A User Friendly Preprocessor for Plane and Space Frames and Space Trusses

by

James Christopher Pugh

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Civil Engineering

APPROVED:

Siegfried M. Holzer, Chairman

Thomas M. Murray

William S. Easterling

July 26, 1989

Blacksburg, Virginia

A User Friendly Preprocessor for Plane and Space Frames and Space Trusses

by

James Christopher Pugh

(ABSTRACT)

A user friendly preprocessor was developed and documented for the plane and space frame and space truss structural analysis programs that are based on the matrix displacements method. This preprocessor is comprised of three programs. The main program in the preprocessor is to allow the user to create error free input data files. This program also allows modifications of existing input data files. The two other programs are the library manager and the graphics presentation. The library manager is used to manage the libraries of the element and material properties. The graphics presentation is used to display a plane structure on the graphics display. In Chapter 2, the development of a user friendly preprocessor is discussed. After a short review of the extension of the analysis program from plane frame to space frame in Chapter 3, the preprocessor and its supporting programs are described in detail in the user manual in Chpter 4. Possible extensions to the preprocessor are discussed in Chapter 5. The appendix contains examples of input data files for these structural analysis programs.

To my parents

Acknowledgements

The author wants to thank professor Siegfried M. Holzer for serving as the author's major adviser and for his wisdom, guidance, and the time he spent proof reading this masters thesis.

The author would also like to give thanks to professor Thomas M. Murray and William S. Easterling for their helpful comments and encouraging comments and for serving on the author's committee.

The author would proudly dedicate this work to his parents for their encouragement and continuing love who have always supported me in my efforts.

Table of Contents

Listing of Figures

Listing of Tables

Chapter 1

Introduction

1.1 Objectives

This thesis is concerned with the development and documentation of a user friendly preprocessor for the structural analysis programs of plane and space frames and space trusses that are based on the matrix displacement method (Holzer, 1985). The main objective of this preprocessor is to provide a way of creating error free input data files for these structural analysis programs. The preprocessor is also to provide a way of editing existing input data files. Another objective is to provide the user with a program to manage the libraries the preprocessor requires to execute. These libraries include the material and element properties

that are used to develop an input data file. The last objective was to provide a graphics program that would draw a plane structure on the graphics display which would allow the user to double check the input data.

1.2 Overview

In Chapter 2, the development of a user friendly preprocessor is given. This chapter deals with the selection of a user interface software package and a programming language. The preprocessor and its supporting programs (library manager, and graphics presentation) are shown in tree chart form. In chapter 3, a short review of the extension of the analysis program from plane frame to space frame is given. This short review addresses the additional information required to define a space frame. In the latter part of this chapter, a short review of the space truss is given. The organization of these programs are illustrated in tree chart form. Chapter 4 deals with the user manual for the preprocessor and its supporting programs. The logical format of the data routines used in the development of an input data file are illustrated and explained in detail. This user manual also explains in detail the functions of the supporting programs: the library manager,

and the graphics presentation. In Chapter 5, a short summary of possible extensions for the preprocessor is given.

Examples of input data files will be presented in Appendix A.

Chapter 2

Development of a User Friendly Preprocessor

2.1 Introduction

As stated earlier, the main objective of this thesis is to develop a preprocessor that can create the input data file for the analysis of plane and space frames and space trusses. This preprocessor was employed because it offers the user conveniences in entering and editing of data, and in the checking for errors in the input data.

To make the preprocessor convenient for users, an interface software package (Section 2.2) was chosen because it allows the creation of data entry screens (Section 2.2.1), pop-up and slug menus (Section 2.2.2), and pop-up and prompt windows (Section 2.2.3). This software package gives the user full control over entering and editing data with full error

checking capabilities. When it is linked with a programming language (Section 2.3), a user friendly preprocessor can be generated. In Section 2.4, the preprocessor and its supporting programs are shown in tree chart form.

2.2 Selecting the User Interface Software

Menus, input screens, and windows are becoming fashionable in commercial software. In addition, they look elegant, and can improve a program's appearance (Olsen, 1988). The appearance and user interaction of the preprocessor plays a large part in determining if it is user friendly. In the article *QuickWindows* Olsen (1988) states that

> "Programs that are not user friendly are less likely to be purchased or used, so programmers need to put a great deal of effort into designing a user interface that is simple, easy to use, and attractive."

A user interface consists of tools to allow the creation of menus, input screens, and windows; thus, the programmer has a choice of designing his own user interface or selecting from a variety of software packages on the market. In order to develop a user friendly preprocessor, a user interface software package was chosen. There were two of them considered for the preprocessor, QuickWindows (**Software Interface, 1987**), and C-Scape 2.1 (**The Oakland Group, 1987**).

Both of these software packages have strengths, and weaknesses.

QuickWindows provides windowing capabilities in BASIC. It is a library of 66 routines written in assembly language that can be linked with a QuickBasic Compiler (by **Microsoft Corporation**). By simply calling the routines, programs can have windows, dialog boxes, input boxes, horizontal and vertical menus, help menus, screen and mouse functions (Olsen, 1988).

C-Scape 2.1 also has the capability of displaying pop-up menus, pull-down menus, prompt and pop-up windows, and nested pop-up (horizontal and vertical) menus called **slugs**. It also contains context-sensitive help (online help system) which allows a help pop-up window to be displayed at any time. It is a library of 200 procedures and functions that will simplify the rapid creation of attractive and flexible text menus (Oakland Group, 1987). This library can be linked with any C language compiler.

C-Scape 2.1 uses object-oriented techniques, based on C's **printf** function which is familiar to C programmers. By using object-oriented techniques, it gives the programmer the

ability to develop simple, flexible, and attractive input screens (data entry screens). When a data entry screen is displayed, the program shifts its control from the program to C-Scape's field functions allowing the user to have full control over entering and editing data. This is an advantage that C-Scape 2.1 has over QuickWindows for data entry screens. For example, a programmer working with QuickWindows would have to design and develop these field functions.

Another advantage of using C-Scape 2.1 is that it incorporates a **visual interface**. What this means is that the user can see something meaningful on the screen.

C-Scape 2.1, for the programmer and user, is the choice for the user interface software package. While, QuickWindows provides the capabilities for producing pop-up menus and windows, its weakness lies in the area of data entry screens compared to C-Scape 2.1. The preprocessor's main purpose is to create an error-free input data file for a structural analysis program; therefore, data entry screens are most important to the preprocessor.

2.2.1 Data Entry Screens

In order to achieve a user friendly preprocessor, data entry screens, pop-up and slug menus (Section 2.2.2), and pop-up and prompt windows (Section 2.2.3) routines are considered and developed using C-Scape 2.1.

A data entry screen routine can be called from another part of the program and will return the data entered by the user (Oakland Group, 1987). They can be considered as pop-up windows requesting data to be entered by the user. When it is activated, it overwrites what is currently on the screen. After the user completes the input or if ESCAPE is pressed, the data entry screen will exit and the display is returned to its previous state.

C-Scape 2.1 treats each menu as an object, or data object, to which input is sent. A data object is an abstract data type that consists of a data structure and various functions that act on the data structure. The Oakland Group (1987) states that "for each data object there is a set of functions to create, observe the contents of, and destroy the object." C-Scape 2.1 consists of three data objects, menu, field, and sed (screen editor), which are used to develop a data entry screen.

The menu object controls the structure and format of a data entry screen. It can be considered as the blueprint of the screen containing normal text and data entry areas called fields.

Another data object is the field. A field consists of writable and nonwritable positions. Writable positions allow characters to overwrite a specific position. These positions are noted as "#". Nonwritable fields do not allow characters to overwrite a specific position. These positions can be any character but not the "#"'s because they denote writable positions. A field also has a variable pointer and a field function bound to it. The variable pointer is the information entered by the user, while the field function determines if the data entered is in the correct format (i.e., string function only allows string input). A field function also processes keystrokes that are entered in a field, handles data editing of a field, and converts data to and from the fields required for display (Oakland Group, 1987). If there is more than one field present in a data entry screen the user will be allowed to move among them using arrow keys.

C-Scape 2.1 allows the programmer to make new field functions or modify existing ones. Therefore, it is possible to

customize the behavior of fields. Field functions are considered C-Scape's main strength because they give fields their personality or "visual interface."

The last object is a sed. The sed object contains and manages a screen's active display information. A sed is said to act as a "frame through which the menu can be seen and manipulated" (Oakland Group, 1987), and it also functions as an editor because it allows movement between and in fields, editing of fields, and validation of fields. The sed does the work of putting the menu up on the display and getting information from it. The data in a sed object consists of the menu colors, menu title, the menu's position on the screen, the height and width of a menu, the menu's border, the number of fields present, and the location of the fields within a menu.

To understand how these data objects work together, an example is shown below on how to build a data entry screen used in the preprocessor.

Input Filename	
Filename :	beam1
Enter the name of the input file	

A data entry screen is used to get a filename for a data file from the user. The routine in C is the following :

```
void get_filename(char *Filename_Dat)
{
  /*  Declare sed, menu objects */
  menu_type    menu;
  sed_type      sed;

  /* create the menu */
  menu = menu_Open();
  menu_Printf(menu, "@c[\x13]@p[1,1]Filename : @fd[########]",
            Filename_Dat, &string_funcs,
           "Enter the name of the input file");
  menu_Flush(menu);

  /* create the sed   */
  sed = sed_Open(menu);
  sed_SetColors(sed, '\x13', '\x13', '\x13');
  sed_SetBorder(sed, bd_std);
  sed_SetBorderTitle(sed, "Input Filename");
  sed_SetPosition(sed, 7, 25);    sed_SetWidth(sed, 25);
  sed_SetHeight(sed, 3);

  /* activate sed, and display it  */
  sed_Repaint(sed);  sed_Push(sed);
  sed_Go(sed);       sed_Pop(sed);

  /* release memory for use for sed, and menu   */
  menu_Close(menu);
  sed_Close(sed);
}
```

The get_filename works as follows: By calling the **menu_Open** function, the menu object is created. The **menu_Printf** function is called in order to define the contents of the menu. The fields are defined using the "**@fd[]**" command, which allows a prompt string, "enter the name of the input file", to

be displayed at the bottom of the menu, while the color of the menu is defined by the "**@c[]**" command. The field has a variable, Filename_Dat, associated with it where the entered data is stored. This field also has a field function bound to it that determines what type of field it is. The field function defined in this menu is **string_funcs**, which defines a field that only allows string input. The menu object is finished by the **menu_Flush** function.

The sed object is created by the **sed_Open** function. It displays the menu's border by **sed_SetBorder**, title by **sed_SetBorderTitle**, position on the screen by **sed_SetPosition**, height and width by **sed_SetHeight and sed_SetWidth**, and the colors of the borders and prompt area by **sed_SetColors**. The **sed_Repaint** function paints the menu on the screen. Calling the **sed_Go** passes the program control over to the field functions. The user can move through the fields if there is more than one and enter data. When the user has completed the entering of data or if ESCAPE is pressed, the control is passed back to the program.

The **sed_Push and sed_Pop** functions are used to store the previous screen and redisplay it after the routine is completed. After the user enters the name of the input file,

the menu and sed data objects are deactivated by calling the **menu_Close and sed_Close** functions. The name of the input file is returned to the calling program.

2.2.2 Pop-up and Slug Menus

One of the most common ways of allowing a user to choose one of many options is the menus. Menus are better than queries because if a menu is designed properly it leaves no room for user error (Jamsa and Nameroff, 1988). The preprocessor contains two types of menus, supported by C-Scape 2.1. These menus are pop-up and **slug**. It is important to understand how these menus function and how to generate them using C-Scape 2.1.

When a pop-up is activated, it overwrites what is currently on the screen. After a selection is made, the pop-up menu disappears and the screen is returned to its original state. There are two methods by which a user can select an option from a pop-up menu.

The first method to select an option is by pressing a **hot key**; a **hot key** is the first letter associated with one of the various menu options. By pressing a **hot key**, a highlight will

be placed on the option the user wants and pressing ENTER to execute it. A highlight informs the user which option is active, and it will display this option in reverse video.

The second method to select an option is by using the up and down arrow keys to move the highlight to the option and pressing ENTER to execute it.

However, by pressing ESCAPE, the active pop-up menu disappears and no option will be selected. The main purpose of the ESCAPE is to exit a specific menu.

In order to create a pop-up menu using C-Scape, the following function is called:

pop_Menu(text,choices,row,col,height,width,color,label,border)

This function creates and displays a pop-up menu with the given title and menu choices. The pop-up menu is displayed on the screen at a position (**row,col**) with the given **height** and **width**. It is drawn with a **color** and displayed with a specific **border**. The parameter **label** is bound to the pop-up menu and specifies which chapter of help to call if F1 is pressed. In the preprocessor, the help system is deactivated. An example

of a pop-up menu used in the preprocessor is shown below:

Do you want to quit ?
Yes
No

The pop-up menu is self explanatory because when the user selects "yes", the program quits. If the ESCAPE is pressed or "no" is selected the program will resume.

C-Scape 2.1 also allows the use of **slug** menus. **Slug** menus are nested pop-up menus. The two methods listed above also relate to **slug** menus; however, when **ENTER** is pressed one of three things happen: (1) Another pop-up menu is called (i.e., submenu), (2) a user supplied function is called, and (3) a value is returned to the program (Oakland Group, 1987). When ESCAPE is pressed, the previous menu is called; if no submenu is found it will exit and no selection will be made.

In order to create a **slug** menu using C-Scape 2.1, the following functions must be called:

slug_Go(slug, start, row, data)

slug_Open(slug_def, dir_flag, border)

slug_Open creates a **slug** menu object from a slug definition structure, **slug_def**. This definition consists of a list of menu options, a list of prompt strings bound to the options, user supplied functions, submenus to call, and a title. The direction flag in **slug_Open** determines in which direction the menu is displayed (i.e., horizontal or vertical). The last parameter in **slug_Open** is **border** which is the border of the slug menu. To activate the **slug** menu system, the **slug_Go** function is called. **Slug_Go** determines the position (**row, col**) on the screen to display the menu. **Start** in **slug_Go** determines which option is to be highlighted when the menu is activated. The data parameter in **slug_Go** is used to pass a data pointer to a user supplied function. An example of a **slug** menu is shown below with its submenu:

Main menu:

Main Menu
Editor
Preprocessor
Create an input data file

Submenu:

Preprocessor
Plane frame
Space frame Space Truss

When the slug menu becomes activated, the "main menu" will be displayed. By selecting the editor option, an editing routine will be called allowing the user to edit an input data file. However, if the user selects the preprocessor option a submenu is displayed allowing the creation of an input data file for a plane frame, space frame, or space truss. If ESCAPE is pressed with this submenu the main menu is redisplayed. However, if it was pressed in the main menu the menu is removed and the program continues.

2.2.3 Pop-up and Prompt Windows

Pop-up and prompt windows are used to display information concerning the operation of the preprocessor. This information can be a message to inform the user while some other activity is carried out, or any errors detected by the preprocessor.

In the preprocessor, pop-up windows are used to show a message to the user while some other activity is carried out. In the following example the user is told to wait while the element library is loaded.

Loading the element library file.

After the element library is loaded, the pop-up window disappears and the program continues. A pop-up window function in C-Scape 2.1 is shown below:

pop_Message(text,row,col,height,width,color,border)

This displays text in a window at a position (**row,col**) with given **height** and **width**. The **color** and **border** parameters are used to define the color and the border of the pop-up window. The **text** is word-wrapped to fit the pop-up window, but you can force a line break in the text by using the character '\n'. To clear a pop-up window from the display, it is recalled again with **text** set to NULL.

Prompt windows are used to inform the user of important messages (i.e., errors). The preprocessor might encounter errors related to the activity of it, or from the user's input. The errors created by the preprocessor are associated

to the disk input/output operations, while the errors generated by the user's input can be related to incorrect entries (such as joint and member numbers out of range or equal to zero). The difference between pop-up and prompt windows is that prompt windows requires the user to press a **hot key**. A **hot key** for prompt windows is ESCAPE. The operation of the preprocessor is suspended until this **hot key** is pressed. When ESCAPE is pressed, the prompt window disappears, the screen returns to its previous state, and the program continues. A prompt window function is shown below:

pop_Prompt(text,row,col,height,width,color,border)

The parameters in the prompt windows are the same as for the pop-up window. An example of a prompt window is shown below:

```
Error :

      Joint number can not be zero.

              Press  ESC
```

After the user presses ESCAPE, the program will continue.

2.3 The Programming Language C

The user interface software package C-Scape 2.1 dictated which

programming language to choose. This programming language was C.

In 1970, Ken Thompson designed the language B from a stripped-down version of BCPL, designed by Martin Richards. The language B is a "typeless language"; that is, it operates on a single data type, the machine word. The language C, designed by Dennis Ritche around 1972, was an attempt to deal with a variety of types of data by adding a "date type" to the language B. Shildt defines "a data type as a set of values that a variable can store along with a set of operations that can be performed on that variable" (Schildt, 1988).

C supports numerous data types, including both basic and aggregate data types. There are four basic data types in C: *char*, *int*, *float*, and *double*. The *char* and *int* are for storing characters and integers, and *float* and *double* are for floating-point numbers. While, aggregate data types are arrays, structures, and unions. These aggregate variables may be made up of multiple components, which may be the same data type, or of different types (Feibel, 1988). An array is a convenient way to organize a large number of identical data items, and they can be single- and multidimensional. The structure in C is a collection of variables that are

referenced under one name, which provides a convenient means of keeping related information together, while unions enable the same piece of memory to be defined as two or more different types of variables (Schildt, 1988). Another data type is a pointer. A pointer is a special data type that can hold the address of an object. This object can be either a variable or a function. The main advantage of using pointers is that they can improve the speed and efficiency of certain routines; however, pointers can be most dangerous; for example, uninitialized or wild pointers can cause the program to crash. A wild pointer can be most difficult to debug.

C can be described as a middle-level language, having characteristics and capabilities of both high- and low-level languages. For high-level languages, it offers powerful control constructs (for, while, if-then), and complex data structures as Pascal, and Modula-2 (Fiebel, 1988). A complex data structure in C, for example, is a structure, while in Pascal this can be referred as a Record. For low-level languages (such as assembly language), C provides access to individual addresses and bits of memory. Because of these multilevel features, C was initially used in writing "system" programming, such as operating systems, editors, and compilers, but it is quite satisfactory for application

programs, such as engineering programs, and numerical analysis programs (Schildt, 1988).

C is a small language compared to Pascal, Modula-2, and even BASIC: it only contains 30 keywords, or reserved words. In contrast, IBM BASIC (by Microsoft Corporation) contains 159 reserved words. This does not mean that C is less powerful than higher-level languages. Many of its actions are close to assembly-level (Schildt, 1988).

C is also a structured language which allows the programmer a variety of programming possibilities. It supports several looping constructs (such as, while, for, do-while) which are part of the control structures of C. In a structured language, the use of **goto** is a forbidden form of program flow. The programming method used for this preprocessor was structured programming because it is a method of combining control constructs in a way to reflect the meaning of the program.

2.4 Finial Presentation of Language and Software

By using a user interface software package, C-Scape 2.1, and the programming language C, a user friendly preprocessor was

achieved. In order to achieve a user friendly preprocessor, (pop-up and slug) menus, data entry screens, and (pop-up and prompt) windows were used.

The preprocessor is a collection of three programs. The main program in the preprocessor is entering and editing data for the input data files. The two other programs, graphics presentation and library manager, are the preprocessor's supporting programs. Graphics presentation is a program to display a plane frame structure. The library manager is used to manage the libraries (such as, the element and material properties) used by the preprocessor. The functions of these programs are explained in detail in the user manual, Chapter 4.

The tree charts for the three programs are presented in Figures 2.1 through Figure 2.25.

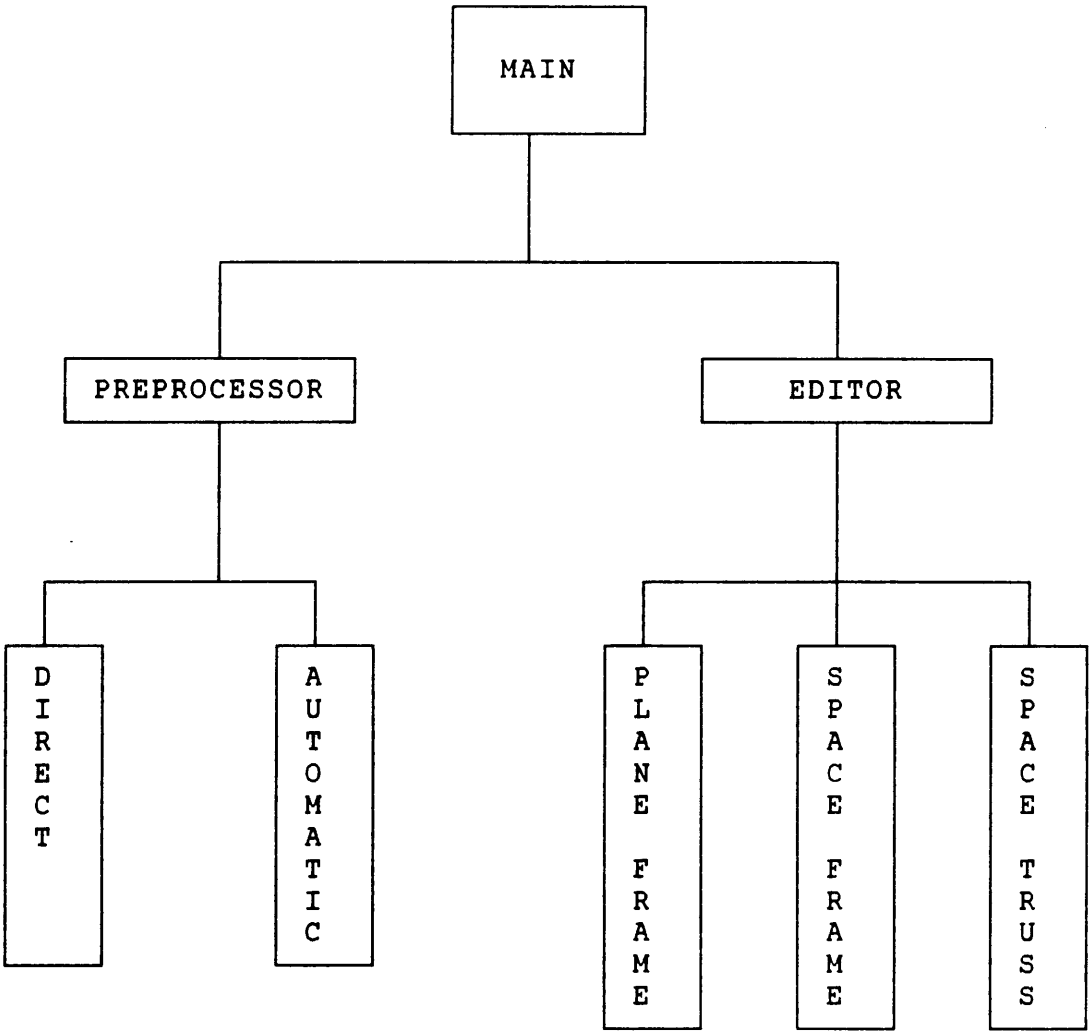
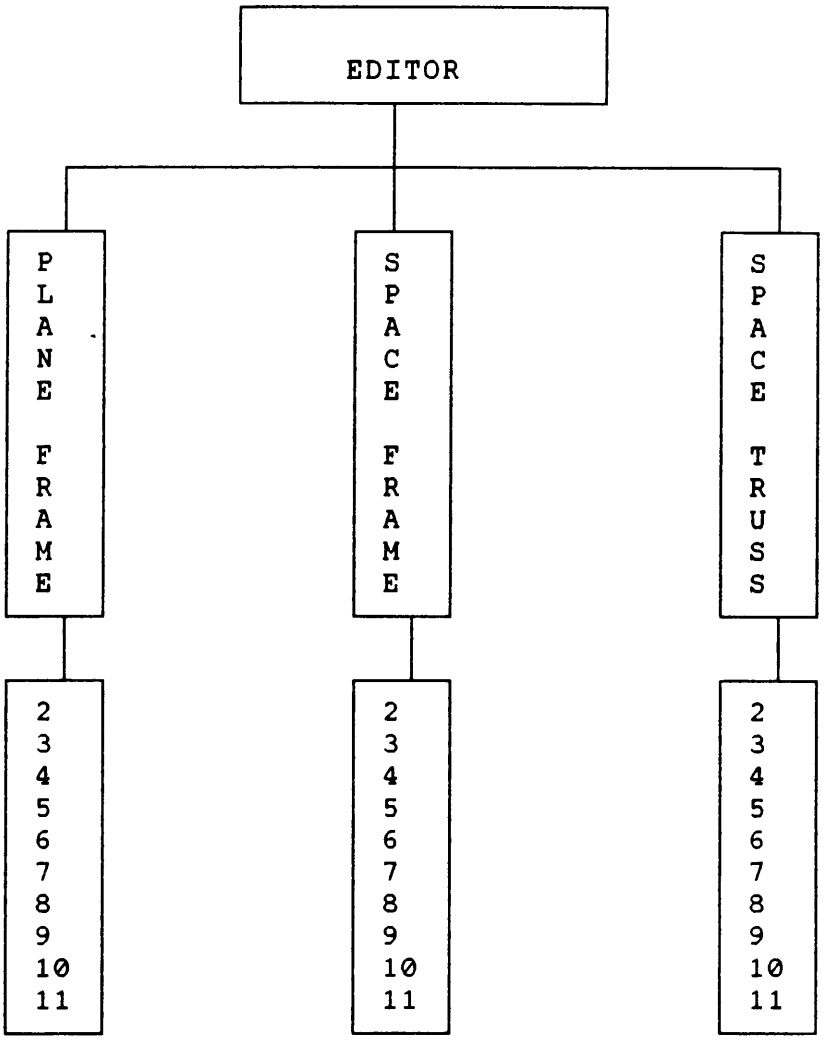
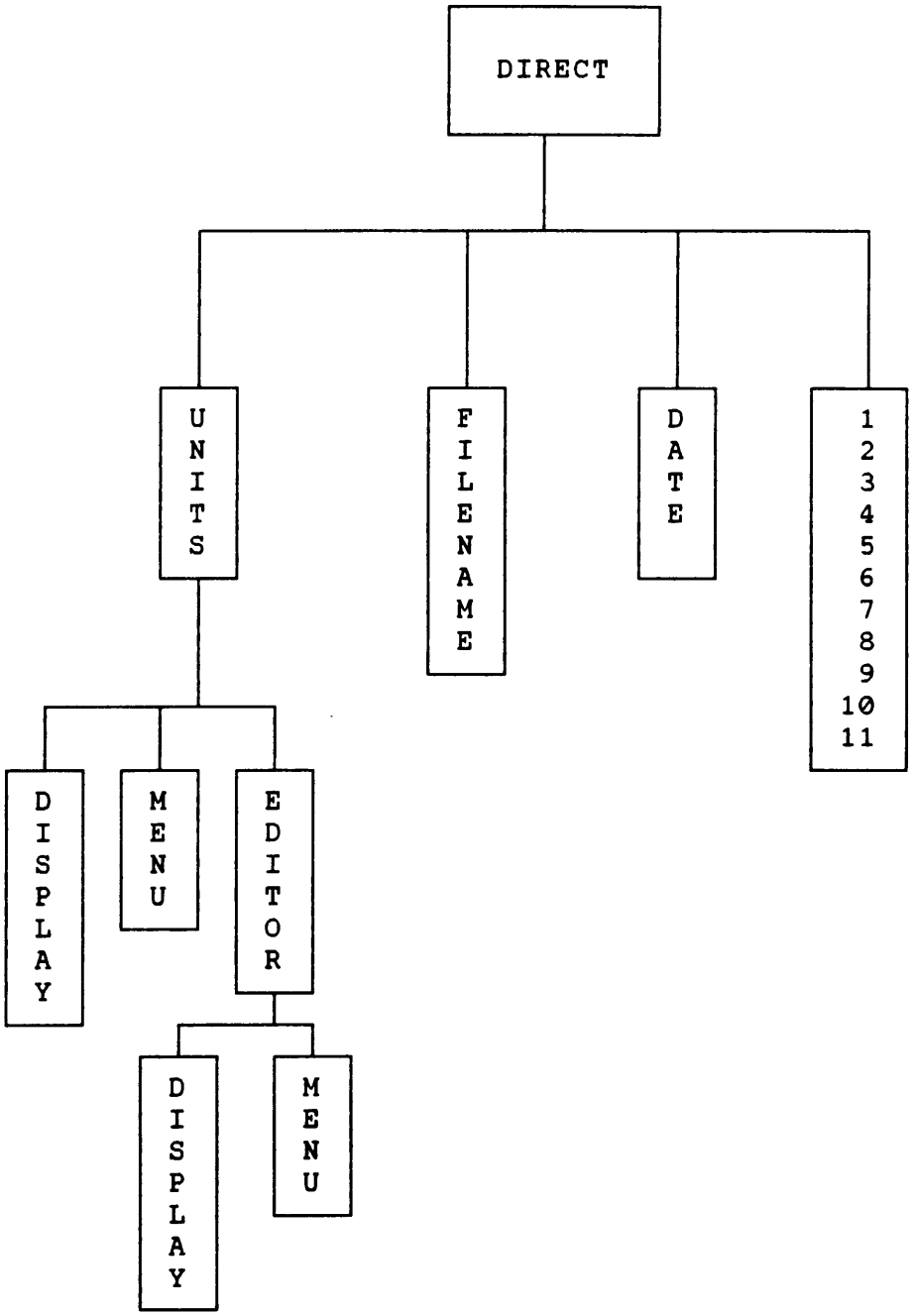


Figure 2.1: Tree Chart of Preprocessor



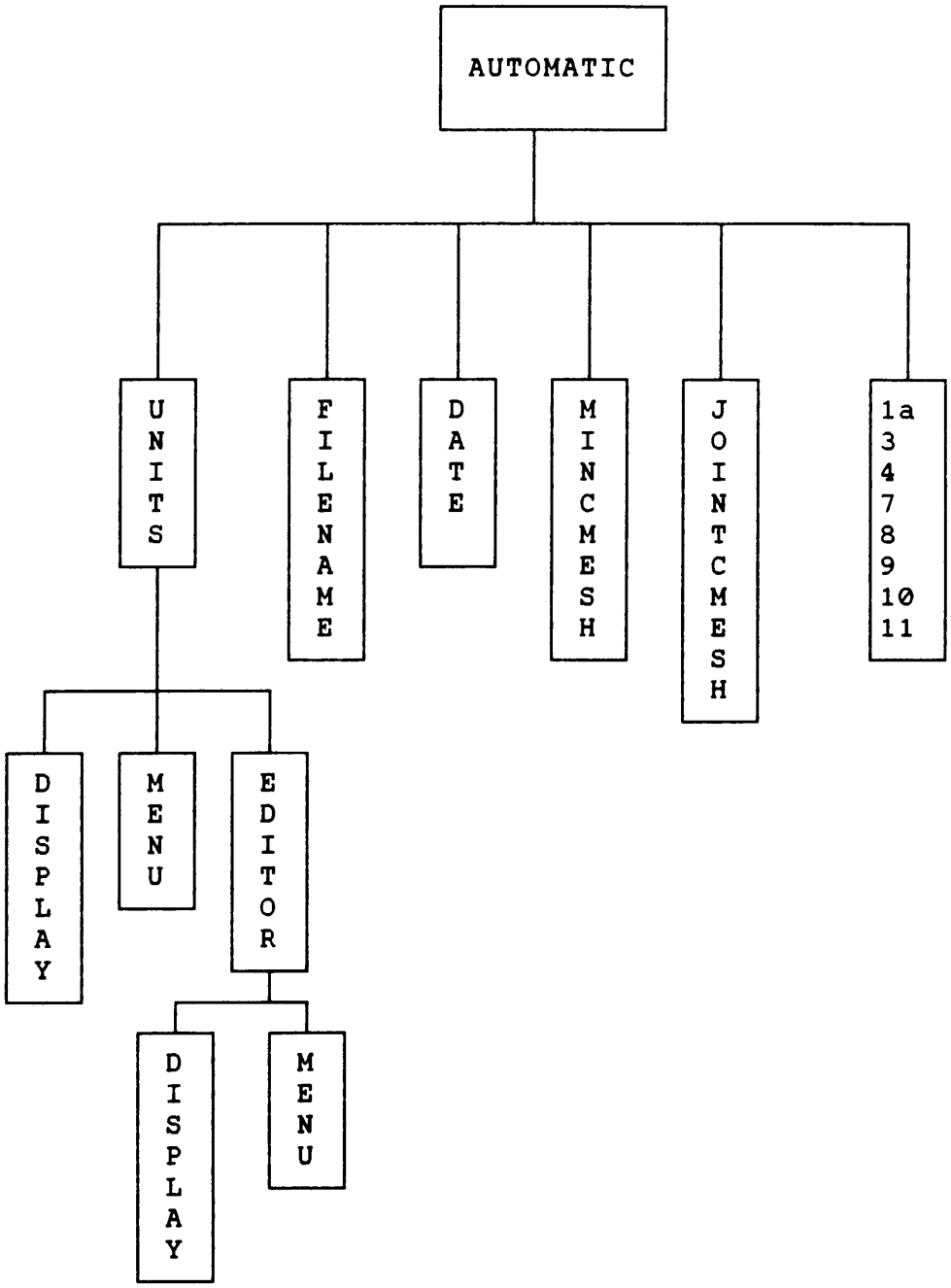
Note: Numbers (1 - 11) denote general procedures (see Figures 2.6 - 2.11). The editor routine is only used.

Figure 2.2: Tree Chart of Editor



Note: Numbers 1 through 11 denote general procedures (see Figures 2.5 - 2.11)

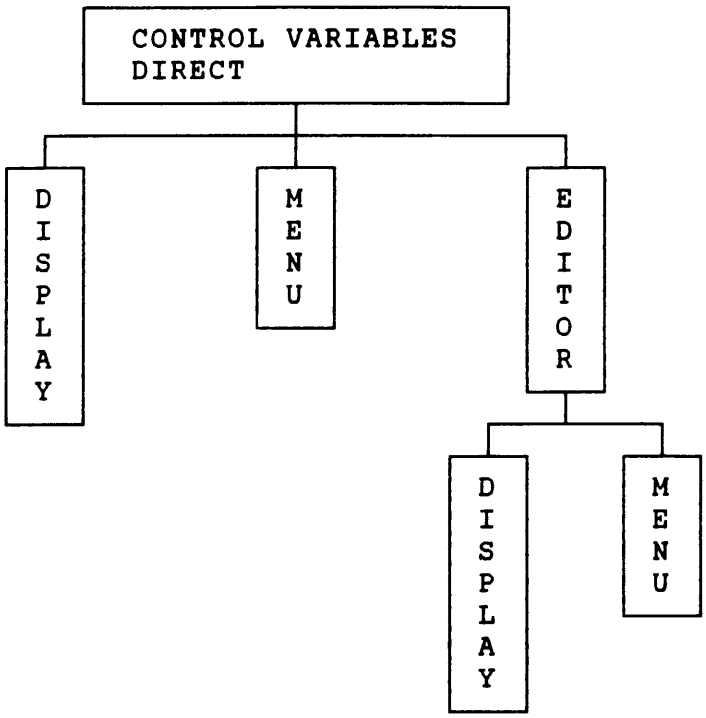
Figure 2.3: Tree Chart of Direct Mesh Generation



Note: Numbers 1a through 11 denote general procedures (see Figures 2.5 - 2.11). Automatic is only available for plane frames.

Figure 2.4: Tree Chart of Automatic Mesh Generation

Procedure 1:



Procedure 1a:

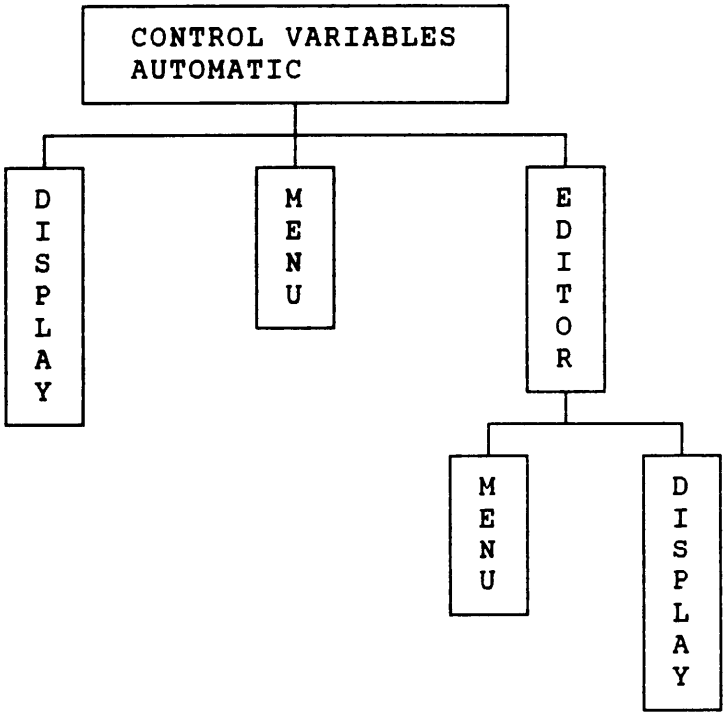
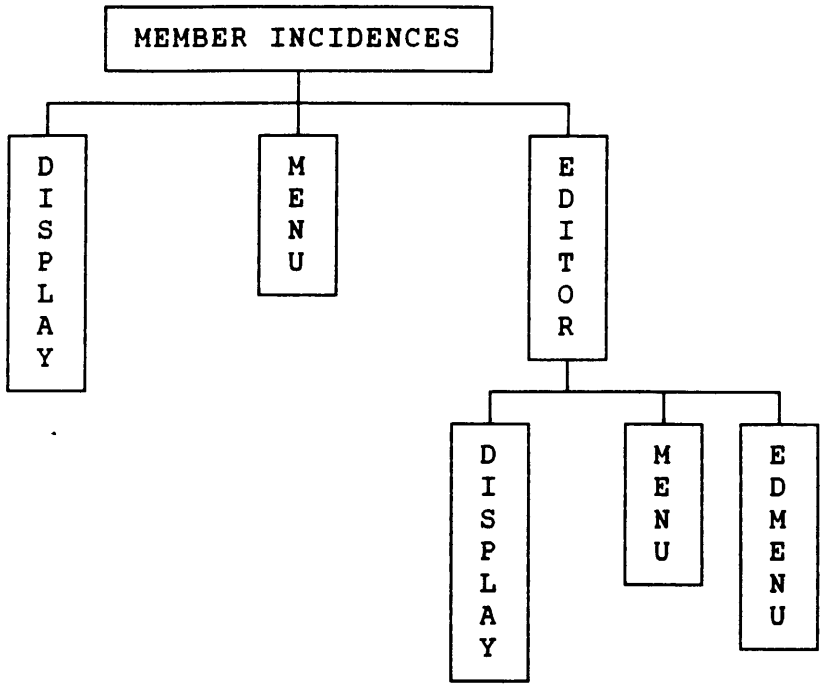


Figure 2.5: Tree Chart of Control Variables

Procedure 2:



Procedure 3:

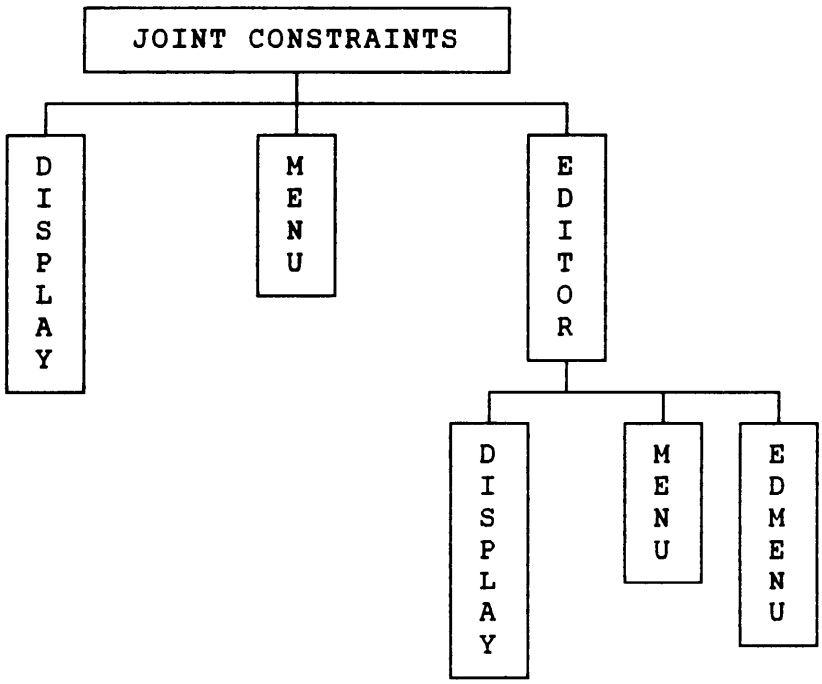
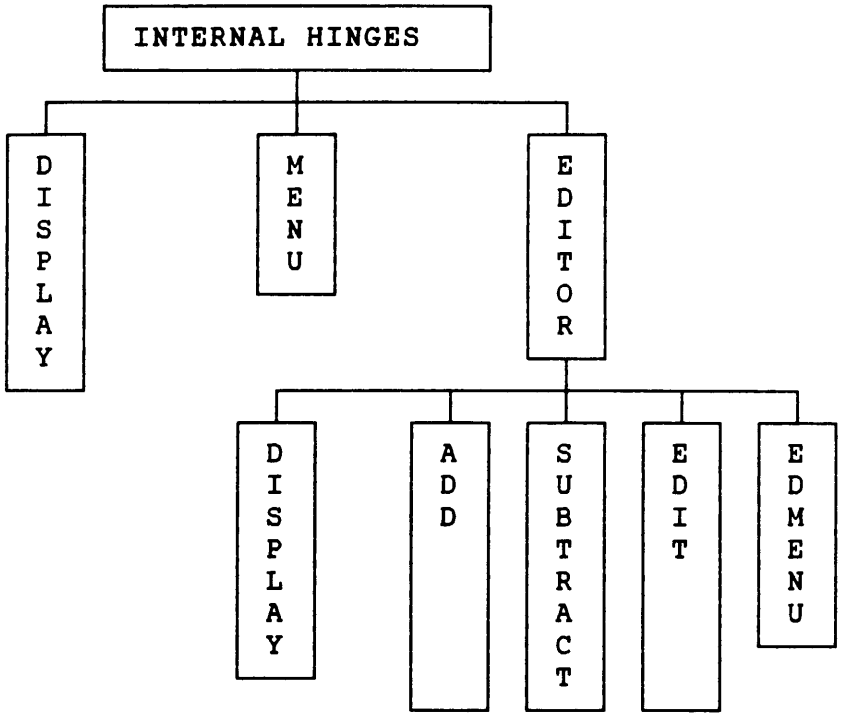


Figure 2.6: Tree Chart of Member Incidences and Joint Constraints

Procedure 4:



Procedure 5:

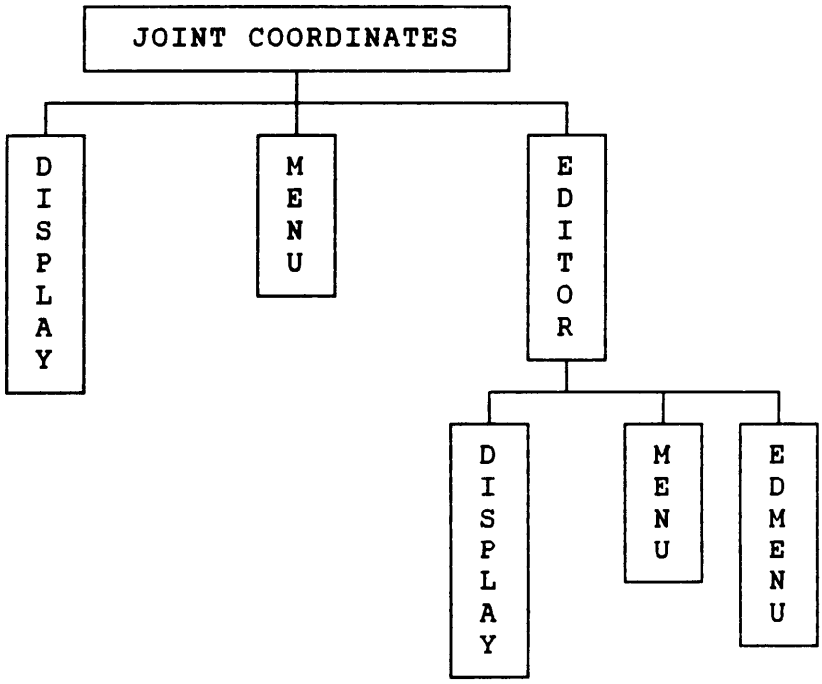
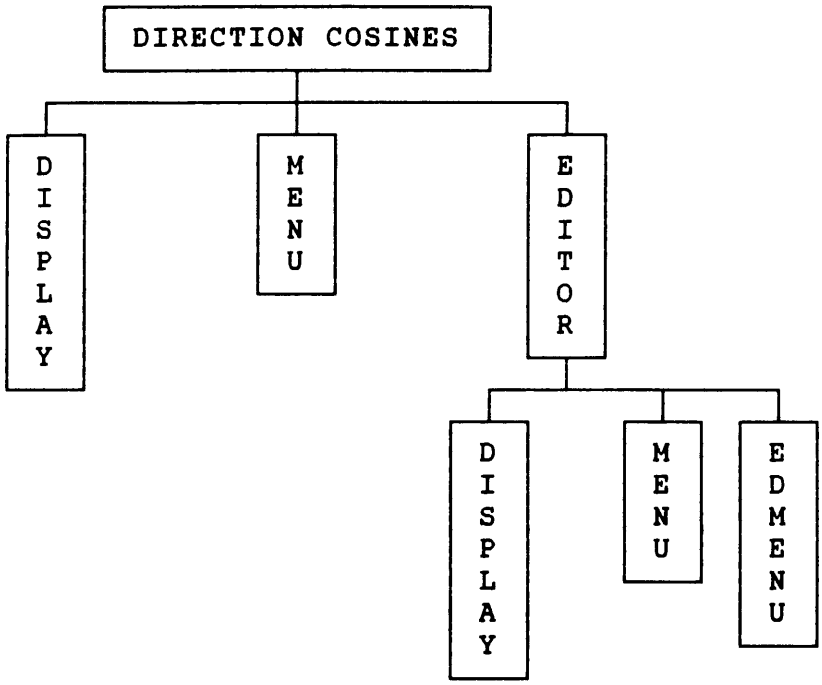


Figure 2.7: Tree Chart of Internal Hinges and Joint Coordinates

Procedure 6:



Procedure 7:

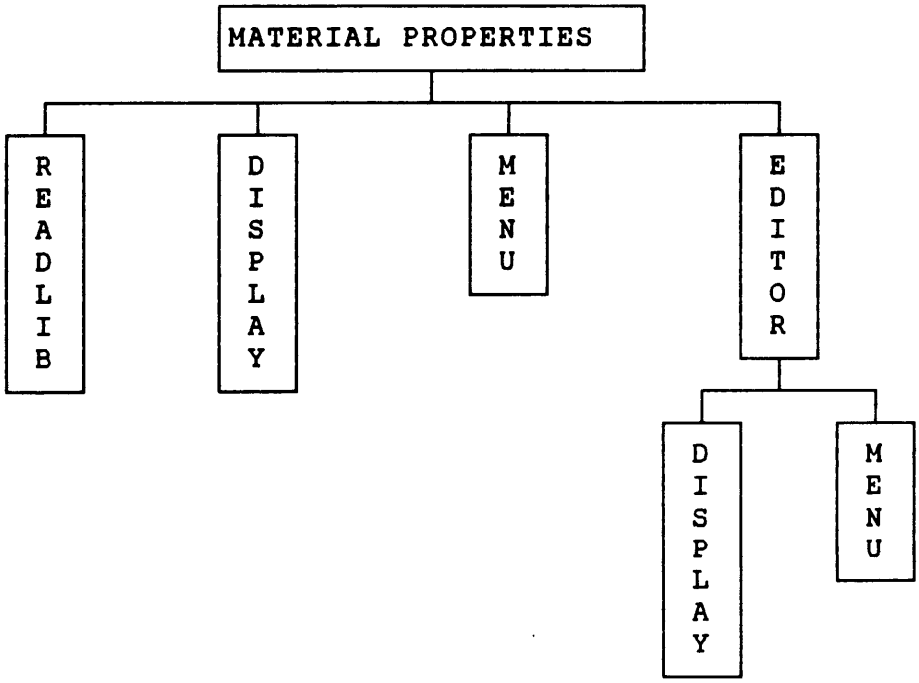
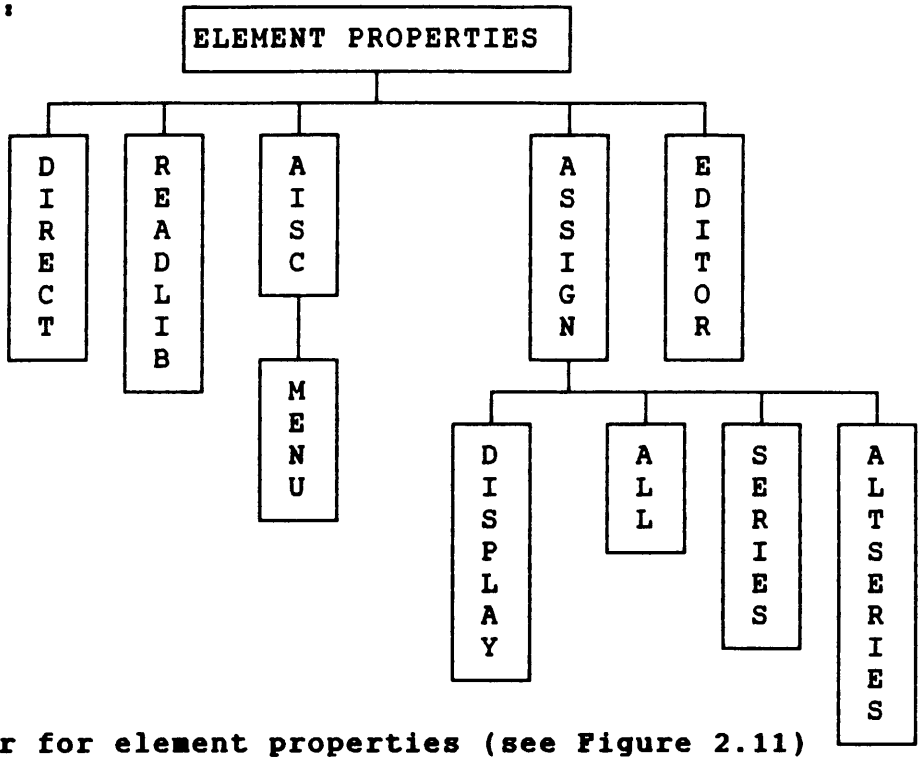


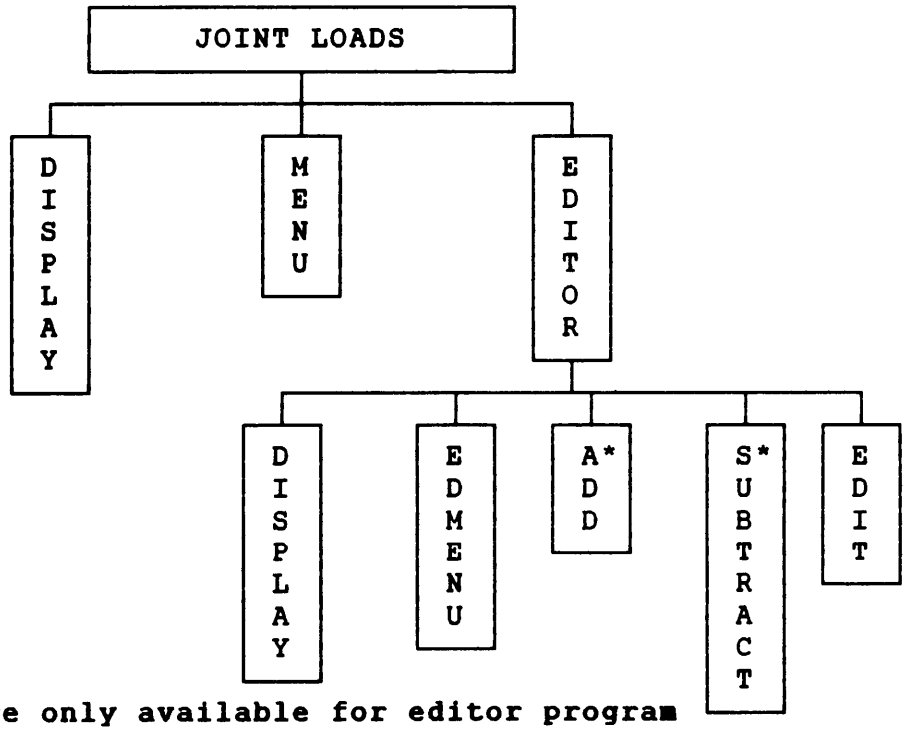
Figure 2.8: Tree Chart of Direction Cosines and Material Properties

Procedure 8:



Note: Editor for element properties (see Figure 2.11)

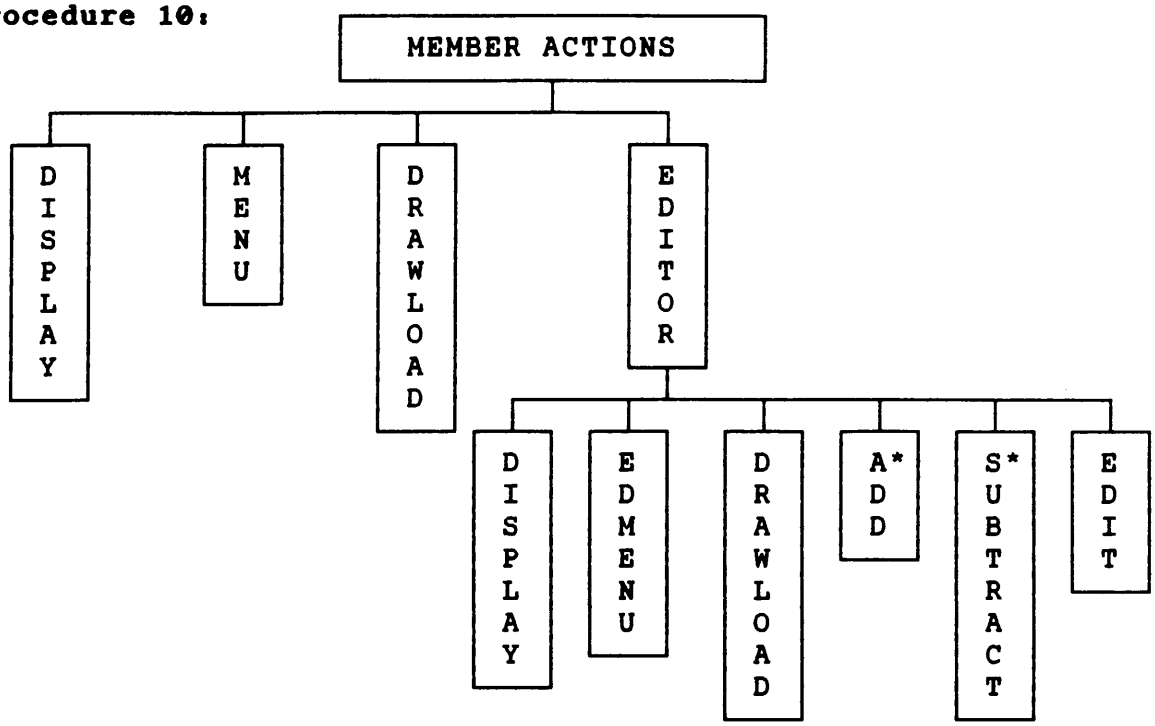
Procedure 9:



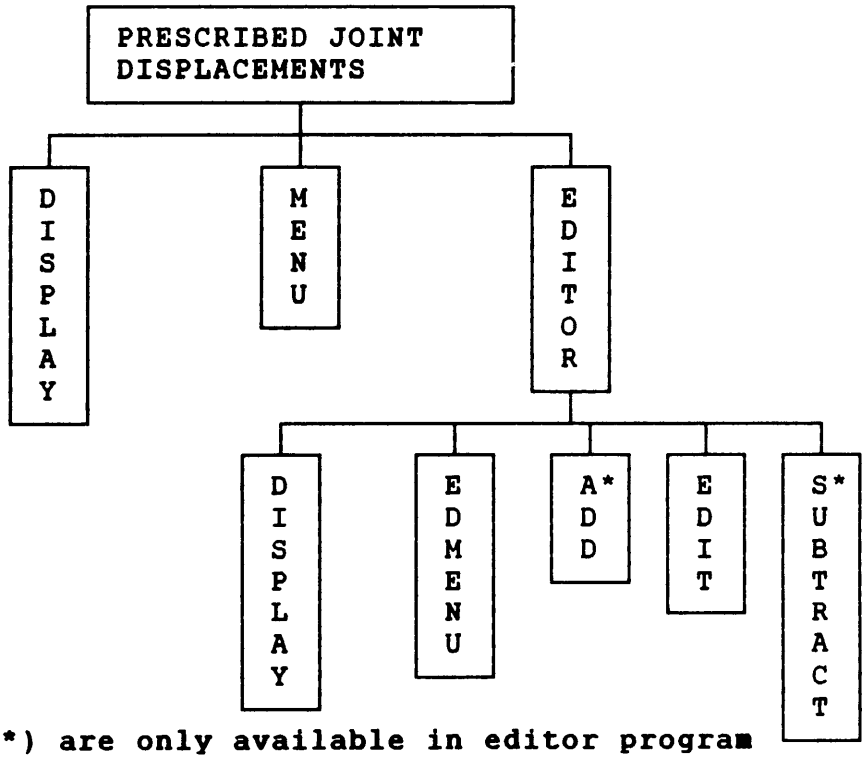
Note: (*) are only available for editor program

Figure 2.9: Tree Chart of Element Properties and Joint Loads

Procedure 10:



Procedure 11:



Note: (*) are only available in editor program

Figure 2.10: Tree Chart of Member Actions and Prescribed Joint Displacements

Editor for Element Properties:

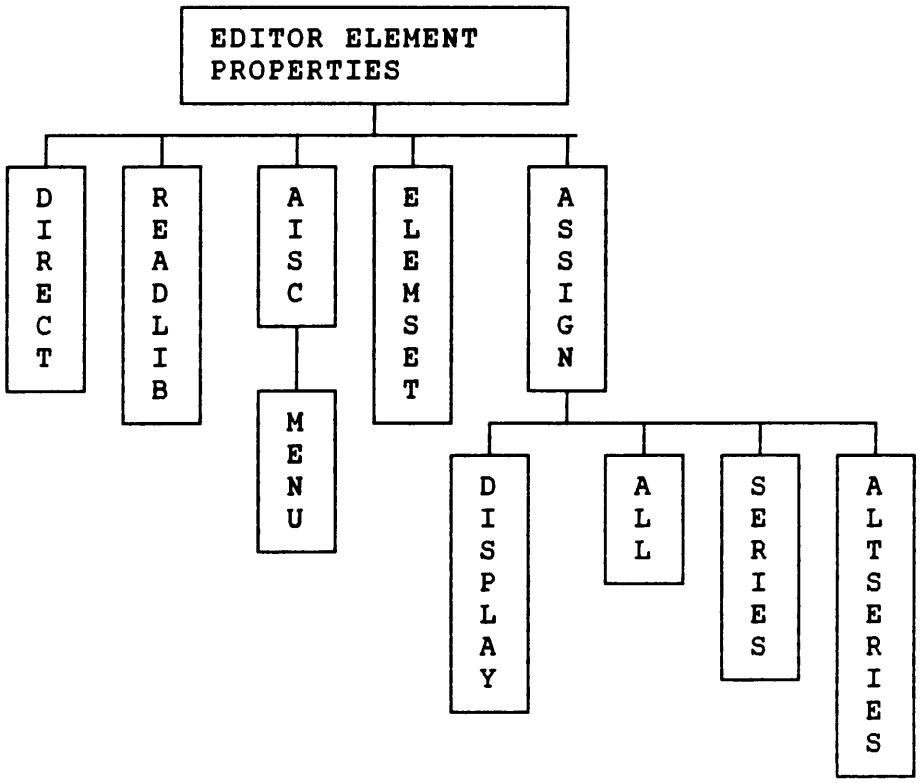


Figure 2.11: Tree Chart of Editor of Element Properties

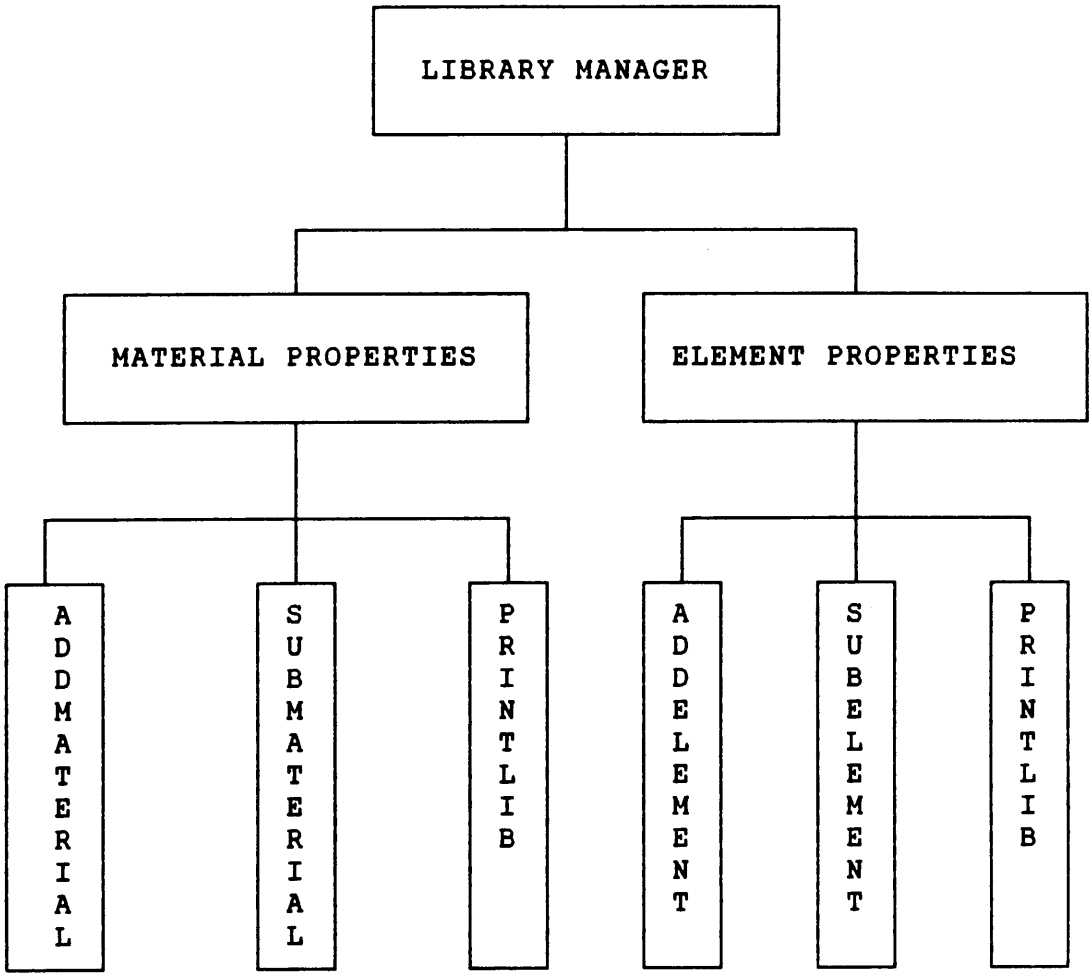


Figure 2.12: Tree Chart of Library Manager

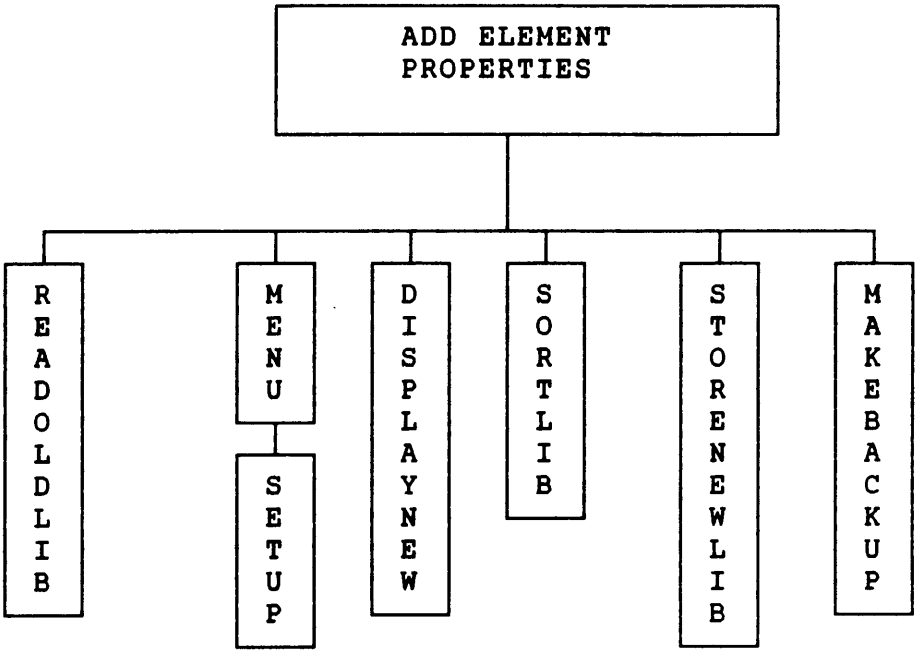


Figure 2.13: Tree Chart of Adding Element Properties

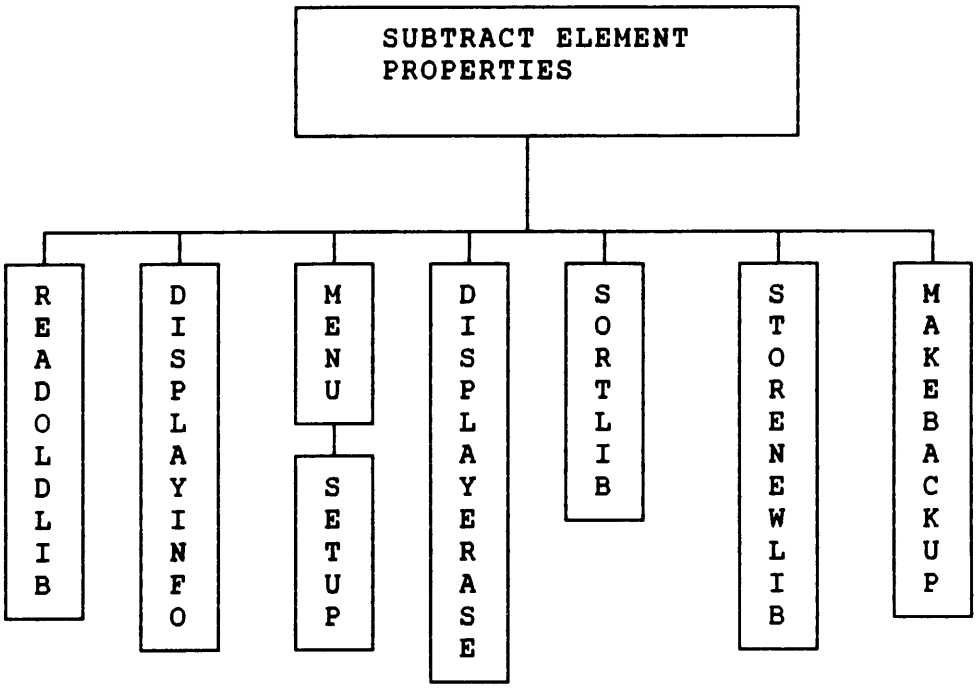


Figure 2.14: Tree Chart of Subtracting Element Properties

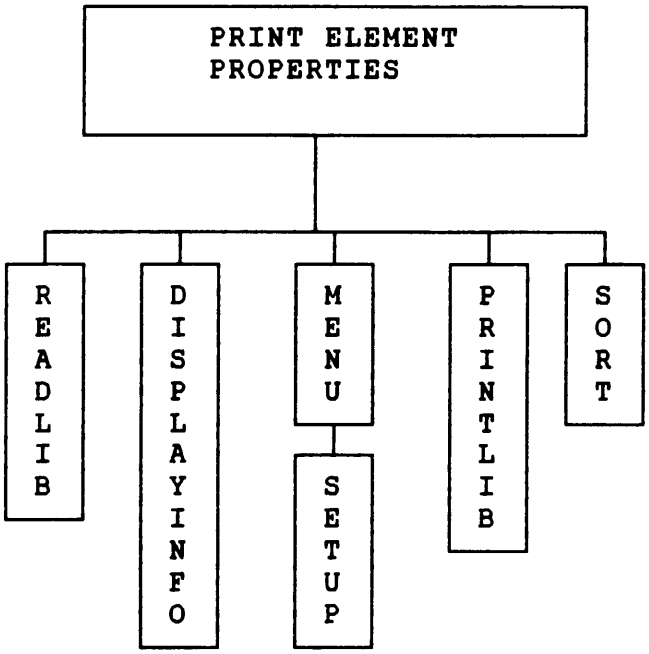


Figure 2.15: Tree Chart of Printing Element Properties

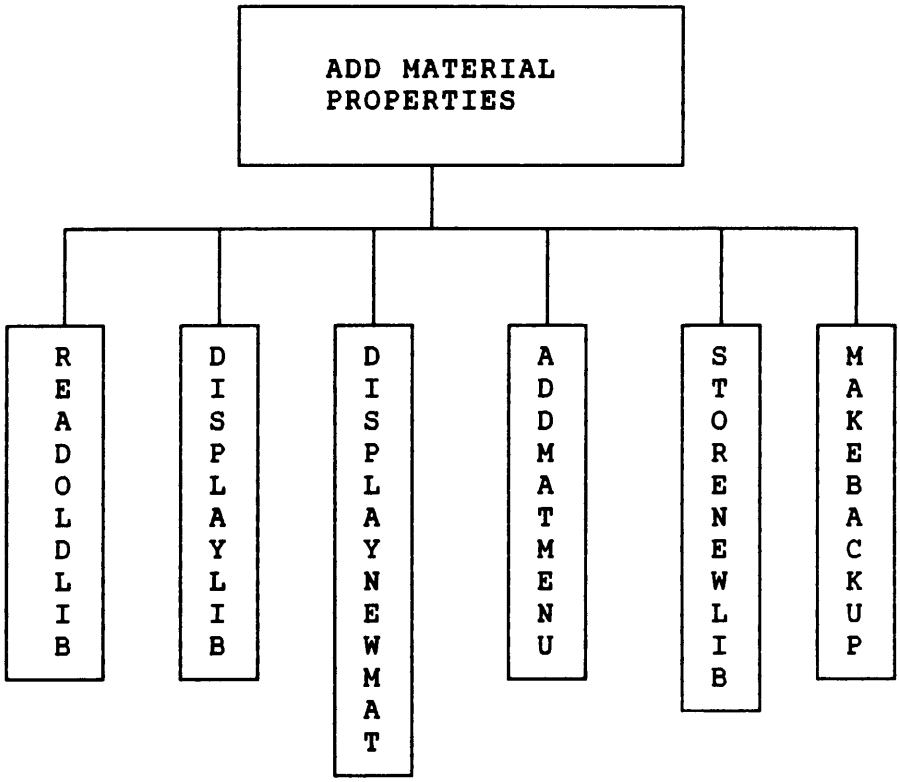


Figure 2.16: Tree Chart of Adding Material Properties

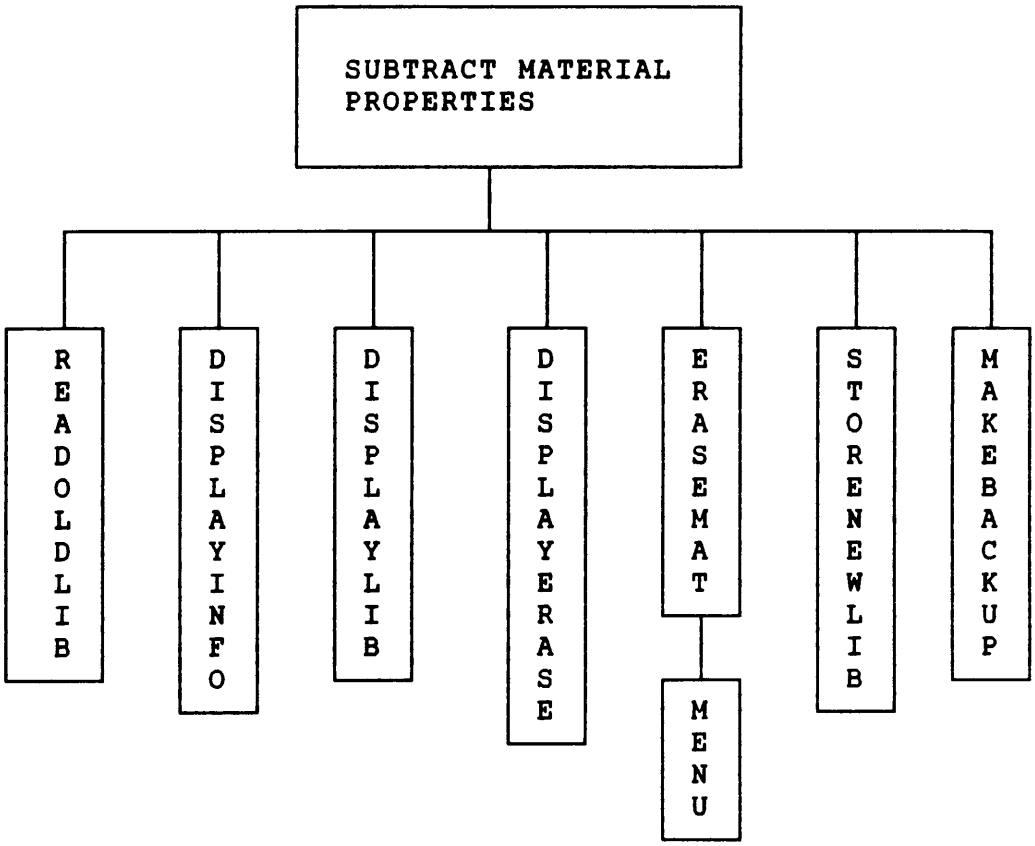


Figure 2.17: Tree Chart of Subtracting Material Properties

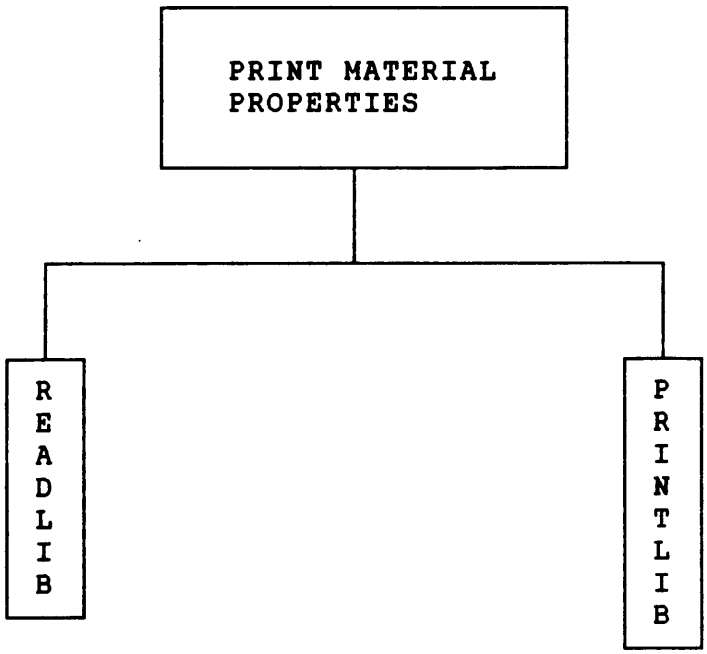
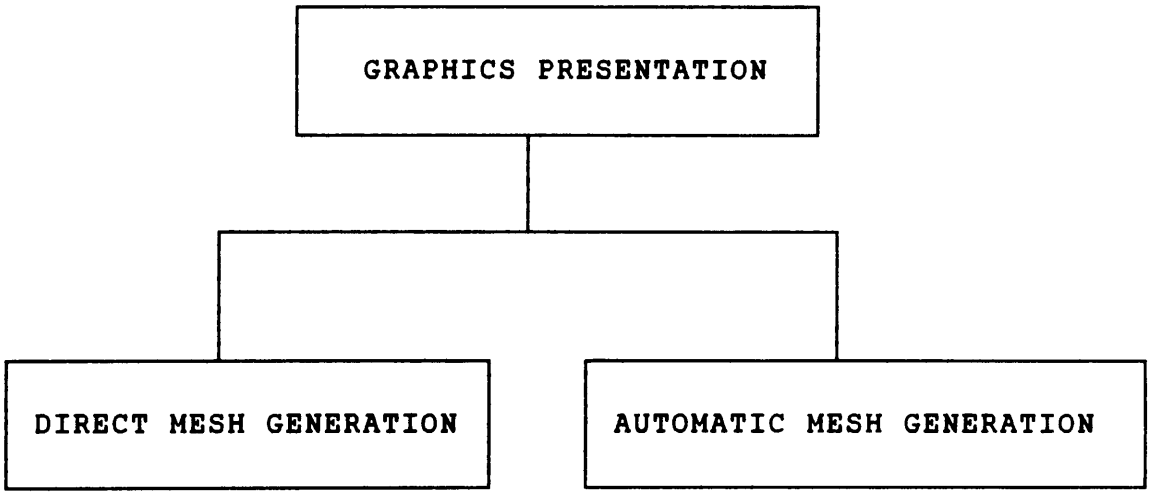
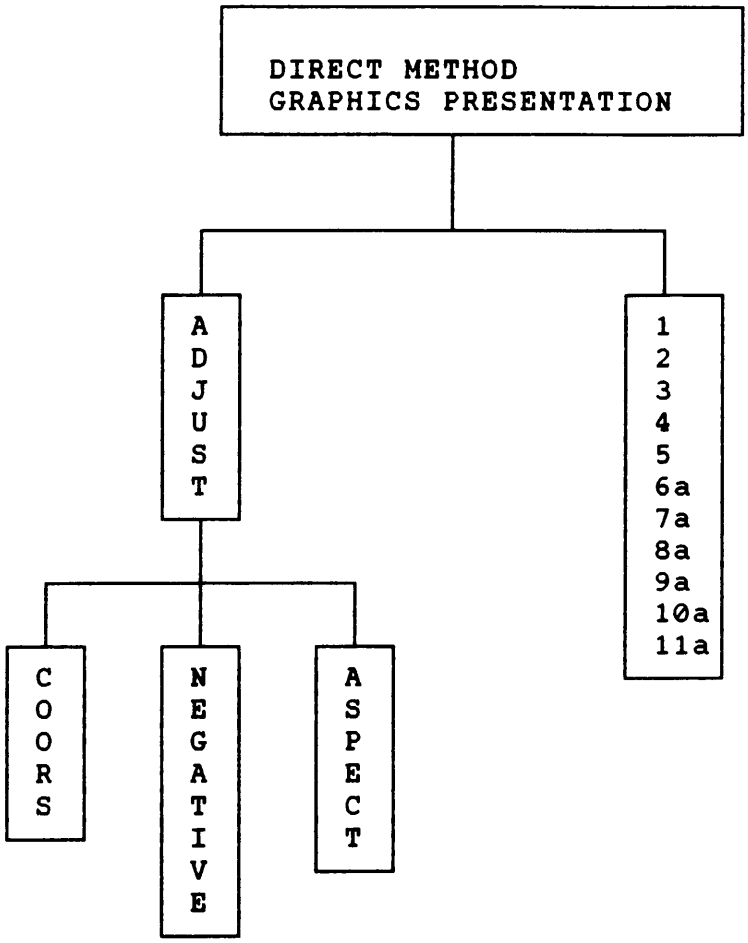


Figure 2.18: Tree Chart of Printing Material Properties



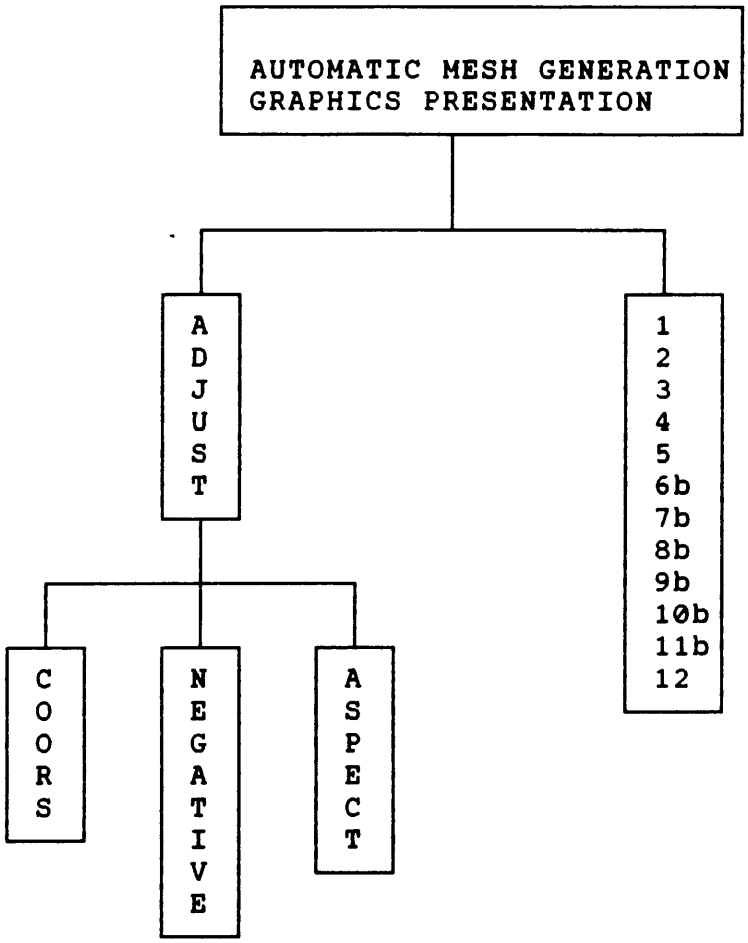
Note: Only plane structures allowed.

Figure 2.19: Tree Chart of Graphics Presentation



Note: Numbers 1 through 11 are for general procedures (see Figures 2.22 - 2.24)

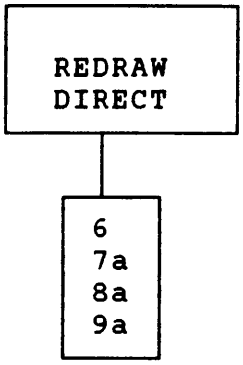
Figure 2.20: Tree Chart of Direct Method of Graphics Presentation



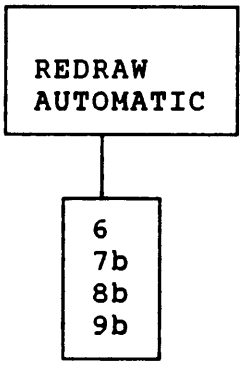
Note: Numbers 1 through 12 are for general procedures (see Figures 2.22 - 2.25)

Figure 2.21: Tree Chart of Automatic Mesh Generation of Graphics Presentation

Procedure 1a:



Procedure 1b:



Note: Numbers (6 - 9) are general procedures

Procedure 2:

Procedure 3:

Figure 2.22: General Procedures (1 - 3) of Graphics Presentation

Procedure 4:

HELP

Procedure 5:

DISPLAY SIZE

Procedure 6:

DRAW JOINT CONSTRAINTS

Procedure 7a:

DRAW MEMBER NUMBERS

Procedure 7b:

DRAW MEMBER NUMBERS MESH

Figure 2.23: General Procedures (4 - 7) of Graphics Presentation

Procedure 8a:

DRAW JOINT NUMBERS

Procedure 8b:

DRAW JOINT NUMBERS MESH

Procedure 9a:

DRAW MEMBERS

Procedure 9b:

DRAW MEMBERS MESH

Procedure 10a:

ZOOM

Procedure 10b:

ZOOM MESH

Figure 2.24: General Procedures (8 - 10) of Graphics Presentation

Procedure 11:

Figure 2.25: General Procedures (11) of Graphics Presentation

Chapter 3

Extension of Analysis Program from Plane Frames to Space Frames

3.1 Space Frames

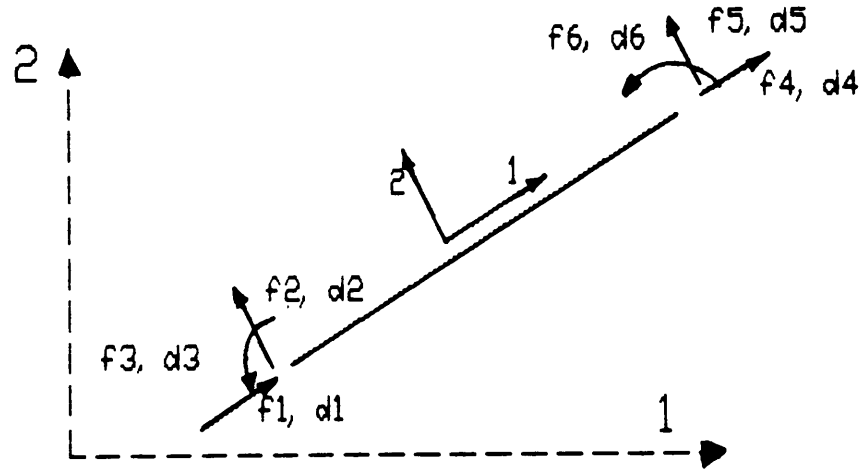
As stated in the previous Chapter, the Preprocessor was developed to create the input data file for a structural analysis program. The analysis program performs the matrix displacement method (Holzer, 1985) to compute the displacements and forces of a structure.

The space frame analysis program, which was developed in a special study class (Holzer, 1988), is an extension of the plane frame analysis program (Holzer, 1985).

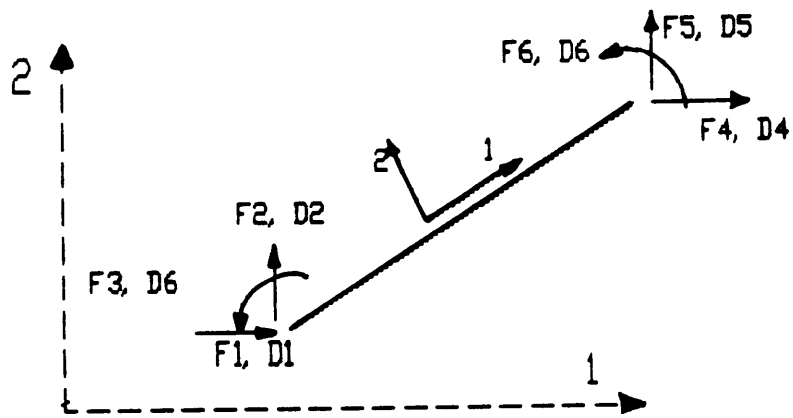
The analysis of plane structures is confined to prismatic elements that are symmetric about the local coordinate plane. The plane structure is assumed to lie in the 1-2 global

coordinate plane. The global coordinate plane is represented by broken lines, while the local coordinate plane is represented by solid lines (Figure 3.1). A plane frame element, Figure 3.1, is represented by axial and flexural deformations; thus, a joint has only three degrees of freedom. A restriction is that the loads and responses must lie in a plane defined by the local coordinate plane.

The analysis of space structures is confined to prismatic elements with bisymmetrical cross section that are free to warp. For a cross section with biaxial symmetry, the shear center and the center of twist occur at the centroid of the cross section. The shear center is a point where an applied load causes bending without twisting, while the center of twist is a point where an applied torque causes twisting without bending. If warping is constrained, the deformation of the cross section out of its plane causes normal stresses and strains to arise. For these normal stresses and strains to be negligible, the torsional and axial deformation must remain uncoupled (i.e., independent). The cross section is free to warp if the deformations are independent. A space frame element, Figure 3.2, represents axial deformations, flexural deformations about two principle axes, and torsional deformation; thus, a joint has six degrees of freedom.



(a) Local



(b) Global

Figure 3.1: Element Displacements and Forces for Plane Frames

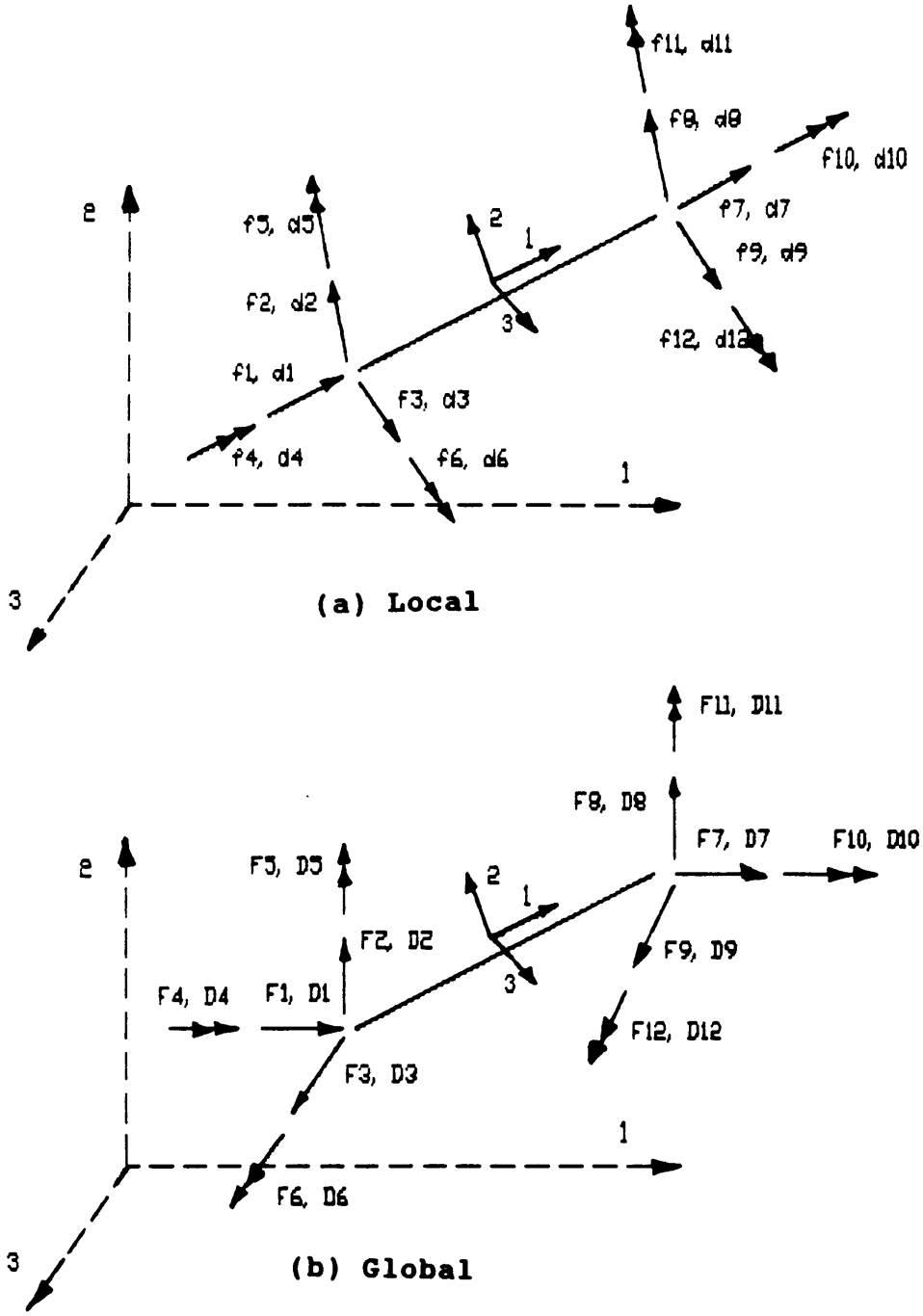


Figure 3.2: Element Displacements and Forces for Space Frames

The preprocessor for space frames requires the user to enter additional information that is not required in a plane frame analysis. This additional information is the orientation of a principle axis of the cross section, additional material and element data, and the loading (refer to Section 3.1.1).

The design of the plane and space frames analysis program is illustrated in Figures 3.3 through 3.7 in tree chart form.

3.1.1 Additional Information Necessary to Define Space Frames

3.1.1.1 Orientation of a Principle Axis of the Cross Section

The joint coordinates of plane frames are sufficient to define the direction cosines of the local reference axes of each element. However, the joint coordinates for space frames define only the components of the unit vector i_1 on the local 1 axis (Figure 3.8). Thus, additional information is needed to define the components of the unit vectors i_2 and i_3 on the local 2 and 3 axes. There are several approaches that can be used to define i_2 and i_3:

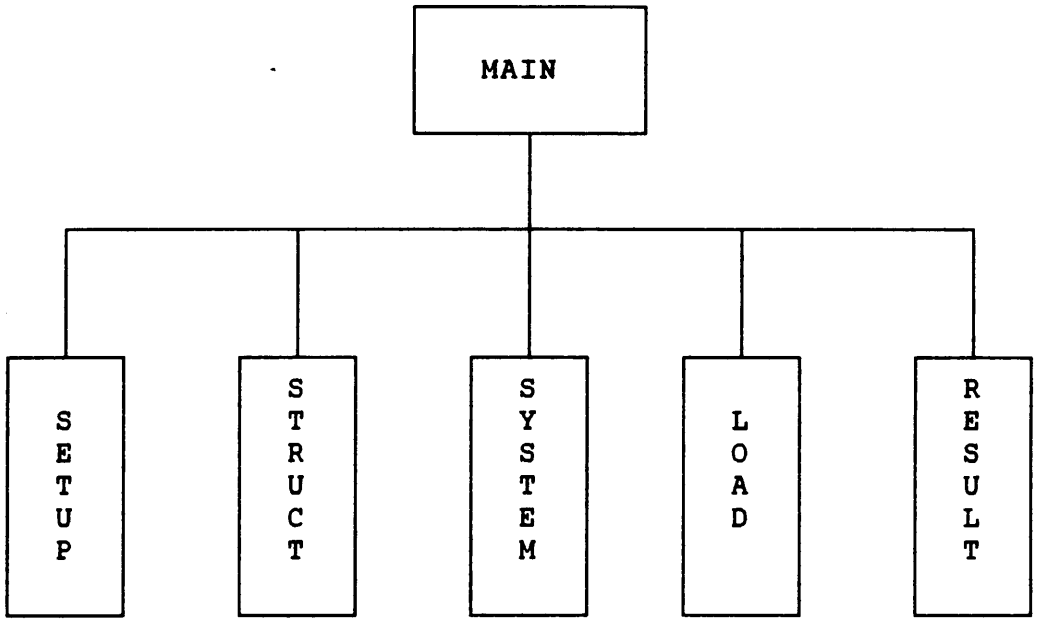


Figure 3.3: Tree Chart of Plane and Space Frame Structural Analysis Program

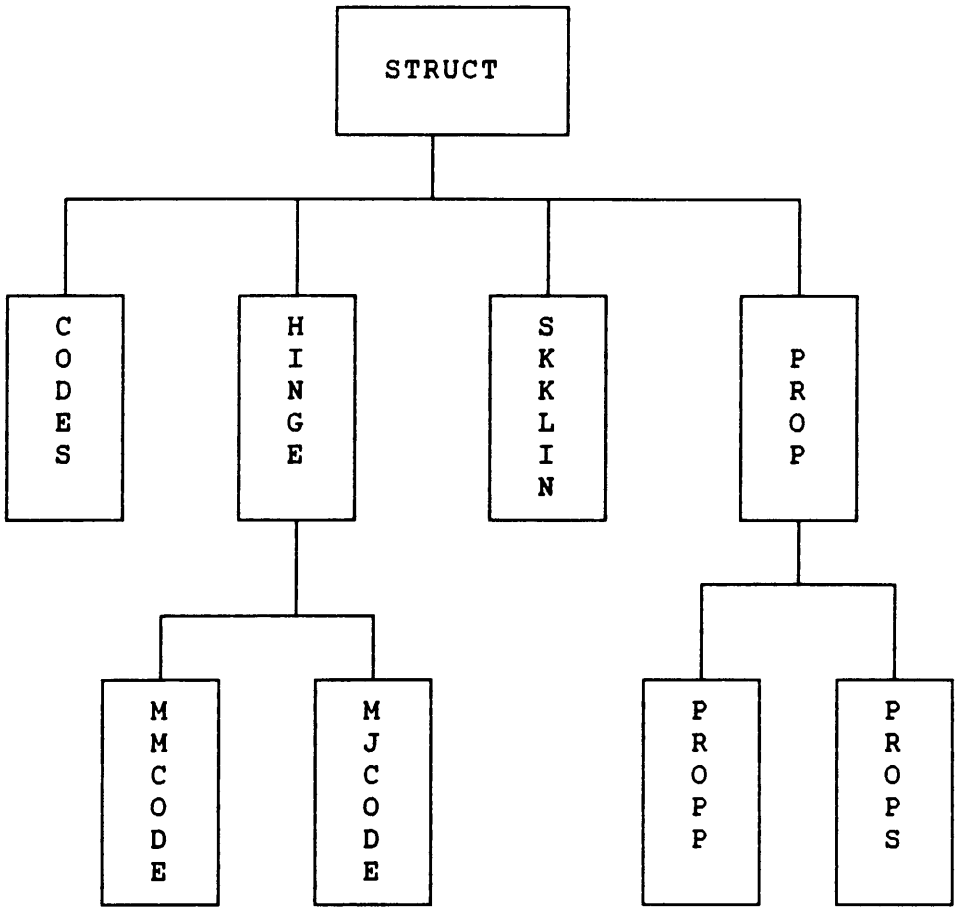


Figure 3.4: Tree chart of Subroutine STRUCT

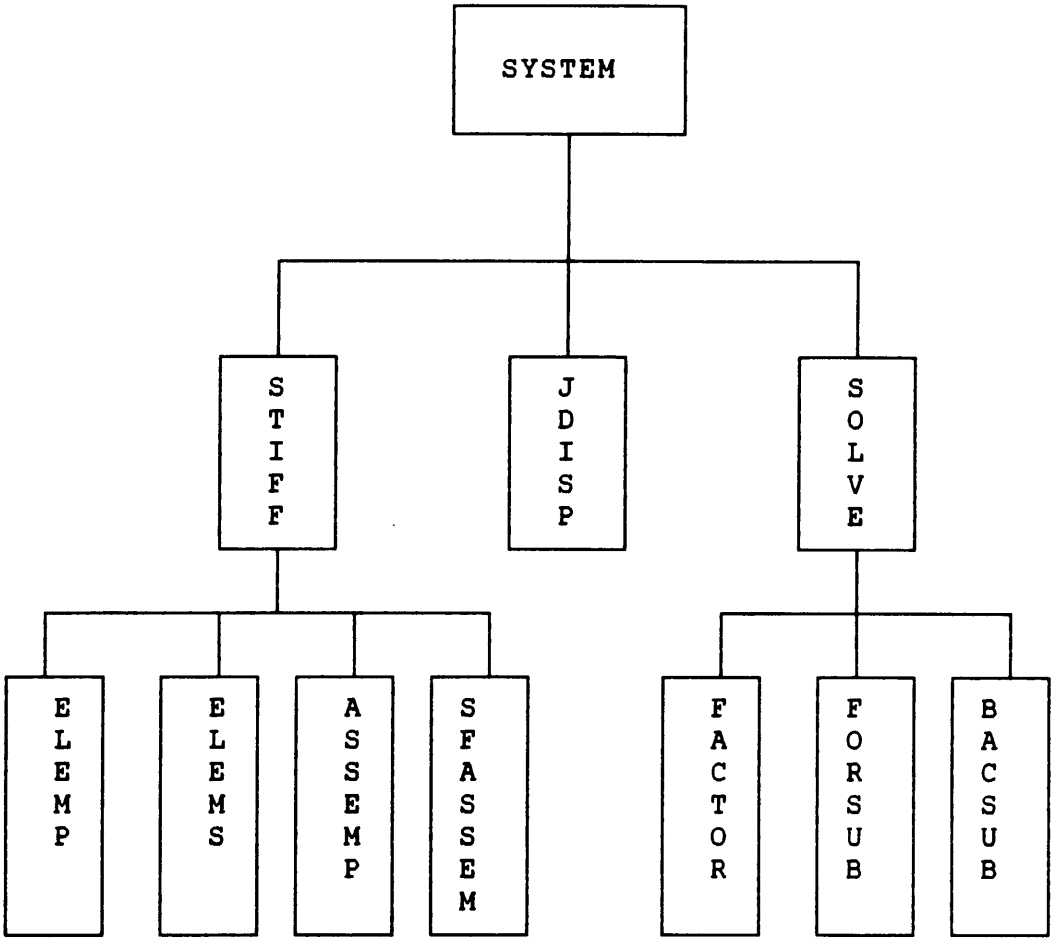


Figure 3.5: Tree Chart of Subroutine SYSTEM

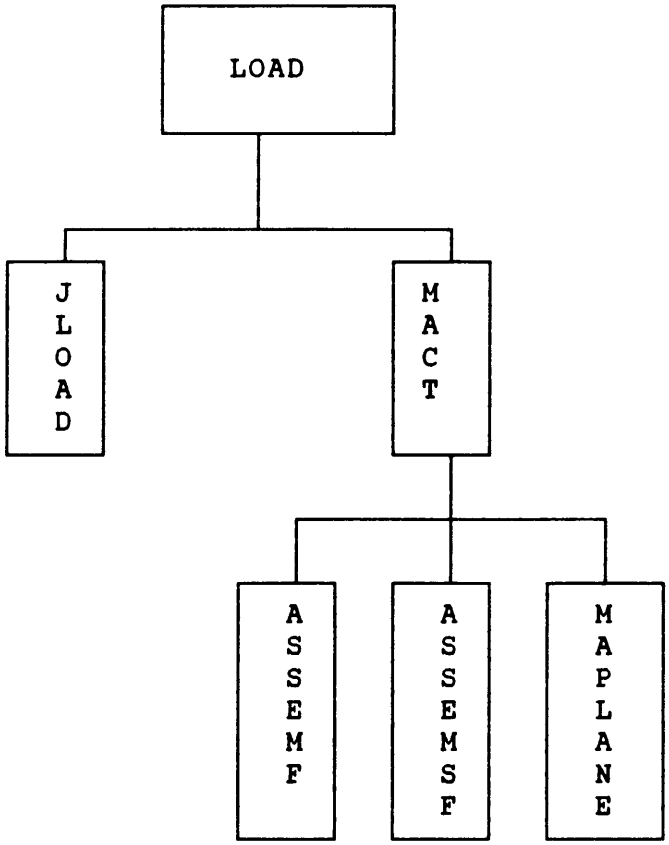


Figure 3.6: Tree Chart of Subroutine LOAD

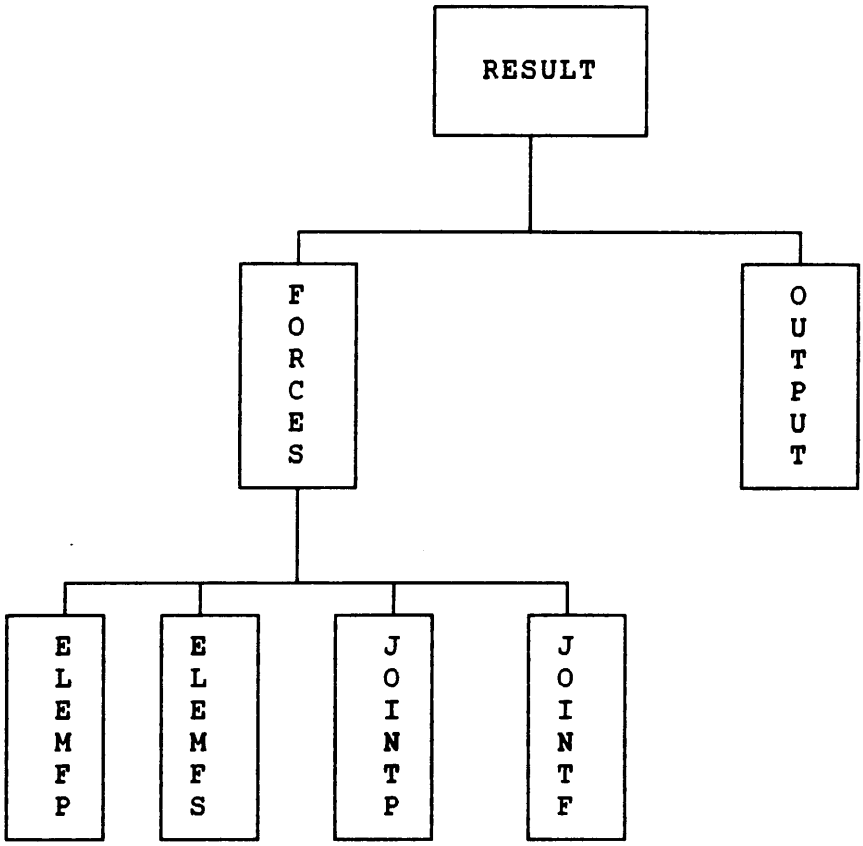


Figure 3.7: Tree Chart of Subroutine RESULT

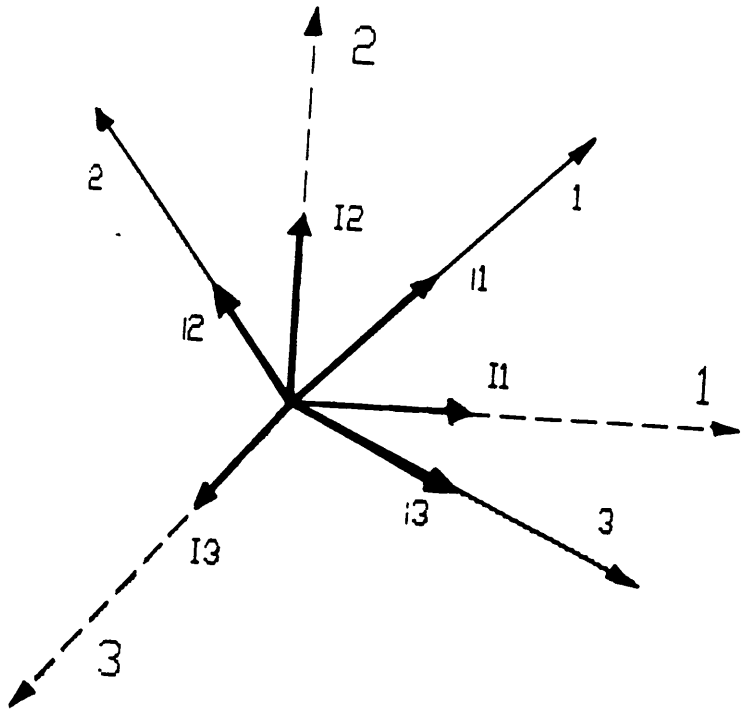


Figure 3.8: Local and Global Coordinates Axes

1. Specify a point P in the local 1-2 plane. The point P may not lie on or near the local 1 axis.
2. Specify either i_2 or i_3.
3. Specify the roll angle.

This Postprocessor uses the second option for space frames; it requires the components of i_2. The components of i_3 can be determined by the cross product $i_1 \times i_2$.

Note: For information regarding approaches 1 and 3 refer to Appendix C of Holzer, 1985.

3.1.1.2 Additional Material and Element Properties

The material and element properties that a user must define for plane frames are the modulus of elasticity, the coefficient of thermal expansion, the cross sectional area, and the moment of inertia about the local 3 axis. The additional information the user must define for the space frame is the shear modulus, moment of inertia about the local 2 axis, and the torsional constant.

3.1.1.3 Loading

The plane frame analysis program can handle three types of loading: joint loads, member actions, and prescribed joint displacements.

There are six different types of member actions, illustrated in Sections 4.5.1.2.1 through 4.5.1.2.6, that can be specified in the direction of the local coordinate system.

1. Concentrated transverse load.
2. Concentrated lateral load.
3. Uniformly distributed transverse load.
4. Uniformly distributed lateral load.
5. Trapezoidally distributed transverse load.
6. Temperature change.

These member actions can be extended to space frames. However, the actions that cause flexural deformation (such as, concentrated transverse load, uniformly distributed transverse load, and trapezoidally distributed transverse load) must lie in one of two local planes, the local 1-2 plane, or the local 1-3 plane defined by the principle axes of the cross section.

The additional information the space frame analysis program requires for the member actions that cause flexural deformations is the local plane in which they act (local 1-2 plane, or local 1-3 plane).

Prescribed joint displacements, caused for example by support settlements, have not been extended to the space frame program.

3.2 Space Trusses

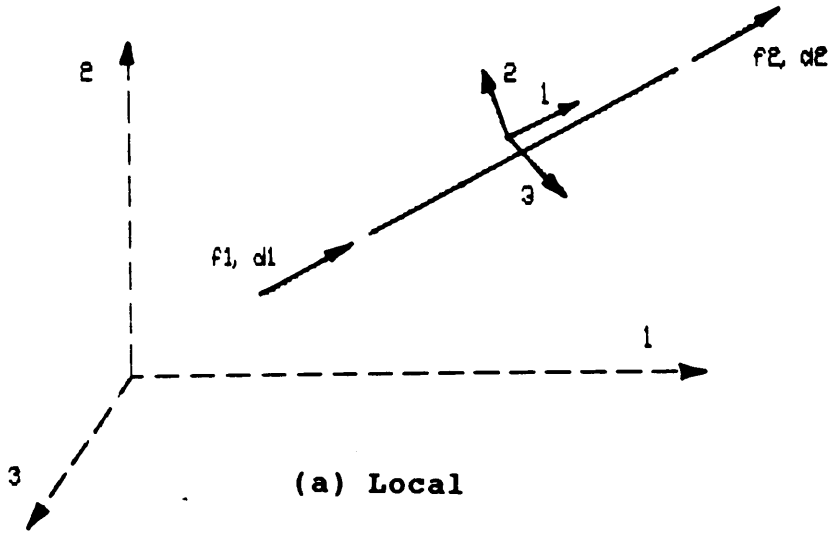
The space truss analysis program was developed as a special study project at VPI & SU. This program performs the matrix displacement analysis of trusses (Niezgoda, 1985) composed of prismatic elements which may have distinct geometric and material properties. A space truss element, illustrated in Figure 3.4, can only experience axial deformation; therefore, the twelve degree of freedom space frame element (Figure 3.2) is reduced to a six degree of freedom space truss element in Figure 3.9.

The element and material properties of a space truss are the **cross sectional area, modulus of elasticity,** and the **coefficient of thermal expansion.**

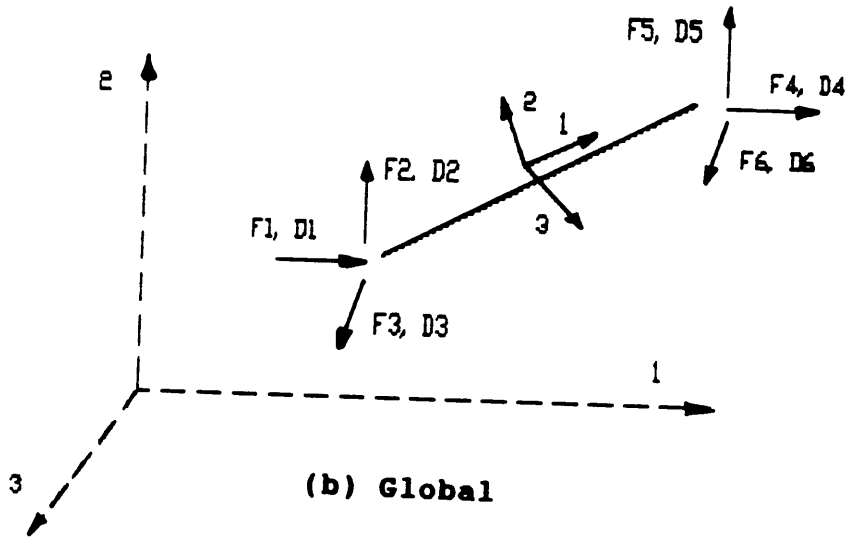
The Postprocessor for space trusses can handle multiple load conditions, and each load condition may consist of joint loads, member actions, and prescribed joint displacements. The only actions (Section 3.1.1.3) that space truss program

can handle are the ones that cause axial deformations (such as, concentrated lateral load, uniformly distributed lateral load, and temperature changes).

The Postprocessor for the space truss is illustrated in tree chart form in Figures 3.10 through 3.13.



(a) Local



(b) Global

Figure 3.9: Element Displacements and Forces for Space Trusses

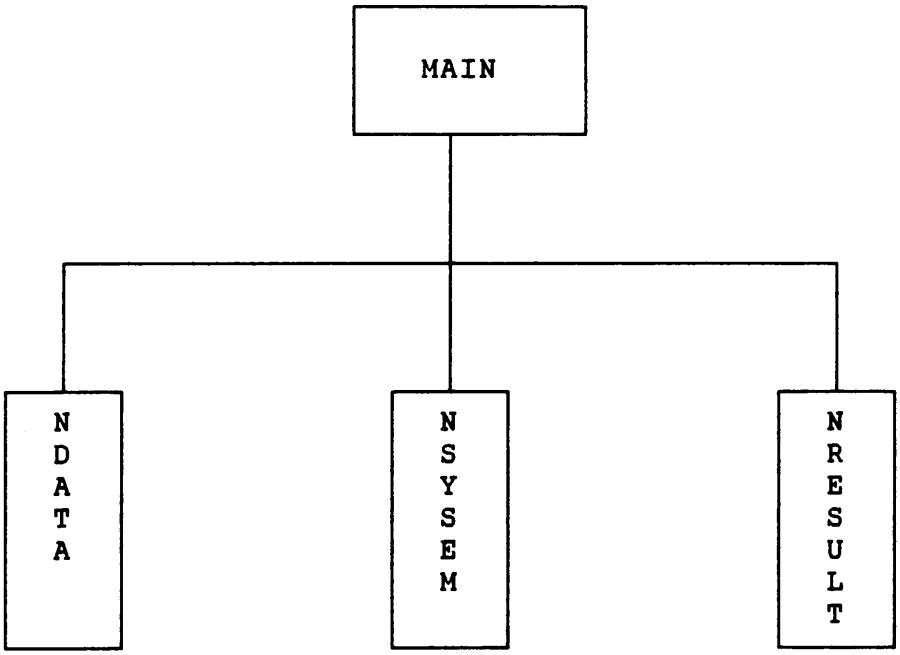


Figure 3.10: Tree Chart of Space Truss Structural Analysis Program

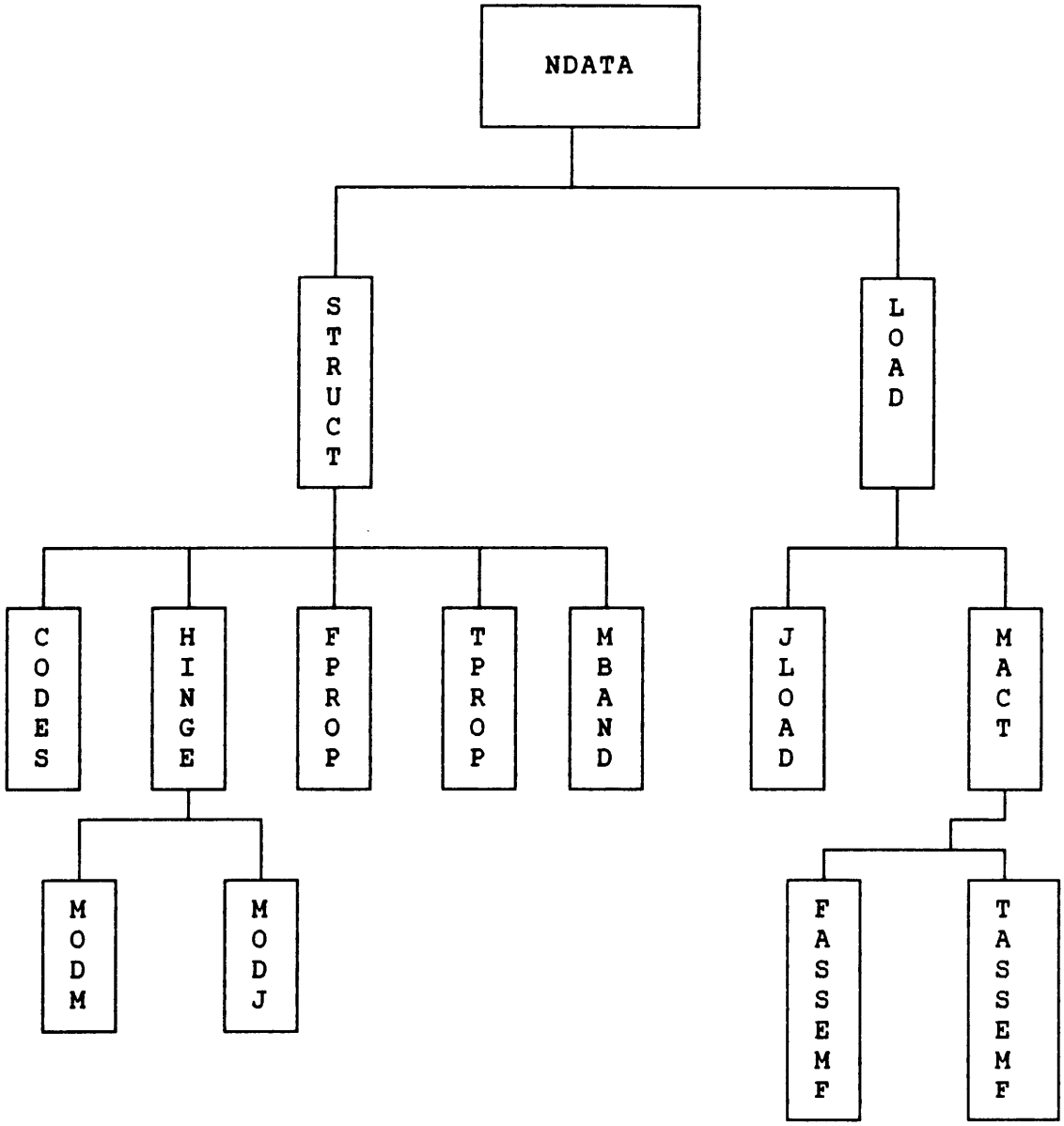


Figure 3.11: Tree Chart of Subroutine NDATA

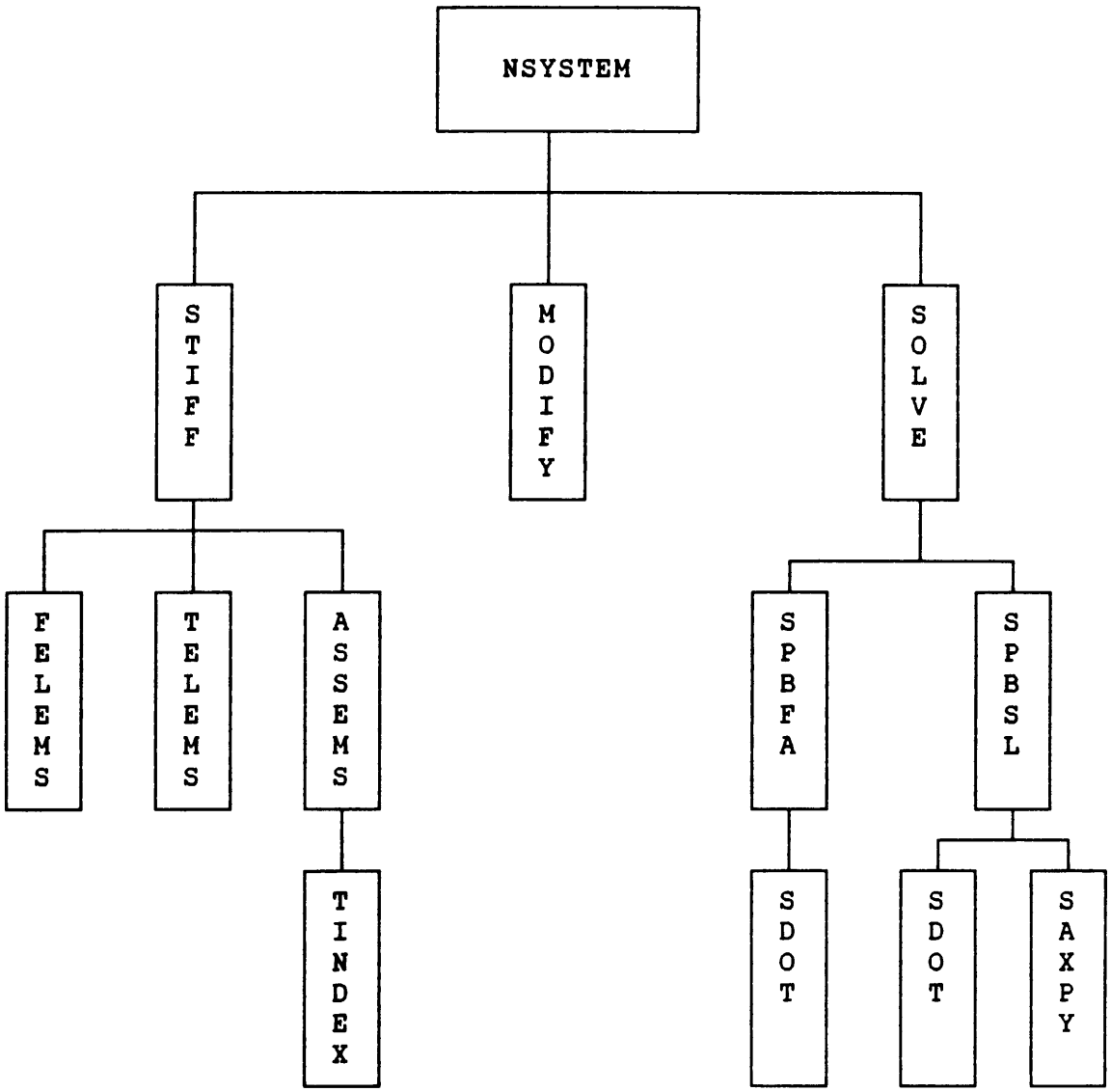


Figure 3.12: Tree Chart of Subroutine NSYSTEM

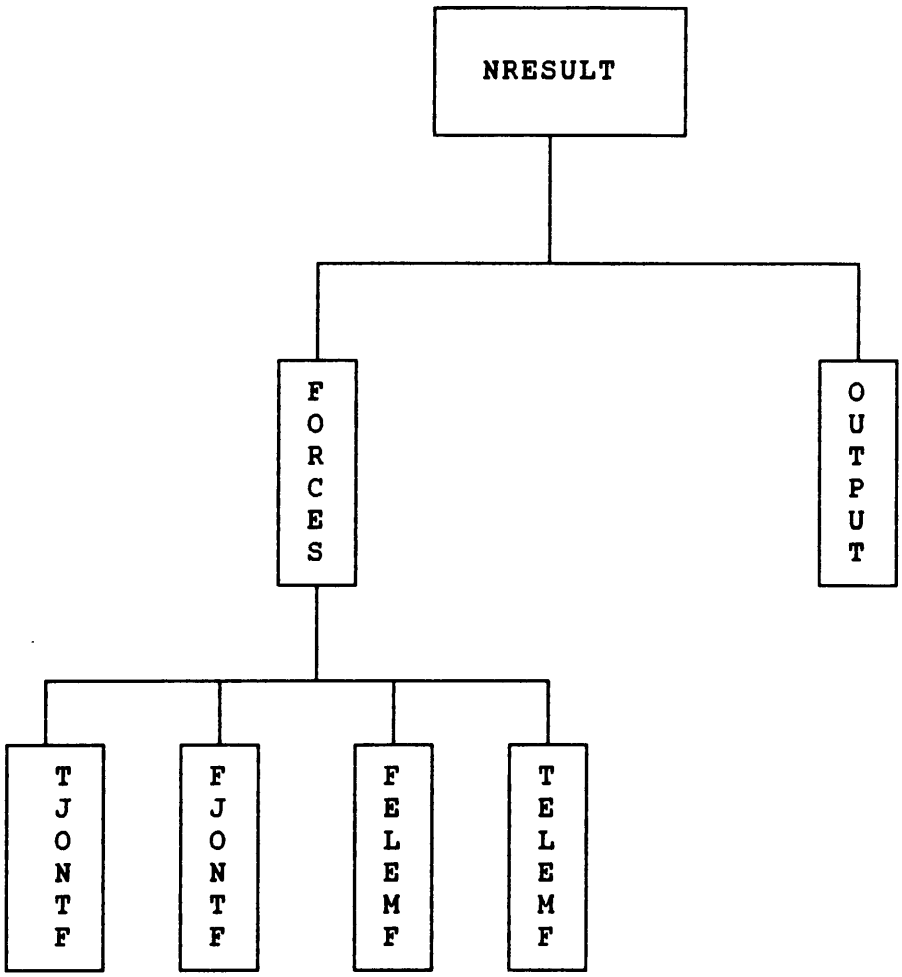


Figure 3.13: Tree Chart of Subroutine NRESULT

Chapter 4

User Manual

4.1 Introduction

4.1.1 Overview

This chapter is comprised of the general descriptions of the preprocessor and its supporting programs and how to use them. As mentioned in Section 2.4, the preprocessor is an interactive program allowing the user to enter and modify data for an input data file. This chapter will also present examples of pop-up menus and data entry screens that was developed by using an interface software package C-Scape 2.1 (Section 2.2). These examples will be used to show the user the appearance of the display when selecting the information

from pop-up menus or entering it within a data entry screen.

An example of a plane frame is given to promote the user's understanding on the operation of generating and modifying input data files using this preprocessor (Section 4.2.3). The data structure of the preprocessor is described in Section 4.2.2. This structure shows what kind of information is required to be defined by the user in order to create an input data file. Sections 4.3 through 4.5 describe the data routines used by the preprocessor to create or modify (edit) the input data of a structure.

The preprocessor consists of a subprogram known as the editor that gives the user the capability of modifying existing input data files. This subprogram allows the user to select the data routine(s) to modify with a pop-up menu. The functions of this editor are described in detail in Section 4.6.

The supporting programs are the library manager and graphics presentation. Their functions are described in detail in Sections 4.6 and 4.7.

4.2 Preprocessor

4.2.1 Main Menu

Main Menu
Preprocessor
Editor
Create a new input file

4.2.1.1 Submenu Preprocessor

Preprocessor
Plane frames
Space frames
Space Trusses

4.2.1.1.1 Plane Frames

Mesh Generation for plane frames
Direct method
Automatic

4.2.1.1.2 Space Frames

Mesh Generation for plane frames
Direct method
Automatic

Note: Automatic mesh generation has not been extended for space frames.

4.2.1.1.3 Space Trusses

Mesh Generation for plane frames
Direct method
Automatic

Note: Automatic mesh generation has not been extended for space trusses.

4.2.2 Data Structure

The data configuration (structure) of the preprocessor is divided into four regions: the control variables, the structural and element data, the material and element properties, and the loading. Each area consists of data

routines. The functions of a data routine are: to retrieve information (by the use of pop-up menus or data entry screens) from the user, to display the information entered in tables, to determine if any errors occurred while entering the information, and to store the information on the input data file. A flow diagram of a data routine is illustrated in Figure 4.1.

The following describes the data that is required to define each of the data routines in the four regions:

1. Control variables
    - Direct method
        - Number of members
        - Number of joints
        - Number of load conditions
    - Automatic method
        - Number of stories
        - Story height
        - Number of bays
        - Bay width
        - Number of load conditions
2. The structural and element data
    - Member incidences

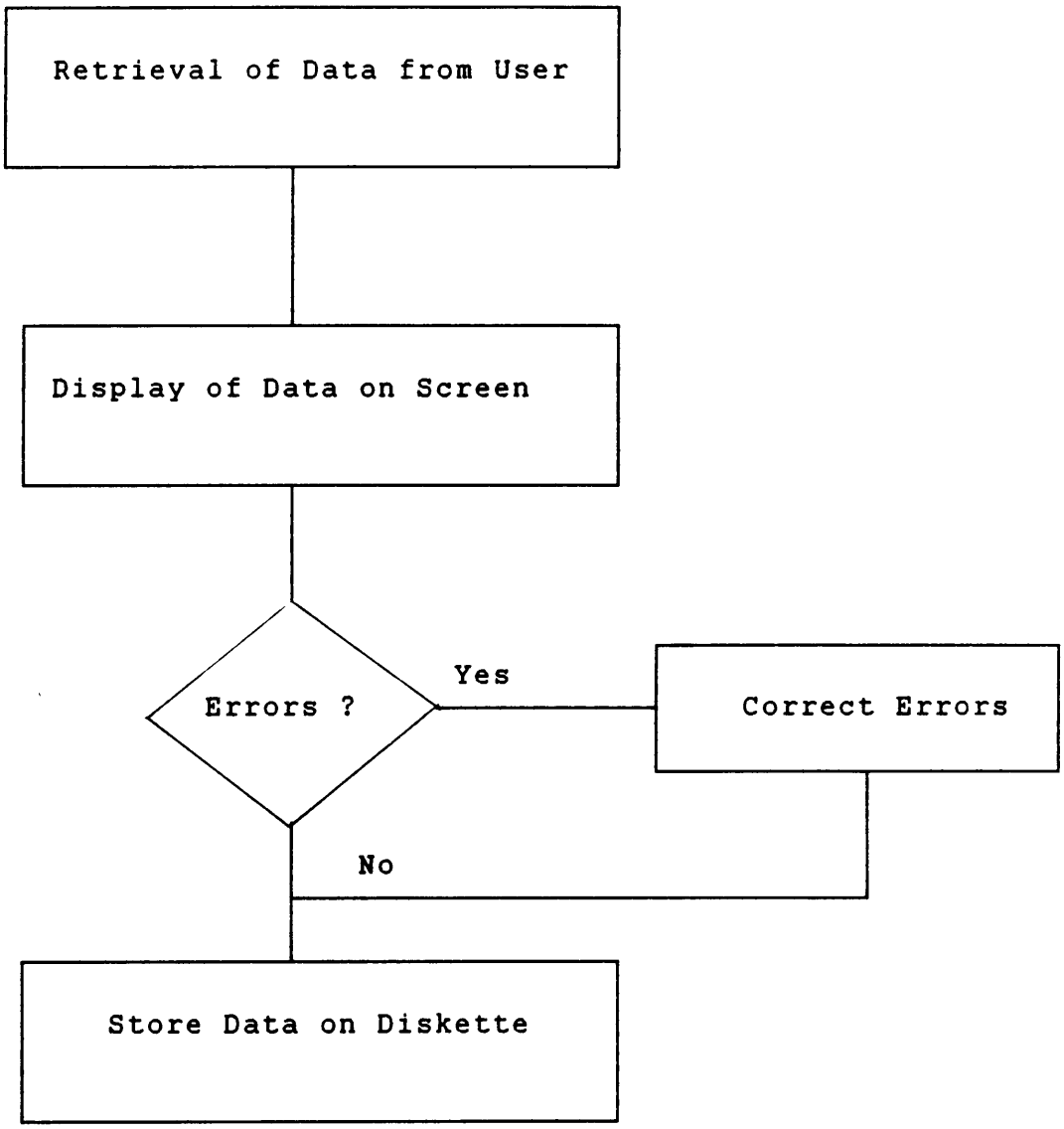


Figure 4.1: Flow Diagram of a Data Routine

A end, b end member incidences

Joint constraints

Constrained joint number

Type of constraint (Tables 4.1 through 4.3)

Joint Coordinates

Two dimensional structures

Global 1, global 2 directions

Three dimensional structures

Global 1, global 2, global 3 directions

Directions cosines (only space frames)

Direction cosines of the local 2 unit vector

Global 1, global 2, global 3 directions

3. The material and element properties

Material properties

Modulus of elasticity

Shear modulus

Coefficient of thermal expansion

Element properties

Area

Moment of inertia Ixx

Moment of inertia Iyy

Torsional constant J

4. The load data

- Joint loads
  - Joint number
  - Joint direction
  - Joint force or moment
- Member Actions
  - Member number
  - Load type (Table 2.1)
  - Distance(s) where load occurs
  - Planes in which loads act (only space frames)
    - 1 - 2 Plane
    - 1 - 3 Plane
- Prescribed joint displacements
  - Joint number
  - Joint direction
  - Joint deflection or rotation

4.2.3 Entering an Input Data File

When the preprocessor is selected from the **slug** menu in Section 4.2.1, a pop-up menu in Section 4.2.1.1 is displayed allowing the user to determine what type of input data file to generate. After selecting the type, another pop-up menu in Sections 4.2.1.1.1, 4.2.1.1.2, or 4.2.1.1.3 is displayed to

determine the type of mesh generation.

Note: The user can move between the submenus in Section 4.2.1.2 to the main menu for the preprocessor by pressing ESCAPE. If ESCAPE is pressed within the menu for the preprocessor, the **slug** menu in Section 4.2.1 is redisplayed allowing further selection.

After selecting the type of mesh generation from the pop-up menu, the preprocessor will prompt the user to enter all of the information necessary to build an input data file for the structural analysis program chosen.

As mentioned in Section 4.2.2, the preprocessor organizes its data into four regions that consist of data routines. On entering the data into these data routines, there are two methods for the user to follow:

One method of entry is simple because the user must enter the information and press **ENTER** after each entry. This is because the number of entries are known once the control variables (Section 4.2.3.2) (such as the number of members, the number of joints, and the number of load conditions) have been specified. Once these have been determined, the data routines

member incidences (Section 4.3.1.1), joint coordinates (Section 4.3.1.4), and direction cosines (Section 4.3.1.5) can be determined by cycling through from one to the number of members and joints. If ESCAPE is pressed within these data routines, the data screen is cleared and the user must reenter the information for that member or joint. The other method has an unknown number of entries for the data routines joint constraints (Section 4.3.1.2), internal hinges (Section 4.3.1.3), and the loads (Section 4.5). In the latter situation, the data entry must be terminated by pressing ESCAPE.

Since the Preprocessor is an interactive program, an example of generating an input data file for a plane frame (illustrated in Figure 4.2) is given. Before creating this input data file, the user must enter a valid file name (Section 4.2.3.1) after which a pop-up menu is displayed to determine the units of input.

Units
US
SI

Thus, the two types of units the preprocessor accept are US and SI. US units are kips, inches, radians, and Fahrenheit,

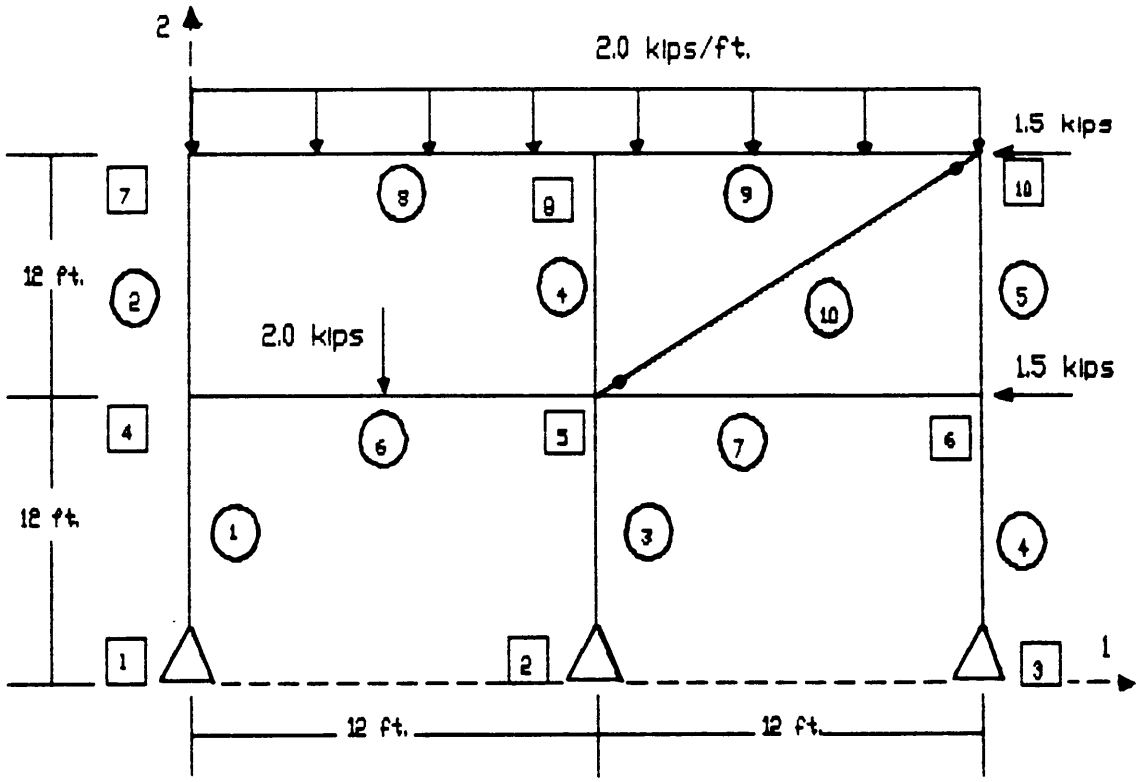


Figure 4.2: An Example of a Plane Frame

while SI units are kN, meters, radians, and Celsius.

To complete the data entry for this input data file, the user must enter the control variables (Section 4.3.3.2), the structural and element data (Section 4.3), the material and element properties (Section 4.4), and the load data (Section 4.5). In each of these areas, the user will be exposed to the pop-up menus and data entry screens that will be used to retrieve the information to build this input data file. These areas also have modifying routines that allow the user to edit (modify) any incorrect data entries.

4.2.3.1 Valid Names of Input Data Files

After selecting the type of mesh generation, a data entry screen is displayed for the user to enter the file name of the input data file.

Data Filename	
Filename :	test
Enter the name of input file	

A valid file name for a user file is any combination of eight alphanumerics and numeric characters (accept 0). Special characters like period, comma, colon, etc. are not accepted. For example, consider the input data file called 'test'. On the diskette the input data file specification is 'TEST.DAT'. When the user enters a file name, the preprocessor will search for a name identical to that. If it exist the preprocessor will rename the old input file as a backup file. This backup file is distinguished by the filename extension '.BAK'.

There are only two areas where the user must enter a valid file name that is the preprocessor and the graphics presentation. For the library manager, it uses distinct file names. The file name for the material library is 'MATERIAL.LIB', while the file name of the element library is 'MEMBER.LIB'. **Note:** The user must not erase these libraries because the preprocessor will display an error message and terminate its execution.

4.2.3.2 Control Variables

<u>Limitations</u>

Maximum allowable number of members	$\leq$ 200
Maximum allowable number of joints	$\leq$ 200

Maximum allowable number of load conditions $\leq$ 5

4.2.3.2.1 Direct Method Mesh Generation

Data entry screen:

Control Variables		
Number of members	:	10
Number of joints	:	9
Number of load conditions	:	2
Consider dead load ?	:	Yes
Enter the number of elements		

Output display:

Control Variables

Number of members	:	10
Number of joints	:	9
Number of load conditions	:	2
Consider dead load ?	:	Yes

4.2.3.2.2. Automatic Mesh Generation

Limitations

A mesh can not exceed the maximum allowable number of joints, members, and load conditions in Section 4.3.2.2. Automatic mesh generation can only be used for plane frames.

Data entry screen:

```
Control Variables
------------------------------------------------------------
   Number of stories      :  [2              ]

   Story height           :  144        in

   Number of bays         :  2

   Bay width              :  144        in

   Number of load conditions  :  1

   Consider dead load ?       :  Yes
------------------------------------------------------------
Enter the number of stories
```

Output display:

```
                Control Variables :
            Number of members          :  10
            Number of joints           :  9
```

```
Number of load conditions :  1
Consider dead load ?       :  Yes
  Mesh Generation :
Number of stories    :  2
Story height         : 144.0000
Number of bays       :  2
Bay width            : 144.0000
```

For automatic mesh generation, the preprocessor creates the necessary information for the member incidences (Section 4.3.1.1) and the joint coordinates (Section 4.3.1.4); thus, the user is not required to enter them. The numbering scheme used for a plane structure using automatic mesh generation is described in Appendix A.

4.3 The Structural and Element Data

This Section consists of the structural and element data routines that are used in the development of an input data file. The structural and element data routines consist of member incidences (Section 4.3.1.1), joint constraints (Section 4.3.1.2), internal hinges (Section 4.3.1.3), joint coordinates (Section 4.3.1.4), and direction cosines (Section 4.3.1.5).

The method of entry for each of these data routines was discussed in Section 4.2.3; that is, for member incidences, joint coordinates, and direction cosines, the user must press ENTER after every entry, while for joint constraints, and internal hinges, the user must press ESCAPE to terminate the data entry.

To modify the structural and element data routines, the user can refer to Section 4.3.2; that is, the member incidences, joint coordinates, and direction cosines (Section 4.3.2.2), joint constraints (Section 4.3.2.3), and internal hinges (Section 4.3.2.4).

4.3.1 Entering the Structural and Element Data

4.3.1.1 Member Incidences

Data entry screen:

```
Member Incidences

   Member number  :    1

     a end        :  1

     b end        :  3

Enter joint number at a end of member
```

Output display:

Member Incidences

Member Number	a end	b end
1	1	4
2	4	7
3	2	5
4	5	8
5	3	6
6	6	9
7	4	5
8	5	6
9	7	8
10	8	9

4.3.1.2 Joint Constraints

Data entry screen:

```
Joint Constraints

     Joint number  :  1

Enter the joint to be constrained
```

Output display:

Joint Constraints

Joint Number	Constrained Directions
1	1
1	2
2	1
2	2
3	1
3	2

4.3.1.2.1 Pop-up Menus for Selecting Joint Constraints

4.3.1.2.1.1 Plane Frames

Main pop-up menu:

Choose joint constraints
Fixed
Pinned
Roller
Shear Release
Other

Submenus:

Roller	
Horizontal direction	(1)
Vertical direction	(2)

Shear Release	
Horizontal direction	(1)
Vertical direction	(2)

Horizontal and vertical directions denote the roller's and shear release's direction of movement (i.e., in (1), or in (2) global direction). The roller and shear release joint constraints are illustrated in Table 4.1.

4.3.1.2.1.2 Space Frames

Choose joint constraint
Fixed
Pinned
Other

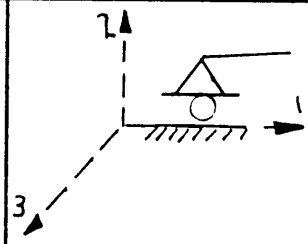
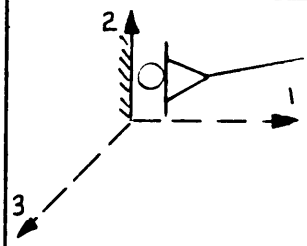
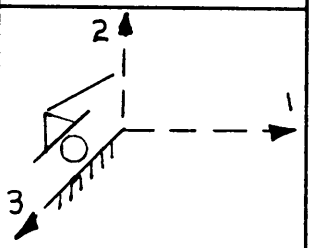
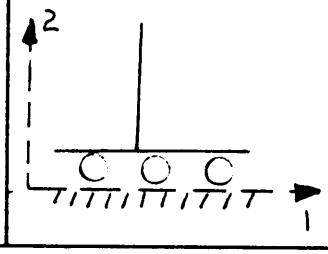
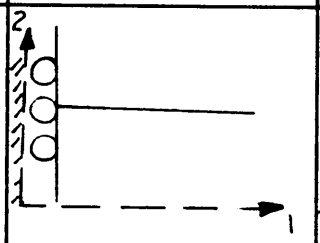
4.3.1.2.1.3 Space Trusses

Main pop-up menu:

Choose joint constraints
Pinned
Roller
Other

Table 4.1

Roller and Shear Release Joint Constraints

Name of Constraint	Global Directions		
	1	2	3
Roller			
Shear Release			

Submenus :

Roller	
Horizontal direction	(1)
Vertical direction	(2)
In the plane	(3)

The horizontal, vertical, in the plane directions (i.e., in (1), in (2), or in (3) global direction) denote the roller's direction of movement. The roller joint constraint is illustrated in Table 4.1.

4.3.1.2.2 Other Constraints

If the joint constraint is not listed within the pop-up menus in Sections 4.3.1.2.1.1 through 4.3.1.2.1.3 , the user has a choice of directly entering it by selecting the "Other" option from these pop-up menus (Section 4.3.1.2.1).

4.3.1.2.2.1 Plane Frames and Space Trusses

Data entry screen:

```
Other Joint Constraint
Press Y key for Yes and N key for No
                                  Constrained in the
                                global 1 direction : [Yes]
 Constrained
 joint number :  1              global 2 direction : Yes
                                global 3 direction : Yes
Is it constrained in the global 1 direction
```

4.3.1.2.2.2 Space Frames

Data entry screen:

```
Other Joint Constraints
  Press Y key for Yes and N key for No
                                  Constrained in the
                                global 1 direction : [Yes]
                                global 2 direction : Yes
```

Constrained joint number : 1	global 3 direction : Yes
	global 4 direction : Yes
	global 5 direction : Yes
	global 6 direction : Yes
Is it constrained in the global 1 direction	

4.3.1.3 Internal Hinges

<u>Limitations</u>

Maximum allowable number of internal hinges $\leq$ 20

Data entry screen:

Internal Hinges	
Member number :	7
Member end :	a end
Enter the member number	

Output display:

Internal Hinges

Hinge Number	Member Number	Hinge Location
1	11	a end
2	11	b end

4.3.1.4 Joint Coordinates

Data entry screen: (two dimensional structures)

```
Joint Coordinates

  Joint number  :   1

  Global  1     : 0.0

  Global  2     : 0.0

 Enter the joint coordinate in global 1 direction
```

Data entry screen: (three dimensional structures)

```
Joint Coordinates

  Joint number  :   1

  Global  1     :  0.0
```

Global 2 : 0.0	
Global 3 : 0.0	
Enter the joint coordinate in global 1 direction	

Output display for two dimensional structures:

Joint Coordinates

Joint Number	Global 1	Global 2
1	0.0000	0.0000
2	144.0000	0.0000
3	288.0000	0.0000
4	0.0000	144.0000
5	144.0000	144.0000
6	288.0000	144.0000
7	0.0000	288.0000
8	144.0000	288.0000
9	288.0000	288.0000

4.3.1.5 Direction Cosines

Direction cosines are only required for space frames (Section 3.1.1.1).

Data entry screen:

Direction Cosines
Enter the direction cosines of the unit vector in the local 2 direction

```
Member number  :   1

Global   1     :  0.0

Global   2     :  0.0

Global   3     :  0.0

Enter the direction cosine in the global 1 direction
```

Output display for the structure in Figure A.3:

Direction Cosines

Member Number	Global 1	Global 2	Global 3
1	0.0000	1.0000	0.0000
2	0.0000	1.0000	0.0000
3	0.5539	0.3846	0.7385

4.3.2 Modifying the Structural and Element Data

4.3.2.1 Methods of Terminating Data Entry Screens When Modifying

This Section describes how to exit a data entry screen after modifying the contents in them. After the data entry screen

is displayed, the user can modify or change the incorrect entry within this data entry screen by using the arrow and editing (such as the insert, the backspace, and the delete) keys. Thus, the user can terminate the data entry by placing a highlight on the field that is at the bottom of the data entry screen and press ENTER, or by pressing ESCAPE which will display a pop-up menu to determine if the previously entered data is to be saved.

Save previous entered data ?
Yes
No

4.3.2.2 Modifying Member Incidences, Joint Coordinates and Direction Cosines

To modify either member incidences (Section 4.3.1.1), joint coordinates (Section 4.3.1.4), or direction cosines (Section 4.3.1.5), a data entry screen will be displayed allowing the user to enter a member or a joint number to edit. An example of a data entry screen for member incidences is shown:

```
Member Incidences

    Member number  :  1

Enter the member to edit
```

Another data entry screen in either Sections 4.3.1.1, 4.3.1.4, or 4.3.1.5 is displayed after the user enters the member or joint number to edit. An example of data entry screen for member one is shown for member incidences allowing the user to change the incorrect entry.

```
Change the incorrect entry
```

```
Member Incidences

  Member Number    :  1

    a end      :  1

    b end      :  4

Enter joint at the a end of the member
```

Thus, the user can modify or change the data entry by referring to Section 4.3.2.1. After terminating the data entry, the data entry screen for determining additional member

or joint numbers is redisplayed. If ESCAPE is pressed within this data screen, the modification of member incidences, joint coordinates, and direction cosines is terminated.

4.3.2.3 Modifying Joint Constraints

To modify joint constraints, a data entry screen is displayed allowing the user to enter the joint number to edit (Section 4.3.1.2) after which a pop-up menu in Section 4.3.1.2.1 is displayed allowing a new joint constraint to be selected. After selecting a new constraint, a data entry screen previously mentioned is redisplayed to allow additional joint numbers to be edited. As stated in Section 4.3.1, the user can terminate the data entry for joint constraints by pressing ESCAPE.

Note: The preprocessor will allow additional joint constraints to be added only if the joint selected is not already constrained.

4.3.2.4 Modifying Internal Hinges

To modify internal hinges, a pop-up menu will appear below the table for internal hinges (Section 4.3.1.3) allowing the user to manipulate them by adding additional hinges (Section 4.3.2.4.1), erasing hinges (Section 4.3.2.4.2), and edit previously entered ones (Section 4.3.2.4.3).

Edit
Add
Erase
Edit

To terminate the data entry for modifying the internal hinges, the user must press ESCAPE.

4.3.2.4.1 Adding Internal Hinges

When this option is selected, a data entry screen in Section 4.3.1.3 is displayed allowing additional internal hinges to be added to the previously ones entered in Section 4.3.1.3. As stated in Section 4.3.1, the user must terminate the data entry screen by pressing ESCAPE which will display a table of internal hinges along with the pop-up menu for selecting additional internal hinges to edit.

4.3.2.4.2 Erasing Internal Hinges

This option allows the user to erase internal hinges that have been previously entered in Sections 4.3.1.3 or 4.3.2.4.1. When this option is selected, a data entry screen is displayed to determine which hinge numbers the user wants to erase.

```
Erase Internal Hinges
------------------------------------------------------
  First hinge number   : [1          ]

  Last hinge number    :  2

  hinge increment number :  1
------------------------------------------------------
Enter the first hinge number to erase
```

To terminate the data entry, the user must press ESCAPE which will display the erased internal hinges in a table. By selecting these erased internal hinges, a new table is created along with the pop-up menu (Section 4.3.2.4) for selecting additional internal hinges to edit.

4.3.2.4.3 Editing Internal Hinges

This option allows the user to edit previously entered internal hinges in Section 4.3.1.3 or 4.3.2.4.1. When this option is selected, a data entry screen is displayed to determine which hinge number to edit.

Internal Hinges
Hinge Number :
Enter the hinge number to edit

After entering the hinge number, a data entry screen in Section 4.3.1.3 is displayed allowing the user to change or modify the data entry within it. After terminating the data entry (Section 4.3.2.1), the data entry screen for selecting the additional hinge numbers to be modified is redisplayed. Thus, if ESCAPE is pressed, the edit option for internal hinges will be completed which will display the hinges in a table along with the pop-up menu in Section 4.3.2.4.

4.4 The Material and Element Properties

This Section consists of material (Section 4.4.1.1) and element (Section 4.4.1.2) properties that are used in the

development of an input data file.

The data entry for material properties is simple because the user only has to select the material type from the menu and press ENTER. On the other hand the data entry for element properties is complex because the user must enter or select the element properties and assign them to the members of the structure.

To modify the material and element properties, the user can refer to Section 4.4.2, that is material properties (Section 4.4.2.1), and element properties (Section 4.4.2.2).

4.4.1 Entering the Material and Element Properties

4.4.1 Material Properties

After selecting the units in Section 4.2.3, the preprocessor will search the material library for the material types (i.e., the names of the material) with the selected units and place these in a data entry screen.

Material Properties	
Material Type:	
steel	concrete_4ksi

To select the material type, a highlight can be moved within this data entry screen by using the arrow keys and selecting it by pressing ENTER.

Output display:

Material Properties

Units : ksi and strain / degree of Fahrenheit

Material Type	Modulus of Elasticity	Shear Modulus	Temperature Coefficient
steel	29000.0000	11200.0000	6.50E-05

4.4.1.2 Element Properties

Limitations

Maximum allowable number of element sets can be defined $\leq$ 20.

The preprocessor requires the user to develop element sets that contain the information about the element properties and

the members which have these properties. An element set contains the element properties the user has chosen by either entering them directly, or by selecting them from the element library that contains the AISC steel section properties (Section 4.4.1.2.1). Thus, the user can choose these from a pop-up menu for a element set.

Element set number : 1

Method of entry ?
Direct
AISC Manual

The number of element sets the user must define depends on how many different element properties the structure requires; for example, in Figure 4.2, the user must define four element sets containing the following element properties: for all columns W30x132, for all beams W36x300, for the roofing M14x18, and for the bracing W16x100. All of these element properties can be selected from the element library. Thus, the method of entry for each element set is the 'AISC Manual' (Section 4.4.1.2.1.2).

By pressing ESCAPE, the selection of element properties for the element sets is terminated after which the user must

determine how they are assigned to the members of the structure (Section 4.4.1.2.2).

4.4.1.2.1 Method of Entry

Limitations

Only symmetric cross sections can be entered.

4.4.1.2.1.1 Direct Method

When this option is selected, a data entry screen is displayed allowing the user to enter the element properties directly.

Data entry screen: (example W 36x300 section)

```
Direct Entry

  Weight    : [300.0        ]

  Area      :  83.30

  Moment of inertia Ixx :  20300.0
                    Iyy :  1300.0

  Torsional constant J  :  64.20

Enter the weight of the cross section
```

If ESCAPE is pressed within this data entry screen, the pop-up menu for selecting the method of entry is redisplayed (Section 4.4.1.2) for the same element set.

4.4.1.2.1.2 AISC Manual

This option allows the user to select element properties from an element library containing AISC steel cross sections. In order to select properties from the library, the user must choose the shape of the cross section from a pop-up menu.

Select section shape
W Shape
M Shape
MT Shape
WT Shape
C Shape
Other Shape

After determining the shape of the cross section, a data entry screen is displayed showing the names of the steel cross sections that exist in this library for the selected shape. By using the arrow keys, a highlight can be positioned over

the name of the cross section, and the user can select it by pressing ENTER.

Data entry screen: (W Shapes)

W Shapes			
36x300	32x241	30x132	27x102
24x76	21x68	18x60	18x35
16x100	16x26	14x132	14x53
12x152	12x53	12x14	10x112
10x60	10x45	8x40	8x15
6x25			

If ESCAPE is pressed within this data entry screen, the pop-up menu for selecting the section shape is redisplayed allowing the user to reselect another shape; otherwise, if ESCAPE is pressed within the pop-up menu, a pop-up menu in Section 4.4.1.2.1 is re-displayed to determine the 'method of entry' for the same element set.

4.4.1.2.2 Assignment of Member Properties

When ESCAPE is pressed to end the selection of the element sets, a pop-up menu for assigning them to the members is displayed.

Assignment of Member Properties
To all members
To members in a series
To members in alternating series

4.4.1.2.2.1 To All Members

Limitations

The preprocessor assumes strong axis bending when assigning the element properties to the members. The preprocessor assigns properties to the members from the first element set.

When this option is selected, the preprocessor will assign members from one to the number of members (Section 4.2.3.2) the same element properties.

4.4.1.2.2.2 To All Members in a Series

Limitations

Maximum allowable number of series can be defined ≤ 20.

When this option is selected, a data entry screen is displayed allowing members that have the same element properties to be assigned in a series.

Data entry screen (two dimensional structures)

```
Enter members in a series

  Element set number :  1

  First member number :  1

  Last member number  :  6

  Axis Bending        : Strong

Enter the element set number
```

Data entry screen: (three dimensional structures)

```
Enter members in a series

  Element set number :  1

  First member number :  1
```

Last member number : 6
Enter the element set number

Note: To complete the series or terminate the data entry for this option to all member in a series, the user must press ESCAPE.

Output display for one series containing the element properties of a W36x300 (strong axis bending):

Member Properties

Member : 7 through member : 8

Weight : 300.0000

Area : 83.3000

Moment of inertia Ixx : 20300.0000

4.4.1.2.2.3 To All Members in Alternating Series

<u>Limitations</u>

Maximum allowable number of series can be defined $\leq$ 20.

When this option is selected, a data entry screen is displayed allowing members that have the same element properties to be

assigned in alternating series.

Data entry screen: (two dimensional structures)

```
Enter members in alternating series

  Element set number : [1          ]

  First member number  :  1

  Last member number   :  2

  Member increment number :  1

  Axis bending            :  Weak

Enter the element set number
```

Data entry screen: (three dimensional structures)

```
Enter members in alternating series

  Element set number : [1          ]

  First member number  :  1

  Last member number   :  2

  Member increment number :  1

Enter the element set number
```

Output display for one series containing the element properties of a W30x132 (weak axis bending):

Member properties

member increment

Member : 1 through member : 2 number : 1

Weight : 132.0000

Area : 38.9000

Moment of inertia Ixx : 196.0000

4.4.2 Modifying the Material and Element Properties

4.4.2.1 Modifying Material Properties

To modify the material properties, a data entry screen in Section 4.4.1.1 is displayed allowing the user to reselect a new material type.

4.4.2.2. Modifying Element Properties

Limitations

Defining additional element sets is not allowed.

To modify the element properties, a pop-up menu in Section 4.4.1.2.2 is displayed. This pop-up menu will allow re-assignments of the element properties by the element sets (developed in Section 4.4.1.2.1) to the members.

4.5 The Load Data

After entering the number of load conditions in Section 4.2.3.2, the preprocessor will prompt the user for each load condition to determine if there are any joint loads, member actions, and prescribed joint displacements.

For joint loads (Section 4.5.1.1) and prescribed joint displacements (Section 4.5.1.3), a data entry screen is displayed allowing the user to enter the loads. As stated in Section 4.2.3, the user can terminate the data entry by pressing ESCAPE at which the loads will be displayed in a table.

For member actions (Section 4.5.1.2), the user must select the type of member action from a pop-up menu (Section 4.5.1.2). After selecting the member action, a data entry screen is

displayed along with an illustration of the action chosen (Sections 4.5.1.2.1 through 4.5.1.2.6). As stated in Section 4.2.3, the user can terminate the data entry for the selected member action by pressing ESCAPE after which a pop-up menu for selecting the type of member actions is redisplayed allowing further selection; otherwise, by pressing ESCAPE again, the data entry for member action is terminated. The loads will be displayed in a table.

To modify the load data routines, the user can refer to Section 4.5.2; that is, the joint loads and prescribed joint displacements (Section 4.5.2.1), and member actions (Section 4.5.2.2).

4.5.1 Entering the Load Data

4.5.1.1 Joint Loads

Limitations

Maximum allowable number of joint loads for each load condition $\leq$ 50. Also, a joint load can only act at a degree of freedom (Figures 3.1, 3.2, and 3.9).

Data entry screen:

```
Joint Loads
Units :  kips or kip-in.

Joint number     : 6

Joint direction :  1

Force or moment :  -1.50

 Enter the joint number
```

Output display:

Joint Loads

Load condition number : 1

Load Number	Joint Number	Joint Direction	Joint force or moment
1	6	1	-1.5000
2	9	1	-1.5000

4.5.1.2 Member Actions

<u>limitations</u>

Maximum allowable number of member actions for each load condition $\leq$ 50.

For plane and space frames (Section 3.1.1.3), there are six different types of member actions that can be specified in

direction of the local coordinate system. These six different types of member actions are displayed in a pop-up menu.

Select the type of member action
Concentrated transverse load
Concentrated lateral load
Uniformly distributed transverse load
Uniformly distributed axial load
Trapezoidally distributed transverse load
Uniform temperature change

For space trusses (Section 3.2), there are three different types of member actions and these are also displayed in a pop-up menu.

Select the type of member action
Concentrated lateral load
Uniformly distributed lateral load
Uniform temperature change

Output display:

Member Actions

Load condition number : 1

Load Number	Member Number	Load Type	Loads P or P1	Loads P2	Distances D or D1	Distances D2	Loads Plane
1	6	CTL	-2.00	0.00	72.00	0.00	1-2
2	8	DTL	-1.67E-02	0.00	0.00	0.00	1-2
6	9	TTL	-1.67E-02	-1.67E-02	0.00	144.000	1-2

4.5.1.2.1 Concentrated Transverse Load (CTL)

Data entry screen: (plane frames)

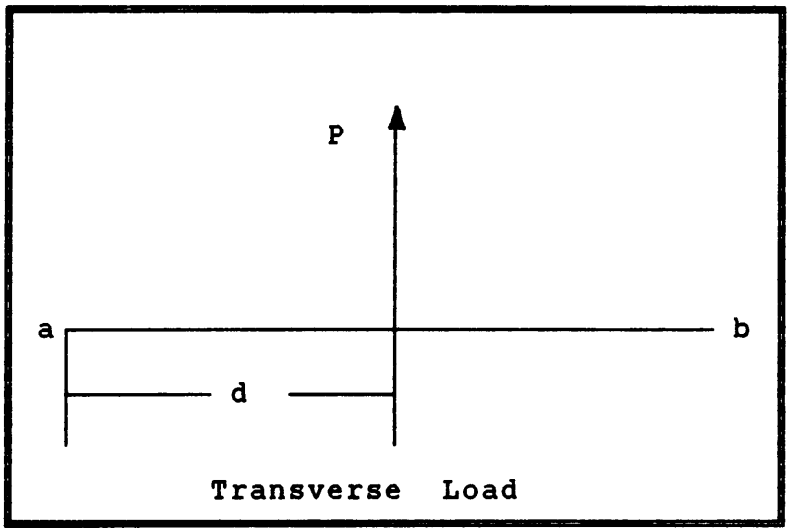
```
 Concentrated Transverse Load
Units :  kip, in.

Member number  : 6

Load     P     : -2.0

Distance d     :   72.0

Enter the member number
```

Data entry screen: (space frames)

```
 Concentrated Transverse Load
Units :  kip, in.

Member number  : 1
```

```
Load      P      :   -5.0

Distance d       :  100.0

Local plane      :  1-2

Enter the member number
```

Illustration of action:

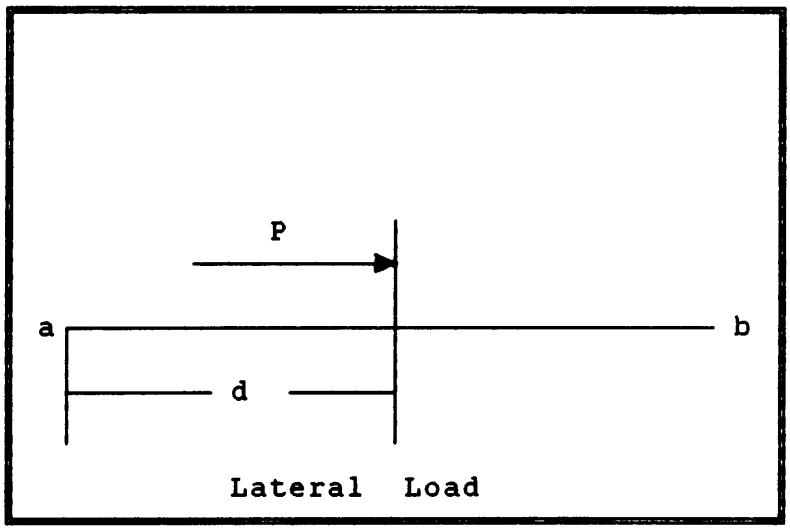


4.5.1.2.2 Concentrated Lateral Load (CLA)

Data entry screen:

Concentrated Lateral Load
Units : kip, in.
Member number : 6
Load P : -2.0
Distance d : 72.0
Enter the member number

Illustration of action:



4.5.1.2.3 Uniformly Distributed Transverse Load (DTL)

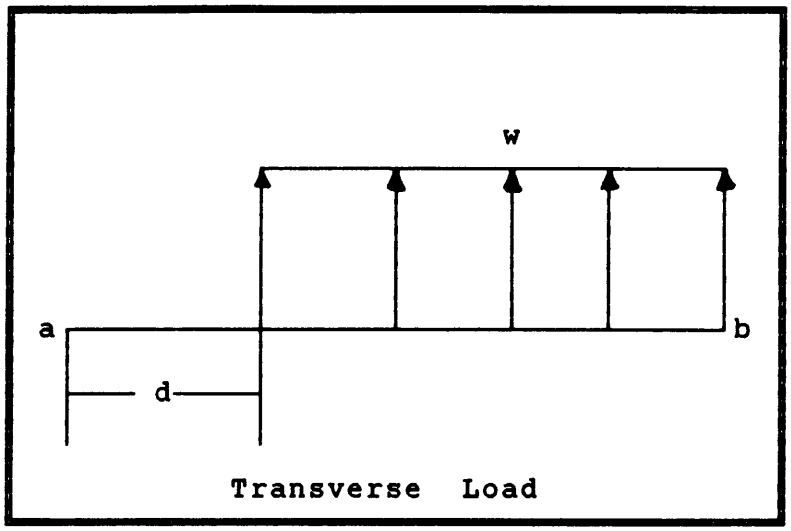
Data entry screen: (plane frames)

```
Distributed Transverse Load
Units :  kip / in., in.

Member number  : 8

Load      P     : -1.67E-02

Distance d      : 0.0

Enter the member number
```

Data entry screen: (space frames)

```
Distributed Transverse Load
Units :  kip / in., in.

Member number  : 5

Load      P     : -1.67E-02

Distance d      :  0.0

Local plane     :  1-2

Enter the member number
```

Illustration of action:

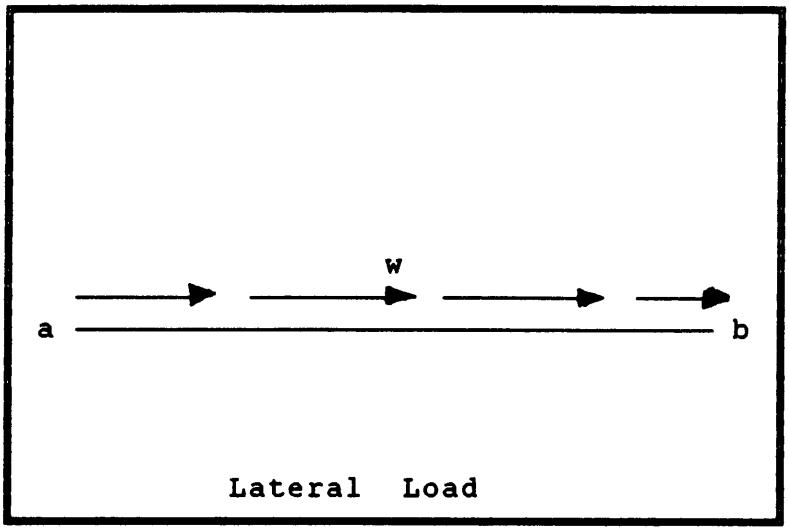


4.5.1.2.4 Uniformly Distributed Lateral Load (DLT)

Data entry screen:

Distributed Lateral Load
Units : kip / in., in.
Member number : 6
Load w : 1.67E-02
Enter the member number

Illustration of action:



4.5.1.2.5 Trapezoidally Distributed Transverse Load (TTL)

Data entry screen: (plane frames)

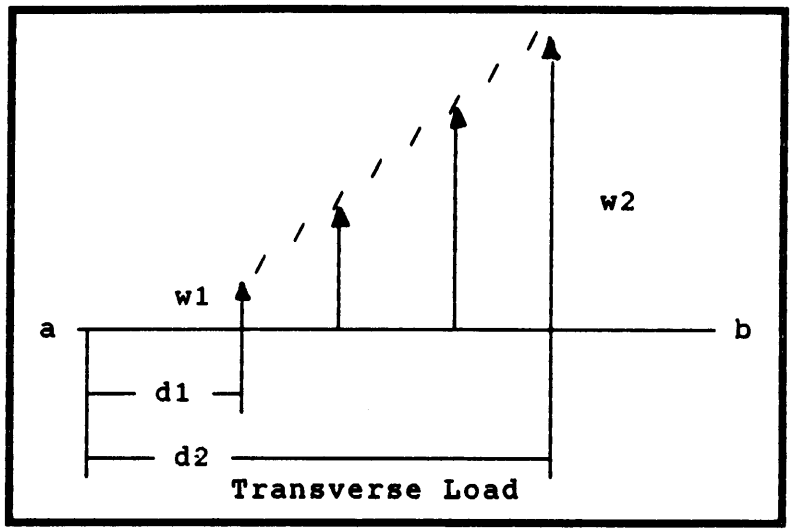
Trapezoidal Transverse Load		
Units : kip / in., in.		
Member number	:	8
Load w1	:	-1.67E-02
Distance d1	:	0.0

```
Load      w2     :  -1.67E-02

Distance d2      : 144.0

Enter the member number
```

Data entry screen: (space frames)

```
Trapezoidal Transverse Load

Units :  kip / in., in.

Member number   : 2

Load      w1     : -1.67E-02

Distance d1      :  0.0

Load      w2     : -8.33E-02

Distance d2      : 144.0

Local plane      :  1-2

Enter the member number
```

Illustration of action:

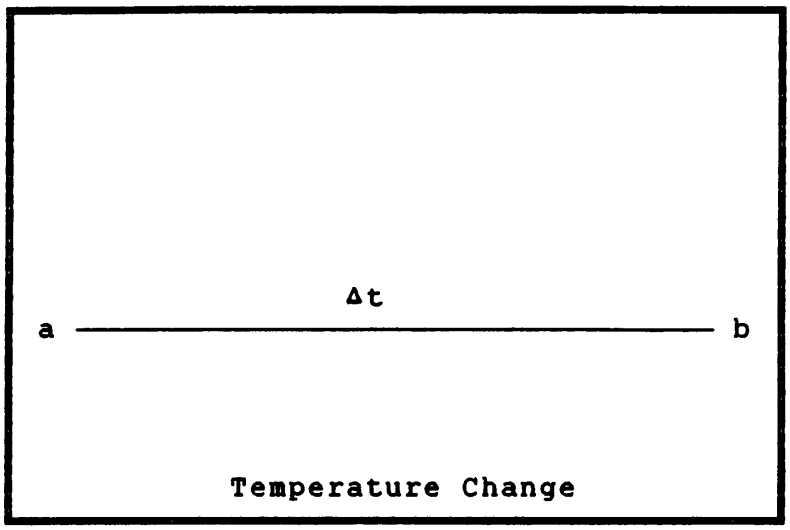


4.5.1.2.6 Uniform Temperature Change (UTC)

Data entry screen:

Uniform Temperature Change
Units : degree Fahrenheit Member number : 3 Temperature Change : 3.0
Enter the member number

Illustration of action:



4.5.1.3 Prescribed Joint Displacements

Limitations

Maximum allowable number of prescribed joint displacements for each load condition $\leq$ 50. A prescribed joint displacement can only act at a degree of freedom (Figures 3.1, and 3.9). Also, they are not allowed for space frames.

Data entry screen:

```
Prescribed Joint Displacements
Units : in. or radian

Joint number      : 2

Joint direction :  2

Deflection or rotation : 3.0

Enter the joint number
```

Output display:

```
Prescribed Joint Displacements

Load condition number :   1

Load      Joint       Joint       Joint deflection or
Number    Number      Direction   rotation
  1         2           2             3.0000
```

4.5.2 Modifying the Loads

To modify any of the loads (such as joint loads, member actions, and prescribed joint displacements), a data entry screen is displayed to determine which load number to edit. An example of data entry screen for joint loads is shown.

Joint Loads
Load number : 1
Enter the load number to edit

Note: The only difference in this data entry screen for member actions and prescribed joint displacements is the title.

To terminate the modifying routine for the loads, the user only has to press ESCAPE within this data entry screen to determine the load number to edit after which the loads will be displayed in a table.

4.5.2.1 Modifying Joint Loads and Prescribed Joint Displacements

The procedure for modifying joint loads and prescribed joint displacements is identical. After entering the load number to edit, a data entry screen in either Section 4.5.1.1 or 4.5.1.3 is displayed containing the information about the load number. For example, a load number of two is entered for joint loads. Thus, a data entry screen in Section 4.5.1.1 is displayed along with a message.

```
Change the incorrect entry

Joint Loads
Units :  kips or kip-in.

Joint number    :  6

Joint direction :  1

Force or moment :  -1.50

 Enter the joint direction
```

When the data entry screen is terminated (refer to Section 4.3.2.1) for joint loads or prescribed joint displacements, the data entry screen discussed in the previous section is re-displayed allowing the user to edit another load number or exit the modifying routine.

4.5.2.2 Modifying Member Actions

After selecting the load number to edit, a pop-up menu in Section 4.5.1.2 is displayed allowing the user to select the type of member action. If ESCAPE is pressed within this pop-up menu, the data entry screen (Section 4.5.2) for member

actions is redisplayed allowing the user to reselect another load number to edit or exit. When a user selects the type of member action, a data entry screen will appear along with a load illustration. **Note:** The preprocessor will not display the load information for the selected load number in the data entry screen because the incompatibilities in data entry screens in Sections 4.5.1.2.1 through 4.5.1.2.6. These incompatibilities can be related to the units used for each data entry screen (such as the loads), and the amount of information needed to define each member action. Thus, the user must define the complete information for the chosen member action.

After terminating the data entry (Section 4.3.2.1), the data entry screen in Section 4.5.2 is redisplayed allowing the user to edit additional member actions.

4.6 Editor

Limitations

The number of members and joints previously entered in Section 4.2.3.2 and the units selected for the input data can not be changed within this editor.

This editor is a subprogram of the preprocessor, which can be selected from the main menu in Section 4.2.1, that gives the user the capability of modifying the input data files that have been previously entered. After selecting the editor from this menu, a data entry screen (Section 4.2.3.1) is displayed allowing the user to enter the file name of the input data file to edit. If ESCAPE is pressed within this data entry screen, the main menu in Section 4.2.1 is redisplayed allowing further selection.

After a valid file name is entered, the editor will load the input data file after which a pop-up menu in Section 4.6.1 is displayed. From this pop-up menu, a user can select to modify the structural and element data (Section 4.6.1.1), the material and element properties (Section 4.6.1.2), and the loading (Section 4.6.1.3). The user can exit the editor by pressing ESCAPE which will save the input data file and redisplay the main menu in Section 4.2.1.

4.6.1 Main Menu

Editor
Structural and Element Data
Material and Element Properties
Loading

4.6.1.1 The Structural and Element Data

When this option is selected, a pop-up menu (Section 4.6.1.1.1) is displayed allowing the user to select the data routines to modify: the member incidences, the joint constraints, the internal hinges, the joint coordinates, and the direction cosines. For modifying these data routines with the editor, the user is to refer to Section 4.3.2. After completing the data entry, the pop-up menu in the previous section is redisplay allowing further selection.

4.6.1.1.1 Structural and Element Data Pop-up Menus

Direct Mesh Generation (plane frames):

Editor
Member Incidences
Joint Constraints
Internal Hinges
Joint Coordinates

Automatic Mesh Generation (plane frames)

Editor
Joint Constraints
Internal Hinges

Direct Mesh Generation (space frames)

Editor
Member Incidences
Joint Constraints
Joint Coordinates
Direction Cosines

Direct Mesh Generation (space trusses)

Editor
Member Incidences
Joint Constraints
Joint Coordinates

4.6.1.2 Material and Element Properties

When this option is selected, a pop-up menu will be displayed. Thus, the user will be allowed to modify the material (Section 4.6.1.2.1) and element (Section 4.6.1.2.2) properties on the input data file.

Editor
Material Properties
Element Properties

After completing the data entry, the main pop-up menu in Section 4.6.1 is redisplayed allowing further selection.

4.6.1.2.1 Material Properties

When this option is selected, the user will be allowed to modify the material properties that are presently on the input data file. For modifying the material properties, the user is to refer to Section 4.4.2.1.

4.6.1.2.2 Element Properties

The element properties presently stored in the input data file are listed on the display after which the editor organizes

them into element sets for the structure. After listing the element sets in a table, a pop-up menu will appear below it to determine if the user wants to generate new elements sets. An example display is shown:

Element Sets

Element set Number	Weight	Area	moment of inertia
1	300.0000	83.3000	203000.0000
2	132.0000	38.9000	196.0000

Generate new element sets ?
Yes
No

If the user decides to generate new element sets, the routine in Section 4.4.1.2 is called allowing additional element sets to be added to the end of the list of already developed ones. Thus, the user can reassign the new element sets to the members of the structure (Section 4.4.1.2.2).

By selecting not to generate new element sets, the pop-up menu in Section 4.4.1.2.2 is displayed allowing the user to assign the element properties by the element sets to the members.

4.6.1.3 Loading

When the loading option is selected from the pop-up menu in Section 4.6.1, a data entry screen is displayed allowing the user to enter the number of load conditions.

Load Conditions
Number of load conditions : 2
Enter the number of load conditions

When the data entry screen is displayed, the number of load conditions that was previously entered in Section 4.2.3.2 is shown. Thus, the user may choose not to modify this entry or reenter a new load condition within this data entry screen. If ESCAPE is pressed within this screen, the pop-up menu in Section 4.6 is displayed allowing further selection.

After the user enters the number of load conditions to edit, the editor will cycle from one to the number of load conditions to determine the loads (such as joint loads, member actions, and prescribed joint displacements) the user wants to

edit. If the editor can not find any loads for that load condition, the data routine in either Section 4.5.1.1, 4.5.1.2, or 4.5.1.3 is executed allowing the user to add new loads. This situation occurs when the user adds load conditions in addition to the previous one entered, or the user did not enter them when creating the input data file in Section 4.5.

However, if the editor determines there are loads for a given load condition, they will be displayed in a table (Section 4.5.1.1, 4.5.1.2, or 4.5.1.3) after which a pop-up menu will appear below this table determining if the user wants to add (Section 4.6.1.3.1), erase (Section 4.6.1.3.2), or edit (Section 4.6.1.3.3) any loads. An example of this pop-up menu for joint loads is shown:

Joint Loads
Add
Erase
Edit

After completing the data entry of the selected option, the loads are redisplayed in a table along with this pop-up menu. To terminate the pop-up menu used to manipulate the loads, the

user must press the ESCAPE after which the next loading is displayed in a table, or the pop-up menu in Section 4.6.1 is re-displayed completing the loading routine.

4.6.1.3.1 Add Loads

When this option is selected, a data entry screen in Sections 4.5.1.1, 4.5.1.2, or 4.5.1.3 is displayed allowing the user to enter additional loads. For adding these additional loads and terminating the data entry, the user is to refer to Section 4.5.

4.6.1.3.2 Erase Loads

When this option is selected, a data entry screen is displayed allowing the user to enter the range of load numbers to erase.

```
Erase Loads

  First load number   : [1          ]

  Last load number    :  3

  Load increment number :  1

Enter the first load number to erase
```

After completing the data entry, the user can press ESCAPE which will display the erased loads in a table along with a pop-up menu to determine if these are the loads the user wants to erase. If not the user can reselect them again using this data entry screen. An example of the table used to display the erased joint loads is shown:

Erase Joint Loads

Load condition number : 1

Load Number	Joint Number	Joint Direction	Joint Force or Moment
1	6	1	-1.50000
2	9	1	-1.50000

After selecting the loads to erase, a new table is displayed with the erased ones deleted.

Note: If the user erases all of the loads within a given load condition, the routines in Section 4.5.1.1, 4.5.1.2, or 4.5.1.3 are executed.

4.6.1.3.3 Edit Loads

When this option is selected, the user will be allowed to modify (edit) the loads. For modifying the loads, the user is

to refer to Sections 4.5.2.1 and 4.5.2.2.

4.7 Library Manager

Library Manager
Element properties
Material properties
Modify the element properties

The library manager is an interactive program used to manage the element and material libraries used by the preprocessor. This program is equipped with full error-checking capabilities to ensure that the preprocessor can read these libraries correctly. The user is recommended to use the library manager when modifying the libraries because the preprocessor does not error-check these libraries when reading them.

Note: To exit any of the menus for the library manager, element properties (Section 4.7.1) and material properties (Section 4.7.2), the user must press ESCAPE.

4.7.1 Element Properties

<u>Limitations</u>

Maximum allowable number of element properties that can be stored in the element library ≤ 250.

Element Properties
Add
Erase Print
Add element properties

4.7.1.1 Add Element Properties

After the user selects the add option from the slug menu for element properties (Section 4.7.1), a data entry screen is displayed.

```
Element Properties from the AISC manual

Shape            :  W-Shape
Name             :  36x300
Weight           :  300.0        lbs./ft.
Area             :  83.30        in.^2
Moment of inertia Ixx :  20300.0      in.^4
                  Iyy :  1300.0       in.^4
Torsional constant J  :  64.20        in.^4

Select the shape: Press the space bar to change.
```

After the data entry screen is activated, the user must determine the shape of the cross-section. The element library consists of six steel shapes. These shapes used for this library are W, M, MT, WT, C, and Other. Other is a shape that the user develops or a shape that is not listed above. The user can cycle through these shapes by pressing the space bar and enter a choice. The next entry must be in the form specified by AISC (AISC, 1985) for a name of a cross section that is 'depth x nominal weight per unit length'; for example, a 36 x 300 W section can be entered as '36x300' (refer to data entry screen). The next entries weight, cross sectional area, moment of inertia Ixx, and Iyy, and torsional constant J can be determined from the AISC steel construction manual (AISC, 1985).

After all element properties have been entered, the user can press ESCAPE to exit the data entry screen. By pressing ESCAPE, newly added element properties are displayed, and a new library is created.

4.7.1.2 Erase Element Properties

This option allows the user to erase element properties from the library. After the user selects this option, a pop-up menu (Section 4.4.1.2.1.2) for selecting the steel shapes is displayed. By selecting a steel shape, a data entry screen is displayed allowing the user to select one or more cross sections to erase. By pressing ESCAPE, which will redisplay the pop-up menu previously mentioned, the user can select additional cross sections to erase. However, if ESCAPE is pressed within this pop-up menu, the names of the erased element properties will be displayed after which a new library is created.

4.7.1.3 Print Element Properties

This option allows the user to print selected element properties on the line printer. The general procedure for selecting the cross sections to print is identical in Section 4.7.1.2. After pressing ESCAPE within the pop-up menu in Section 4.7.1.2, the element properties will be printed.

4.7.2 Material Properties

Limitations

Maximum allowable number of material properties that can stored in the material library ≤ 50.

Material Properties
Add
Erase Print
Add material properties

4.7.2.1 Add Material Properties

After selecting the add option from the slug menu for material properties (Section 4.7.2), a data entry screen is displayed allowing the user to add additional material properties to the library.

Material Properties Additions
Material type Material name : steel Material units : US Modulus of elasticity : 29000.0 Shear modulus : 11200.0 Temperature coefficient : 6.5E-06
Enter the name of the material type

The material name in material properties must be one word; that is, if two or more words are used, there must not be any spaces separating them, i.e., the '_' character can be substituted for a space. After entering the name of the material type, the user must select the units of input. There are two types of units this library accepts: US and SI. The US units are ksi and strain per degree Fahrenheit, while SI units are kPa and strain per degree Celsius. The user can choose the units of input by pressing the space bar to cycle through US and SI and pressing **ENTER** to select. The last three data items needed to define a material for the library are modulus of elasticity, shear modulus, and coefficient of thermal expansion. After all of the input has been entered, the user has a choice of entering additional material properties or pressing ESCAPE to exit the data entry screen. By pressing ESCAPE, the newly added material properties are displayed in a table after which a new material library is then created.

4.7.2.2 Erase Material Properties

After selecting the erase option from the **slug** menu for material properties (Section 4.7.2), a pop-up menu appears

determining which units of the library the user wants to erase.

Choose the units of material properties to erase
US
SI

When the units have been selected, a data entry screen (Section 4.4.1.1) is displayed showing the names of the material properties with the selected units. The user can select one or more material types to erase. To leave the data entry screen, press ESCAPE which will redisplay the pop-up menu for selecting the units. The user can erase additional material properties or press ESCAPE. If ESCAPE is pressed, the names of the erased material properties are displayed after which a new material library is created.

Note: The material library for each unit must consist of at least one material type. If not the preprocessor will terminate its execution when it encounters the material library.

4.7.2.3 Print Material Properties

When the print option is selected from the **slug** menu for material properties (Section 4.7.2), the entire material library is printed on the printer.

4.8 Graphics Presentation

Limitations

This program can only draw two dimensional structures (plane frames).

When this program is executed, a data entry screen (Section 4.2.3.1) is displayed allowing the user to enter the file name of a plane structure to display. By pressing ESCAPE within this data entry screen, the graphic presentation program is terminated. After a valid file name (Section 4.2.3.1) is entered, the program will read from the input data file the control variables, the member incidences, the joint constraints, and the joint coordinates. **Note:** this program will not display any loadings or internal hinges.

Before a structure is displayed, the program must determine how to display the joint constraints for this structure. This program only allows four types of joint constraints to be

displayed (Section 4.3.1.2.1.1). Thus, when a user or the program requests to draw them, the program must determine for each joint the following: the type of constraint, the direction to draw the constraint (such as horizontal, or vertical), and at which end of the member the joint is located (such as a end, or b end). However, there are situations where the program can not determine the type of constraint or its orientation. If a constraint is not listed in Section 4.3.1.2.1.1, the program will use a default constraint named 'Other'. This constraint is denoted by a rectangular box drawn around the joint. The program will also use the default directions if it can not determine the direction to draw the joint constraint (i.e. the orientation of it). For example, the default direction of a roller is in the global one direction (illustrated in Table 4.1).

This program can handle two types of plane structures generated by the preprocessor in Section 4.2.3.2 (Direct and Automatic) each with its own set of graphic commands to manipulate the structure on display (Sections 4.8.1, and 4.8.2).

Note: In order to get a print of a structure, the user must have entered from DOS 'Graphics.Com' after which by pressing

Print Screen, the structure will be inverted and printed.

4.8.1 Direct Graphic Commands

Direct method has eleven graphics commands giving the user the capability of manipulating the display of a structure on the screen. These graphic commands are entered on a command line. This command line, noted as 'Command', appears on top of the graphics display from which the user can enter one of eleven graphic commands. These graphic commands for the Direct Method are discussed below:

Display joint constraints (db) : display the joint constraints for a structure.

Display member numbers (dm) : display the member numbers for a structure.

Display joint numbers (dj) : display the joint numbers for a structure.

Display size (ds) : determines the character size for the joint and member numbers. The character sizes range from one (small) to four (large). The default display size is four.

Help : displays the graphic commands and their descriptions.

Help color (hc) : displays the color information (see set color).

Redraw all (all) : redraws the entire structure on display with its member and joint numbers, and joint constraints.

Redraw off (off) : redraws the entire structure on display without its members and joint numbers, and joint constraints.

Set color (sc) : the user must enter an integer ranging from (1) to (15) to designate the color to display. The graphics presentation program allows up to fifteen colors to display the structure or the background color in. These colors are (1) blue, (2) green, (3) cyan, (4) red, (5) magenta, (6) brown, (7) light gray, (8) dark gray, (9) light blue, (10) light green, (11) light cyan, (12) light red, (13) light magenta, (14) yellow, and (15) white. **Note:** If the user's computer has a color graphics adapter (CGA), the structure is displayed in the color selected; where as, with an enhance graphics adapter (EGA), the background color is displayed.

Quit : quit the graphic program.

Zoom : When zoom is selected, a crosshair will appear on the display. This crosshair can be manipulated about the display by using the arrow keys. It represents the location of the upper left corner of the zoom window and can be set

by pressing ENTER. The user can select the size of the zoom window by using the arrow keys and pressing ENTER to select.

The user can manipulate a 'zoomed' structure by using the graphic commands: **redraw off**, **display member and joint numbers**, and **joint constraints**. To magnify the structure to its original size, the user must enter the graphic command **redraw all** or (**all**).

4.8.2 Automatic Graphic Commands

The second set of graphic commands to manipulate the structure on the display are automatic. Automatic has twelve commands to manipulate the structure; however, eleven of these commands are identical to those for Direct (Section 4.8.1). These eleven commands are **display member and joint numbers**, **display joint constraints**, **display size**, **help color**, **help**, **set color**, **redraw all**, **redraw off**, and **zoom**. The other command is **display**.

Display : the user is instructed to enter the start, ending story, and bay numbers. Display is another form of **zoom** (Section 4.8.1) allowing the user to enter a specific region to view. After the structure is redisplayed, it can be

manipulated by using the graphic commands **redraw off**, **draw member and joint numbers**, and **display joint constraints**. To zoom the structure back to its original size, the graphic command **redraw all** or **(all)** must be entered.

Chapter 5

Extensions

This final chapter addresses some possible extensions for the Preprocessor. In Section 2.2, it was mentioned that a user interface software package, C-Scape 2.1, was used to develop the popup menus, popup and prompt windows, and data entry screens used for the Preprocessor. One possible extension is to update the user interface software package to C-Scape 3.0. The updated version allows the use of graphics within a data entry screen and the installation of a mouse for further input selection. C-Scape 2.1 permits only the display of text characters within data entry screens. However, C-Scape 3.0 allows full-bit map support giving the program the capability of displaying graphics within a data entry screen. This gives the programmer the ability to present the loading information (as shown in Sections 4.5.1.2.1 through 4.5.1.2.6) with more elegant and accurate drawings. By allowing the use of a mouse

in the Preprocessor, the user will have another method of selecting options from menus and data entry screens (refer to Section 2.1.2).

In Section 4.5.2, the possibility of refining the plane frame automatic mesh generation was suggested. At present, the user can only enter a mesh specifying the "number of stories" and the "number of bays". It would be desirable to give the user more flexibility to modify the existing mesh by allowing the addition of members and joints.

References

C-Scape 2.1 by the Oakland Group, Inc. (1987), Cambridge, Massachusetts.

Feibel, Werner (1988), Advance Quick C, Osborne McGraw-Hill, Inc., Berkeley, California.

Holzer, S. M. (1985), Computer Analysis of Structures, Elsevier, New York, New York.

Holzer, S. M. (1988), "Advance Topics in Matrix Structural Analysis", Blacksburg, Virginia.

IBM BASIC by Microsoft Corporation (1982), Redmond, Washington.

Jamsa, Kris and Nameroff, Steven (1988), Turbo Pascal: Programmer's Library, 2nd edition, Osborne McGraw-Hill, Berkeley, California.

Niezgoda, Steve (1985), "Space Truss Project in CE4980 at VPI & SU", Blacksburg, Virginia.

Oakland Group, Inc. (1987), The C-Scape Interface Management System Manual and Reference, Oakland Group, Inc., Cambridge, Massachusetts.

Olsen, Charles (1988), "QuickWindows," Computer Shopper, Coastal Associates Publishing, L.P., Titusville, Florida, pp. 319-320.

QuickBasic Compilier 4.0 by Microsoft Corporation (1988), Redmond, Washington.

QuickWindows by Software Interface, Inc (1987), Providence, Rhode, Island.

Schildt, Herbert (1988), Turbo C: The Complete Reference, Osborne McGraw-Hill, Inc., Berkeley, California.

Steel Construction Manual (1983), 8th edition, ASIC, Chicago, Illinois.

Appendix A

Examples of Input Data Files

In this Appendix, examples of input data files for plane frames (Section A.1) and space frames (Section A.2) and space trusses (Section A.3) are presented.

A.1 Plane Frames

A.1.1 Example 1

An example of an input data file for a plane frame (illustrated in Figure A.1) using the direct method mesh generation is listed below:

```
6/17/89
PF
US
2     3     1                 control variables
1     2                       member incidences
2     3
1     1                       joint constraints
1     2
```

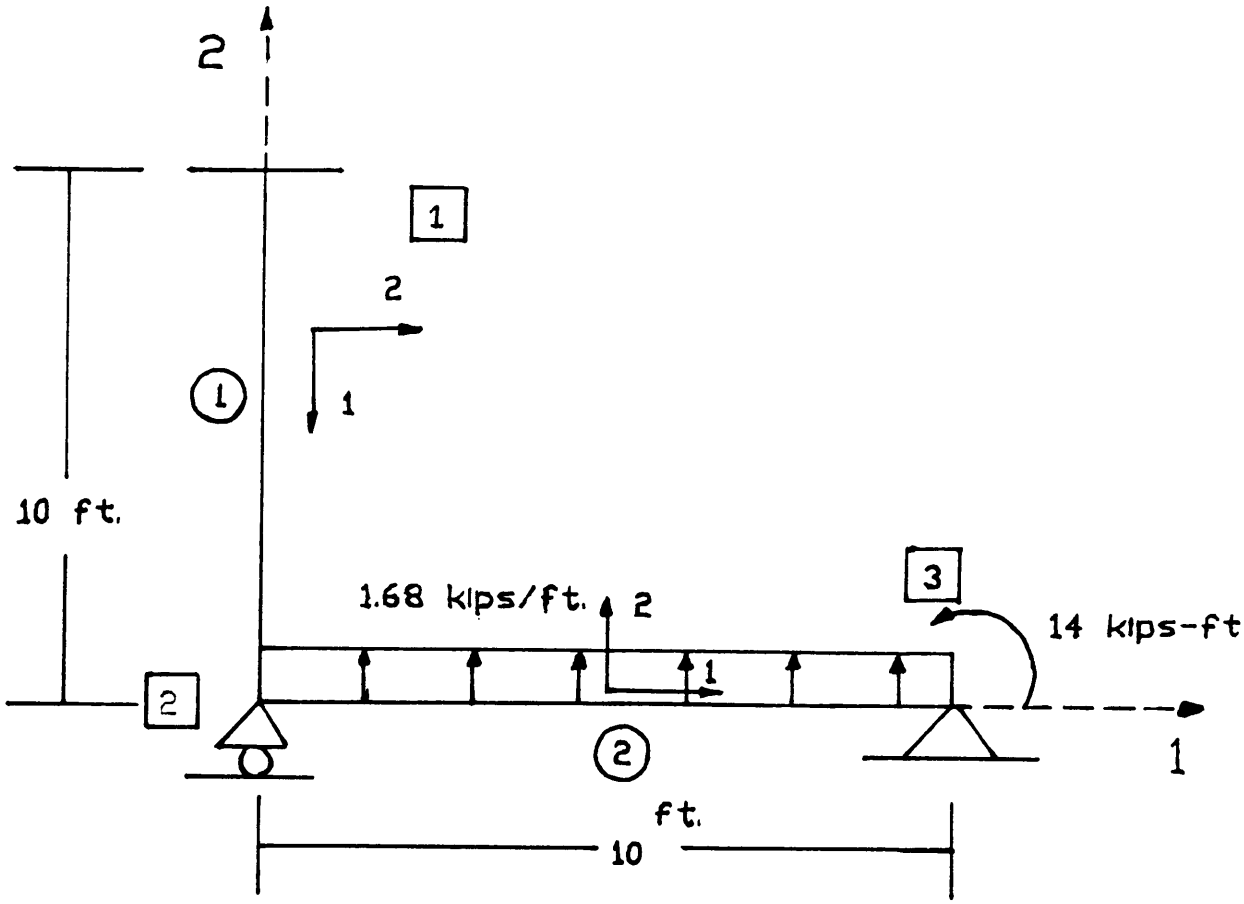


Figure A.1: An Example of a Plane Frame using Direct Method Mesh Generation (Holzer, 1985, pg. 139)

```
1      3
2      2
3      1
3      2
0      0
0      0                          internal hinges
0.0000      120.0000              joint coordinates
0.0000      0.0000
120.0000    0.0000
steel                             material properties
29000.0000      6.5E-006
1                                 element properties
1      2       1
4.430       48.0000
3      3     168.0000              joint loads
0      0     0.0000
2      3    -0.1400 0.0000 0.0000 0.0000  3   member actions
0      0     0.0000 0.0000 0.0000 0.0000  0
0      0     0.0000                prescribed joint displacements
```

A.1.2 Example 2

An example of an input data file for a plane frame (illustrated in Figure A.2) using automatic mesh generation is listed below:

```
6/17/89
MESH
US
6      6      1                   control variables
1      3                          member incidences
3      5
2      4
4      6
3      4
5      6
1      1                          joint constraints
1      2
2      1
2      2
0      0
```

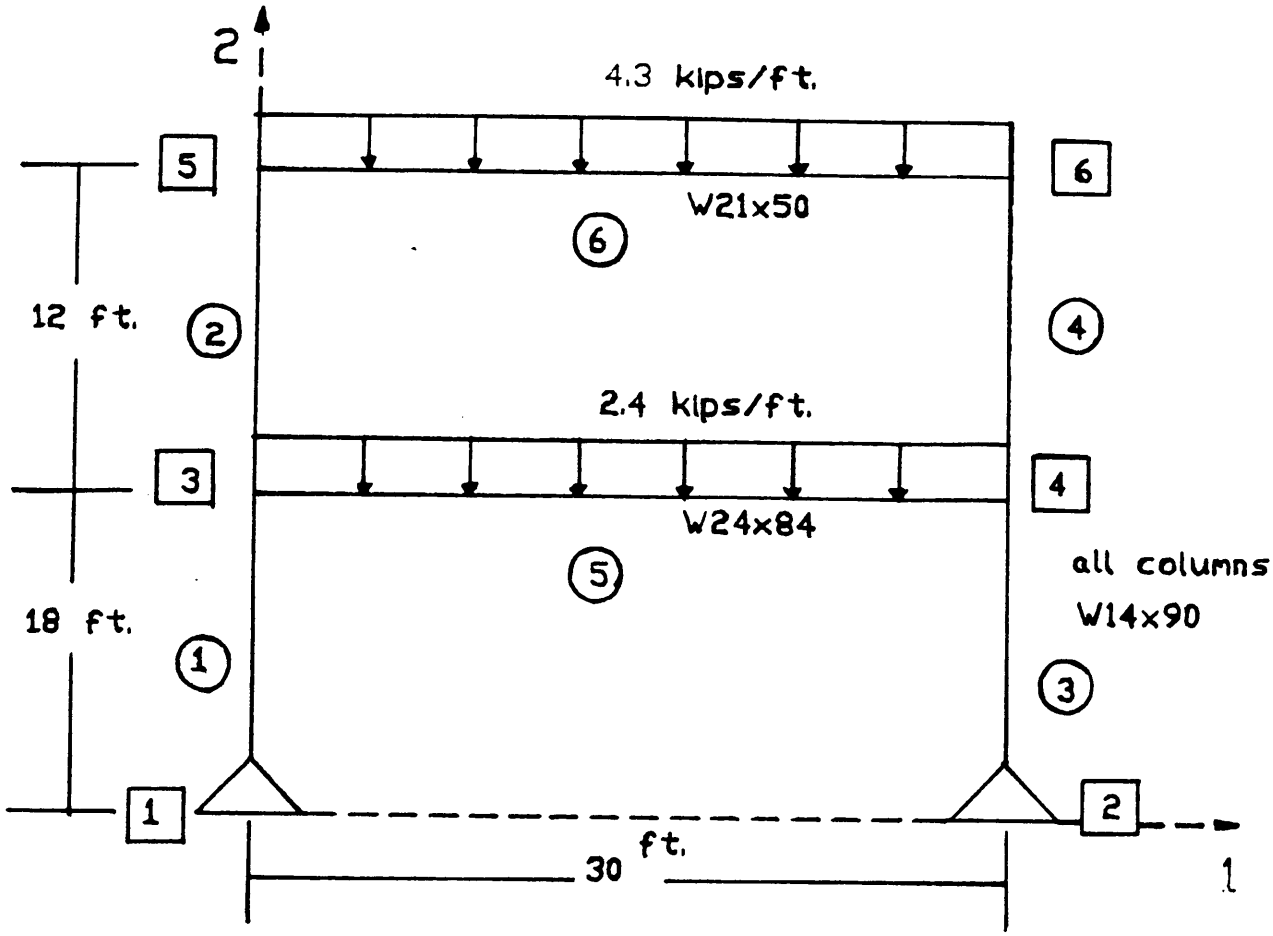


Figure A.2: An Example of a Plane Frame using Automatic Mesh Generation

```
0      0                          internal hinges
0.0000    0.0000                  joint coordinates
0.0000    216.0000
0.0000    360.0000
360.0000 0.0000
360.0000 216.0000
360.0000 360.0000
steel                             material properties
29000.0000      6.5E-006
3                                 element  properties
1      4     1
26.5000     362.0000
5      5     1
24.7000     2370.0000
6      6     1
14.7000     984.0000
0      0     0.0000                joint loads
5      3     -0.3583 0.0000 0.0000 0.00000 3  member actions
6      3     -0.1670 0.0000 0.0000 0.00000 3
0      0      0.0000 0.0000 0.0000 0.00000 0
0      0      0.0000               prescribed joint displacements
```

Note: The preprocessor uses the following scheme to number the joints and members in the automatic mesh generation. The joint numbers are numbered from the left to right, starting at bay one. The columns are numbered from bottom to top, starting at bay one after which the beams are numbered from left to right.

A.2 Space Frames

An example of an input data file for a space frame is presented in this Section. As mentioned in Section 3.1.1.2, the preprocessor requires the direction cosines of the unit

vector in the local 2 direction (the Y-axis according to AISC) to be defined for each member.

In the following example of a space frame (illustrated in Figure A.3), the direction cosines of each member for the unit vector in the local 2 direction are determined:

For member 1 and 2, the local and global axis coincide; therefore, the unit vector in the local 2 direction for each member is I_2, which is the unit vector in the global 2 direction. Thus,

$$i^1_2 = i^2_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

For member 3, the unit vector in the local 2 direction can be found by the following procedure (Holzer, 1985): First, the unit vector i_1 of the member is computed:

$$i^3_1 = \frac{\Delta X}{|\Delta X|} \qquad \text{(C.1)}$$

where

$$\Delta X = X_4 - X_2 = \begin{bmatrix} -3 & 12 & -4 \end{bmatrix}$$

$$|\Delta X| = 13$$

Thus,

Examples of Input Data Files

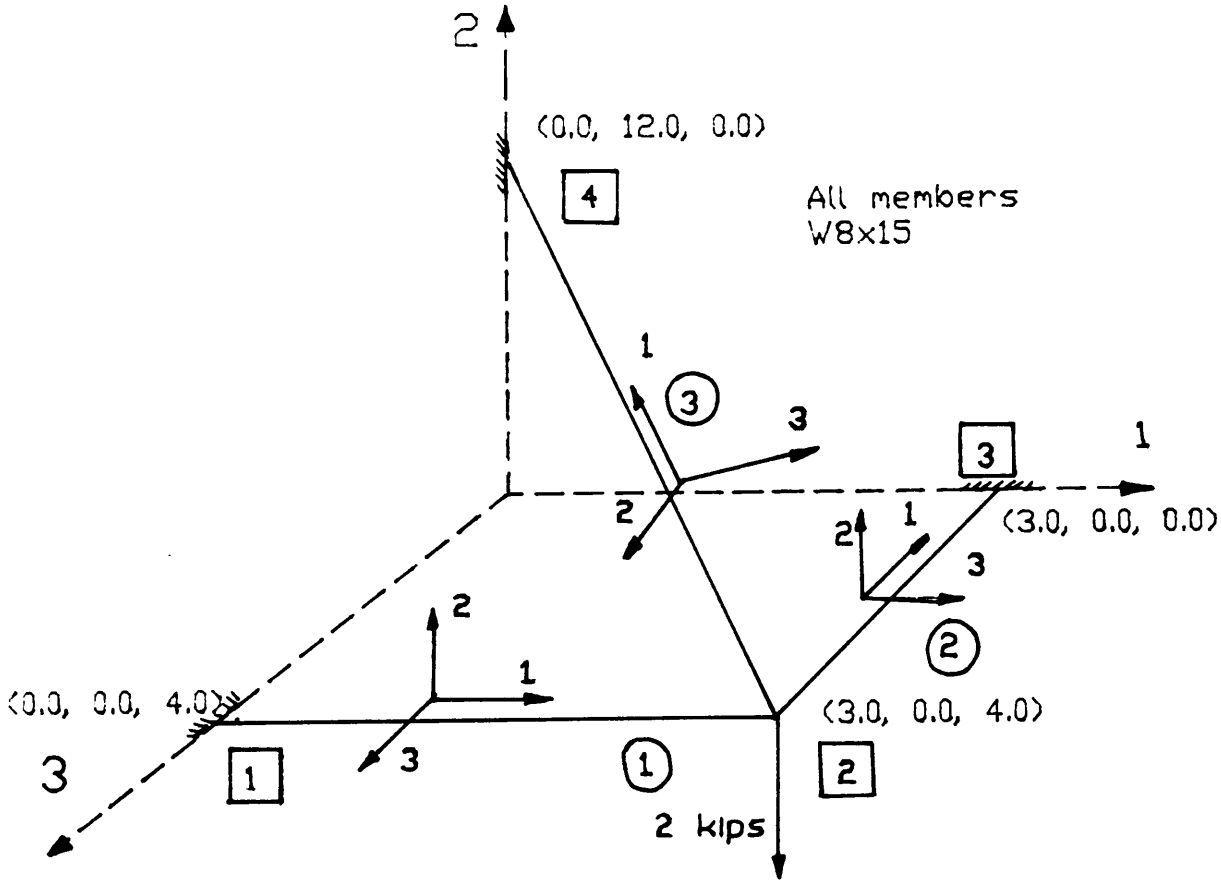


Figure A.3: An Example of a Space Frame

$$i^3_1 = \frac{1}{13}\begin{bmatrix} -3 & 12 & -4 \end{bmatrix} \qquad (C.2)$$

In order to determine the unit vector in the local 2 and 3 directions, additional information must be given about the orientation of the cross section. For this problem, the web is assumed to be vertical. Therefore, the principle plane is vertical, and the local 3 axis is perpendicular to the vertical plane that contains the vectors i_1 and I_r Thus, the vector i^3_3 can be obtained as follows:

$$\bar{v} = i^3_1 \times I_1 \qquad (C.3)$$

where

$$\bar{v} = \begin{vmatrix} I_1 & I_2 & I_3 \\ \frac{-3}{13} & \frac{12}{13} & \frac{-4}{13} \\ 0 & 1 & 0 \end{vmatrix} \qquad (C.4)$$

$$\bar{v} = \frac{1}{13}\begin{bmatrix} 4 & 0 & -3 \end{bmatrix}$$

and

$$i^3_3 = \frac{\bar{v}}{|\bar{v}|} \qquad (C.5)$$

Examples of Input Data Files

$$i^3_3 = \frac{1}{5}\begin{bmatrix} 4 & 0 & -3 \end{bmatrix} \qquad (C.6)$$

The unit vector in the local 2 direction can be determined by the cross product of i_3 and i_1:

$$i^3_2 = i^3_3 \times i^3_1 \qquad (C.7)$$

where

$$i^3_2 = \begin{vmatrix} I_1 & I_2 & I_3 \\ \frac{4}{5} & 0 & \frac{-3}{5} \\ \frac{-3}{13} & \frac{12}{13} & \frac{-4}{13} \end{vmatrix} \qquad (C.8)$$

which yields

$$i^3_2 = \frac{1}{65}\begin{bmatrix} 36 & 25 & 48 \end{bmatrix} \qquad (C.9)$$

After determining the direction cosines for the unit vectors in the local 2 direction for each member, the user can create an input data file for the space frame in Figure A.3. The input data file for this structure is listed below:

Examples of Input Data Files

```
6/17/89
SF
US
3      4      1                 control variables
1      2                        member incidences
2      3
2      4
1      1                        joint constraints
1      2
1      3
1      4
1      5
1      6
2      1
2      2
2      3
2      4
2      5
2      6
3      1
3      2
3      3
3      4
3      5
3      6
0      0
0      0                        internal hinges
0.0000    0.0000    48.0000     joint coordinates
36.0000   0.0000    48.0000
36.0000   0.0000     0.0000
0.00000   144.0000   0.0000
0.0000    1.0000     0.0000     direction cosines
0.0000    1.0000     0.0000
0.5539    0.3846     0.7385
steel                            material properties
29000.0000  11200.0000 6.5E-006
1                                element properties
1   3    1
4.43  48.0000   3.41   0.1400
2   2   -2.0000                  joint loads
0   0    0.0000
0   0    0.0000  0.0000  0.0000 0.0000 0  member actions
0   0    0.0000
```

A.3 Space Trusses

An example of an input data file for a space truss (illustrated in Figure A.4) is listed below:

```
6/17/89
ST
US
3     4     1                 control variables
1     4                       member incidences
2     4
3     4
1     1                       joint constraints
1     2
1     3
2     1
2     2
2     3
3     1
3     2
3     3
0     0
0     0                       internal hinges
0.0000 0.0000  36.0000
0.0000 36.0000  0.0000
0.0000 0.0000  -36.0000
48.0000 0.0000 0.0000
steel                         material properties
29000.0000     6.5E-006
1                             element properties
1   4    1
4.43    48.00000
0   0    -2.0000              joint loads
0   0    0.00000
0   0    0.00000 0.00000 0.0000 0.0000  member actions
0   0    0.0000               prescribed joint displacements
```

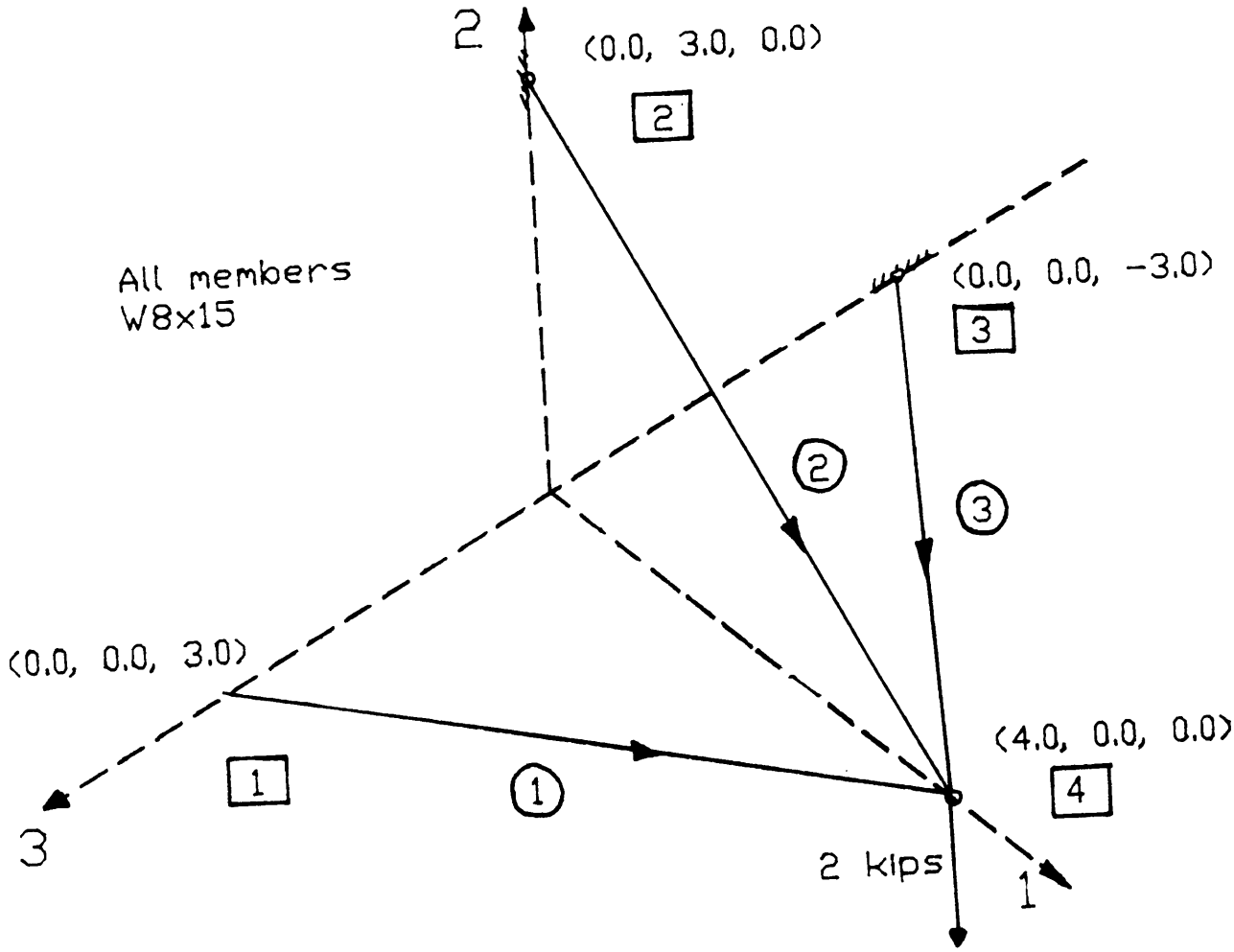


Figure A.4: An Example of a Space Truss (Holzer, 1985, pg. 276)

The vita has been removed from the scanned document