would allow no translations or rotations in any direction for all of the 33 joints.

A description of the global axes for SPAR is shown in Figure 2.

3.1.0 TAB Sub-Processors

Once the START command is inserted, any number of sub-processors within TAB may be called. Table 6 gives a list of these sub-processors and their functions. In order to call up a sub-processor, the short or long name is entered, as described in Table 6. Note that for a particular problem, only a few of the listed sub-processors are actually needed.

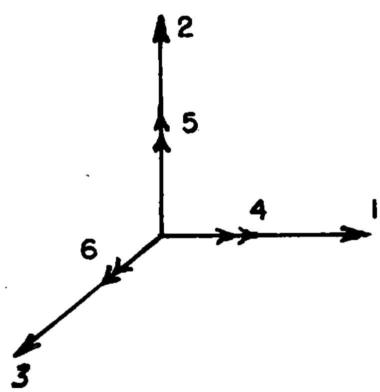


Figure 2. SPAR Global Reference Frame or Axes (Default).

Table 6. TAB Sub-processors for the Definition of Geometric and Material Properties

Short Name	Reference Section Number	Long Form Name and Function
TITLE	3.1.1	TITLE. Sixty character alphanumeric title for Lib.1.
TEXT	3.1.2	TEXT. Creates a data set containing the alphanumeric text documenting the analysis.
MATC	3.1.3	MATERIAL CONSTANTS. These entries define the linear material constants, e.g., E, ν, etc.
NSW	3.1.4	DISTRIBUTED WEIGHT. These entries define the non-structural distributed weight parameters, e.g., weight/length or weight/area.
ALTREF	3.1.5	ALTERNATE REFERENCE FRAMES. These entries describe the reference frames selected by the analyst for convenience in defining the joint locations, etc.
JLOC	3.1.6	JOINT LOCATIONS.
JREF	3.1.7	JOINT REFERENCE FRAMES. These entries define the orientation of the reference frames associated with the joints. Also used in defining the constraints, applied loadings, etc.,
MREF	3.1.8	BEAM ORIENTATION. These entries define the beam cross-section orientation.
BRL	3.1.9	BEAM RIGID LINKS. These entries define the rigid links off-setting the end points of the elastic, two-node elements from the joints they connect.
BA	3.1.10	E21 SECTION PROPERTIES.
BB	3.1.11	BEAM S6x6. These entries define the elastic characteristics of Type E22 and E26 elements.
BC	3.1.12	E23 SECTION PROPERTIES.
BD	3.1.13	E24 SECTION PROPERTIES.
SA	3.1.14	SHELL SECTION PROPERTIES.

Table 6 (Cont'd)

Short Name	Reference Section Number	Long Form Name and Function
SB	3.1.15	PANEL SECTION PROPERTIES.
CON	3.1.16	CONSTRAINT DEFINITION.
JSEQ	3.1.17	JOINT ELIMINATION SEQUENCE.
RMASS	3.1.18	RIGID MASSES.

MATC, JLOC, CON: any one of the BX sub-processors for beam elements; and SX sub-processors for plate elements, if present, are the minimum number of sub-processors required.

The following sub-sections explain the data associated with each sub-processor. There are a number of element types that are available in SPAR, and which can be accessed by a number of TAB sub-processors. A list of these element type definitions has already been given in Table 3.

3.1.1 Title Printout-Sub-processor: TITLE

Allows a title to be printed out with the TOC (Table of Contents) listing of Library 1.

3.1.2 Text Printout-Sub-processor: TEXT

Allows a text to be printed out with the output. The format for the TEXT is:

'60 character alphanumeric string.

(Note: the character ' must appear in Column 1).

3.1.3 Material Constants-Sub-processor: MATC

Allows for the input of Young's Modulus (E); Poisson's ratio (ν); mass density (ρ); thermal expansion coefficients (α_1 in the x-direction, α_2 in the y-direction); angle of rotation (θ) between the local and global reference frames (Note: θ must be in degrees). If θ and α_2 are omitted during the input, the program sets $\theta=0$ and $\alpha_2 = \alpha_1$. The α_1 and α_2 are referred to the local x and y directions.

The format for the MATC input is:

k, E, ν, ρ, α_1, α_2, θ

where k is the material number.

3.1.4 Non-Structural Weights - Sub-processor: NSW

Allows for the input of non-structural distributed weights. The general format for the input is:

k, w

where: k: entry number

w: weight per unit length, for two-node elements

w: weight per unit area for three and four-node elements

3.1.5 Alternate Reference Frame - Sub-processor: ALTREF

In some cases, it might be convenient to set up a local coordinate system (local reference frame), and this reference frame is identified by an integer. Although the global frame is always FRAME 1, the user has the option to define other frames: 2, 3, 4 --- etc.

The format for the input is:

k, I1, A1, I2, A2, I3, A3, X1, X2, X3

where: X1, X2 and X3 are coordinates (relative to the global frame) of the origin of the frame, k, being defined,

I1, A1, I2, A2, I3, A3 are the ordered rotations defining the frame, k.

Within the sub-processor ALTREF, there are two commands available to allow the user to define ordered rotations:

FORMAT = 1 and FORMAT = 2

Using the command: FORMAT = 1 (default value), the rotational sequence is as follows:

(a) rotate the local frame A1 degrees about the local axis, I1, (b) from this new position, rotate the local frame A2 degrees about the axis, I2, and then (c) from this position rotate the local frame A3 degrees about the axis, I3.

If FORMAT = 2 is used, the I's and A's indicate rotation of the frame, k, relative to the global frame.

For an example and a description of the axis rotation, see Figure 3.

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

27   3,30.   1,10.$

3.1.6 Joint Locations - Sub-processor: JLOC

JLOC allows an input for the joint coordinates of the structure. Two separate coordinate systems: rectangular and cylindrical, are available to the user. Unless otherwise specified by the user, the data appears in rectangular coordinates. To use cylindrical coordinates, insert the FORMAT = 2 card before any joint location data is inputted.

There are two main methods of inserting data into JLOC. The user may define each joint by a joint number and its location. The form of this input is:

K, X1, X2, X3

where the X's are the coordinates for joint K. Spaces instead of commas may be used to separate the data.

The second method is to use an automatic grid generator.

3.1.6.1 One Dimensional Generation

This defines the locations of multiple joints in one direction. As an example, when defining the nodes of a beam, the input sequence is:

K, Xa_1, Xa_2, Xa_3, Xb_1, Xb_2, Xb_3, ni, ijump.

where K: starting node number;

Xa's: coordinates of starting node;

Xb's: coordinates of ending node;

ni: number of equally-spaced nodes (including origin and terminus);

ijump: joint number increment for each node (default = 1).

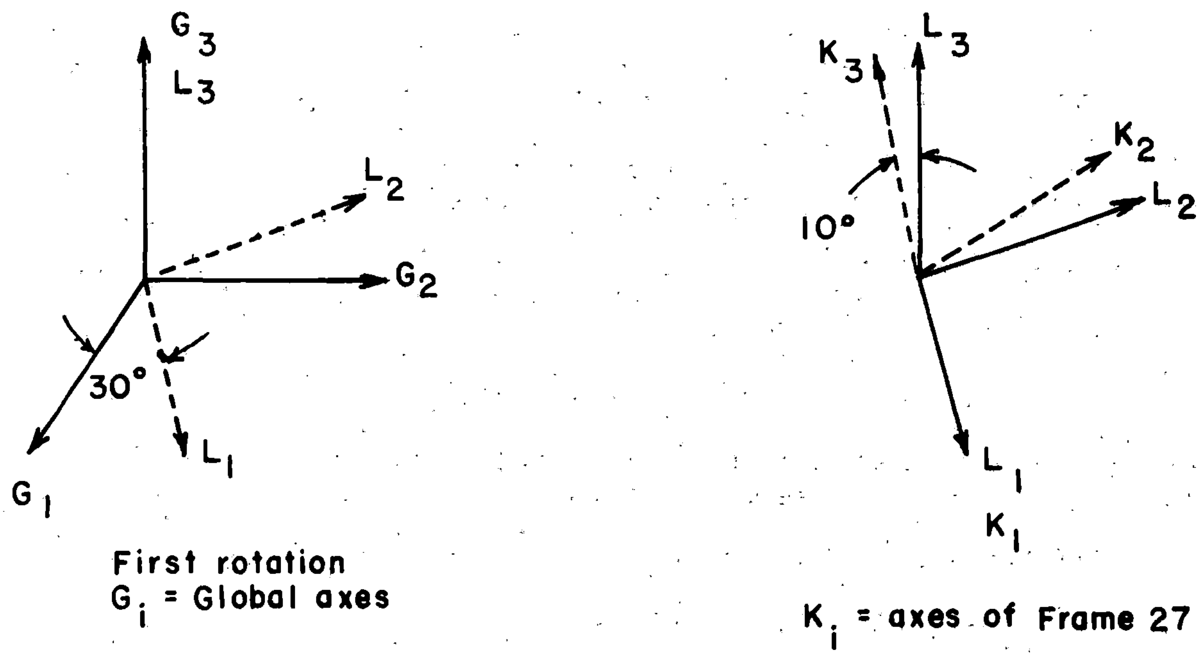


Figure 3. Sample Axis Rotation.

3.1.6.2 Two Dimensional Generation

This involves a grid that is defined by the four corner nodes. Two cards are required for the input sequence:

Card 1: K, Xa_1, Xa_2, Xa_3, Xb_1, Xb_2, Xb_3, ni, ijump, nj

Card 2: jjump, Xc_1, Xc_2, Xc_3, Xd_1, Xd_2, Xd_3

where:

K:	starting node number;
Xa's--Xd's:	Locations of the four corner nodes;
ni:	number of equally-spaced nodes in the x-direction;
nj:	number of equally-spaced nodes in the y-direction.
ijump:	joint numbering increment, in the x-direction;
jjump:	joint numbering increment, in the y-direction.

Combinations of any of the three input sequences may be used.

If the joint locations that appear are not relative to the global frame, but to some other frame, the command:

NREF = n

must be inserted with the joint data. This dictates that the following joint location data is relative to some frame n, until another NREF command is encountered. NREF is defined in sub-processor JREF.

3.1.7 Joint Reference Frames - Sub-processor: JREF

JREF allows the user to change the orientation of any number of joints. This is done by declaring the orientation of the joints perpendicular to some reference frame n.

To expedite the changes, a loop limit format is used for inputting the data, and has the form:

J1, J2, INC.

This implies the transition from J1 to J2 by an incremental amount INC (default value of INC is 1).

The format for JREF is:

NREF = n; (associated joints)

As an example,

NREF = 1Ø : 5,8 : 19 : 3Ø, 34, 2

implies that joints 5 through 8 inclusive, joint 19, and joints 3Ø through 34, in increments of 2, have orientations parallel to frame 1Ø.

3.1.8 Beam Orientation - Sub-processor: MREF

For two-node elements, there is a two-node element reference frame. The 3-axis of this frame is always the axis from the origin to the terminus. MREF determines the relative orientations of the 1-and 2-axes.

The input sequence is (FORMAT=1):

k, NB, NG, ISIGN, C

where k: table entry number, to be used in processor ELD to specify all elements defined by k;

C: cosine of the smallest angle between a beam axis, NB, and a global axis, NG:

ISIGN: keeps track of whether the sign of the cosine of the angle between the beam axis, 3-NB and NG, is plus or minus one.

Since Axis 3 has already been determined, the legal values for NB are 1 and 2. The legal values for NG are 1, 2 and 3.

3.1.9 Beam Rigid Links - Subprocessor: BRL

Should the case arise in a structure where rigid links are needed for the origin and terminus of a two-node element, sub-processor BRL may be used. The sequence is:

$K, I^1, X_1^1, X_2^1, X_3^1, I^2, X_1^2, X_2^2, X_3^2$

where: K: entry number;

X_1^1, X_2^1, X_3^1: coordinates (rectangular) at joint JL (origin) minus position coordinates of the beam origin;

X_1^2, X_2^2, X_3^2: coordinates of J2 (terminus) minus coordinates of beam terminus.

Coordinates must be in rectangular coordinates relative to reference frames: I^1, I^2. Joint numbers are assigned, using coordinate information from TAB and element information from ELD.

3.1.10 Section Properties of Type E21 General Beam Elements - Sub-processor: BA

Sub-processor BA generates a table of section properties for E21 elements.

E21 elements are defined as general beam elements, e.g., angles, WF-sections, T-sections, Z-sections, tubes, etc. Nine different types of sections are allowed and the input varies depending upon the section type.

Table 7 lists typical section types and their corresponding data input sequence. The associated input parameters are defined in Figure 4.

GIVN and DSY provide for general beam sections that are not classified, as shown above. Notice that DSY sections require two lines of input.

Table 7. Types of Beam Sections Associated with Sub-processor BA

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

In the above, k identifies the table entry number. The b's and t's are cross-sectional dimensions, as defined in Figure 5.

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

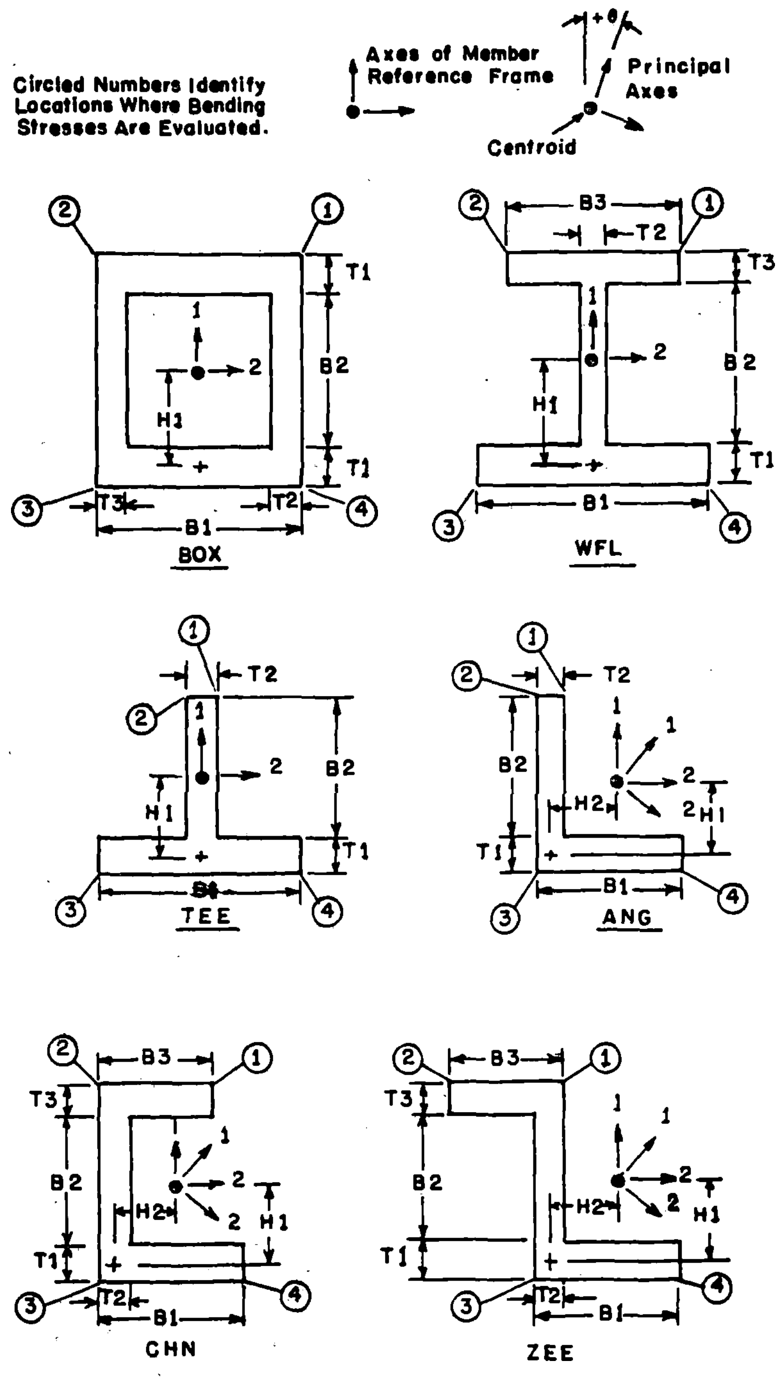


Figure 4. Dimensional Input Parameters for Structural Elements Associated with Sub-processor BA.

For GIVN and DSY sections:

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

α_1, α_2: Transverse shear deflection constants, associated with I_1 and I_2, respectively. For no shear deflection, set $\alpha_i = 0$.

a: cross-sectional area.

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

z_1, z_2: Shear center - centroid offsets.

θ: Inclination of principal axes relative to the element reference frame (see Figure 5). θ is in radians.

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

y_{i1}, y_{i2}: Location relative to the element reference frame of the ith point, at which the My/I combined bending stresses are to be calculated. Up to four such points may be specified.

3.1.11 Section Properties of Type E22 and E25 Beam Elements - Sub-processor: BB.

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

$k,\ s_{11}$

$s_{21},\ s_{22}$

$s_{31},\ s_{32},\ s_{33}$

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

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

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

In the above, k identifies the table entry, and the s_{ij}'s are the elements of a symmetric intrinsic stiffness matrix, S, which is defined as follows:

<u>If the terminus (see ELD, MREF, BRL discussions) of the element is completely fixed</u>,

$$F = SU,$$

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

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

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

3.1.12 Section Properties of Type E23 Bar Elements with Axial Stiffness - Sub-processor: BC.

Sub-processor BC generates a table of axial element section constants, for use with type E23 elements (bar elements having only axial stiffness).

Input sequence: K, a

where: a is the cross-sectional area.

3.1.13 Section Properties of Type E24 Planar Beam Elements- Sub-processor: BD.

Sub-processor BD is used whenever Type E24 plane beam elements are used. Reference is made to BD in processor ELD when handling these types of elements, since BD generates a table of plane beam sectional properties. Figure 5 is a diagram pertaining to the use of sub-processor BD.

The data sequence is: K, A, I_1, α_1, h, c, q_1

where: A: cross section area;

I_1: principal moment of inertia;

α_1: Transverse shear deflection constant associated with I_1;

h: see Figure 5;

c: see Figure 5;

q_1: section shape factor, such that the maximum shear stress from V_1 is $q_1\ V_1$.

3.1.14 Section Properties of Type E31, E32, E33, E41, E42, and E43 Shell Elements - Sub-processor: SA.

SPAR is able to handle six types of shell sections: isotropic, membrane, plate, uncoupled, coupled and laminate.

Only isotropic sections will be covered here, however, a description of the other five sections may be found in Appendix B.

Sup-processor SA generates tables of shell section properties used in processor ELD, for element types E31, E32, E33, E41, E42, and E43 (triangular and quadrilateral elements).

The user should declare the isotropic section in a format command:

FORMAT = ISOTROPIC (DEFAULT)

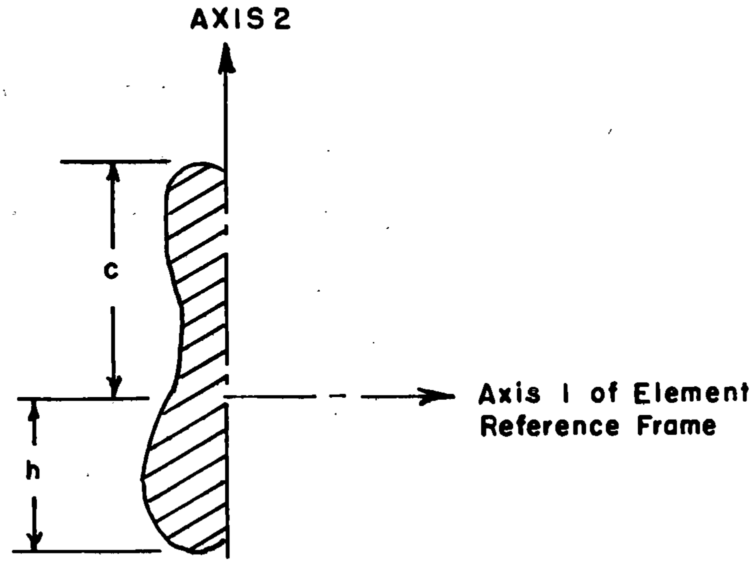


Figure 5. Diagram for Sub-processor BD.

This should then be followed by the input:

K, t $

where: K: section or element number;

t: shell thickness;

$: used to declare the end of input (optional).

It is possible to declare different materials for different sections. The command is:

NMAT = mat (default value is 1).

As an example:

SA	(Invokes the execution of sub-processor SA).
1,0.10 $	using default NMAT = 1 defines Section 1 and
2,0.25 $	2 to be made of Material 1, with thickness
	of 0.1 and 0.25, respectively.
NMAT = 2	Sections 3 and 4 are made of material 2, with
3,0.15 $	respective thicknesses of 0.15 and 0.05.
4,0.05 $	

The type of material used must also be identified in processor ELD.

It should be noted here that the material properties of each defined section are also contained in the sub-processor MATC (TAB) under the respective entry number.

Also listed in Appendix B is a list of possible error codes which might be produced by SA.

3.1.15 Section Properties of Type E44 Shear Panel Elements - Sub-processor: SB.

If shear panel elements are used, sub-processor SB should be used to calculate the section properties. The output is used in processor ELD, for type E44 elements.

The data sequence is: K, t

where: K: section or element number;

t: panel thickness.

3.1.16 Constraint Case n - Sub-processor: CON

Sub-processor CON is used to define any number of different constraint cases that might arise in a structure.

Legal commands to call the CON sub-processor are CONSTRAINT CASE n, CONSTRAINT n and CON = n, where n is an integer representing the constraint case.

Once the sub-processor is called, several commands may then be used: ZERO, NONZERO, SYMMETRY PLANE, ANTISYMMETRY PLANE, and FIXED PLANE. Most analysts will use ZERO and NONZERO. For explanations of the other commands, the user is referred to Reference [1].

The form of the ZERO command is:

ZERO, M1, M2, ...

This specifies that the joint motion components M1, M2 ...etc. are unconditionally ZERO at certain points. A list of constrained joints should follow the ZERO command. They may be individually listed, or the loop limit format may be used (see JREF).

The joint motion components listed for the ZERO command should not repeat the constrained joint motions listed for the START command.

Example: Set the direction-3 displacement and the direction-2 rotation equal to zero for Joints 1, 2, 3, 17, 34, 36 and 38. Assume the START card, as follows:

```
@RUN TAB
START 1Ø2, 4, 6
```

```
CONSTRAINT CASE 1
    ZERO 3, 5
1,3     $ Loop Limit format
17
34,38,2    $ Loop Limit format
```

The command:

NONZERO M1, M2.....

describes those joint motion components which will have some nonzero value. Again, the affected joints should follow the command, as with the ZERO command. The NONZERO command acts as an override to the START command.

3.1.17 Joint Elimination Sequence - Sub-processor: JSEQ

This sub-processor allows user control over the joint elimination process during a factoring sequence.

For most users, this is not a necessary procedure, and it is mentioned here only for reference.

3.1.18 Rigid Mass Definition - Sub-processor: RMASS

This sub-processor allows the definition of rigid masses, and two formats are available. If FORMAT = 1, the input sequence is:

K, M, I_1, I_2, I_3

where: K: joint number to which the rigid mass is attached;
M: mass;
I_1: mass moments of inertia (the joint reference frame must be coincident with the principal axes of the rigid mass).

If FORMAT = 2, the data sequence is:

$K, M_1, M_2, M_3, I_1, I_2, I_3$

where: K, I's: same as above;

M_i: mass that is effective in direction i of the joint reference frame.

It is possible to define the same mass at several joints. The REPEAT command is used:

REPEAT n, j

The same mass is defined at n joints (K, K + j, K + 2j....)

It is also possible to multiply masses by a constant. The CM command is used:

CM, p, q

where: p: constant that multiplies the M or M_i's;

q: constant that multiplies the I's.

Note: The CM commands are not cumulative.

3.2 ELD: Element Definition Processor

Processor ELD accepts the element definition input and data cards, and forms data sets to be used by subsequent processors. Input data takes the form of element types, their interconnections and respective nodes.

As ELD processes the data, it also performs checks on the data already received from TAB, in order to make sure that there are no unreferenced table entry numbers, joint numbers or other erroneous data present.

ELD sets up pointers to applicable entries in the tables of section properties, material constants, etc., for each element type, e.g., E21, E41, etc. There are thus twelve sub-processors and they have been listed in Table 3. Sub-processors may be called in any order, and a typical input stream is as follows:

@RUN ELD

(RESET Cards)

E43

NMAT = 2

NSECT = 1

(Data Cards defining all of the E43 elements in the structure that belong to Section Type 1 and Material Type 2.

NSECT = 1

E21

NSECT = 2

NREF = 1

(Data Cards defining all of the E21 elements in the structure)

NSECT = 3

Seven types of data-sets are produced in ELD. A list of all of the processors and the names of their respective output data-sets may be found in Table 4. These are important to the user, because only the data-set names are listed in the TOC (Table of Contents) (see DCU). It is, therefore, possible to determine which sub-processors were executed and which ones failed.

SPAR makes available to the user the ability to group together several elements and assign to them a group number. This is not needed in the analysis of small systems, however, it may be useful (particularly for graphics and design work) in analyzing structures with many nodes. Users who feel that the GROUP command would be useful are referred to the SPAR System Reference Manual [1].

3.2.1 Input Data Cards

Table 8 shows the form of the input for two, three and four-node elements. The optional parameters are part of a built-in mesh generation system and will be described later.

Table 8. Data Inputs for Two, Three and Four Node Elements.

Two-node Elements	Optional
J1, J2	NETOPT, NET(1), NET(2) ---
Three-node Elements:	
J1, J2, J3	NETOPT, NET(1), NET(2) ---
Four-node Elements:	
J1, J2, J3, J4	NETOPT, NET(1), NET(2) ---

Figure 6 shows the element reference frames that result from following the above nodal sequence. It should be noted that the order in which J1, J2 ... appear on the element definition cards is important, because positive pressures act in the direction of the three-axis of the element reference frame. All element-related quantities, such as stiffness coefficients, stresses, etc., are relative to these reference frames.

If desired, the user can list each element and its corresponding nodes individually, or use the mesh generator.

3.2.2 Mesh Generation Commands

There are eight mesh generation options available in SPAR: three options for two-node elements, three options for three-node elements, and two options for four-node elements.

The next few pages give some examples of the network option plans (NETOPT), including the sequence for the data inputs. The user is advised to look over these examples and then observe some of the sample problems to see how these options are actually used.

3.2.2.1 Two Node Element Mesh Generation

Input Data Format: J1, J2, NETOPT, NET(1), NET(2), NET(3)

(a) Option 1 (NETOPT = 1): Rectilinear Network

As an example, given the following data in the ELD input stream:

J1	J2	NETOPT	NI(default = 1)	NJ(default = 1)	JINC
9	12	1	4	2	1ØØ

the resulting rectilinear network is as shown in Figure 7.

(b) Option 2 (NETOPT = 2): Closed Polynomial Network

As an example, given the following input data:

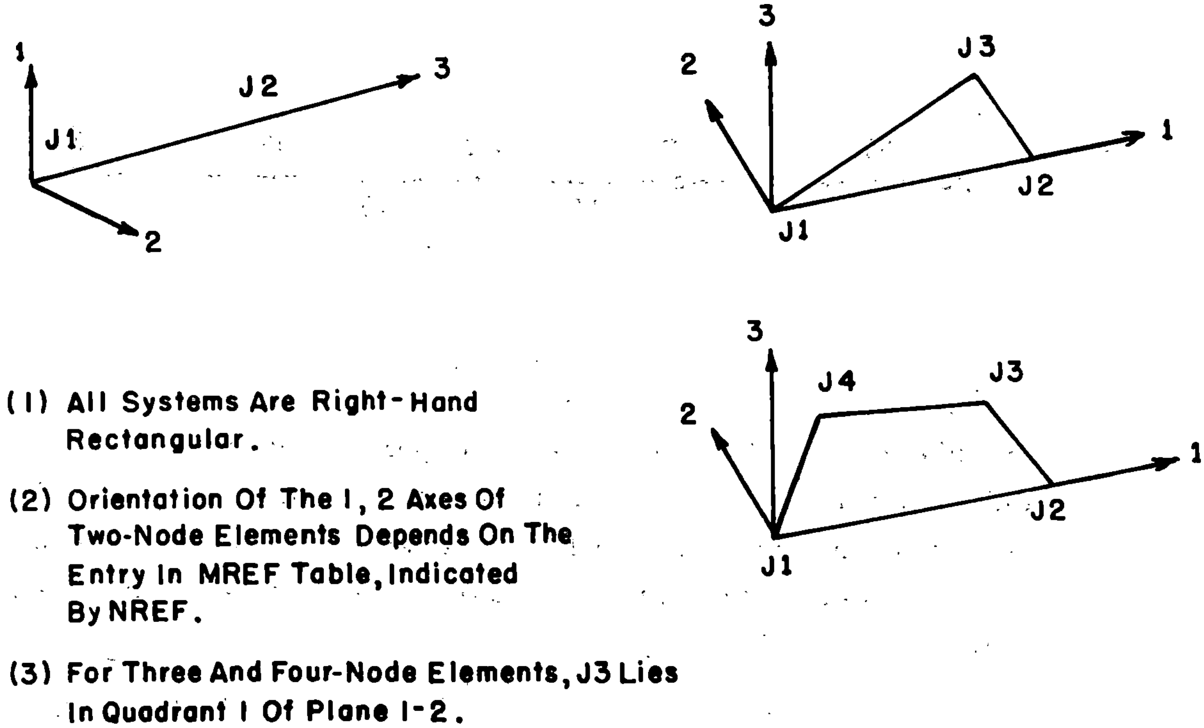
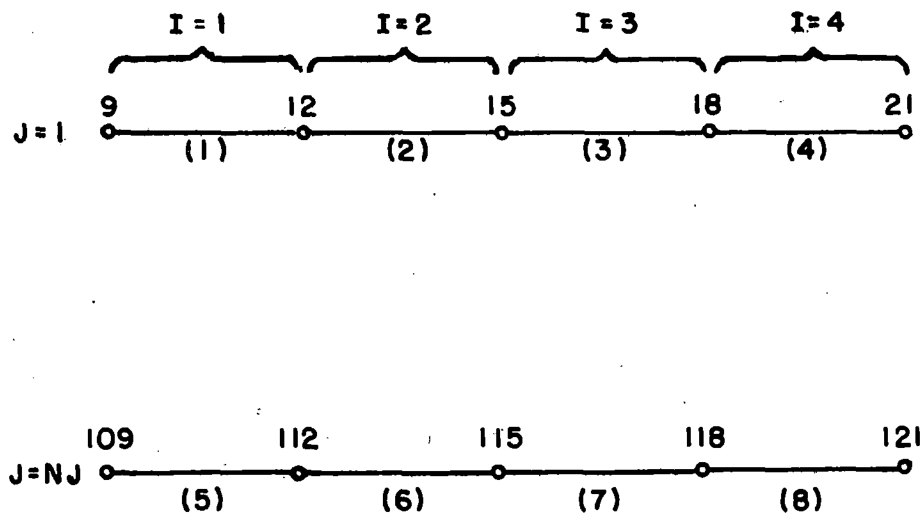


Figure 6. Element Reference Frame Assignments.



NOTE: The element numbers enclosed in parentheses are identified and determined by the order in which the element cards are defined, and the particular strategy followed. The difference between every two consecutive elements is constant, i.e., IINC = J2 - J1.

Figure 7. Example of a Two Node Element Mesh Generation, Using Option 1 to Generate a Rectilinear Network.

J1 J2 NETOPT NI NJ(default = 1) JINC

9 12 2 4 2 1ØØ

the resulting closed polynomial network is as shown in Figure 8. Note that with NI=4, it forms a rhombus. With NI=6, it generates a hexagon, etc. Since JINC=100, it places the element string along a closed path.

(c) Option 3 (NETOPT = 3): Open Intersecting Network

Input Data Format:

J1 J2 NETOPT NI IINC NJ(default = 1) JINC

As an example, given the following input data:

J1 J2 NETOPT NI IINC NJ JINC

4 37 3 4 1Ø 2 1ØØ

a mesh with intersecting straight line members is generated, as shown in Figure 9.

3.2.2.2 Three Node Element Mesh Generation

(a) Option 1 (NETOPT = 1): Triangular Network

Using the following input data format:

J1 J2 J3 NETOPT NI NJ(default = 1)

2 3 1Ø3 1 3 2

a triangular network is generated, as shown in Figure 10. The increments, IINC and JINC, for generating the nodes in two directions (I and J) are calculated as follows:

IINC = J2-J1

JINC = J3-J2

The generation is completed from one rectangle to another in each row, before proceeding along another row. The numbers inside the parentheses in Figure 8 represent the element sequence numbers.

(b) Option 2 (NETOPT = 2): Open Triad Polynomial Network

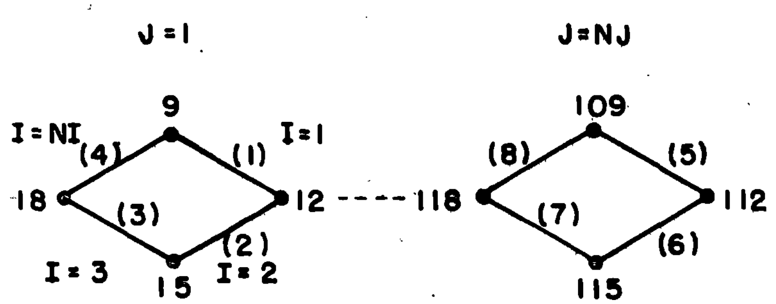
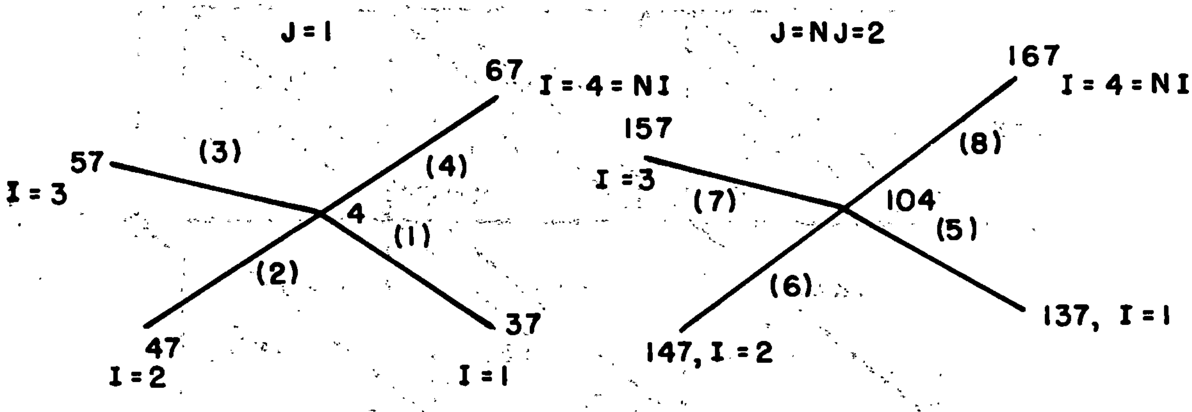


Figure 8. Example of a Two Node Element Mesh Generation, Using Option 2 to Generate a Closed Polynomial Network.



NOTE:

Node J_1 Is Kept Fixed And The Line Members Are Generated By Connecting J_1 With Points J_2, J_2+IINC, ····· J_2+(NI-1)·IINC.

Figure 9. Example of a Two Node Element Mesh Generation, Using Option 3 to Generate an Open Intersecting Network.

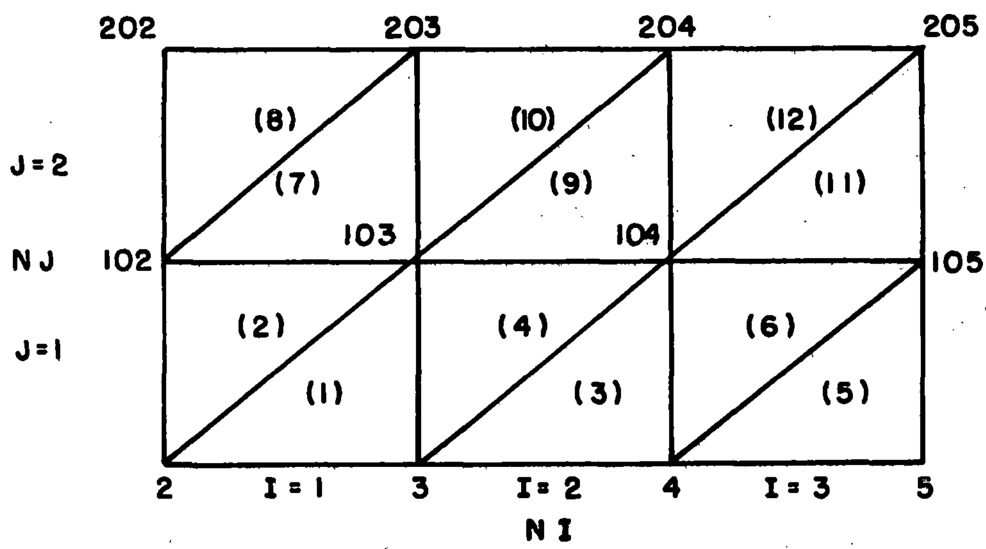


Figure 10. Example of a Three Node Element Mesh Generation, Using Option 1 to Generate a Triangular Network.

This case is the analogy of Option 2 for a one-dimensional element, and the elements are formed around a fixed point, i.e., IINC = 0.

Using the following input data:

J1	J2	J3	NETOPT	NI (must be greater than 1)	NJ (default = 1)	JINC
2	5	7	2	6	2	3Ø

the resulting open triad polynomial mesh network is as shown in Figure 11.

(c) Option 3 (NETOPT = 3): Closed Triad Polynomial Network

Using the following input data:

J1	J2	J3	NETOPT	NI	NJ (default = 1)	JINC
2	5	7	3	6	2	3Ø

the resulting closed triad polynomial network is as shown in Figure 12. This mesh is identified with the one generated under Option 2, but is closed, i.e., has a common line (two common points).

3.2.2.3 Four Node Element Mesh Generation

The following two networks can be used to generate element types E41, E42, E43 and E44. The input definitions and the resulting generated networks are shown below:

(a) Option 1 (NETOPT = 1): Rectangular Network

Using the following input data:

J1	J2	J3	J4	NETOPT	NI	NJ (default = 1)	NK(default = 1)	KINC
2	3	23	22	1	2	3	2	2ØØ

results in the rectangular network shown in Figure 13.

(b) Option 2 (NETOPT = 2): Cylindrical Network

Using the following input data:

J1	J2	J3	J4	NETOPT	NI	NJ (default = 1)
1	11	12	2	2	6	3

results in the cylindrical network shown in Figure 14.

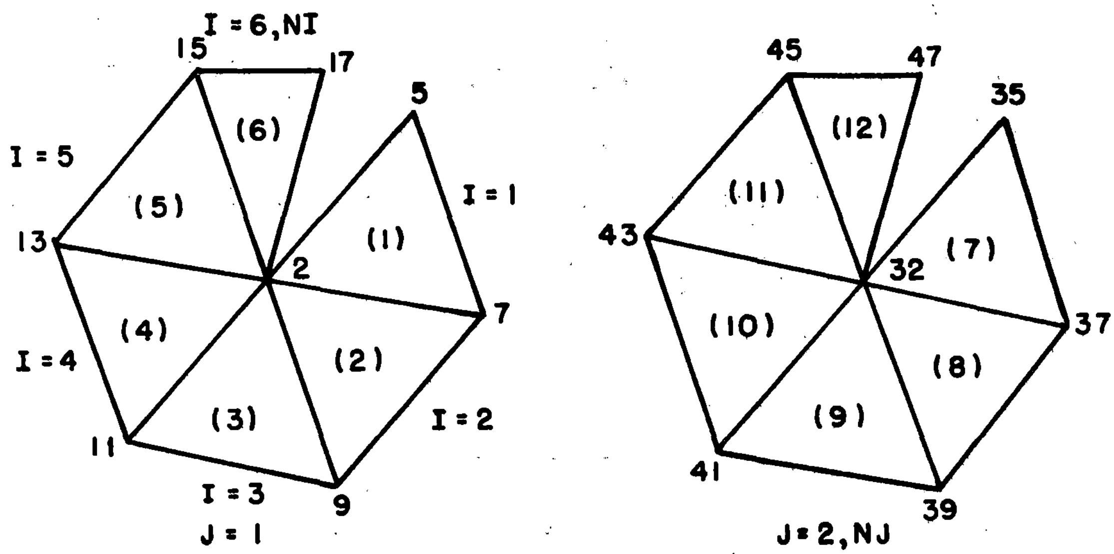


Figure 11. Example of a Three Node Element Mesh Generation, Using Option 2 to Generate an Open Triad Polynomial Network.

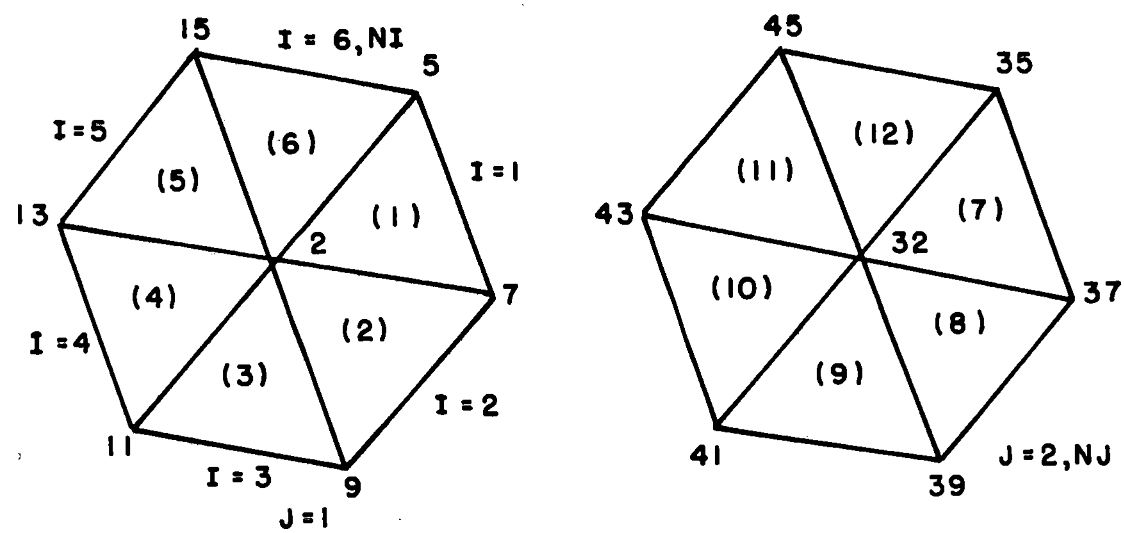


Figure 12. Example of a Three Node Element Mesh Generation, Using Option 3 to Generate a Closed Triad Polynomial Network.

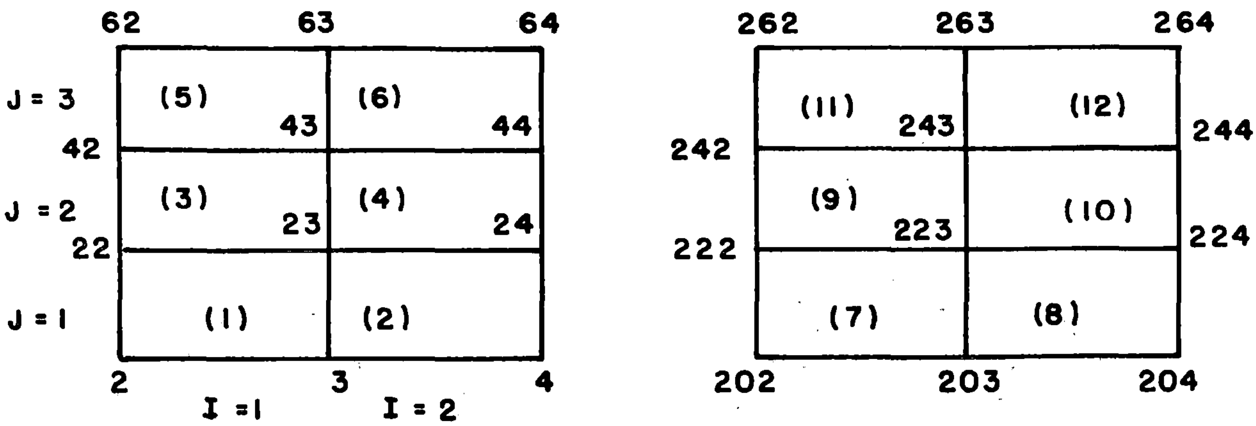


Figure 13. Example of a Four Node Element Mesh Generation, Using Option 1 to Generate a Rectangular Network.

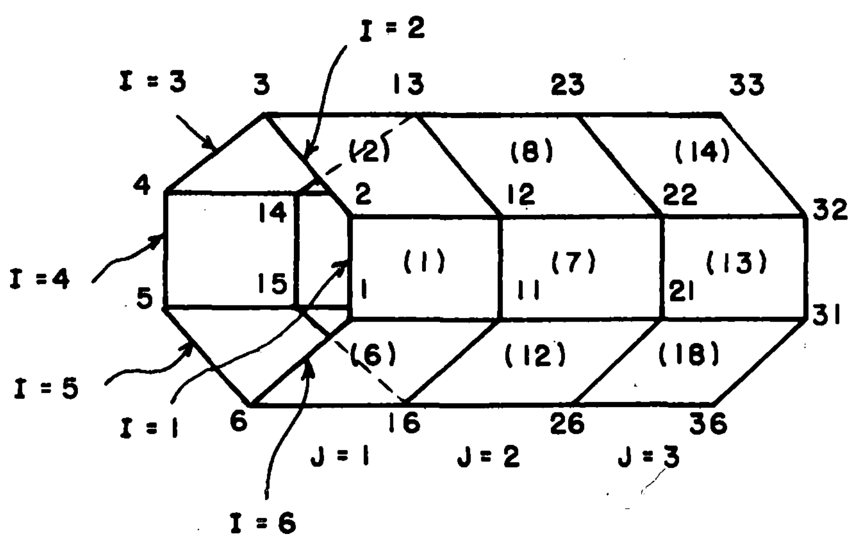


Figure 14. Example of a Four Node Element Mesh Generation, Using Option 2 to Generate a Cylindrical Network.

3.3 E-E State Definition

Processor E constructs the "element information packets" (EIP) for each element type used in the model. The processor extracts the required data and pertinent information from TAB and ELD. From these inputs, an ordered arrangement of information is constructed within EIP, and includes pointers to access it (omitting the intrinsic stiffness and stress data). These informational arrangements are done automatically and without user interaction.

Reset commands exist that can produce internal changes in SPAR, but are normally not needed. This is mentioned here only as information.

3.4 EKS: Element Intrinsic Stiffness and Stress Matrix Generator

Using information supplied by data packets from E, EKS calculates the intrinsic stiffness and stress matrices, and inserts them into the data packets.

The structural weight and dimensions (total length for two-node elements and areas for three and four-node elements) are printed out for each element type used in the model.

Reset values are available to the user, but the default values should more than suffice.

4.0 SYSTEM MATRIX PROCESSORS

There are six processors that assemble, factor and display the system matrices: TOPO, K, INV, M, KG and PS. Only the first three are used for static solutions.

Figure 15 shows the flow of data to and from these processors.

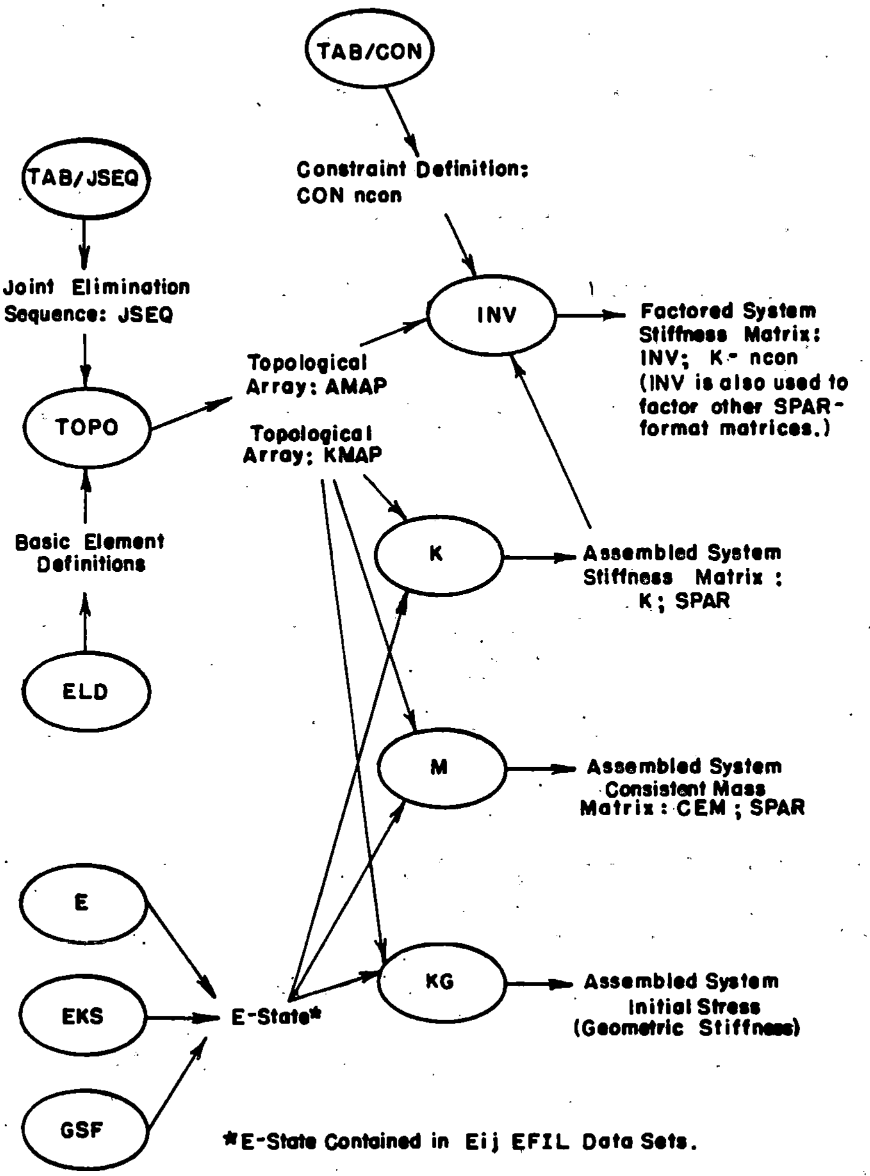


Figure 15. Data Flow Diagram for the SPAR System Matrix Processors.

4.1 TOPO: Element Topology Analyzer

Processor TOPO analyzes the element interconnectivities. Its output contains two data sets: AMAP and KMAP, and both are used for factoring the system and mass matrices. The names of the data sets generated by this processor is listed in Table 4.

Reset values are available, but are not normally used. If needed, the SPAR System Reference Manual [1] should be consulted.

4.2 K: System Stiffness Matrix Assembler

Processor K takes data from E, EKS and TOPO and assembles the unconstrained stiffness matrix K. The output data set is K SPAR. Reset values may be found in Reference [1]. The names of the data sets generated by this processor is listed in Table 4.

4.3 INV: SPAR Matrix Decomposition Processor

Processor INV factors the assembled system matrices. Input data sets to INV are K SPAR (created in processor K), CON (created in processor TAB) and AMAP (created in processor TOPO).

The output data set is in the following form:

INV XXXX n

where XXXX: first word of input stiffness matrix;

n: integer describing a constraint case.

There are two important reset controls which the user might find helpful. The form of the reset command is:

RESET name = value

Name	Default Value
K	K
CON	1

Typical reset for K is:

RESET K = K + KG†

where KG is the geometric stiffness matrix.

CON is an integer representing the constraint case to be used. Other reset controls exist [1], but are not normally needed by the user for static stress analyses.

An optional command is offered with INV. This is the ONLINE command. If ONLINE = 2 is used, printouts of joints with negative diagonal terms during factoring will occur.

5.0 UTILITY PROCESSORS

There are three utility programs built into SPAR,

AUS: Arithmetic Utility System

DCU: Data Complex Utility

VPRT: Vector Printer

AUS does all of the matrix arithmetic, altering the data sets to accomodate different loadings and operations related to substructural analysis.

DCU is the processor that allows user access to a given problem's library files. It allows a display of the table of contents (a list of all of the data-set names produced thus far), along with a print-out of the contents of any data set. More functions are available with DCU, but these will be discussed later.

VPRT edits and displays data sets, such as eigenvectors, static displacements, reactions and nodal load vectors.

5.1 AUS: Arithmetic Utility System

There are four sub-processors in AUS that have the ability to construct data sets: ALPHA, TABLE, ELDATA and SYSVEC. Each subprocessor has the ability to produce specialized data sets, as summarized below:

---

†For a discussion of methods of combining system matrices (e.g., K+KG, K-CM, adding rigid masses to CEM or DEM, etc.), see the SUM sub-processor of AUS [1].

- ALPHA — Creates or modifies data sets containing arrays of alpha-numeric character strings, used as load case titles, or to describe eigenvectors.
- TABLE — Creates or modifies data sets associated with nodal pressures, nodal temperatures and data produced in TAB.
- ELDATA — Creates or modifies data sets associated with element applied load data, such as pressures, temperatures and dislocations.
- SYSVEC — Creates or modifies data sets associated with system vectors, such as applied nodal forces, motions, moments and eigenvectors.

In addition to the above, there are sixteen sub-sub-processors to carryout the general arithmetical operations.

Examples

K+KG=SUM(K, 4.7 KG)	System stiffness matrix, including the effects of prestress.
M1=SUM(RMASS, DEM)	Diagonal system matrix, composed of rigid mass data plus the lumped-mass equivalent of all of the distributed element mass.
M2=SUM(CEM, RMASS)	SPAR-format, consistent mass matrix, plus rigid-mass data.
K24=SUM(K, -2400.M)	Shifted stiffness matrix to be used in EIG to compute eigenvalues near 24,000.

The list of the available sub-sub-processors in AUS are given in Table 9, and additional details can be found in Reference [1].

Table 9. AUS Sub-processors Used for General Arithmetical Operations.

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

5.1.1 Sub-processor: ALPHA

The general form for ALPHA is:

```
ALPHA
N1  N2  n3  n4
      data
```

N1, N2 refer to data-set names (i.e., CASE TITLES, etc.), n3 and n4 refer to the associated block numbers. If n4 is omitted, n4 is zero.

The data for ALPHA consists of the case (or block) number, followed by a ('), and up to a 60 character alphanumeric string.

Example

```
@ RUN AUS


ALPHA
CASE TITLES 2 $
1' THIS DEFINES A TITLE FOR CASE 1
2' THIS DEFINES A TITLE FOR CASE 2
```

5.1.2 Sub-processor: TABLE

The general form for TABLE is:

```
TABLE, u
N1 N2 n3 n4
data
```

SPAR has very elaborate input rules for TABLE, and the user may find it easier not to take short cuts at this point. Note that N1 N2 n3 n4 is a data set that is to be created.

(a) Example No.1:

(1) @RUN AUS	Execute the processor AUS
(2) TABLE	Call sub-processor Table

(3)	NODAL TEMPERATURE 1$	Create a data set called NODA TEMP 1 1
(4)	CASE 1$	First case of joint temperature is to follow (this input may be omitted if CASE 2 does not exist)
(5)	I=1$	Temperature loads act in direction 1 only. (Default value for I is 1)
(6)	JOINTS = 12, 16	Loads act on Joints 12, 13, 14, 15, and 16
(7)	0.5, 0.5, 0.5625, 0.5625, 0.6245 $	Load at Joint 12 in Direction 1 is 0.5
		Load at Joint 13 in Direction 1 is 0.5
		Load at Joint 14 in Direction 1 is 0.5625
		Load at Joint 15 in Direction 1 is 0.5625
		Load at Joint 16 in Direction 1 is 0.6245

(b) Example No. 2:

(1)	RUN AUS	Execute the processor AUS
(2)	TABLE	Call the sub-processor TABLE
(3)	NODAL TEMPERATURE 1$	Create a data set called NODA PRES 1 1
(4)	I = 1,3: J=2,5$	Pressure loads are in Directions 1 and 3 on Joints 2, 3, 4 and 5
(5)	7.,8.: 7.2,8.1: 7.4,8.2: 7.6,8.3$	Load on Joint 2 in Direction 1 is 7
		Load on Joint 2 in Direction 3 is 8
		Load on Joint 3 in Direction 1 is 7.2
		Load on Joint 3 in Direction 3 is 8.1
		Load on Joint 4 in Direction 1 is 7.4
		Load on Joint 4 in Direction 3 is 8.2
		Load on Joint 5 in Direction 1 is 7.6
		Load on Joint 5 in Direction 3 is 8.3

The alternate form to input the pressure load is:

```
7., 8. $          Loads on Joint 2 in Direction 1 and 3
7.2, 8.1 $        Loads on Joint 3 in Direction 1 and 3
7.4, 8.2 $        Loads on Joint 4 in Direction 1 and 3
7.6, 8.3 $        Loads on Joint 5 in Direction 1 and 3
```

5.1.3 Sub-processor: ELDATA

The general form for ELDATA is:

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

The data is similar to that in TABLE, except that ELDATA uses elements instead of joints. N1 N2 n3 refers to a portion of a data set name. The CASE n4 completes the name. The case statement refers to a distinct input data set.

Along with using I = XXX to determine the directions of the acting loads, two other lists are needed, as follows:

$$E = e_1, e_2, e_3$$

where: E refers to the joints, and e_1, e_2..... are the element number indices.

$$G = g_1, g_2 \ldots$$

where: g_1, g_2.... are the group numbers. (See Processor ELD for the group number definitions).

Note that if E = is omitted, the default is to list all elements. <u>The G = statement must not be omitted!</u> This will produce an error.

To modify an already-existent ELDATA data set, use:

```
ELDATA, u
  N1  N2  n3  $
```

To cause an integer 'gshift' to be added to each group number in subsequent lists of groups, the command:

GSHIFT = gshift should be used.

Likewise, to shift element indices, use:

ESHIFT = eshift

For a more rigorous discussion of ELDATA, the user is referred to Reference [1].

5.1.4 Sub-processor: SYSVEC

The general form of SYSVEC is:

```
SYSVEC, u
N1  N2  n3  n4
```

SYSVEC prepares <u>SPAR</u> for input data, concerning loads or moments acting at a joint.

The input rules are similar to TABLE. This is the sub-processor of AUS that will be used most often.

The following example concerns nodal force*inputs.

```
        @RUN AUS

        SYSVEC                      Enter SYSVEC

        APPLIED FORCES 88$          Name of output data set.

CASE 1$
     I = 2,3$                  Direction 2 and 3 forces.
     J = 7,9$                  List of joint numbers.
              7.2  7.3$        f2, f3 at joint 7.

              8.2  8.3$        f2, f3 at joint 8.

              9.2  9.3$        f2, f3 at joint 9.

CASE 2$
     I = 1: J = 10: 14.6$           f1 at joint 10.

     I = 3: J = 7: 19.2$            f3 at joint 7.
```

---

*<u>Note</u>: When determining the direction of the load, I = 1, 2 or 3 indicates a load acting in the positive 1, 2 or 3 direction. I = 4, 5 or 6 indicates applied moments about the joint.

5.2 DCU: Data Complex Utility

For each processor that is run, one or more data sets are created. These data sets are inserted into the binary data base SPARA. The user may wish to examine these data sets for errors, or to determine how far the program went during execution.

DCU has the ability to list all of the data sets in the order in which they were created. It is also able to print out the contents of any individual data set, such as joint locations: JLOC, stiffness matrix: K SPAR, etc.

The counterpart to DCU is VPRT (VECTOR PRINTER). VPRT is limited in the types of data sets that it can print out, however, the output format from VPRT is much neater than for DCU.

Since DCU can reproduce the contents of any of the data sets, it is an extremely powerful processor.

5.2.1 DCU Commands

The most commonly used command in DCU is the request for a listing of all data sets created, i.e., a table of contents.

The commands are:

    TOC lib

where lib = 1 corresponds to data sets in SPARA;

lib = 2 corresponds to data sets in SPARB, etc.

This command will cause all of the data set names to be listed.

    TOC 1 K SPAR

This command causes one line of the table of contents of Library 1 to be printed. This line will correspond to the first line in the TOC that contains the data set name K SPAR.

    TOC 1 27

This causes line 27 of the table of contents of Library 1 to be printed out.

TOC 1 27, 32

This command prints out lines 27 through 32 of the table of contents of Library 1.

In order to print individual data sets, the command is

PRINT 1 JLOC

This particular command will print the entire contents of the data set JLOC which contains the joint locations.

An example of creating a table of contents is shown in Table 2.

5.2.2 TOC Information Summary

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

TOC ITEM	MEANING
NJ	See above.
NI*NJ	See above.
TY	Type code: 0 = integer, -1 = real, -2 = double precision, 4 = alphanumeric.

5.3 VPRT: Vector Printer

Processor VPRT is used to display certain data sets, such as displacements, eigenvectors and reactions.

To print an allowed data set, the command is:

@RUN VPRT

PRINT N1, N2, n3, n4' (Optional Heading)

For example, to print the static displacements found in data set STAT DISP 1 1:

PRINT STAT DISP 1 1' STATIC DISPLACEMENTS - CASE 1

To print out those data sets containing ALPHA characters:

TPRINT N1, N2, n3, n4

In the output of VPRT, an asterisk (*) adjacent to a number implies that this particular joint component has some constraint case associated with it.

It is possible to be selective if the user wants only portions of a data set to be printed. The following is a list of commands that give VPRT this flexibility.

Note: These are not RESET controls.

Control Statement	Meaning
LIB=m	Library m contains the data to be printed. Default Lib = 1.
VECTORS v_1, v_2	Print only vectors (blocks) v_1 through v_2. Default is all vectors.

Control Statement	Meaning
JOINTS j_1, j_2, j_3, ---	Print data only for joints j_1, j_2, j_3, --- Default is all joints.
COMPONENTS = i_1, i_2, ---	Display only components i_1, i_2, ---, where each i is between 1 and 6. Default is all components.
FILTER = e_1, e_2---e_6	Data for a given joint will be printed only if the absolute value of some component, v_i, exceeds e_i. All six e_i's must be given for this command to be valid. Default $e_i = 1. \times 10^{-20}$ for all six components.
FORMAT = k	Values of k from 1 to 4 select one of the following FORTRAN print out formats: E9.3, F9.5, F9.1 and E15.7, respectively. Only the first three are suitable for a 72-character teletype output. Default FORMAT = 1.
LINES = n	Print n lines per page. Default LINES = 50.
HEADINGS = m	Any non-zero value causes headings to be repeated at the top of each page. Default HEADING = 0.

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

It is now suggested that the user review one of the sample problems that contains VPRT. Notice the summary of constraint conditions and compare with the asterisks in the VPRT output.

VPRT can only be used to print out such data sets as static deformations, reactions, vibrational and buckling eigenvectors, mechanical loading applied to joints and various equivalent joint loads.

6.0 STATIC SOLUTIONS

The normal execution sequence for static solutions is summarized as follows:

1) Sub-processors are used to form tables of applied loading data for the following categories:
   a) Point forces acting at joints;
   b) Moments acting at joints;
   c) Directly specified joint motion components;
   d) Inertial loadings, etc.,
2) SSOL is executed to calculate the static displacements and reactions,
3) GSF is executed to calculate the stresses,
4) PSF is executed to print the stresses,
5) VPRT is executed to print the displacements and reactions.

Figure 16 shows the flow of data during the static solution process.

The following two sections give some examples of the type of data required in the AUS processor for static solution computations.

6.1 Point Forces and Moments Acting on Joints

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

In the following example, Set 20 contains two cases:

@RUN AUS

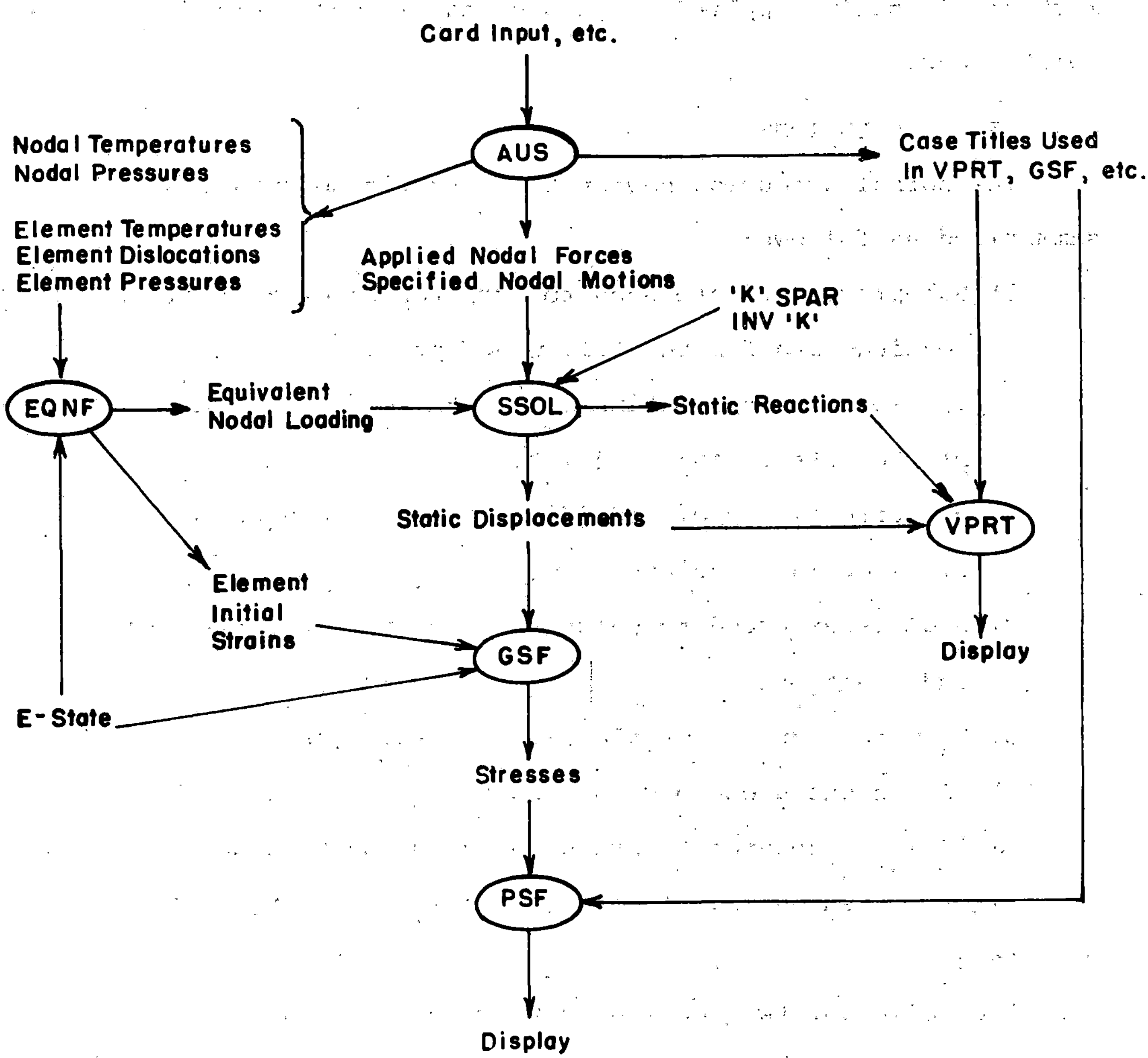


Figure 16. Data Flow Diagram for the Calculation of Static Solutions.

```
SYSVEC
APPLIED FORCES 20
CASE 1
    I = 2:J = 4,10:420.$        Direction 2: Joints 4
                                through 10, 420 lb.

    I = 1:J = 7:3500.$          Direction 1: Joint 7, 3500 lb.

CASE 2
    I = 1,2,3:J = 7,9$
    17.,27.,37.$                Direction 1,2,3: forces at Joint 7.

    18.,28.,38.$                Direction 1,2,3: forces at Joint 8.

    19.,20.,39.$                Direction 1,2,3: forces at Joint 9.
```

6.2 Specified Joint Motions

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

In the following example, Set 14 contains two cases.

```
@RUN AUS
SYSVEC
APPLIED MOTIONS 14
CASE 1
    I = 2:J = 110:-4.2$   Direction 2: displacement of Joint
                          110 is -4.2.
CASE 2
    I = 1:J = 2,8:3.2$    Direction 1: displacement of Joints
                          2 through 8 = 3.2.
```

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

If any of the following cases are present: inertial loadings; nodal temperatures; nodal pressures; thermal loading, defined for individual elements; dislocational loading of individual elements;

pressure loading of individual elements, then processor EQNF must be executed to calculate the equivalent nodal loadings. More information on these cases can be found in Reference [1].

6.3 SSOL: Displacement Data Generator

Processor SSOL is used to calculate static displacements and reactions, due to point loads applied at the nodes.

Output data sets are:

STAT DISP iset ncon

STAT REAC iset ncon

where iset is a load set identifier and ncon is a designated constraint case (see SSOL reset controls).

Components of the STAT REAC iset ncon set, corresponding to constrained or specified joint motion components, are reactions. All other items are residual error forces (F - KU), where F = total applied forces, K = stiffness matrix and U = calculated displacements.

Three items will be printed out during the execution of SSOL for each load case: F*U, U*KU and ERR. F and U are the applied force and computed joint motion vectors, respectively. In reality, F*U corresponds to F^TU, U*KU corresponds to U^TKU and ERR corresponds to $\left| (F^TU - U^TKU)/(\text{the larger of } F^TU \text{ or } U^TKU) \right|$.

ERR should not be taken as a measure of error, if joint motions are specified in APPL MOTI iset.

6.4 GSF: Stress Data Generator

Processor GSF is used to generate data sets containing element stresses and internal load details.

Output data sets are:

STRS Eij iset, icase

where Eij designate an element type, icase and iset have the same meaning as before.

6.4.1 RESET Controls for GSF

The following RESET controls are available in GSF:

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

If a specific data set is to be used as an input to GSF (normally STAT DISP iset icon is used by default), then the statement:

SOURCE = N1 N2 n3 n4 may be used.

This statement names a data set in QLIB that will replace STAT DISP iset icon as an input data set.

The data set entitled: STRS Eij iset icase will be produced for all cases between $icase_1$ and $icase_2$ (defined through RESET Controls by L1, L2). If L1, L2 are not RESET, output data will be supplied for all cases.

It is possible to have the stresses for certain elements as follows:

E21:4,7:10 $	Element E21, Groups 4 thru 7 and Group 10
E33 $	All E33 elements
E43:1:7:10,12 $	E43, Groups 1,7,10,11 and 12
E44:1,5:7,17 $	E44, Groups, 1,2,3,4,5 and 7,8,9,...17

If these types of cards are not present, stresses will be calculated for all of the elements.

6.5 PSF: Stress Table Printer

PSF prints the element stresses and internal load information contained in the STRS Eij iset icase data set produced by GSF. If present, load case titles contained in CASE TITL iset are displayed.

6.5.1 RESET Controls for PSF

Name	Default Value	Meaning
QLIB	1	Data source library for STRS Eij iset, icase and CASE TITL iset.
SET	1	iset (load set identified).
L1	1	$icase_1$
L2	1	$icase_2$ - Default for $icase_2$ is the number of cases (blocks) in CASE TITL iset.
DISPLAY	1	DISPLAY = 1 produces a standard stress printout. DISPLAY = 2 designates the output of end force data for beams, bars, etc., and stress resultants for two-dimensional elements. DISPLAY = 3 produces a detailed stress display for E21 elements and the bending stress resultants for three and four node elements.
NODES	1	For three and four node elements, set NODES = 0 to eliminates the printout of stresses, etc., at the element corners.
CROSS	1	For three and four node elements, set CROSS = 0 to limit the printout to the mid-surface stresses.

Name	Default Value	Meaning
LINES	56	Lines per page.
IEA	1	Set IEA = 0 to cause the run to be aborted if an error occurs (e.g., the designated source data sets do not exist).

6.5.2 PSF Output for a Type E21 General Beam Element and DISPLAY = 1 (Default Value)

See Figure 17 for a sample output.

6.5.3 PSF Output for a Type E21 General Beam Element and DISPLAY = 2

See Figure 18 for a sample output.

6.5.4 PSF Output for a Type E21 General Beam Element and DISPLAY = 3

See Figure 19 for a sample output.

6.5.5 PSF Output for a Type E23 Bar Element with Axial Stiffness

See Figure 20 for a sample output.

6.5.6 PSF Output for a Type E24 Planar Beam Element

See Figure 21 for a sample output.

6.5.7 PSF Output for a Type E41 Shell Element and DISPLAY = 1 (Default Value)

See Figure 22 for a sample output.

6.5.8 PSF Output for a Type E43 Shell Element and DISPLAY = 1 (Default Value)

See Figure 23 for a sample output.

6.5.9 PSF Output for a Type E44 Shear Panel Element and DISPLAY = 1 (Default Value)

See Figure 24 for a Sample Format.

TRANSVERSE SHEAR + TWISTING MOMENT CASE 7- 2

RING STIFFENER, Z=60 ◆GROUP 1

E21 STRESSES, DIVIDED BY 1000.0000

INDEX	CONNECTED JOINTS		MAX COMBINED P/A +BENDING TENSION	COMP	P/A	TRANSVERSE SHEAR STRESS S1	S2	TWIST SHEAR
1	25	26	74.21	-65.42	4.40	.00	.00	.00
2	26	27	67.87	-51.77	3.05	.00	.00	.00
3	27	28	61.20	-57.20	2.00	.00	.00	.00
4	28	29	70.70	-69.08	.81	.00	.00	.00
5	29	30	78.58	-64.64	6.37	.00	.00	.00
6	30	31	63.24	-51.73	5.75	.00	.00	.00
7	31	32	59.68	-67.33	-3.63	.00	.00	.00
8	32	25	61.52	-71.78	-5.13	.00	.00	.00

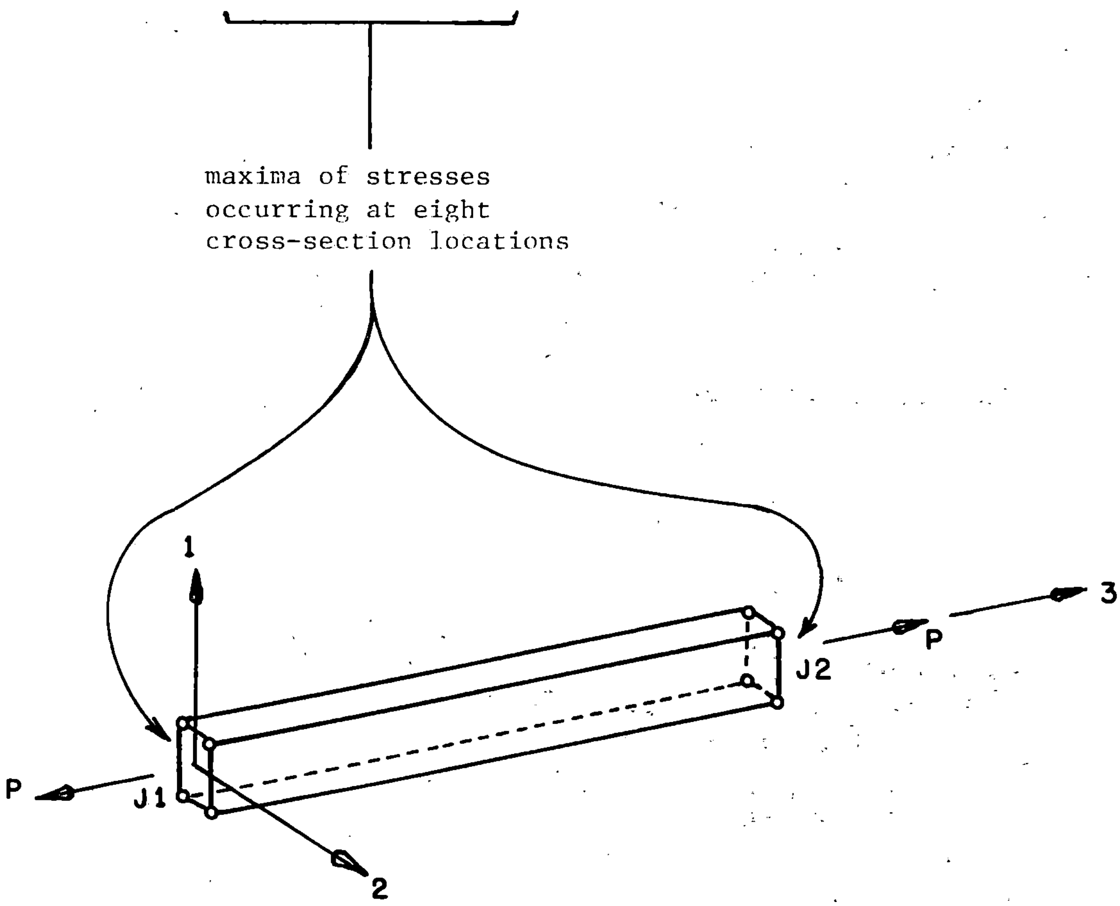


Figure 17. Sample PSF Output for a Type E21 General Beam Element and DISPLAY = 1 (Default Value).

TRANSVERSE SHEAR + TWISTING MOMENT CASE 7- [illegible]

RING STIFFENER, Z=60 *GROUP 1

E21 FORCES, DIVIDED BY 1000.0000

INDEX	JOINT	P1	P2	P3	P4	P5	P6
1	25	-2.89	-5.34	-87.92	26.66	2.45	-.00
	26	2.89	5.34	87.92	185.74	-117.41	.00
2	26	6.97	2.09	-160.99	-120.35	197.67	.00
	27	-6.97	-2.09	160.99	37.35	75.87	-.00
3	27	7.28	2.04	-40.02	34.21	94.38	.00
	28	-7.28	-2.04	40.02	-115.33	205.00	-.00
4	28	-.34	-6.58	-16.26	205.57	-66.48	.00
	29	.34	6.58	16.26	56.46	54.97	.00
5	29	.54	-7.19	-139.45	66.40	68.89	.00
	30	-.54	7.19	139.45	219.68	-47.58	-.00
6	30	7.82	1.66	-115.03	-105.36	315.66	.00
	31	-7.82	-1.66	115.03	39.34	95.60	.00
7	31	5.19	3.29	76.52	15.68	43.56	.00
	32	-5.19	-3.29	-76.52	-145.46	163.06	-.00
8	32	-4.23	-4.39	102.59	154.38	-144.44	.00
	25	4.23	4.39	-102.59	10.4-	-24.06	-.00

Similar output is generated for E22 and E25 elements.

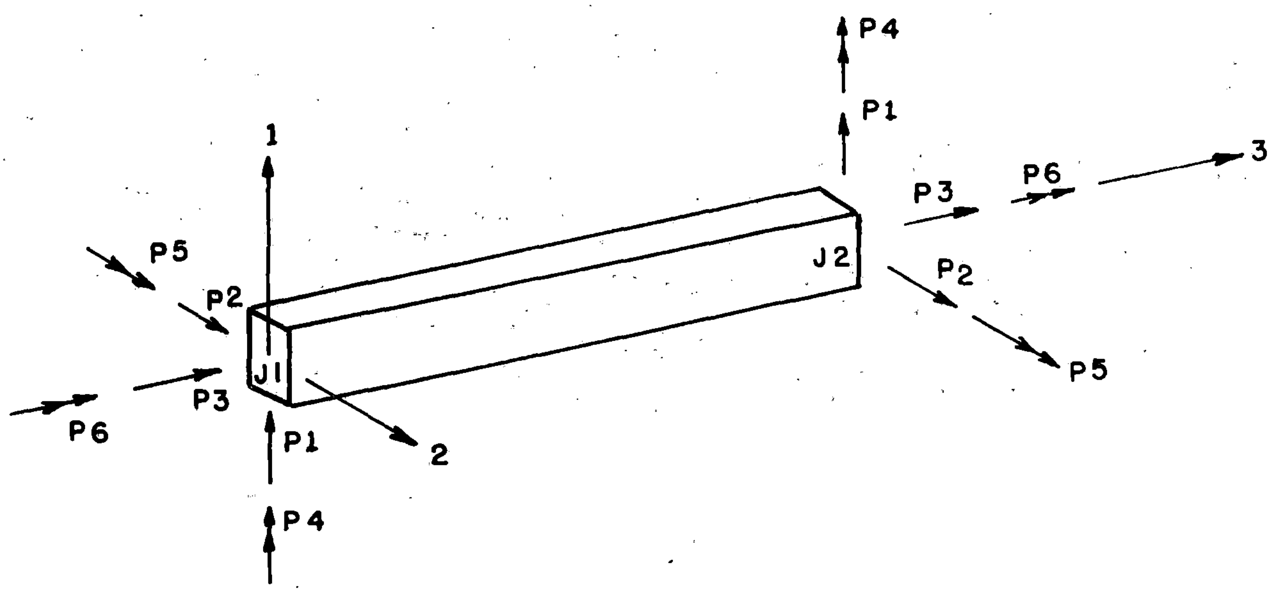


Figure 18. Sample PSF Output for a Type E21 General Beam Element and DISPLAY = 2.

TRANSVERSE SHEAR + TWISTING MOMENT CASE 7- 2

RING STIFFENER, Z=60 +GROUP 1

E21 STRESSES, DIVIDED BY 1000.0000

INDEX	LOC	MY2/I1	MY1/I2	MY2/I1 +MY1/I2	P/A	MY/I'S +P/A
5						
	29-1	-19.92	8.27	-11.65	6.97	-4.68
	29-2	19.92	8.27	28.19	6.97	35.16
	29-3	19.92	-8.27	11.65	6.97	18.62
	29-4	-19.92	-8.27	-28.19	6.97	-21.21
	30-1	65.90	5.71	71.61	6.97	78.58
	30-2	-65.90	5.71	-60.19	6.97	-53.22
	30-3	-65.90	-5.71	-71.61	6.97	-64.64
	30-4	65.90	-5.71	60.19	6.97	67.17
6						
	30-1	31.61	25.88	57.49	5.75	63.24
	30-2	-31.61	25.88	-5.73	5.75	.02
	30-3	-31.61	-25.88	-57.49	5.75	-51.73
	30-4	31.61	-25.88	5.73	5.75	11.48
	31-1	11.80	-11.47	.33	5.75	6.08
	31-2	-11.80	-11.47	-23.27	5.75	-17.52
	31-3	-11.80	11.47	-.33	5.75	5.42
	31-4	11.80	11.47	23.27	5.75	29.02
7						
	31-1	-4.70	5.23	.53	-3.83	-3.30
	31-2	4.70	5.23	9.93	-3.83	6.11
	31-3	4.70	-5.23	-.53	-3.83	-4.35
	31-4	-4.70	-5.23	-9.93	-3.83	-13.76
	32-1	-43.94	-19.57	-63.51	-3.83	-67.33
	32-2	43.94	-19.57	24.37	-3.83	20.54
	32-3	43.94	19.57	63.51	-3.83	59.68
	32-4	-43.94	19.57	-24.37	-3.83	-28.20

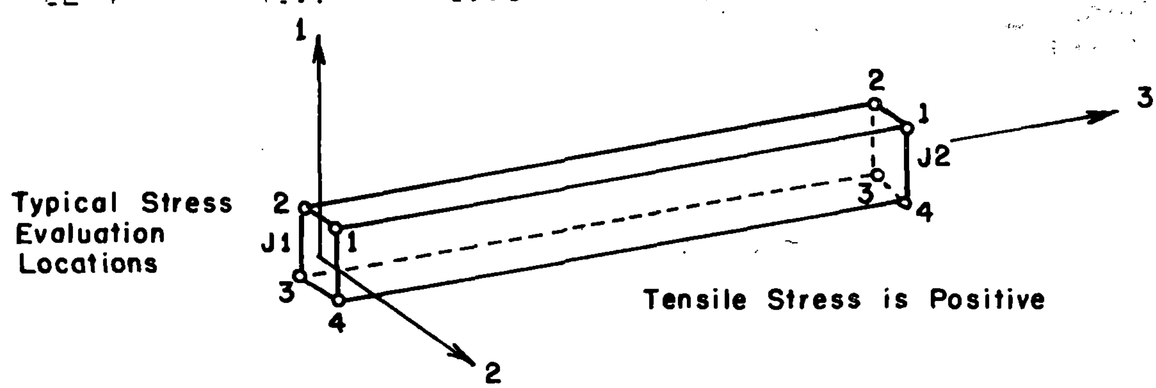


Figure 19. Sample PSF Output for a Type E21 General Beam Element and DISPLAY = 3.

TRANSVERSE SHEAR + TWISTING MOMENT CASE 7- 2

RING STIFFENER, Z=40 *GROUP 1

E23 STRESSES, DIVIDED BY 1000.0000
FORCES, DIVIDED BY 1000.0000

INDEX	CONNECTED JOINTS		FORCE	STRESS
1	17	18	102.97	5.15
2	18	19	135.77	6.79
3	19	20	1.54	.08
4	20	21	-34.74	-1.74
5	21	22	33.85	1.69
6	22	23	-19.22	-.96
7	23	24	-148.53	-7.43
8	24	17	-91.99	-4.60

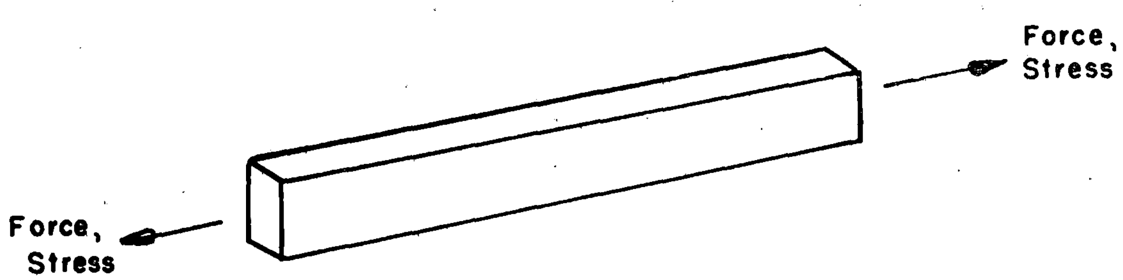


Figure 20. Sample PSF Output for a Type E23 Bar Element with Axial Stiffness.

TRANSVERSE SHEAR + TWISTING MOMENT CASE 7- 2

RING STIFFENER, Z=20 •GROUP 1

E24 STRESSES, DIVIDED BY 1000.0000
FORCES, DIVIDED BY 1000.0000

IND	JOINT	P	V	M	P/A	SS	MC1/I	MH1/I
1	9	-137.75	-4.51	142.49	-6.89	.00	-42.75	42.75
	10	-137.75	-4.51	-147.38	-6.89	.00	44.22	-44.22
2	10	-182.92	-.35	10.05	-9.15	.00	-3.02	3.02
	11	-182.92	-.35	-12.39	-9.15	.00	3.72	-3.72
3	11	-41.65	.42	-3.88	-2.08	.00	1.16	-1.16
	12	-41.65	.42	23.42	-2.08	.00	-7.03	7.03
4	12	5.38	-2.65	93.18	.27	.00	-27.95	27.95
	13	5.38	-2.65	-77.24	.27	.00	23.17	-23.17
5	13	-33.61	-2.70	79.14	-1.68	.00	-23.74	23.74
	14	-33.61	-2.70	-94.16	-1.68	.00	28.25	-28.25
6	14	62.61	.23	-19.00	3.13	.00	5.70	-5.70
	15	62.61	.23	-4.08	3.13	.00	1.22	-1.22
7	15	201.44	-.57	18.38	10.07	.00	-5.51	5.51
	16	201.44	-.57	-18.04	10.07	.00	5.41	-5.41
8	16	103.63	-4.63	153.71	5.18	.00	-46.11	46.11
	9	103.63	-4.63	-144.18	5.18	.00	43.25	-43.25

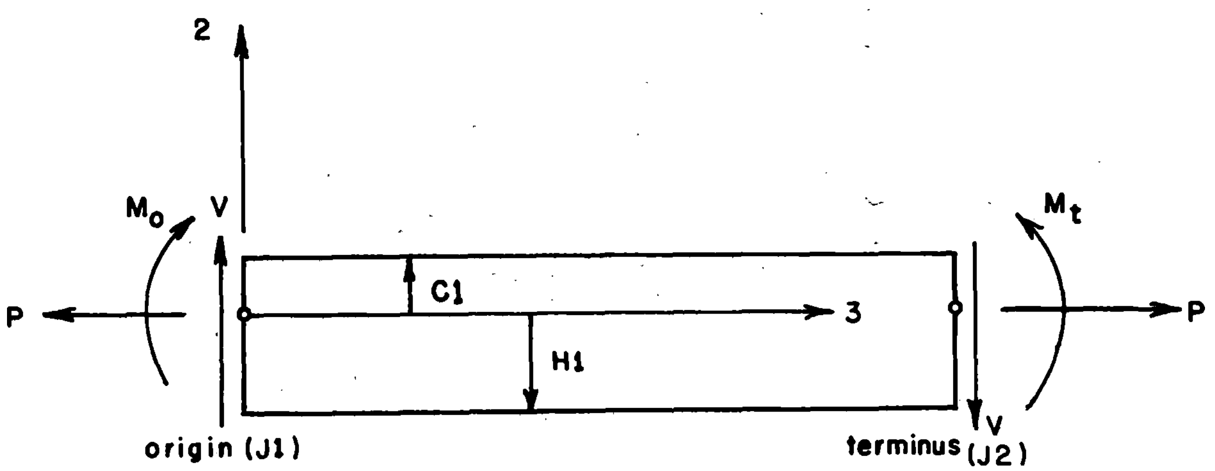


Figure 21. Sample PSF Output for a Type E24 Planar Beam Element.

MAXIMUM COMBINED CONDITION, #810 CASE 7- 1

CONICAL SURFACE, Z=0 TO 20 *GROUP 1

E41 STRESSES, DIVIDED BY 1000.0000

GRP/IND		LOC	SX	SY	TXY	ANG	MAX PS	MIN PS	MAX SHR
1/	1	C	-22.70	-18.12	11.27	51.	-8.91	-31.91	11.50
		90C	8.67	-28.35	11.27	16.	11.83	-31.51	21.67
		170C	8.67	-15.09	11.27	22.	13.17	-19.59	16.38
		180C	-47.47	-8.65	11.27	75.	-5.62	-50.50	22.44
		100C	-60.67	-20.39	11.27	75.	-17.46	-63.61	23.08
1/	2	C	-50.28	-14.35	-6.95	101.	-13.05	-51.58	19.26
		100C	-83.71	-35.02	-6.95	98.	-34.05	-84.69	25.32
		180C	-83.71	-8.23	-6.95	95.	-7.59	-84.35	38.38
		190C	-23.89	4.79	-6.95	103.	6.39	-25.49	15.93
		110C	-9.81	-18.94	-6.95	152.	-6.06	-22.69	8.32
1/	3	C	12.56	15.44	-25.25	133.	39.29	-11.29	25.29
		110C	-13.82	12.30	-25.25	121.	27.67	-29.19	28.43
		190C	-13.82	16.37	-25.25	120.	30.70	-28.14	29.42
		200C	33.38	18.25	-25.25	143.	52.92	-.48	26.35
		120C	44.49	14.74	-25.25	130.	58.93	.31	29.31
1/	4	C	33.48	17.04	-17.88	147.	44.94	5.58	19.68
		120C	66.50	24.20	-17.88	150.	73.04	17.65	27.69
		200C	66.50	14.92	-17.88	163.	72.09	9.33	31.38
		210C	7.41	10.41	-17.88	133.	26.85	-9.03	17.94
		130C	-6.49	18.63	-17.88	117.	27.92	-15.78	21.85

Similar output is generated for E31 elements, default display.

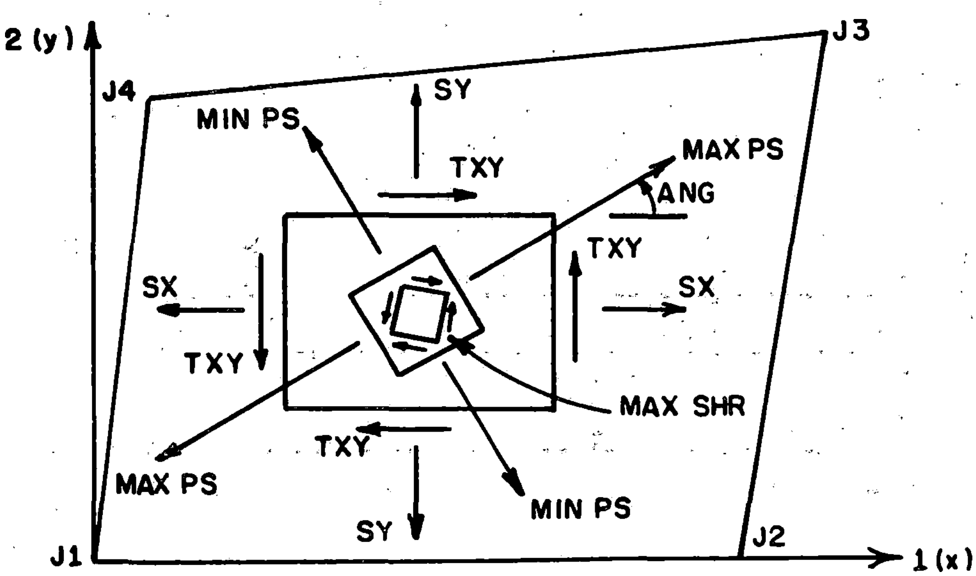


Figure 22. Sample PSF Output for a Type E41 Shell Element and DISPLAY = 1 (Default Value).

MAXIMUM COMBINED CONDITION, #810 CASE 7- 1

CONICAL SURFACE, Z=40 TO 80 *GROUP 1

E43 STRESSES, DIVIDED BY 1000.0000

GRP/IND	LOC	SX	SY	TXY	ANG	MAX PS	MIN PS	MAX SHR
1/ 4	C	38.64	16.21	-20.10	150.	50.44	4.41	23.02
	A	44.49	27.27	-14.43	150.	52.68	19.08	16.80
	B	32.79	5.15	-25.77	149.	48.21	-10.27	29.24
	20C	68.91	13.12	-20.10	162.	75.40	6.63	34.39
	A	47.71	19.27	-17.25	155.	55.84	11.13	22.36
	B	90.12	6.97	-22.95	156.	96.03	1.06	47.49
	29C	68.91	17.44	-20.10	161.	75.83	10.53	32.65
	A	77.17	39.97	-9.88	166.	79.64	37.51	21.06
	B	60.65	-5.09	-30.32	159.	72.50	-16.93	44.72
	29C	16.44	19.05	-20.10	133.	37.89	-2.40	20.14
	A	26.08	16.87	-11.38	146.	33.75	9.19	12.28
	B	6.30	21.23	-28.82	128.	43.72	-15.69	29.71
	21C	.29	15.22	-20.10	125.	29.20	-13.69	21.44
	A	16.10	22.60	-19.21	130.	38.83	-.13	19.48
	B	-15.51	7.84	-20.99	120.	20.16	-27.86	24.02
1/ 5	C	-13.66	10.63	-7.99	107.	13.09	-16.07	14.54
	A	-12.87	-8.11	-6.95	116.	1.30	-16.28	8.79
	B	-14.44	23.37	-9.04	103.	25.42	-16.49	20.96
	21C	15.70	3.51	-7.99	154.	19.69	-.44	10.07
	A	-3.92	-10.60	-4.04	155.	-2.02	-12.50	5.24
	B	35.41	17.61	-11.94	153.	41.41	11.62	14.89
	29C	15.75	13.48	-7.99	139.	22.69	6.54	8.07
	A	15.67	2.94	2.46	11.	16.12	2.48	6.82
	B	15.82	24.02	-13.44	129.	36.82	1.03	13.89
	30C	-35.22	17.19	-7.99	98.	18.38	-36.41	27.40
	A	-50.62	-14.94	-8.21	102.	-13.15	-52.42	19.63
	B	-19.83	49.32	-7.78	96.	50.18	-20.69	35.44
	22C	-50.91	8.35	-7.99	98.	9.41	-51.96	30.69
	A	-31.09	2.80	-17.99	113.	10.58	-38.32	24.76
	B	-70.76	13.91	2.01	39.	13.95	-70.81	42.38

Similar output is generated for E33, E42, and E32 elements, default display.

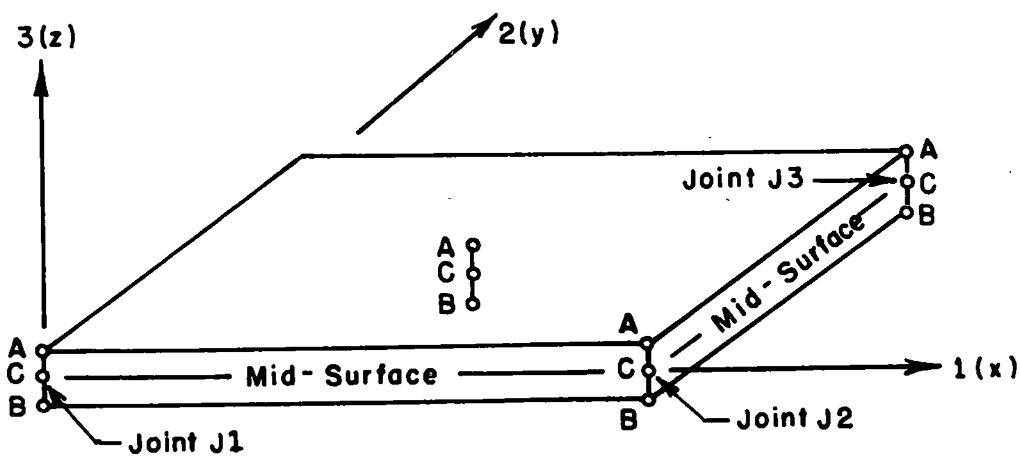


Figure 23. Sample PSF Output for a Type E43 Shell Element and DISPLAY = 1 (Default Value).

MAXIMUM COMBINED CONDITION, #810 CASE 7- 1

CONICAL SURFACE, Z=60 TO 80 ◆GROUP 1

E44 STRESSES, DIVIDED BY 1000.0000

INDEX	CONNECTED JOINTS				THICKNESS	STRESS	SHEAR FLOW
1	25	33	34	26	.87	19.47	16.8[illegible]
2	26	34	35	27	.87	-31.61	-27.40
3	27	35	36	28	.87	-60.16	-52.15
4	28	36	37	29	.87	-43.79	-37.97
5	29	37	38	30	.87	-40.87	-35.44
6	30	38	39	31	.87	-57.51	-49.26
7	31	39	40	32	.87	-36.44	-31.60
8	32	40	33	25	.87	15.65	13.56

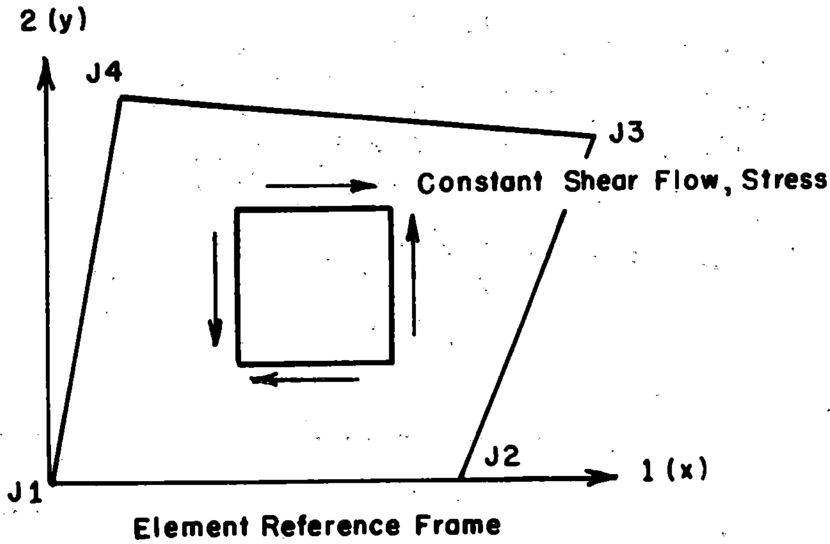


Figure 24. Sample PSF Output for a Type E44 Shear Panel Element and DISPLAY = 1 (Default Value).

7.0 REFERENCES

1. Whetstone, W.D., "SPAR Structural Analysis System Reference Manual - System Level II, Volume I - Program Execution," NASA CR-145096-1, Engineering Information Systems, Inc., San Jose, California, 1977.
2. Whetstone, W.D., "Computer Analysis of Large Linear Frames," Journal of the Structural Division, ASCE, Volume 95, No. ST10, November, 1979, pp. 2401 - 2417.
3. Whetstone, W.D., and Jones, C.D., "Vibrational Characteristics of Linear Space Frames," Journal of the Structural Division, ASCE, Volume 95, No. ST10, October, 1969, pp. 2077 - 2091.
4. Giles, G.L., and Haftka, R.T., "SPAR Data Handling Utilities," NASA Technical Memorandum TM 78701, NASA Langley Research Center, Hampton, Va., September, 1978.
5. Barone, T.R., and Feeser, L.J., "Beginner's User Manual for SPAR," Rensselaer Polytechnic Institute, Department of Civil Engineering Report No. 78-1, Troy, New York, May, 1978.
6. Storaasli, O.O., and Foster, E.P., "Cost Effective Use of Minicomputers to Solve Structural Problems," AIAA Paper No. 78-484, April, 1978.
7. Prasad, B., and Haftka, R.T., "PARS-Program for Analysis and Resizing of Structures-User's Manual," NASA CR-159007, Illinois Institute of Technology, Chicago, Illinois, April, 1979.

8.0 APPENDICES

8.1 Appendix A

Summary List* of SPAR Processor Commands

* NOTE: Only frequently used RESET Controls, etc., are included in this Summary. See Reference [1] for more complete information.

A.1 TAB

START		number of joints; list of excluded joint motion components.
UPDATE	=	1 to enter update mode, 0 to leave.
FORMAT	=	i, to select alternate format i.
NREF	=	n, to identify reference frame n.
MOD	=	m, to modify table entry numbers or joint numbers by m.
		After each sub-processor execution, TAB returns automatically to FORMAT=1, NREF=1, and MOD=0.

A.2 TAB Sub-processors

TEXT:	4/8 - - - 76 characters of text.
MATC:	k, E, Nu, Rho, $Alpha_1$, $Alpha_2$, Theta
NSW:	k, weight/(area or length).
ALTREF:	k, $i_1, a_1, i_2, a_2, i_3, a_3, x_1, x_2, x_3$
JLOC:	k, xa_1, xa_2, xa_3, xb_1, xb_2, xb_3, ni, ijump, nj.
	jjump, xc_1, xc_2, xc_3, xd_1, xd_2, xd_3.
	Formats 1, 2 = rectangular, cylindrical.
	NREF = reference frame.
JREF:	NREF=n: j_1, j_2, inc: k_1, k_2, inc: - - - -
MREF:	k, nb, ng, isign, c (FORMAT 1)
	k, i_1, x_1, x_2, x_3 (FORMAT 2)
	NREF = n allowed under Format 2.
BRL:	k, i^1, x_1^1, x_2^1, x_3^1, i^2, x_1^2, x_2^2, x_3^2.

A.2 TAB Sub-processors Cont'd.

BA: BOX, k, b_1,t_1, b_2,t_2.

TEE, k, b_1,t_1, b_2,t_2.

ANG, k, b_1,t_1, b_2,t_2.

WFL, k, b_1,t_1, b_2,t_2, b_3,t_3.

CHN, k, b_1,t_1, b_2,t_2, b_3,t_3.

ZEE, k, b_1,t_1, b_2,t_2, b_3,t_3.

TUBE, k, inner r, outer r.

GIVN, k, I_1, α_1, I_2, α_2, a, f, f_1, z_1, z_2, θ.

DSY, k, I_1, α_1, I_2, α_2, a, f, f_1. (card 1)

q_1, q_2, q_3, y_{11}, y_{12}, - - - y_{41}. (card 2)

BB: k, s_{11}.

s_{21}, s_{22}.

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

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

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

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

BC: k, a.

BD: k, a, I_1, α_1, h, c, q_1.

A.2 TAB Sub-processors Cont'd.

SA:

FORMAT:	Input data list, section k		Applicable Element types
ISOTROPIC:	k, t \$ t= thickness		All
MEMBRANE:	k, f_{11} f_{21} f_{31} \$	*Record 1*	E31, E41, only
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} \$	*Record 2*	
PLATE:	k, f_{42} f_{52} f_{62} f_{43} f_{53} f_{63} \$	*Record 1*	E32, E42, only
	c_{44} c_{45} c_{55} c_{46} c_{56} c_{66} \$	*Record 2*	
UNCOUPLED:	k, f_{11} f_{21} f_{31} >		All
	f_{42} f_{52} f_{62} f_{43} f_{53} f_{63} \$	*Record 1*	
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} \$	*Record 2*	
	c_{44} c_{45} c_{55} c_{46} c_{56} c_{66} \$	*Record 3*	
COUPLED:	k, f_{11} f_{21} f_{31} f_{41} f_{51} f_{61} >		E33, E43, only
	f_{12} f_{22} f_{32} f_{42} f_{52} f_{62} >		
	f_{13} f_{23} f_{33} f_{43} f_{53} f_{63} \$	*Record 1*	
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} c_{14} >		
	c_{24} c_{34} c_{44} c_{15} c_{25} c_{35} c_{45} >		
	c_{55} c_{16} c_{26} c_{36} c_{46} c_{56} c_{66} \$	*Record 2*	
LAMINATE:	k \$ *Record 1. Additional records, one for each layer, are given in the following form:*		E33, E43, only
	z^i θ^i t^i > \$	*θ is in degrees*	
	c^i_{11} c^i_{12} c^i_{22} c^i_{13} c^i_{23} c^i_{33} >		
	c^i_{44} c^i_{45} c^i_{55} c^i_{46} c^i_{56} c^i_{66} \$	*Record 1 + i*	

SB: k, t.

A.3 <u>CON n: (constraint case n)</u>

ZERO $m_1, m_2,$ - -:j_1^1, j_2^1, inc^1:j_1^2, j_2^2, inc^2: - - - -:

NONZERO (same as above)

RELEASE (same as above)

SYMMETRY PLANE = n

ANTISYMMETRY PLANE = n

FIXED PLANE = n

LZERO= x.

RZERO = r.

A.4 <u>REPEAT n^1, inc^1, $jmod^1$:</u>

j_1/j_2, j_3, j_4/j_5 - -:

j_6, j_7, j_8/j_9 - - -:

REPEAT n^2, inc^2, $jmod^2$:

j_{10}, j_{11}, j_{12}/j_{13} - - -:

etc.

A.5 <u>RMASS</u>: k, M, I_1, I_2, I_3. (Format 1)

k, M_1, M_2, M_3, I_1, I_2, I_3. (Format 2)

REPEAT n, jump.

CM = (M multiplier), (I multiplier).

A.6 <u>ELD</u>

Sub-processor Name	Element type
E21	General beam
E22	Finite length, directly specified K
E23	Axial element
E24	Plane beam
E25	Zero-length, directly specified K

A.6 ELD Cont'd.

Sub-processor Name	Element type
E31	Triangular membrane
E32	Triangular bending element
E33	Triangular membrane + bending element
E41	Quadrilateral membrane
E42	Quadrilateral bending element
E43	Quadrilateral membrane + bending element
E44	Quadrilateral shear panel

Pointer Name	Default Value	Associated Table
NMAT	1	Material constants (MATC)
NSECT	1	Section properties (BA - - - - - SB)
NOFF	0	Rigid link offsets (BRL)
NNSW	0	Nonstructural weight (NSW)
NREF	1	Beam orientation (MREF). Also indicates pressure sense for 3 and 4 node elements.

Data Modifiers:

MOD JOINT NUMBERS n, or MOD JOINT=n

MOD GROUP NUMBERS n, or MOD GROUP=n

MOD NMAT (or NSECT, etc.)=n

INC NMAT (or NSECT, etc.)=n

All MOD and INC parameters are reset to zero upon conclusion of each sub-processor execution.

A.6 <u>ELD Cont'd.</u>

<u>2-node elements:</u>

j_1, j_2

j_1, j_2, 1, ni, nj, jinc. (nj grid lines)

j_1, j_2, 2, ni, nj, jinc. (nj rings)

j_1, j_2, 3, ni, iinc, nj, jinc. (nj sets of spokes)

<u>3-node elements:</u>

j_1, j_2, j_3

j_1, j_2, j_3, 1, ni, nj. (rectangular mesh)

j_1, j_2, j_3, 2, ni, nj, jinc. (nj 'open fans')

j_1, j_2, j_3, 3, ni, nj, jinc. (nj 'closed fans')

<u>4-node elements</u>

j_1, j_2, j_3, j_4

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

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

A.7 AUS/TABLE

TABLE(NI= ni, NJ= nj): N1 N2 n3 n4 $ = Output name

CASE n$ or BLOCK n

OPERATION= SUMS, or XSUM, or MULT, or DIVIDE

I= i_1, i_2,... i_m

DDATA= d_1, d_2,... d_m

J= j_1:j_2:j_3:.. $j_r$$ Loop limit format also allowed.

$e_{j_1}^{i^1}$, $e_{j_1}^{i^2}$ ... $e_{j_1}^{i^m}$$ Data for j= j_1 .

$e_{j_2}^{i^1}$, $e_{j_2}^{i^2}$ ... $e_{j_2}^{i^m}$$ Data for j= j_2 .

. . .

. . .

. . .

$e_{j_r}^{i^1}$, $e_{j_r}^{i^2}$ ... $e_{j_r}^{i^m}$$ Data for j= j_r .

A.8 AUS/SYSVEC

Same as AUS/TABLE, except that NI and NJ are automatically fixed by SYSVEC.

A.9 AUS/ELDATA

ELDATA: N1 N2 n3$ 4th word not given here

CASE n4$ Results in output data set N1 N2 n3 n4.

Other input is the same as in TABLE, except that

J= j_1: j_2, --- is replaced by

G= g_1: g_2:----$ List of groups

E= e_1: e_2:----$ List of element indexes

A.10 AUS/ALPHA

```
ALPHA: N1 N2 n3 n4
1'Title string 1
2'Title string 2
  etc.
```

8.2 Appendix B

Shell Section Properties

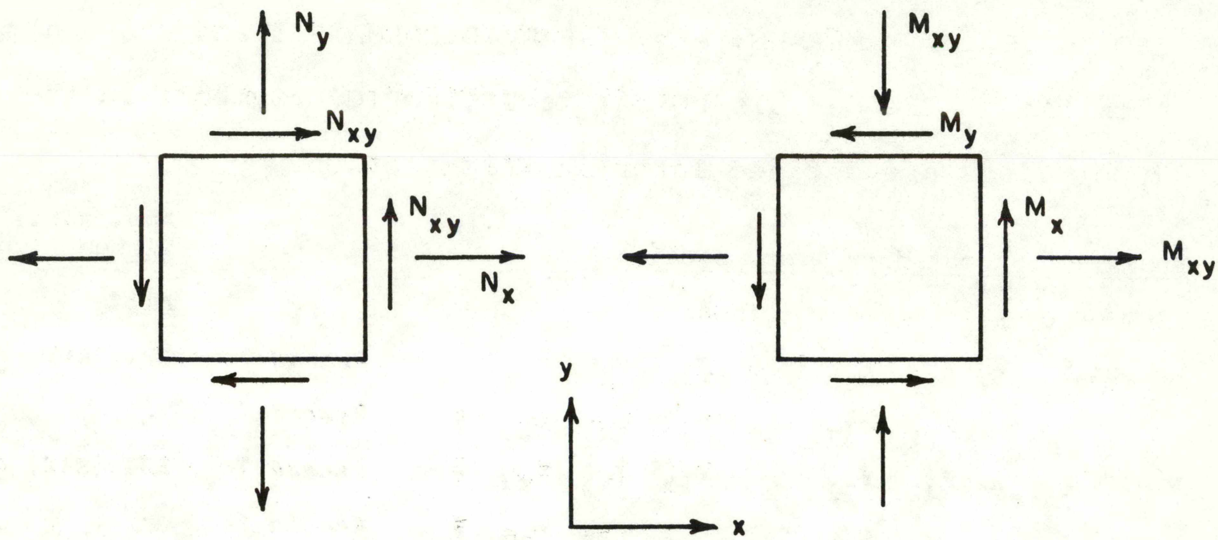
B.1 Formats

Processor SA generates a table of shell section properties, to which reference is made during the definition of Element Types E31, E32, E33, E41, E42 and E43 in processor ELD. The user may select any of the formats indicated below, by setting FORMAT= ISOTROPIC, or MEMBRANE, etc. SPAR/10 FORMATS 1 and 2 are also permitted. The default FORMAT is ISOTROPIC. Symbols in the input lists are defined later in this Section.

FORMAT:	Input data list, section k		Applicable Element types
ISOTROPIC:	k, t \$ t= thickness		All
MEMBRANE:	k, f_{11} f_{21} f_{31} \$	*Record 1*	E31, E41, only
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} \$	*Record 2*	
PLATE:	k, f_{42} f_{52} f_{62} f_{43} f_{53} f_{63} \$	*Record 1*	E32, E42, only
	c_{44} c_{45} c_{55} c_{46} c_{56} c_{66} \$	*Record 2*	
UNCOUPLED:	k, f_{11} f_{21} f_{31} >		All
	f_{42} f_{52} f_{62} f_{43} f_{53} f_{63} \$	*Record 1*	
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} \$	*Record 2*	
	c_{44} c_{45} c_{55} c_{46} c_{56} c_{66} \$	*Record 3*	
COUPLED:	k, f_{11} f_{21} f_{31} f_{41} f_{51} f_{61} >		E33, E43, only
	f_{12} f_{22} f_{32} f_{42} f_{52} f_{62} >		
	f_{13} f_{23} f_{33} f_{43} f_{53} f_{63} \$	*Record 1*	
	c_{11} c_{12} c_{22} c_{13} c_{23} c_{33} c_{14} >		
	c_{24} c_{34} c_{44} c_{15} c_{25} c_{35} c_{45} >		
	c_{55} c_{16} c_{26} c_{36} c_{46} c_{56} c_{66} \$	*Record 2*	
LAMINATE:	k \$ *Record 1. Additional records, one for each layer, are given in the following form:*		E33, E43, only
	z^i θ^i t^i > \$	*θ is in degrees*	
	c^i_{11} c^i_{12} c^i_{22} c^i_{13} c^i_{23} c^i_{33} >		
	c^i_{44} c^i_{45} c^i_{55} c^i_{46} c^i_{56} c^i_{66} \$	*Record 1 + i*	

B.1 Formats Cont'd.

The sign convention for stress resultants is shown below. The x and y axes shown are those of an element reference frame (see ELD discussion).



Membrane stress resultants　　　　Bending stress resultants

The c_{ij}'s appearing in the input data lists are defined, as follows: Where e_x, e_y and γ_{xy} are membrane strains, and where $k_x = -\frac{\partial^2 w}{\partial x^2}$, $k_y = -\frac{\partial^2 w}{\partial y^2}$, and $k_{xy} = 2\frac{\partial^2 w}{\partial x \quad \partial y}$,

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & & & & c_{55} & c_{56} \\ \text{Symmetric} & & & & & c_{66} \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ \gamma_{xy} \\ k_x \\ k_y \\ k_{xy} \end{bmatrix}$$

B.1 Formats Cont'd.

The f_{ij}'s are stress coefficients, defined as follows:

$$\sigma_x = f_{1j} N_x + f_{4j} M_x$$
$$\sigma_y = f_{2j} N_y + f_{5j} M_y$$
$$\tau_{xy} = f_{3j} N_{xy} + f_{6j} M_{xy}$$

The index j indicates cross-sectional location. In the stress printout produced by processor PSF, locations C, A, and B correspond, in order, to j=1, 2, and 3. For ISOTROPIC sections, j=1 at z=0 (the mid-surface), j=2 at z=t/2, j=3 at z=t/2.

For MEMBRANE, PLATE, COUPLED, AND UNCOUPLED sections, it is permissible to omit trailing zeroes from any input record. For the MEMBRANE, PLATE, and UNCOUPLED sections, all c_{ij}'s and f_{ij}'s, other than those named in the input lists, are zero, with the exception that for MEMBRANE and UNCOUPLED sections: $f_{ij} = f_{i1}$, for j = 2 and 3.

The material properties of the k-th section are contained in the nmat-th entry of the MATERIAL CONSTANTS table produced by TAB/MATC. The value of the pointer nmat is controlled by the following command:

NMAT = nmat (default = 1).

For example,

```
SA
1,   .10 $        Sections 1 and 2 are made of Material 1 (default)
2,   .25 $
NMAT = 3
3,   .15 $        Sections 3 and 4 are made of Material 3
4,   .05 $
NMAT = 2
5,   .33 $        Section 5 is made of Material 2
```

The material of which each section/element is composed must be identified by the NMAT parameter <u>in both TAB/SA and in ELD</u>.

B.2 <u>ISOTROPIC Sections</u>

For istropic sections, SA computes the c_{ij}'s and f_{ij}'s as indicated below, using material constants E, ν, and ρ, from entry nmat of the MATC table:

B.2 ISOTROPIC Sections Cont'd.

$c_{11} = c_{22} = Et/(1 - \nu^2)$

$c_{33} = c_{11}(1 - \nu)/2$

$c_{12} = \nu c_{11}$

$f_{ij} = 1/t$ for i & $j = 1,2,3$

structural weight/area $= \rho t$

$c_{44} = c_{55} = Et^3/12(1 - \nu^2)$

$c_{66} = c_{44}(1 - \nu)/2$

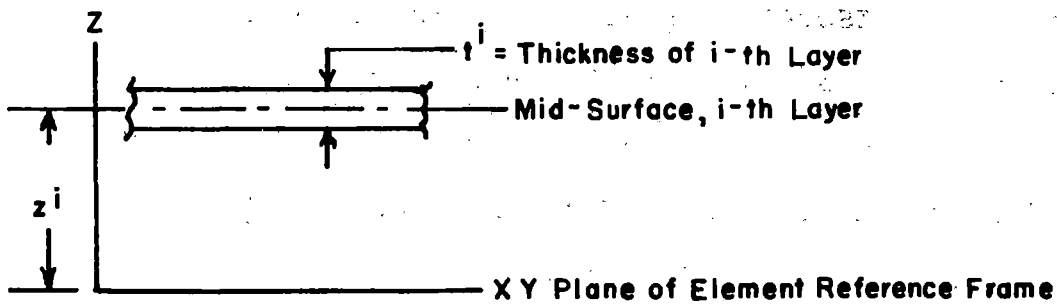
$c_{45} = \nu c_{44}$

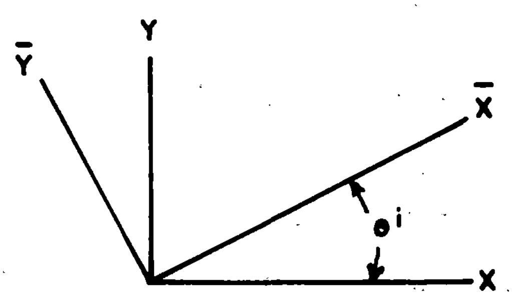
$f_{42} = f_{52} = -f_{62} = 6/t^2$

$f_{43} = f_{53} = -f_{63} = -6/t^2$

B.3 LAMINATE Sections

If the shell section property table is to contain any LAMINATE section data, TAB/SA must be entered via the command SA(maxnl), where maxnl is the greatest number of layers contained in any LAMINATE section. Do not arbitrarily set maxnl to a large number, since this parameter controls the dimensions of the output data set. The parameters z^i, θ^i, and t^i appearing in the LAMINATE input data list are defined below.





X And Y Are Axes Of The Element Reference Frame. $\bar{X}$ And $\bar{Y}$ Are Principal Axes Of The i-th Layer, Relative To Which Are Defined The Stiffness Coefficients c^i_{11}, c^i_{12}, etc.

B.3 LAMINATE Sections Cont'd.

If the c^i_{pq}'s of a layer are the same as those of the proceeding layer, the input for the layer may be abbreviated to z^i, θ^i, t^i. If the layer has negligible bending stiffness, the six words c^i_{44} - -c^i_{66} may be omitted.

The thermal expansion coefficients (α_1, α_2, θ, defined in TAB/MATC and referred to in SA via the NMAT parameter) apply to cross-sections in total, not to individual layers. If thermal stresses in individual layers are to be computed, all layers in each LAMINATE section should have the same θ, α_x, and α_y. This will usually necessitate the use of several elements to model a total cross-section.

B.4 Structural Weight Per Unit Area

SA computes structural weight/area = thickness x weight density for ISOTROPIC sections. For all other section types, the following input is used: W(i): w_i, w_{i+1}, w_{i+2}, - - -(w_i is weight/area for section i). For example, the following indicates that the weight/area for Sections 5, 6, 7 and 11 is .02, .03, .025 and .028, respectively:

W(5): .02, .03, .025: W(11): .028 $.

B.5 INVM and INVB Commands (affecting only MEMBRANE, PLATE, UNCOUPLED and COUPLED)

The parameters invm and invb are controlled by the commands INVM = invm, and INVB= invb. Default for invm and invb is zero. If invm is not zero, flexibility (rather than stiffness) coefficients are inputted for the membrane terms of the MEMBRANE and UNCOUPLED sections, and flexibility coefficients are inputted for the COUPLED sections. If invb is not zero, flexibility coefficients are inputted for the bending terms of the PLATE and UNCOUPLED sections.

B.6 Error Codes

The following is a list of error codes produced in SA:

Code	Meaning
W1 or W3	Structural weight data given for section not previously defined
W2	Structural weight data not in real form
	The following apply only to record 1:
W4	k not in integer form
W5	Trailing data not in real form
	The following apply only to records 2 and 3:
D1	Empty reord
D2	Too many words in record
D3	Non-real data in record
	The following apply only to LAMINATE section layer inputs:
C1	Record contains other than 3, 9, or 15 words
C2	No layers defined, or singular stiffness matrix
C3	Number of layers exceeds maxnl

B.7 Output Data Set Contents

The content of each line of the output data set is as indicated below. The d_{ij}'s are flexibility coefficients, corresponding to the c_{ij}'s on Page 104.

B.7 Output Data Set Contents Cont'd.

Location	Contents
1	ist. The value is 3, 1, 2, 3, 4, or 5, corresponding to the section types: ISOTROPIC, MEMBRANE, PLATE, UNCOUPLED, COUPLED and LAMINATE, respectively.
2	nmat, the applicable entry in the MATERIAL CONSTANTS table.
3	structural weight/area.

Location	For other than COUPLED or LAMINATE sections:	Location	For COUPLED and LAMINATE sections:
4 - 9	d_{11} d_{12} d_{22} d_{13} d_{23} d_{33}	4 - 24	d_{11} d_{12} d_{22} - - d_{66}
10 - 15	d_{44} d_{45} d_{55} d_{46} d_{56} d_{66}		
16 - 25	not used	25	no. of layers
	For other than LAMINATE:		For LAMINATE:
26 - 43	f_{11} f_{21} f_{31} f_{41} f_{51} f_{61}	26 - 43	g^1_{11} g^1_{21} g^1_{31} g^1_{12} g^1_{22} g^1_{32}
	f_{12} f_{22} f_{32} f_{42} f_{52} f_{62}		g^1_{13} g^1_{23} g^1_{33} g^1_{14} g^1_{24} g^1_{34}
	f_{13} f_{23} f_{33} f_{43} f_{53} f_{63}		g^1_{15} g^1_{25} g^1_{35} g^1_{16} g^1_{26} g^1_{36}
		44 - 61	g^2_{11} - - - - g^2_{36}
		62 - 79	g^3_{11} - - - - g^3_{36}
		80 - ?	Eighteen additional terms for each successive layer.

The g^i_{pq}'s are layer stress recovery coefficients. Omitting the superscript i, indicating the i-th layer:

B.7 Output Data Set Contents Cont'd.

$$\begin{bmatrix} \sigma_{\bar{x}} \\ \sigma_{\bar{y}} \\ \tau_{\overline{xy}} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} & g_{15} & g_{16} \\ g_{21} & g_{22} & g_{23} & g_{24} & g_{25} & g_{26} \\ g_{31} & g_{32} & g_{33} & g_{34} & g_{35} & g_{36} \end{bmatrix} \begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix}$$

User's Manual: Spar 10, Structural Analysis, Finite Element Program, For Use on a DEC-2050 Computer System, R Prasad, Som P Singh, 1981, 2 Train Control & Communications

SMEAD OO WP535A