

A STUDY OF SOME FEATURES
OF THE FULLY STRESSED DESIGN ALGORITHM

by

Kevin Haskell Traynor

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:

A. E. Somers, Jr., Chairman

S. M. Holzer

D. A. Garst

May, 1982

Blacksburg, Virginia

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. A. E. Somers for his help and guidance during this project and to Willie Hylton for her excellent job in typing this paper. I would also like to express my appreciation to my parents, David and Lorraine Traynor, to whom this thesis is dedicated, for their love and support through the years.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	vi
1. INTRODUCTION	1
2. LITERATURE REVIEW	2
2.1 DESIGN	2
2.2 FULLY STRESSED DESIGN	5
2.3 PLANE FRAME OPTIMIZATION	7
2.4 MEMBER PROPERTY RELATIONSHIPS	9
3. SOLUTION PROCESS.	11
3.1 GENERAL PROCEDURE.	11
3.2 BEAM-COLUMN ELEMENT MODEL.	13
3.2.1 REFERENCE FRAMES	15
3.2.2 GLOBAL ELEMENT MODEL	17
3.3 AISC CODE REQUIREMENTS	19
3.4 EFFECTIVE LENGTH AND STABILITY ANALYSIS	22
3.5 FULLY STRESSED DESIGN	25
3.6 CROSS SECTION PROPERTY RELATIONSHIPS	26
4. DESIGN STUDIES.	23
4.1 MODIFIED K-FACTORS FOR INELASTIC BEAM-COLUMNS.	28
4.2 CROSS-SECTION PROPERTY EQUATIONS	29
4.3 CURVE-FITTING ALGORITHM	31
4.4 COLUMN EQUATIONS	31

TABLE OF CONTENTS (continued)

	<u>Page</u>
4.5 BEAM EQUATIONS	34
4.6 COMPARISON OF BEAM EQUATIONS	34
5. DESIGN EQUATIONS	39
5.1 DESIGN EXAMPLE 1	39
5.2 DESIGN EXAMPLE 2	49
5.3 DESIGN EXAMPLE 3	56
5.4 CONCLUSIONS	60
6. EXTENSIONS	61
REFERENCES	67
APPENDIX A. COMPUTER PROGRAM LISTING	72
VITA	100
ABSTRACT	

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
3.1	STRUCTURAL DESIGN PROCEDURE	12
3.2	LOCAL BEAM-COLUMN ELEMENT MODEL	14
3.3	ARBITRARY ORIENTATION OF COORDINATE SYSTEMS IN SPACE	16
3.4	FLOWCHART FOR COMPUTATION OF K-FACTORS AND INTERACTION EQUATIONS	20
4.1	SECTION MODULUS VS. BENDING MOMENT OF INERTIA FOR COLUMNS	32
4.2	SECTION MODULUS VS. AREA FOR COLUMNS	33
4.3	SECTION MODULUS VS. MOMENT OF INERTIA FOR BEAMS	35
4.4	SECTION MODULUS VS. AREA FOR BEAMS	36
5.1	DESIGN EXAMPLE 1 - MEMBER AND JOINT NUMBERS . . .	40
5.2	DESIGN EXAMPLE 1 - LOAD CONDITION 1	41
5.3	DESIGN EXAMPLE 1 - LOAD CONDITION 2	42
5.4	DESIGN EXAMPLE 1 - LOAD CONDITION 3	43
5.5	DESIGN EXAMPLE 1 - LOAD CONDITION 4	44
5.6	DESIGN EXAMPLE 2 - MEMBER AND JOINT NUMBERS . . .	50
5.7	DESIGN EXAMPLE 2 - LOAD CONDITION 1	51
5.8	DESIGN EXAMPLE 2 - LOAD CONDITION 2	52
5.9	DESIGN EXAMPLE 2 - LOAD CONDITION 3	53
5.10	NEW SECTION MODULUS VS. AREA EQUATION FOR BEAMS	57
5.11	NEW SECTION MODULUS VS. MOMENT OF INERTIA FOR BEAMS	58
6.1	MEMBER SECTION ROUTINE FLOWCHART	63

LIST OF TABLES

<u>TABLE</u>		<u>Page</u>
5.1	DESIGN RESULTS OF EXAMPLE 1 BY ARORA, HAUG AND RIM	45
5.2	DESIGN RESULTS OF EXAMPLE 1 VIA COMPUTER CODE . .	47
5.3	DESIGN RESULTS OF EXAMPLE 2 VIA BROWN/ANG EQUATIONS	54
5.4	DESIGN RESULTS OF EXAMPLE 2 VIA MEDIAN EQUATIONS	55
5.5	DESIGN RESULTS OF EXAMPLE 3 VIA NEW EQUATIONS	59
6.1	SELECTED SECTIONS BASED ON COMPUTER RESULTS . . .	65
6.2	SELECTED SECTIONS BASED ON COMPUTER RESULTS AND TABLE 6.1	66

CHAPTER 1

INTRODUCTION

This thesis presents the results of a study of the fully stressed design (FSD) of rigidly connected plane steel frames. The fully stressed design condition is defined as the state for which each structural member is at the prescribed allowable (limiting) stress for at least one loading condition. The objective is (1) to formulate the FSD algorithm consistent with the most recent AISC Manual of Steel Construction [23], including new dependent section property functions and the complete set of allowable stresses for beams and beam-columns; (2) to study the effect of some of the features of FSD on the final design; and (3) to compare final designs with previously published results. The iterative design procedure uses the displacement method of analysis and the stress-ratio method of redesign.

A literature survey is presented first. Then, a detailed description of the design and analysis procedure is developed. Next, a series of studies include the selection of an independent design variable, the modification of the elastic beam-column analysis procedure for inelastic behavior, and the selection of a set of dependent section property functions for both the beam and column sections tabulated in the AISC Manual. Several examples are provided as the basis for demonstrations and comparisons. A systematic procedure to select discrete member sections of the AISC Manual, based on the final design of the FSD procedure, is discussed along with other possible extensions to this study.

CHAPTER 2

LITERATURE REVIEW

2.1 Structural Design

The general objectives of optimum structural design are (1) to develop a structure that safely performs its intended function and (2) to minimize the cost required to develop the structure. The cost need not be thought of only in terms of money, but it can also be described by physical parameters, such as the volume of material used. This allows the optimization process to be formulated in terms of design variables of a mathematical model.

It is generally accepted that Galileo [41] was the first to apply the optimum design objectives in the area of structural design. In his studies on beams designed for uniform strength, Galileo used a design method first utilized years before by Leonardo Di Vinci. Commonly referred to as the model method, this is probably the oldest and, perhaps, the best method of optimum design. In this method, full size or scaled down models of the structure are tested for their adequacy in resisting the prescribed loads. This procedure is still used today to test the response characteristics of complicated structural systems such as shells, irregular plates or gridworks [41]. Soon after Galileo's work was published in 1638, other men, including Bernoulli, Lagrange, and Newton began to look into the structural design problem [41].

In general, most of the advances in optimum structural design technology can be categorized into four distinct phases. These are the

(1) theory of layout, (2) simultaneous modes of failure, (3) optimality-criteria-based, and (4) mathematical programming formulations [10].

The theory of layout approach is one in which the geometric orientations of the members of a structure are optimized. This approach mainly deals with uniaxial or truss members. An optimum design using this theory would be an arrangement of members in a structure that would produce the maximum resistance to specified loads with a fixed volume of material. Maxwell developed the basic theorems for this approach in 1854 and Michell applied them to some practical problems in 1904 [10, 26, 41]. However, since these theorems are applied without meaningful constraints, many solutions yielded by this method are impractical [10]. A more detailed review in this area has been presented by Prager and Sheu [31].

The simultaneous mode of failure approach assumes that every component in an optimum structure will reach its limiting strength under a specified loading condition as failure of the entire structure impends. Wasuitynski and Brandt [41] break this category down further into three hypotheses. One hypothesis considers the rupture or plastic strength of the material as the limiting strength, while another considers the acceptable strength safety factor as the limiting criteria. A third hypothesis relates the form of the structure to the internal forces.

The earliest works using a form of this approach, called the uniform strength approach, were by Galileo in his study of beams which was noted earlier. More recent contributions to the approach were made

by Gerard, Cox and Shanley during the 1940's and 1950's [10]. Unfortunately, these studies were conducted prior to the age of the electronic digital computer using classical minimization functions. Also, since all of these works dealt with single loading conditions only, the approach in its present form has very limited practical application to realistic problems. Consequently, little research has been done utilizing the approach since those works were published. If, however, the approach is expanded to include multiple loading conditions, it can be applied to a wide range of practical problems. The fully stressed design method is one such adaptation of this modified approach. A more detailed study of this method will be presented later.

The optimality criteria approach establishes a set of optimality conditions which must be satisfied in order to obtain an optimum design. Stresses and displacements are often used in the formulation of these conditions which are derived from structural mechanics and variational principles. This approach has mostly been limited to simple structural forms and loading conditions in past studies. Major contributors to the development of the approach have been derived by Venkayya, Gellatly and Berke [10].

The most recent approach in optimum structural design uses mathematical programming techniques to define an optimum structure. These procedures locate the extreme of a function consisting of many variables that are subject to constraints. These constraints are represented by a set of equalities and inequalities. A unique feature of this method

is that the inequality constraint enables the designer to specify which constraints do not have to be met by all of the members in a specific situation. This approach fits well in design problems which are represented by finite element models [10]. Some of the more common approaches in structural design are the linear, nonlinear, and dynamic programming approaches. The earliest utilization of mathematical programming to structural design was by Livesly and Pearson in the late 1950's [10]. Foulkes recognized that structural design in the elastic range was a nonlinear programming problem while plastic structural design could be a linear programming problem [31]. Extensive surveys of mathematical programming procedures in optimum structural design have been presented by Schmit [35, 36, 37].

2.2 Fully Stressed Design

In the previous section it was noted that the fully stressed design method of structural design grew out of the simultaneous mode of failure approach. It is an analysis-based design method in which the objective or merit function to be extremized is replaced by an optimum stress constraint for each member. Also, the geometric configuration of the structure is specified and does not vary. The method is an iterative analysis-design process. First, initial section properties are assigned to each member in the structure. Then, the structure is analyzed and the maximum stress in each member is computed. This stress is then compared to an allowable stress specified for that member. If the member is overstressed or understressed, the design variable, usually the area or section modulus, is increased or

decreased, respectively, according to some function of the stress differential. The other section properties of the member are also modified by using empirical dependent section property relationships. This is done for all of the members in the structure. The process is repeated until the stress differential in every member is so small that further design is impractical. The final design is one in which every member of the structure is fully stressed under at least one of its loading conditions.

Gallagher [11] believes that, although it is a classical concept that has been around for a long time, the fully stressed design method is still highly regarded and widely used because it has several favorable characteristics.

First of all, the fully stressed design procedure is usually less expensive, computationally, than the more complex mathematical programming procedures. This becomes significant when large structures with many members are considered. The most dramatic example of this was presented by Venkayya [40] who claimed that a series of trusses with up to two hundred elements converged to a fully stressed design in less than twenty iterations. Johnson [15], Melosh and Luik [25], and Young and Christiansen [42] also came up with similar results for large structures.

Secondly, even though a fully stressed design is not, in general, an optimum design, it can be used as a starting point for more rigorous design procedures based on nonlinear programming concepts. This is especially significant for indeterminate structures in which the

critical member stress is a function of the applied loading and a highly nonlinear function of the section properties of all of the members in the structure [32].

A third characteristic exhibited by the method is that a fully stressed design is a minimum weight design under certain conditions. Patnaik and Dayaratnam [5] found this to be true for a statically determinate truss subjected to a single load condition. Razani [32] extended this to include all statically determinate structures. Although Kicher [20] concluded that an indeterminate structure subjected to multiple load conditions would not be, in general, both a fully stressed and a minimum weight design, the weight difference has been found to be small in some situations [32].

Since it has been shown [20] that a fully stressed design is not always a minimum weight design, Kuhn and Tucker [22] developed a test in which the optimality of any design problem can be determined. A similar test for elastic trusses was provided by Razani [32].

Although the fully stressed design method does not always yield an optimum design, its simplicity, computational economy, and high adaptability to a wide range of large and small problems certainly justifies it as a useful tool to the designer.

2.3 Plane Frame Optimization

In the early stages of investigation into optimum design of plane frames, most works were devoted to either one of two optimality conditions. One was minimum weight and the other was uniform strength. Initial interest in plane frame design seemed to originate in Michell's

paper [26] in 1904. In it, he suggested the possibility that framed structures could be designed for minimum weight. However, more rigorous investigations in this area did not appear until 1953 when Foulkes published two works on minimum weight design of plane frames [41]. Uniform strength design of frames was studied chiefly by two Russian scientists, Rabinovitch and Vinogradov, and their colleagues in the 1930's [41].

In the early 1960's, more sophisticated design techniques were developed when the use of the high speed digital computer became practical for solving large problems. The most significant contribution in this area was the development and application of optimization techniques based on mathematical programming formulations. Moses [27], Romstad and Wang [34], Cornell, Reinschmidt, and Brothie [4], and others used a popular linear programming technique called the Simplex method. This method approximates the nonlinear programming problem by a series of linear programming problems. Gellatly and Gallagher [12] and Brown and Ang [2] used feasible direction methods of nonlinear programming on frame structures. Another popular method that has been used in optimum frame design is the penalty function method. Schmidt and Fox [38], Kavlie, Kowalik and Moe [23], and Marcal and Gellatly [24] have obtained good results with this technique. Palmer [30] and Twisdale and Khachaturian [39] have also applied dynamic programming techniques with favorable results. Detailed descriptions of these various techniques can be found in books on structural optimization.

2.4 Member Property Relationships

Most of the optimization routines used in structural design attempt to minimize the overall volume or weight of the structure since minimization of weight can be directly related to the minimization of the cost of the structure. In order to do this, it is necessary to choose a set of design variables that adequately describe the structure. These design variables are modified in the design procedure. Most of the work on structural frames use the cross-sectional properties of the members as design variables.

One approach to this problem was used by Kavlir and Moe [19] for the design of grillage structures and plate girders. It calls for the optimization of the cross-section dimensions. This could be done by breaking down the total area into a web area and two flange areas. In the case of the web, the height could be specified and the thickness designated as the design variable. The same could also be done with the flanges, where the width is specified and the thickness is varied. The flanges are optimized for moments while the web is optimized for shear. The remaining cross-sectional properties such as moment of inertia and section modulus could be computed directly from the area.

A more popular approach deals with fixed section properties of standard rolled sections. In this approach a section property such as moment of inertia or section modulus is chosen as the independent design variable and the other properties are calculated from empirically derived equations. Clarkson [3] was one of the first to do this for British Standard sections. In his study, the area was the design

variable and equations for moment of inertia and section modulus were developed. These equations were later used by others [13,18,19, 21,28,33]. Nakamura [29], Brown and Ang [2] and Arora, Haug and Rim [1] also developed empirical equations for AISC standard rolled sections [23].

CHAPTER 3

SOLUTION PROCESS

3.1 General Procedure

A computer program has been developed for this study that produces plane frame structures that meet the objectives of the fully stressed design approach; that is, every member is subjected to a limited stress under one of the prescribed loading conditions. It is based on an iterative design procedure. Figure 3.1 is a flowchart representation of the procedure.

In steps 1 through 6 in Figure 3.1, the displacement method of structural analysis is used to determine the response behavior of a structure. The object of this analysis is to solve a set of simultaneous equilibrium equations which represent the structural system under a specified load condition. The solution is a set of generalized displacements of joints in the structure. The mathematical model of the structure can be represented in matrix notation by

$$\begin{matrix} K & \cdot & q & = & Q \\ (n,n) & (n,1) & (n,1) & & \end{matrix} \quad (3.1)$$

where K is the system stiffness matrix, q is the joint displacement matrix, Q is the joint load matrix, and n is the number of degrees of freedom for the structure. From here, the member end forces are computed from the generalized displacements. In step 7, member stresses are found and compared to allowable stresses set by the American Institute of Steel Construction (AISC). If the computed stresses are

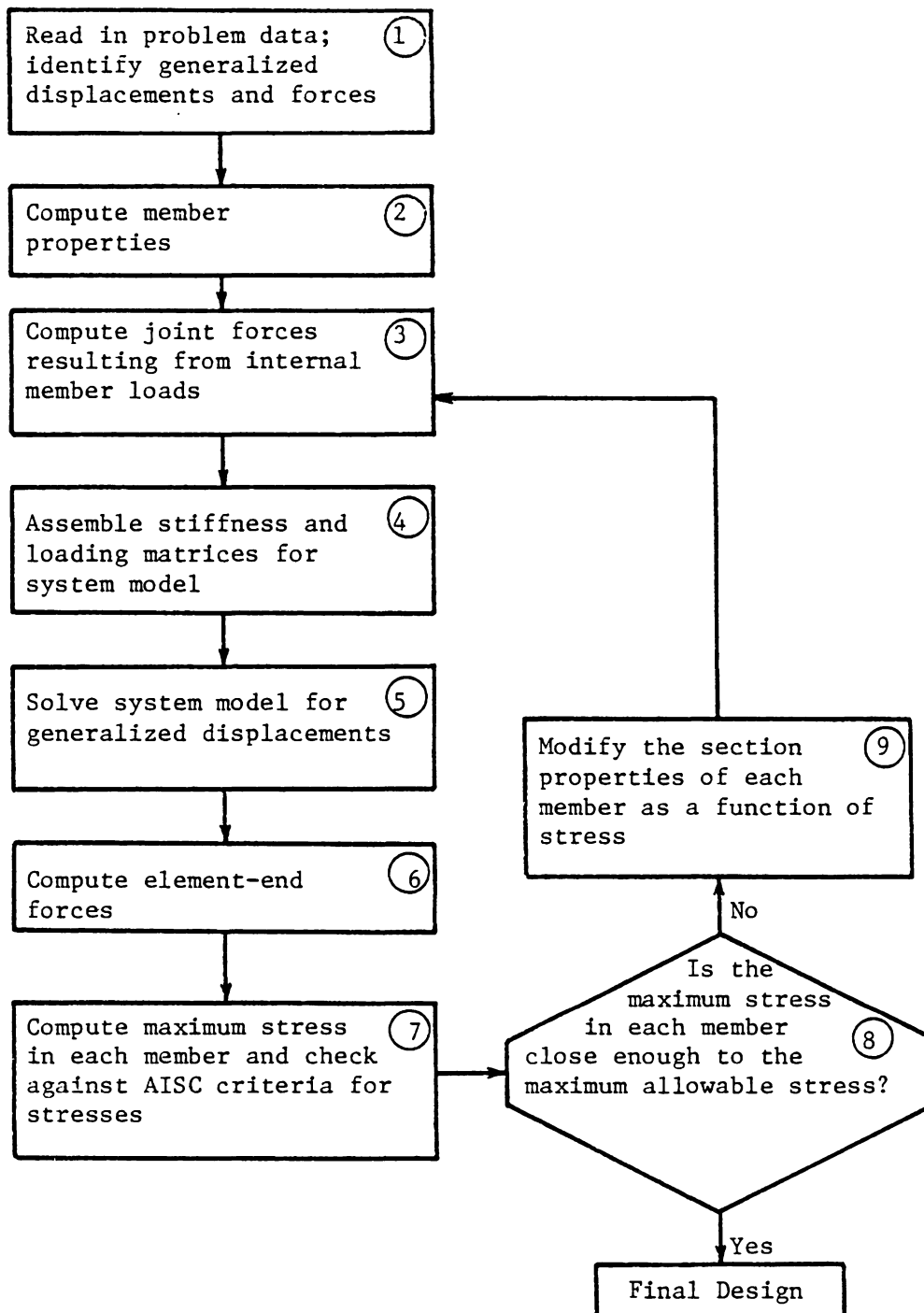


Figure 3.1 Structural Design Procedure

within an acceptable tolerance of the allowable stresses for every member in the structure, the design is considered to be fully stressed. If not, each member is reportioned according to some function of the stress. This is performed in step 9 of the flowchart. The revised structure is then reanalyzed. This is repeated until a fully stressed design is achieved.

Plane frame structures are studied in the investigation. These structures are composed of beam-columns with in-plane loads. A mathematical model of a beam-column element is presented in the next section. The formulation of this model is based on derivations given in class notes from CE 4001 [44].

3.2. Beam-Column Element Model

Figure 3.2 illustrates a linear beam-column element with six generalized element-end displacements and their corresponding forces. This gives the element three degrees of freedom at each end: an axial displacement, a shear displacement, and a rotation. The corresponding forces are an axial force, a shear force, and a moment. The relationship between the forces and displacements can be found by subjecting the governing differential equations of the continuum model to the displacement boundary conditions of the element. Since the derivation of this model can be done using different approaches and many texts on the subject present these derivations in detail, it is not presented here. Instead, the final result of these derivations are shown below in matrix form:

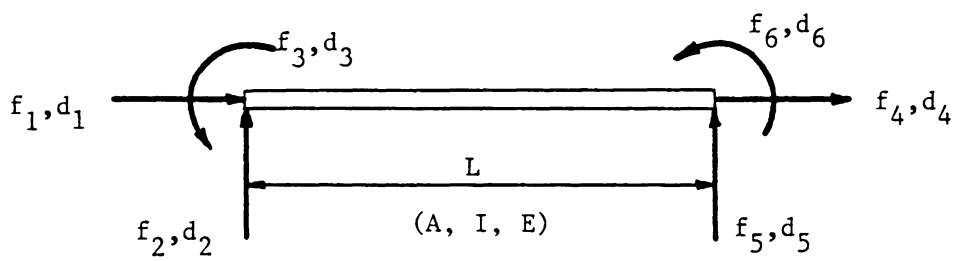


Figure 3.2 Local Beam-Column Element Model

$$\begin{array}{ccc}
 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} & = \alpha & \begin{bmatrix} \beta & 0 & 0 & -\beta & 0 & 0 \\ & 12 & 6L & 0 & -12 & 6L \\ & & 4L^2 & 0 & -6L & 2L^2 \\ & & & \beta & 0 & 0 \\ \text{symmetric} & & & & 12 & -6L \\ & & & & & 4L^2 \end{bmatrix} & \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \\
 f & & k & & d \\
 (6,1) & & (6,6) & & (6,1)
 \end{array} \tag{3.2}$$

$$\alpha = \frac{EI}{L^3} \quad \beta = \frac{AL^2}{I} \tag{3.2a}$$

In this case, k is the stiffness matrix of the local beam-column element.

3.2.1 Reference Frames

In order to assemble the system model represented by equation 3.1 from the individual element models, it is necessary to relate the reference frame of each element to the reference frame of the entire structure. For this discussion, refer to Figure 3.3. The local or x -coordinate system describes the geometry of the element while the global or X -coordinate system describes the geometry of the structure as a whole. A relationship between the two reference frames is found by establishing two intermediate coordinate systems, a and b , and relating all of the coordinate systems with a set of angles. A matrix representation of these angular relationships is found by

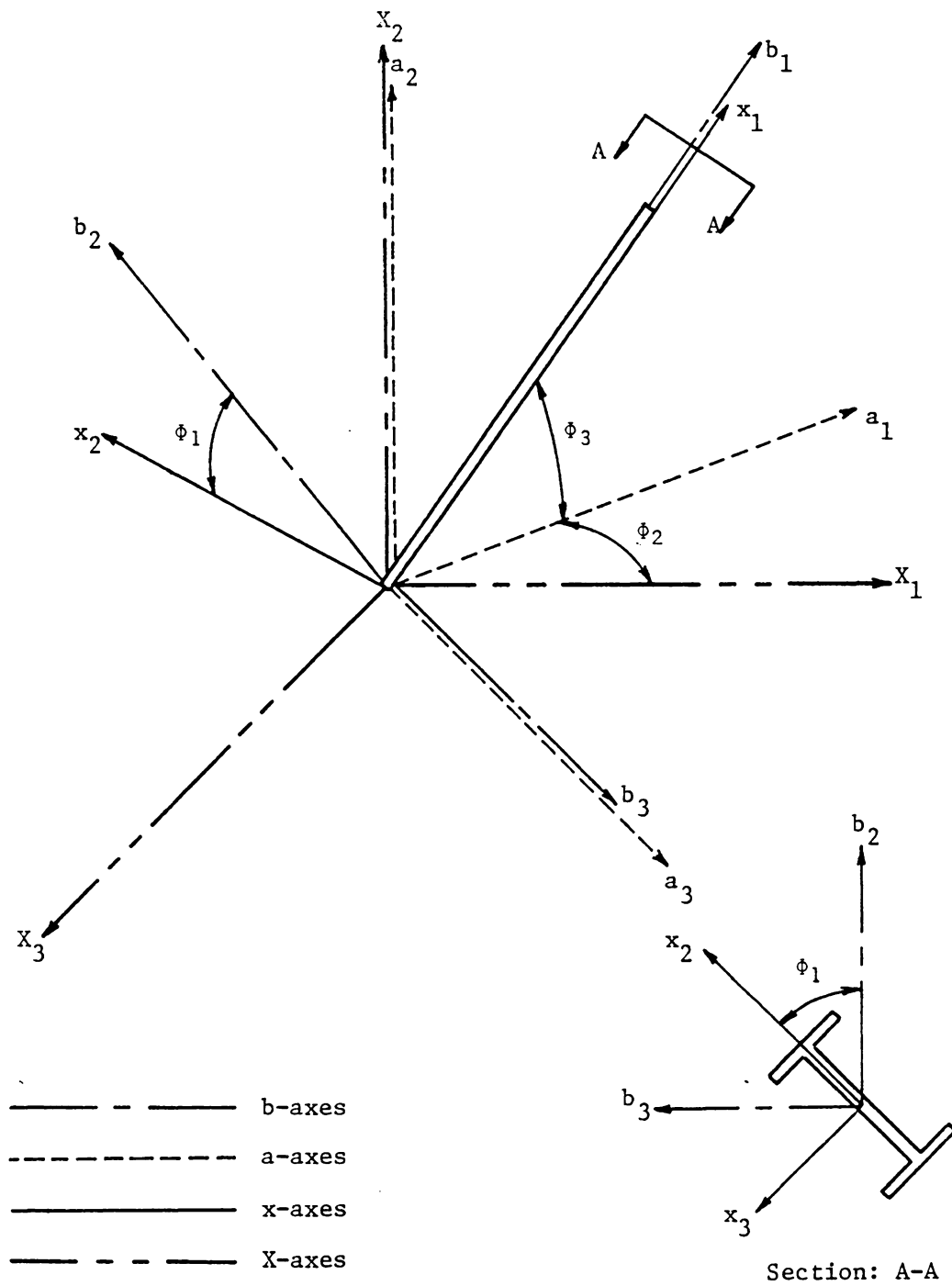


Figure 3.3 Arbitrary Orientation of Coordinate Systems in Space

establishing two intermediate coordinate systems, a and b, and relating all of the coordinate systems with a set of angles. A matrix representation of these angular relationships is found by projecting a unit vector of the coordinate axes of one reference frame onto the corresponding coordinate axes of another reference frame. In Figure 3.3, ϕ_1 relates the local x-axes to the b-axes while ϕ_3 relates the b-axes to the a-axes and ϕ_2 relates the a-axes to the global X-axes. This procedure ultimately yields

$$X = \lambda_2 a \quad (3.3)$$

$$a = \lambda_3 b \quad (3.4)$$

$$b = \lambda_1 x \quad (3.5)$$

where λ_i is the transformation submatrix and i is the angle number. Relating the X-axes to the x-axes directly is accomplished by multiplying equations 3.3, 3.4 and 3.5 together which leads to:

$$\begin{matrix} X = [\lambda_2 \lambda_3 \lambda_1] x = \Lambda x & (3.6) \\ (3,1) & (3,3)(3,1) \end{matrix}$$

where Λ is the global transformation submatrix. A detailed derivation of equation 3.6 can be found in class notes from CE 4002 [45].

3.2.2 Global Element Model

The derivation of the transformation submatrix, Λ , for the beam-column element model is provided in the CE 4001 notes [44] so it will not be presented here. However, it can be shown that the Λ submatrix for a beam-column element is

$$\Lambda_{(3,3)} = \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

where C_3 and S_3 are the cosine and sine of the unit vector of the local x-axes projected onto the global X-axes with respect to the angle ϕ_3 . The subscripts can be dropped from equation 3.7 because the ϕ_3 angle is the only angle of the three that is non-zero for the derivation. This is due to the fact that ϕ_2 and ϕ_1 become zero for plane frame structures.

Equation 3.7 can now be used to transform the element model from the local to the global reference frame. This is accomplished by multiplying the stiffness matrix, k , by

$$R_{(6,6)} = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix} \quad (3.8)$$

and its transpose,

$$R_{(6,6)}^T = \begin{bmatrix} \Lambda^T & 0 \\ 0 & \Lambda^T \end{bmatrix} \quad (3.9)$$

to yield

$$\begin{matrix} K_{\text{global}} & = & R & k & R^T \\ (6,6) & & (6,6) & (6,6) & (6,6) \end{matrix} \quad (3.9)$$

The stiffness matrix for the global beam-column element model now becomes

$$K_G = \begin{bmatrix} h_1 & h_2 & h_4 & -h_1 & -h_2 & h_4 \\ & h_3 & h_5 & -h_2 & -h_3 & h_5 \\ & & h_6 & -h_4 & -h_5 & -h_7 \\ & & & h_1 & h_2 & -h_4 \\ & \text{symmetric} & & & h_3 & -h_5 \\ & & & & & h_6 \end{bmatrix} \quad (3.10)$$

where

$$h_1 = \alpha(\beta c^2 + 12s^2) \quad (3.10a)$$

$$h_2 = \alpha cs(\beta - 12) \quad (3.10b)$$

$$h_3 = \alpha(\beta s^2 + 12c^2) \quad (3.10c)$$

$$h_4 = -\alpha 6sL \quad (3.10d)$$

$$h_5 = \alpha 6cL \quad (3.10e)$$

$$h_6 = \alpha 4L^2 \quad (3.10f)$$

$$h_7 = \alpha 2L^2 \quad (3.10g)$$

3.3 AISC Code Requirements

All of the structures that are designed in this investigation comply with the 1978 AISC specifications [23] for stresses and stability. These requirements are implemented in steps 7 and 8 in the computer program flowchart in Figure 3.1.

Figure 3.4 is a flowchart for computing maximum allowable stresses. Section 1.6 of the AISC specifications states that members subjected to both axial compression and bending stresses must satisfy one of the following equations. For the general case,

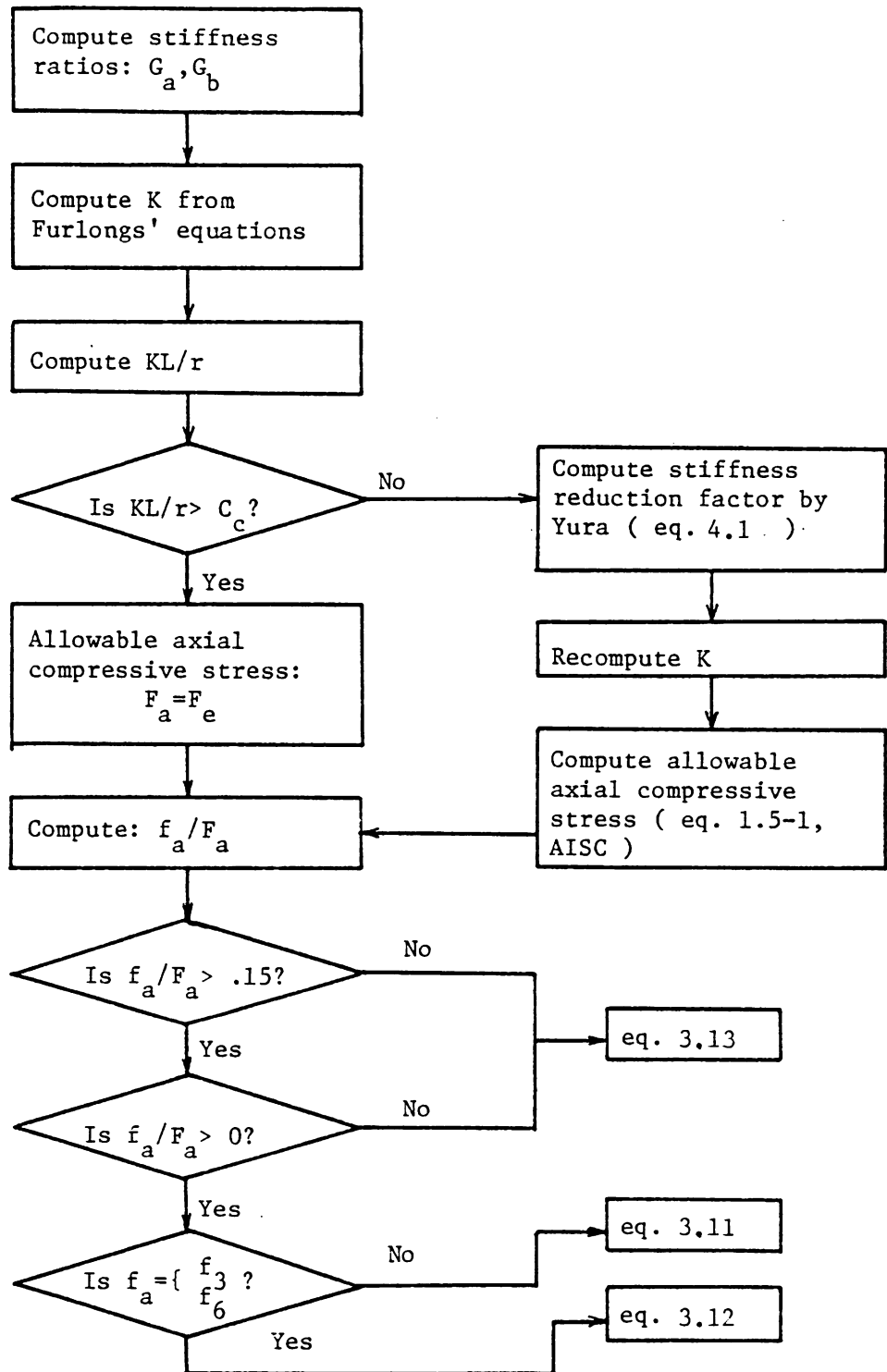


Figure 3.4 Flowchart for Computation of K-factors and Interaction Equations

$$\frac{f_a}{F_a} + \frac{C_m f_b}{\left(1 - \frac{f_a}{F_e}\right) F_b} \leq 1.0 \quad (3.11)$$

where

f_a = computed axial stress, ksi

f_b = computed bending stress, ksi

F_a = allowable axial compressive stress per
AISC section 1.5.1.3

F_b = allowable bending stress per AISC
section 1.5.1.4

F_e = Euler buckling stress divided by a
factor of safety of 1.92

c_m = secondary moment reduction factor

However, if the maximum combined stress occurs at either end of the member, then the equation

$$\frac{f_a}{0.6F_y} + \frac{f_b}{F_b} \leq 1.0 \quad (3.12)$$

where F_y is the yield stress of the steel, may be used instead of Equation 3.11. Then again, if the ratio of the computed axial stress to the allowable axial stress, f_a/F_a , is greater than 0.15 or if the computed axial stress is tensile, then the equation

$$\frac{f_a}{F_a} + \frac{f_b}{F_b} \leq 1.0 \quad (3.13)$$

may be used instead of Equation 3.11 or Equation 3.12.

The allowable bending stress used in these equations is

$$F_b = 0.66F_y \quad (3.14)$$

which is the allowable stress for compact sections. In this study, the assumption that all members are compact is a valid one because all frames are assumed to be adequately braced laterally. Another reason for this lies in the fact that the cross-sectional property relationships used in the design procedure are derived from the cross-sectional properties of wide flange sections which are considered compact sections [23].

Axial compressive stress controls the axial component of the above interaction equations. The allowable axial stress for these equations can be calculated from equations in AISC section 1.5.1.3. The formulas express the allowable compressive axial stress as a function of the members effective unbraced length and slenderness. These parameters take into account the stability aspect of the element and the structure. A more detailed discussion on the significance of the effective length of a member with respect to its strength and stability is presented in the next two sections.

3.4 Effective Length and Stability Analysis

Since 1961 AISC has required that a stability analysis be incorporated into the design of columns in unbraced frames. In practice this is usually done by evaluating the critical stress for a column as though it was a pinned-end braced column. To do

this, an equivalent or effective column length is substituted for the actual length in the Euler buckling equation so that the equation

$$F_e = \frac{\pi^2 E}{(L/r)^2} \quad (3.15)$$

becomes

$$F_e = \frac{\pi^2 E}{(KL/r)^2} \quad (3.16)$$

where K is the effective length factor for the column. In general K-factors can be found by performing a buckling analysis on all or part of the structure. However, an exact analysis can become very complex when the number of members in a structure becomes large. For this reason it is more desirable to get an approximation of the buckling effects in a structure.

The simplest method for the determination of K-factors is proposed by the Column Research Council's guide [15]. It approximates the critical load of a column in a frame by using a subassemblage of the structure for the stability analysis [9]. An equation is derived that relates the effective length to the relative stiffnesses of the members that frame into both ends of the column. Julian and Lawrence [16] transformed the equation into easy to read nomographs that are available in many structural design texts. Furlong [8] developed a set of equations that closely approximates Julian and Lawrence's equation and these are used to compute K-factors in the computer program. They are:

$$K = \frac{20 - R'}{20} \sqrt{1 + R'} \quad [R' < 2.0] \quad (3.17)$$

$$K = 0.9 \sqrt{1 + R'} \quad [R' \geq 2.0] \quad (3.18)$$

where

$$R' = \frac{G_a + G_b}{2} \quad (3.19)$$

and G_a and G_b are the beam-to-column stiffness ratios at each end of the column. These ratios can be stated algebraically as:

$$G = \frac{\sum_1 (I/L)_{col.}}{\sum_1 (I/L)_{bm.}} \quad (3.20)$$

These equations are valid only if the following assumptions are made in the analysis:

1. Behavior of the structure is purely elastic.
2. All members have a constant cross-section.
3. All joints are rigid.
4. Rotations at the opposite ends of restraining beams are equal and produce reverse curvature bending.
5. All columns buckle simultaneously.
6. Stiffness parameters $L\sqrt{P/EI}$ are equal.
7. Joint restraint that is distributed to the columns above and below the joint are proportional to the I/L of the column.

It should be noted that the above equations are valid under idealized conditions and that actual conditions may result in overly conservative designs.

3.5. Fully Stressed Design

The fully stressed design (FSD) approach is a simple and effective strength optimization method that economically produces low-weight designs for many classes of simple and complex structures. It is an iterative procedure that searches for a design in which each member is subjected to a limiting stress under at least one of the specified loading conditions. In chapter two, it was pointed out that a FSD does not always lead to a minimum-weight design because an objective weight function is absent in the procedure. However, numerical evidence [32] suggests that a FSD is close to a minimum-weight design in many cases. For this reason and for computational simplicity, the FSD method using a stress-ratio modification algorithm is used as the design algorithm in this study. A description of this method is presented in the next section. A detailed description of other FSD member modification methods can be found in Reference 11.

In the simplest of FSD approaches, the ratio of the maximum calculated stress to the allowable stress is used as a scaling factor to adjust the design variable of each member at the end of an analysis cycle. The new value of the design variable is then used in the next analysis cycle. This process is repeated until convergence is achieved. This can be expressed algebraically as

$$X_i^{k+1} = \frac{\sigma_i^k}{\bar{\sigma}_i} X_i^k \quad (3.21)$$

where the value of the design variable of member i in the $(k + 1)^{th}$ cycle is calculated from the stress-ratio found in the k^{th} cycle, σ_i^k is the maximum stress in the member and $\bar{\sigma}_i$ is the maximum allowable stress permitted for that member. If the stress-ratio is greater than unity, the member is overstressed and it's cross-section is increased. Conversely, if the ratio is less than unity, the cross-section is decreased because the member is understressed.

Realizing that an increase in the member size will draw more load to the member and a decrease will reduce the load carried by the member, Gallagher [11] suggested that equation (3.21) be modified by adding an "over-relaxation" factor to account for this behavior. With this addition, the equation becomes

$$X_i^{k+1} = \left[\frac{\sigma_i^k}{\bar{\sigma}_i} \right]^\beta X_i^k \quad (3.22)$$

where β has a value greater than unity. He believed that this modification would increase the convergence rate of the method. However, Equation 3.21 will be the only equation used in this investigation.

3.6. Cross-Section Property Relationships

In Section 3.5.1, it is stated that the design variable of each member is proportioned by the ratio of the computed stress to the allowable stress after each analysis cycle. The reportioned design variable, usually a cross-sectional property such as area or bending moment of inertia, is then used to calculate the remaining section

properties required for analysis through empirically derived dependent section property relationships. This is referred to as a continuous solution which is used in this study. In this application it is more practical than a discrete solution in which the cross-sectional properties of member sections are selected from a list of available standard sections via a search routine. This creates a lack of continuity between the section property relationships which may hinder or even prevent convergence to a fully stressed design.

CHAPTER 4

DESIGN STUDIES

4.1. Modified K-Factors for Inelastic Beam-Columns

In most structures, the assumption of elastic behavior in all members, especially columns, is unrealistic and overly conservative. Yura [43] reports that the stress levels in most multistory structures exceeds $0.5F_y$ which is considered to be the critical stress of the column. Therefore, columns in multistory buildings are subjected to stress levels in the inelastic range. Consequently, the existing K-factor equations yield designs that are very conservative. Yura corrects this by suggesting a procedure that takes into account the inelastic behavior if the situation calls for it. It simply reduces the end restraint factors for a column in the inelastic range by modifying equation 3.20 as shown below:

$$G_{\text{inelastic}} = \frac{F_a}{F_e} G_{\text{elastic}} \quad (4.1)$$

where

F_a = allowable compressive stress

F_e = Euler buckling stress

and F_a/F_e is a reasonably accurate reduction factor due to plasticity. This produces lower K-factors which give higher allowable stresses. If the behavior is elastic, however, $F_a = F_e$ and the ratio has no effect. This method provides a more realistic design. A flowchart representation of this procedure is illustrated in Figure 3.4. A

similar procedure based on Yura's method has been presented by Disque [6].

4.2. Dependent Section Property Functions

The design of plane frames requires three section properties for each member, the cross sectional area (A), the moment of inertia (I), and the section modulus (S), which contains a measure of the section depth. If the possible section properties are those included in the AISC Manual, then it is not possible to consider all three properties as independent of each other. In other words, the tabulated section properties that are available for selection in the design process define some functional relationship between these three quantities. Therefore, one of the properties is considered the independent variable while the other two are the dependent variables. The nature of the dependency is obtained from empirically derived functions based on the tabulated values.

Two past studies of structural designs have produced empirically derived dependent section property functions, Brown and Ang [2], and Arora, Haug and Rim [1]. However, two questions were posed that led to a decision to derive a new set of section property functions. The first concerned the decision to use the moment of inertia (I) as the independent design variable rather than the section modulus (S). Neither study provided a specific reason for this choice. The selection of S as the independent variable instead of I could be justified by noting that the stress in an element is more sensitive to changes in S than to changes in I .

The second question concerns the fact that one set of equations is used in previous works to design both beams and columns in frame structures. This may be acceptable theoretically but it is unrealistic in a practical sense. A quick inspection of the beam and column design tables in the AISC Manual [23] reveals that the section property relationships of beams are vastly different from the column property relationships. This can be explained by examining the functions of each. A beam's primary function is to resist moment while a column's main function is to resist axial load. Therefore, the cross-sectional area for a column section is greater than the area of a beam cross-section with the same section modulus. The development of two distinct sets of section property equations, one for beams and the other for columns, is a more realistic way of approaching the steel frame design problem.

With two sets of section property equations to choose from, some criterion should be established to distinguish which members should be designed as beams and which should be designed as columns. A possible solution would be to say that vertical members are designed as columns and horizontal members are designed as beams. However, geometric orientation within the structure is not always a good rule of thumb since large lateral loads and low gravity loads may produce large bending stresses in members traditionally labelled as columns. Therefore, the fraction of f_a/F_a , where f_a is the computed axial stress in a member and F_a is the allowable axial stress for that member, is used as the segregating criterion consistent with

section 1.6.1 of the AISC specification on combined stresses [23].

This simply infers that members with f_a/F_a ratios less than 0.15 are treated as beams while the rest are treated as columns.

4.3. Curve-Fitting Algorithm

The dependent section property functions were developed by the use of a univariate curvilinear regression model using orthogonal polynomials through the data points. Details of this method are contained in Reference 7.

4.4. Column Equations

Figures 4.1 and 4.2 show the curves produced by the curve fitting algorithm for section properties of columns. The data points consist of the available column sections found in the AISC Manual. The two distinct regions of Figure 4.2 represent the functions for W14 sections and W12 sections. All of the data points in the S vs I function can be represented by the following equations:

For $(5 \text{ in}^3 \leq S \leq 50 \text{ in}^3)$,

$$I = -11.88349 + [4.132184]S + [9.396914 \times 10^{-3}]S^2 \quad (4.4)$$

and for $(50 \text{ in}^3 \leq S \leq 1200 \text{ in}^3)$,

$$I = -9.179416 + [6.954839]S + [2.4126 \times 10^{-3}]S^2 \quad (4.5)$$

The curves produced by these equations are shown in Figure 4.1.

The curves for S vs A in Figure 4.2 are:

$$A = 2.346176 + [2.443544 \times 10^{-1}]S - [3.459238 \times 10^{-5}]S^2 \quad (5 \text{ in}^3 \leq S \leq 50 \text{ in}^3) \quad (4.6)$$

$$A = 6.1575 \times 10^{-1} + [1.871112 \times 10^{-1}]S - [1.518453 \times 10^{-5}]S^2 \quad (50 \text{ in}^3 \leq S \leq 1200 \text{ in}^3) \quad (4.7)$$

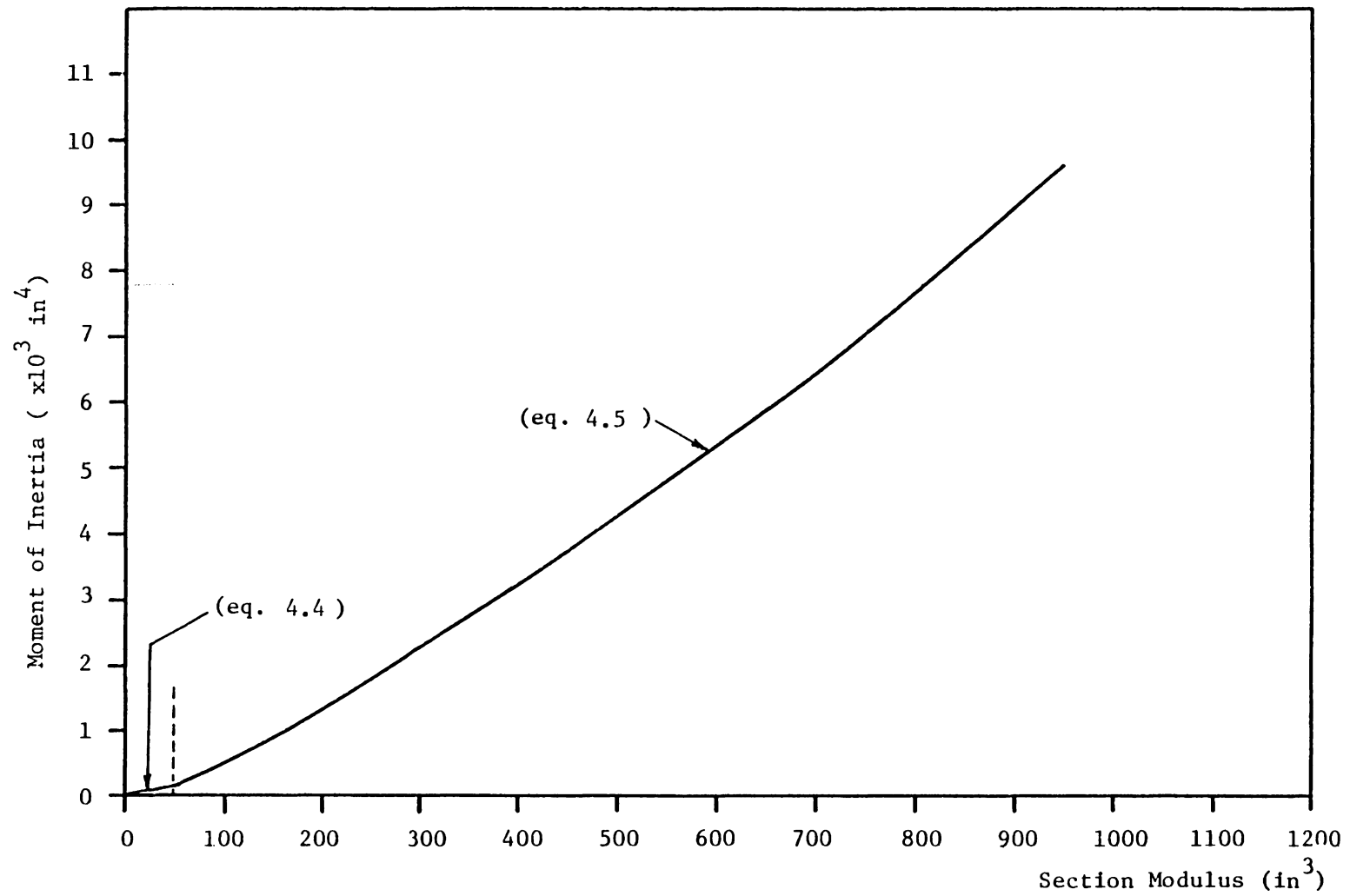


Figure 4.1 Section Modulus vs. Bending Moment of Inertia for Columns

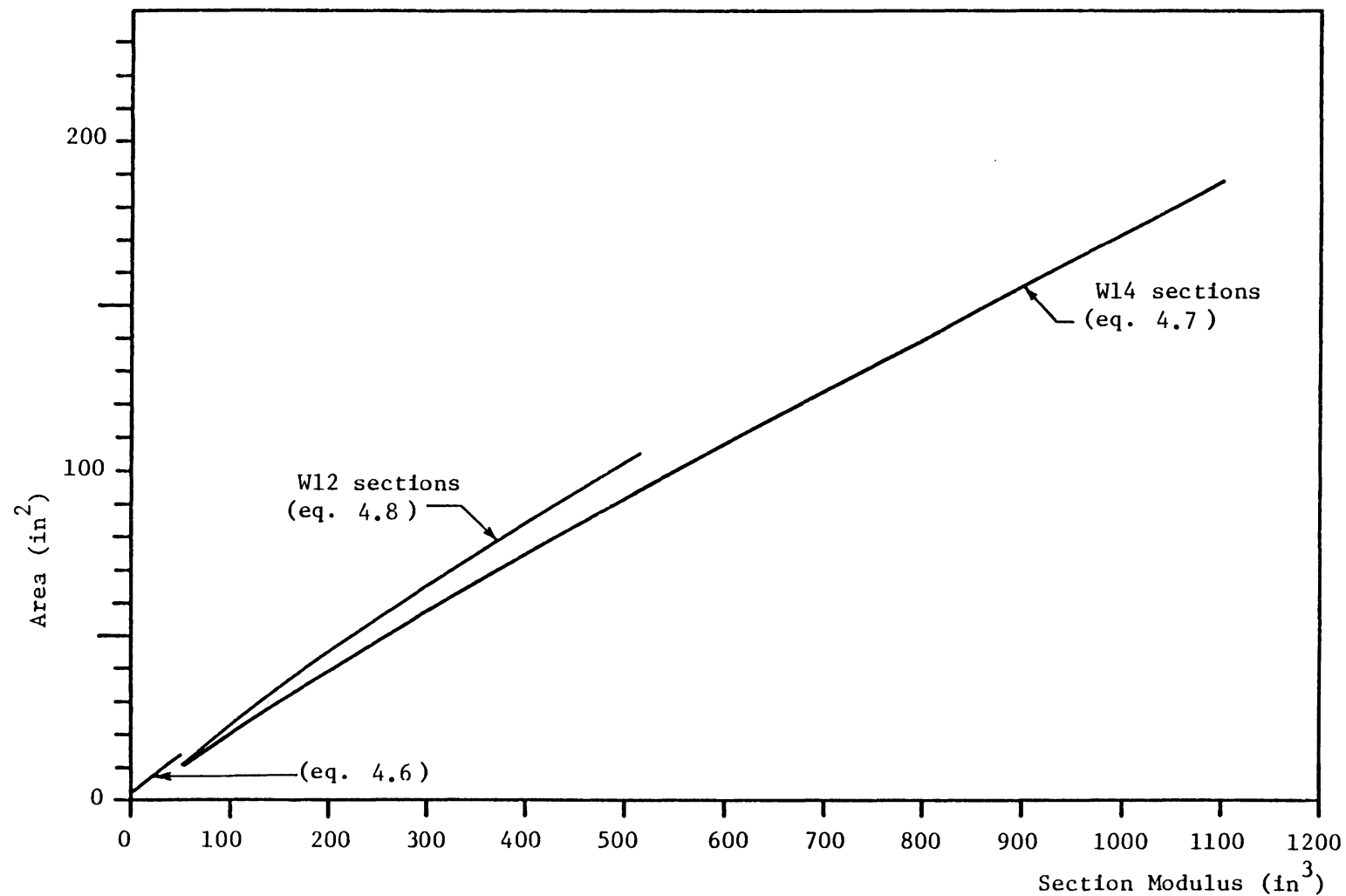


Figure 4.2 Section Modulus vs. Area for Columns

$$A = 2.559928 \times 10^{-1} + [2.192961 \times 10^{-1}]S - [3.082755 \times 10^{-5}]S^2$$

$$(50 \text{ in}^3 \leq S \leq 500 \text{ in}^3) \quad (4.8)$$

Equation 4.7, which is the W14 curve, provides a smaller area for a given section modulus. This, in turn, produces a lighter section design which is desirable. However, equation 4.8 may be used if weight is not critical and S values are not greater than 500 in³. This size limitation is due to the unavailability of W12 sections with higher S values.

4.5. Beam Equations

Equations 4.9, 4.10, and 4.11 are the beam section property functions developed by the algorithm described in Section 4.3 using the properties of all the available beam sections as data points. These are shown as solid curves within the data regions of Figures 4.3 and 4.4.

$$I = -13.15975 + 4.499167S + 4.945434 \times 10^{-2}S^2 \quad (0 \leq S \leq 50 \text{ in}^3)$$

$$(4.9)$$

$$I = -107.5656 + 7.474257S + 2.30901 \times 10^{-2}S^2 - 1.222851$$

$$\times 10^{-5}S^3 \quad (50 \text{ in}^3 \leq S \leq 1200 \text{ in}^3) \quad (4.10)$$

$$A = 3.989864 + 0.1468865S - 1.605061 \times 10^{-4}S^2 + 8.959847$$

$$\times 10^{-8}S^3 \quad (0 \leq S \leq 1200 \text{ in}^3) \quad (4.11)$$

4.6. Comparison of Beam Equations

The data points of the beam cross sectional properties do not fall in a pattern through which a unique curve can be fitted, as did the column section data points. Rather, a data region is defined as shown in Figures 4.3 and 4.4. Such data regions can be represented

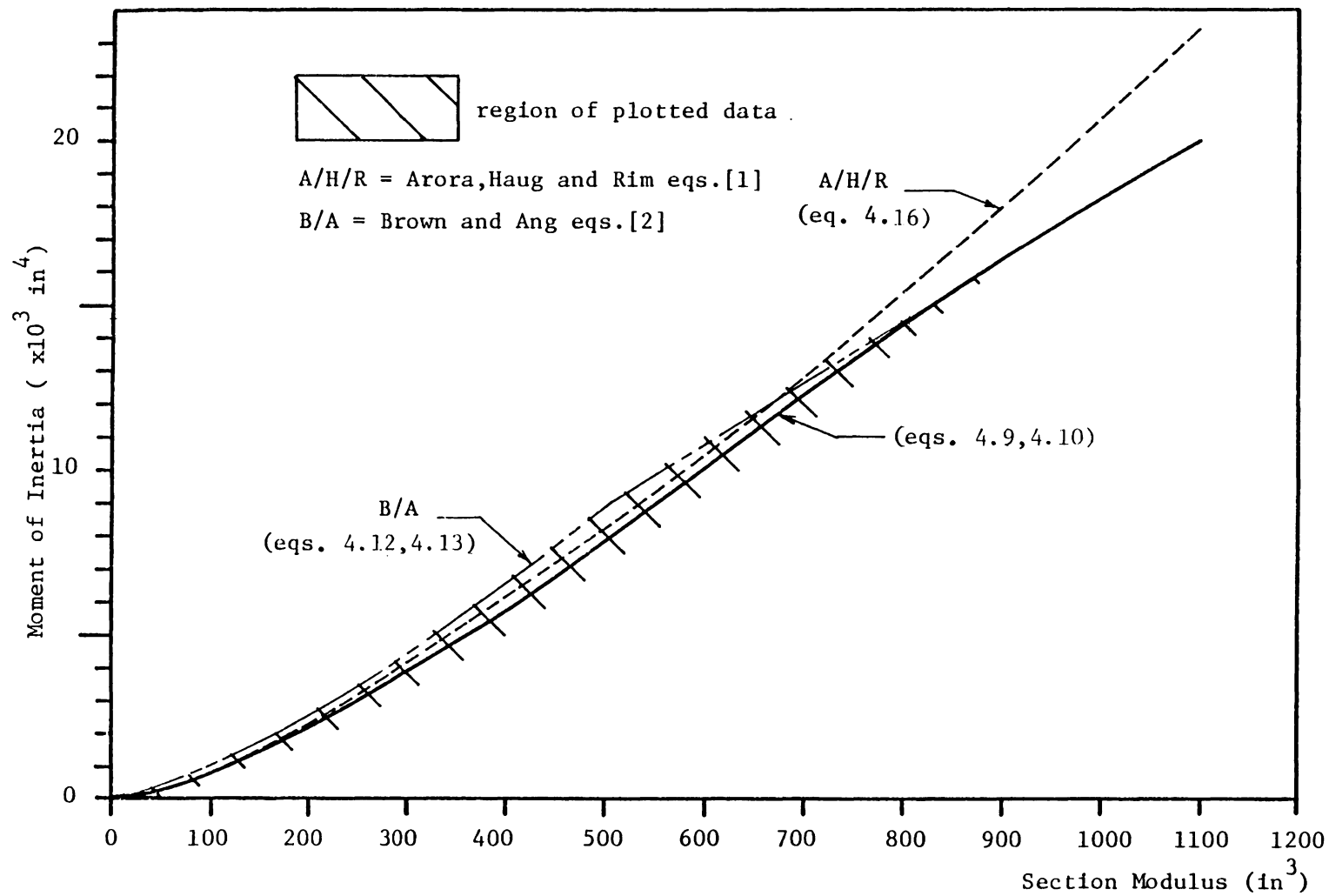


Figure 4.3 Section Modulus vs. Moment of Inertia for Beams

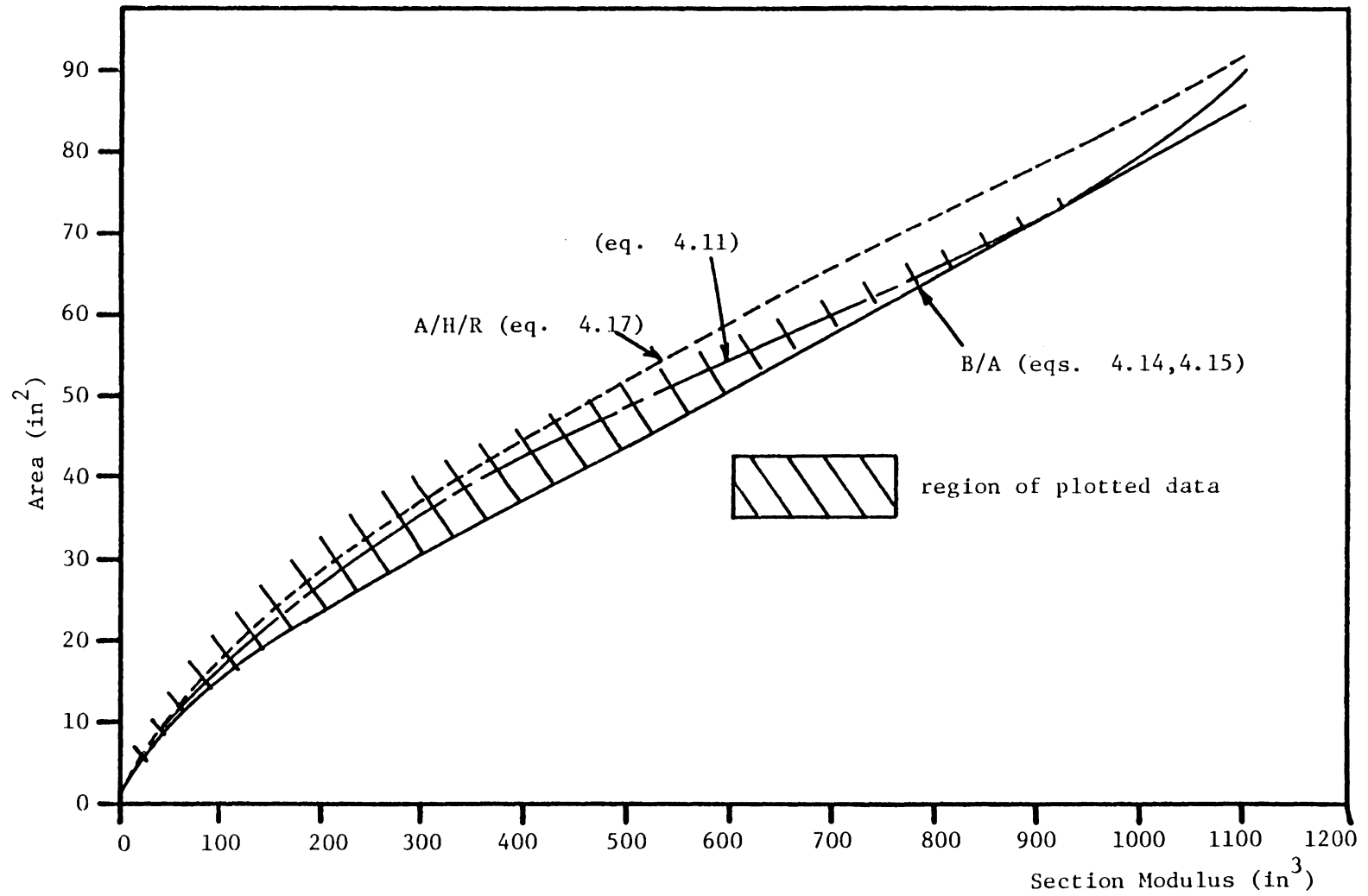


Figure 4.4 Section Modulus vs. Area for Beams

by different functions. Two additional sets of functions are defined in equations 4.12-4.15, (from Brown and Ang (B/A) [2]) and equations 4.16 and 4.17 (from Aurora, Haug, and Rim (A/H/R) [1]).

The B/A equations are:

$$S = \sqrt{60.6I + 84100} - 290. \quad (0 \leq I \leq 9000.\text{in}^4) \quad (4.12)$$

$$S = \frac{I + 311.0664}{18.5111} \quad (9000.\text{in}^4 \leq I \leq 20,300.\text{in}^4) \quad (4.13)$$

$$A = 0.464\sqrt{I} \quad (0 \leq I \leq 9000.\text{in}^4) \quad (4.14)$$

$$A = \frac{I + 2300}{256} \quad (9000.\text{in}^4 < I \leq 20,300.\text{in}^4) \quad (4.15)$$

It should be noted that equation 4.13 does not appear in the publication by Brown and Ang [2]. The published equation did not fit the specified data points, so equation 4.13 was developed to provide a better representation. The A/H/R equations are:

$$S = 0.58 I^{0.75} \quad (0 \leq I \leq 20,300.\text{in}^4) \quad (4.16)$$

$$A = 0.58 I^{0.5} \quad (0 \leq I \leq 20,300.\text{in}^4) \quad (4.17)$$

The two sets of functions are also shown in Figures 4.3 and 4.4 so that they can be seen relative to the derived set, equations 4.9-4.11.

Efficient sections can be defined as those that provide the largest I and the smallest A within the data regions corresponding to a specified S value. This combination of section properties provides smaller bending stresses and smaller weight. The B/A set (equations 4.12-4.15) is the most efficient in this regard compared to the

other two sets. The A/H/R set (equations 4.16-4.17) is seen to be valid only within a limited S range. The derived equation set (equations 4.9-4.11) is a median curve within each region, as intended.

A fully stressed design could be achieved using any one of the three sets of dependent section property functions defined above. However, the total weight of the final design depends on these functions. This feature is discussed further in Chapter 5.

CHAPTER 5

DESIGN EXAMPLES

5.1. Design Example 1

A comparison study is presented to show the effect of the inelastic beam-column modification procedure, outlined in Chapter 4, relative to the procedure that uses the elastic k-factors in terms of the weight of the final design. The structure used in this comparison is a six story, two bay frame with four loading conditions. The structure and its' loading conditions are shown in Figures 5.1 through 5.5. This structure was previously designed by Arora, Haug and Rim [1], using an iterative analysis-design approach. The analysis portion was based on a matrix approach with a stress constraint based on the AISC specifications for combined stresses [23]. The design process was based on the steepest descent algorithm. The design solution for this procedure is shown in Table 5.1. The weight of each member is found by inserting the moment of inertia values into the area equation used by Arora, Haug and Rim in their paper and multiplying the area by the density of steel, 490 lbs./ft^3 , and by the member's length. By doing this, the total weight of the structure is found to be 20876 pounds.

The structure was then designed by the computer code used in this study. The section property equations used by Arora, Haug and Rim were also used in this design so that the only major difference between the two design programs was the procedure for K-factor calculations. It was assumed that the two design algorithms were

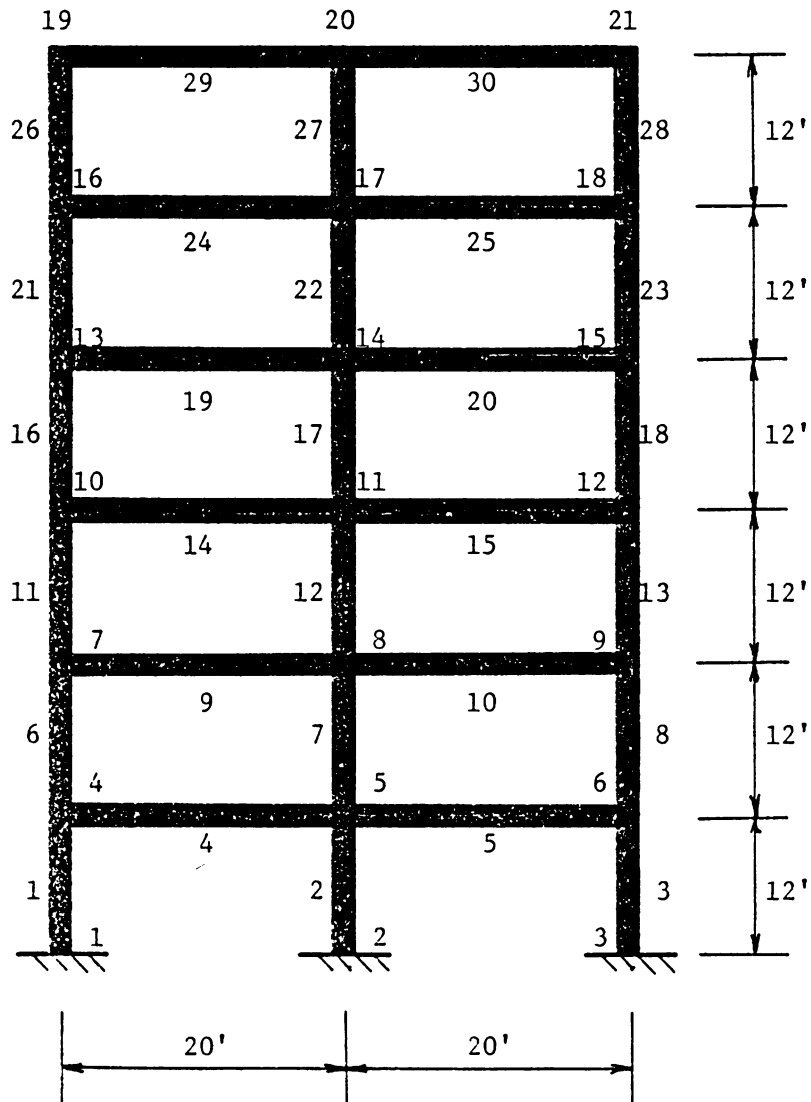


Figure 5.1 Design Example 1 - Member and Joint Numbers

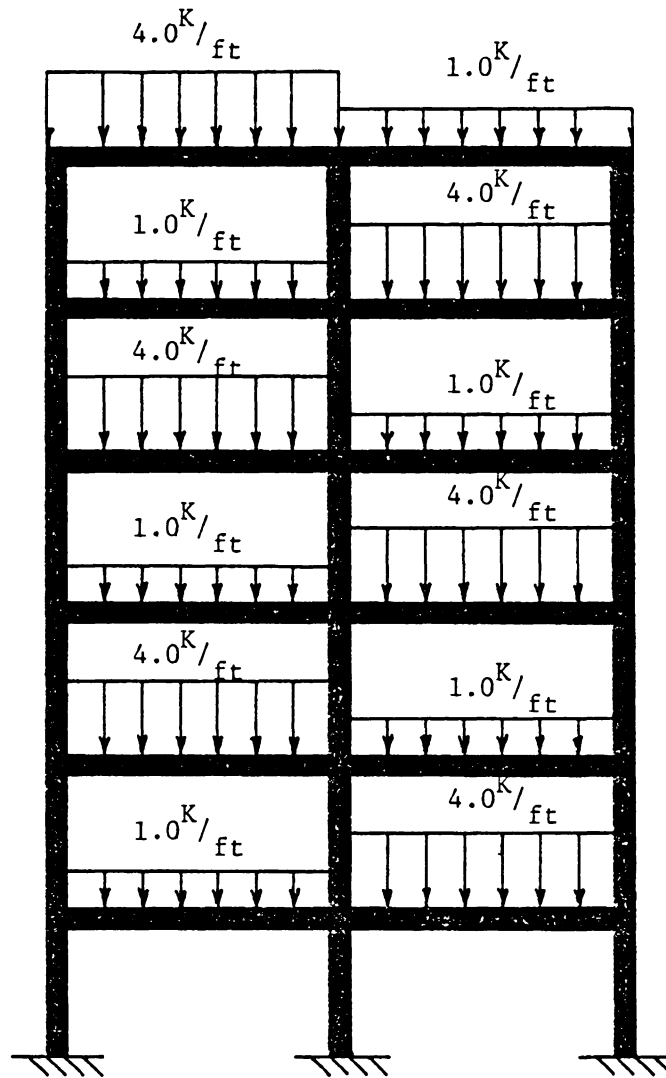


Figure 5.2 Design Example 1 - Load Condition 1

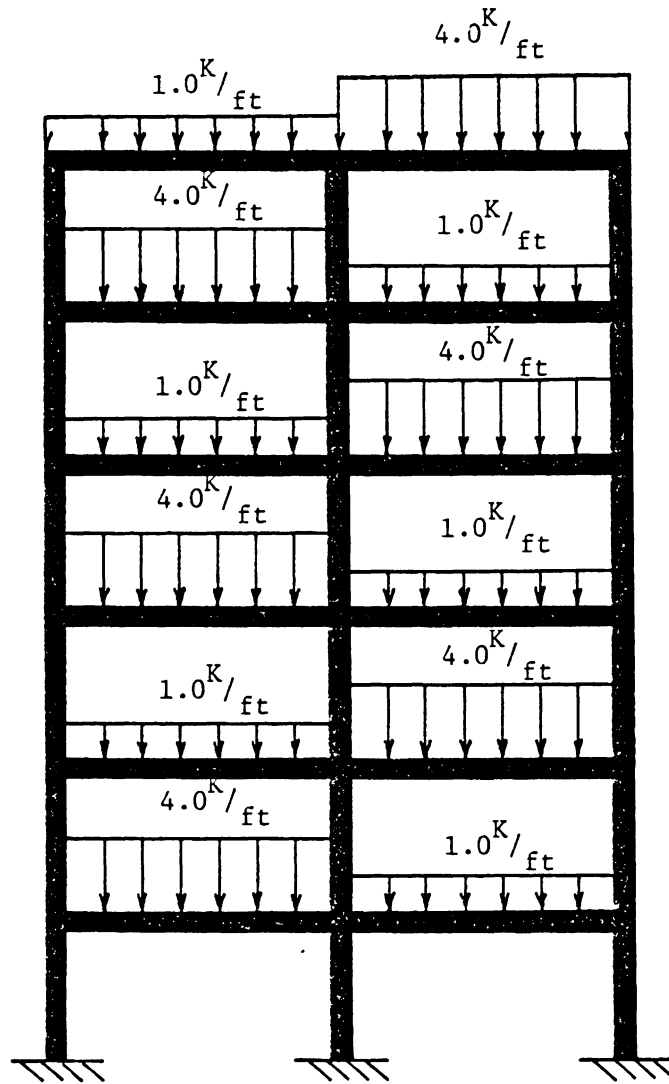


Figure 5.3 Design Example 1 - Load Condition 2

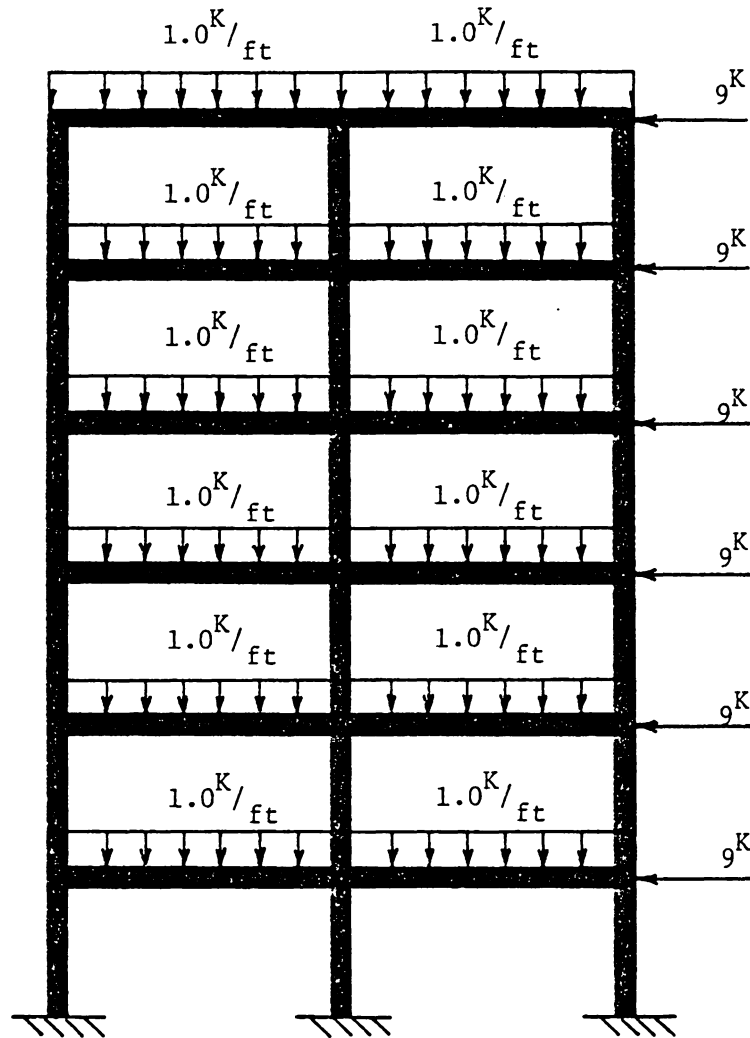


Figure 5.4 Design Example 1 - Load Condition 3

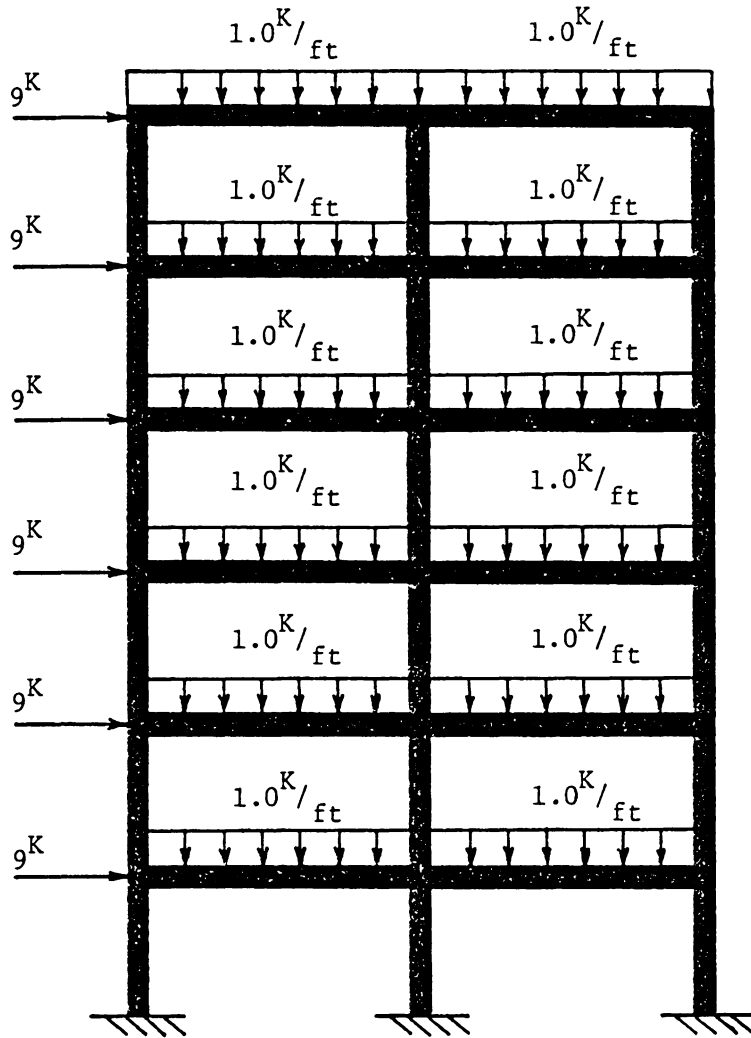


Figure 5.5 Design Example 1 - Load Condition 4

TABLE 5.1
Design Results of Example 1
by Arora, Haug, and Rim

Member Number	I (in. ⁴)	A (in. ²)	S (in. ³)	Length (ft.)	Weight (lb.)
1	1099.0	19.2	110.7	12	784
2	1489.7	22.4	139.1	12	915
3	1099.0	19.2	110.7	12	784
4	666.4	15.0	76.1	20	612
5	666.4	15.0	76.1	20	612
6	646.3	14.7	74.3	12	600
7	1044.5	18.7	106.6	12	764
8	646.5	14.7	74.4	12	600
9	699.1	15.3	78.9	20	1041
10	699.1	15.3	78.9	20	1041
11	468.3	12.6	58.4	12	514
12	723.5	15.6	80.9	12	637
13	467.5	12.5	58.3	12	510
14	521.9	13.2	63.3	20	898
15	521.9	13.2	63.3	20	898
16	425.3	12.0	54.3	12	490
17	472.7	12.6	58.8	12	514
18	425.3	12.0	54.3	12	490
19	481.8	12.7	59.6	20	864
20	481.8	12.7	59.6	20	864

TABLE 5.1 (continued)
 Design Results of Example 1
 by Arora, Haug, and Rim

Member Number	I (in. ⁴)	A (in. ²)	S (in. ³)	Length (ft.)	Weight (lb.)
21	394.3	11.5	51.3	12	470
22	397.1	11.6	51.6	12	474
23	394.1	11.5	51.3	12	470
24	530.8	13.4	64.1	20	912
25	530.8	13.4	64.1	20	912
26	498.6	13.0	61.2	12	531
27	394.9	11.5	51.4	12	470
28	498.6	13.0	61.2	12	531
29	450.6	12.3	56.7	20	837
30	450.6	12.3	56.7	20	837

Total Weight : 20876

- TABLE 5.2
Design Results of Example 1
Via Computer Code

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	Weight (lb.)
1	60.0	12.8	486.0	96.0	523
2	227.7	31.1	2875.0	99.5	1270
3	60.0	12.8	486.0	96.0	523
4	78.6	15.3	696.0	98.2	521
5	78.6	15.3	696.0	98.2	521
6	71.1	14.3	608.0	98.5	584
7	86.0	16.2	785.0	97.9	662
8	71.1	14.3	609.0	98.5	584
9	81.2	15.6	727.0	97.9	531
10	81.2	15.6	727.0	97.9	531
11	36.7	9.2	252.0	97.5	376.0
12	92.0	17.0	859.0	98.2	694
13	36.7	9.2	252.0	97.5	376
14	67.4	13.8	567.0	98.2	470
15	67.4	13.8	567.0	98.2	470
16	69.3	14.1	588.0	97.8	576
17	47.4	10.9	355.0	98.0	445
18	69.3	14.1	588.0	97.8	576
19	60.9	12.9	495.0	98.1	439
20	60.9	12.9	495.0	98.1	440

* Percent Fully Stressed

TABLE 5.2 (continued)

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	Weight (lb.)
21	13.6	4.8	67.0	97.8	196
22	51.8	11.6	399.0	98.0	474
23	13.6	4.8	67.0	97.8	196
24	60.3	12.8	488.0	97.8	436
25	60.3	12.8	488.0	97.8	436
26	66.6	13.7	558.0	97.9	559
27	10.9	4.1	50.0	98.3	167
28	66.6	13.7	558.0	97.9	559
29	60.0	12.8	485.0	97.9	436
30	60.0	12.8	485.0	97.9	436

Total Weight : 15007

* Percent Fully Stressed

essentially equivalent. The final design details are tabulated in Table 5.2. For this design the total weight of the structure is 15007 pounds. This structure is 28 percent lighter than the previous design which is quite significant. These results seem to verify the claim made in Chapter 4 and by Yura [43] that elastic K-factors in plane frame analysis are overconservative and unrealistic, and that modified K-factors due to inelastic behavior yield more realistic data.

5.2. Design Example 2

The structure in Figure 5.6 was used in a study to evaluate the significance of the dependent section property functions for beams on the total weight of the structure of the final design. The structure is subjected to the three loading conditions shown in Figures 5.7 through 5.9 and it is designed via the computer program developed for this investigation.

Two designs are presented. The first design uses the dependent section property functions developed by Brown and Ang (equations 4.12-4.15). The results are shown in Table 5.3. All members are fully stressed and the total structural weight is 7033 pounds. The second design uses the beam equations developed for this study (equations 4.9, 4.10, and 4.11). The final results for this design are presented in Table 5.4. All members are fully stressed and the total structural weight of this design is 8082 pounds, 15 percent heavier than the previous structure. A comparison of the results also show that while

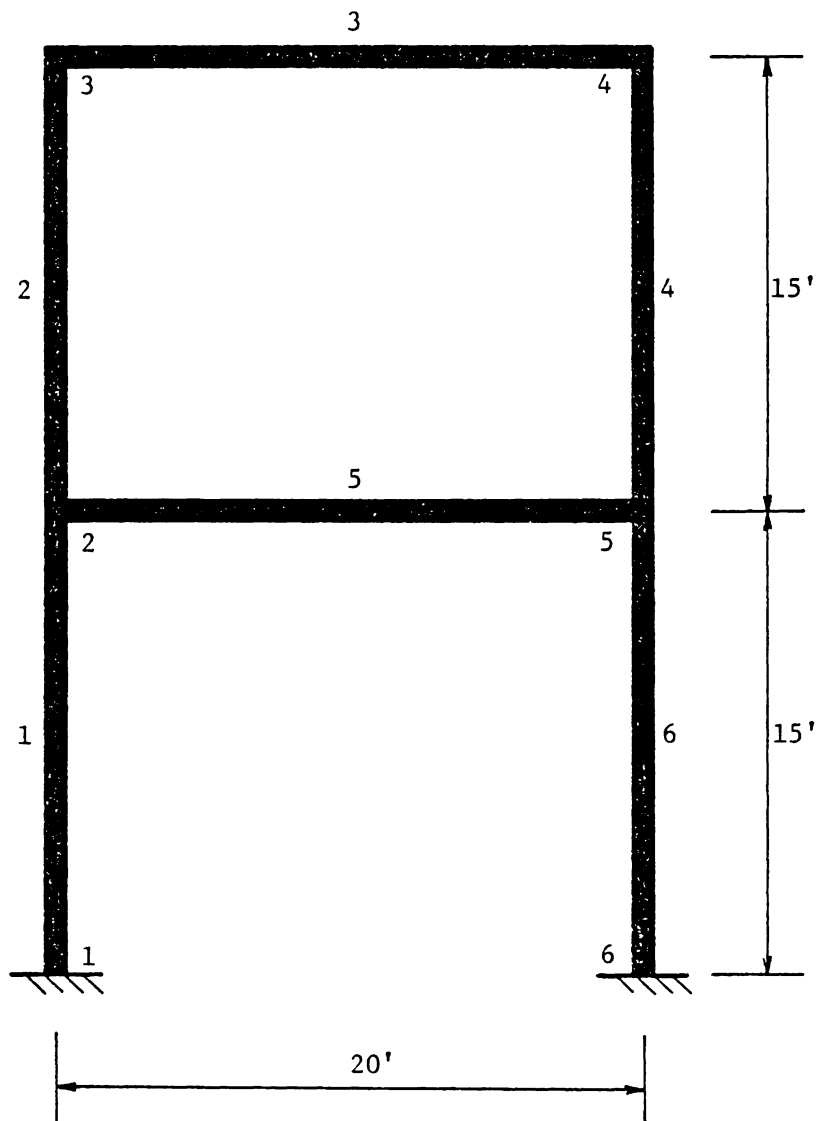


Figure 5.6 Design Example 2 - Member and Joint Numbers

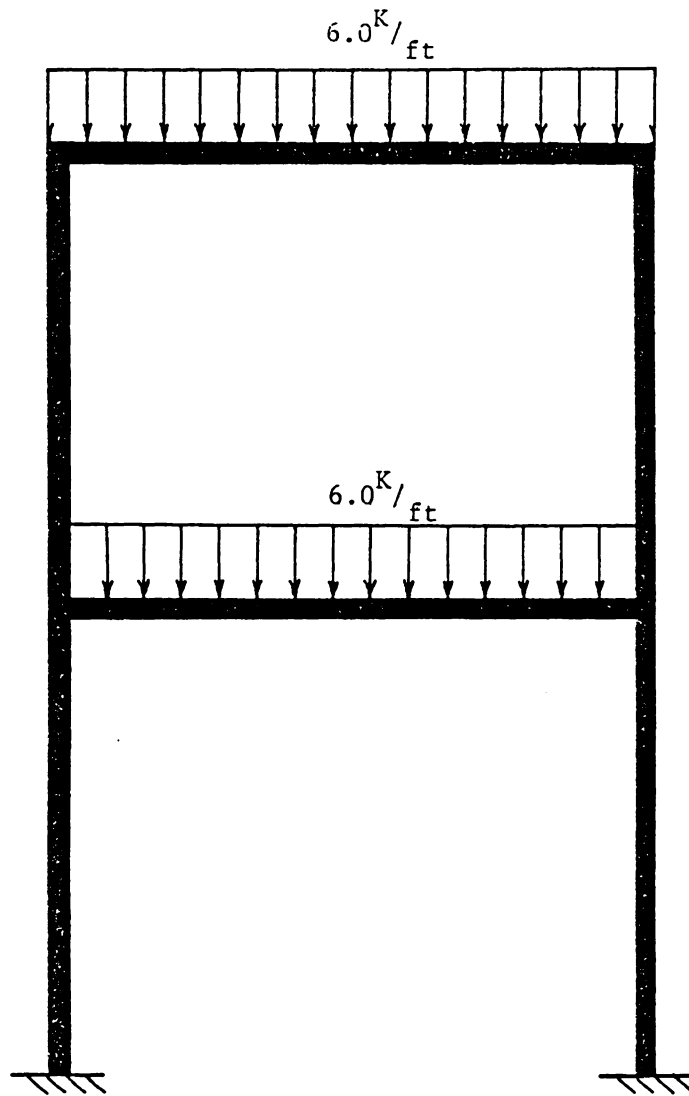


Figure 5.7 Design Example 2 - Load Condition 1

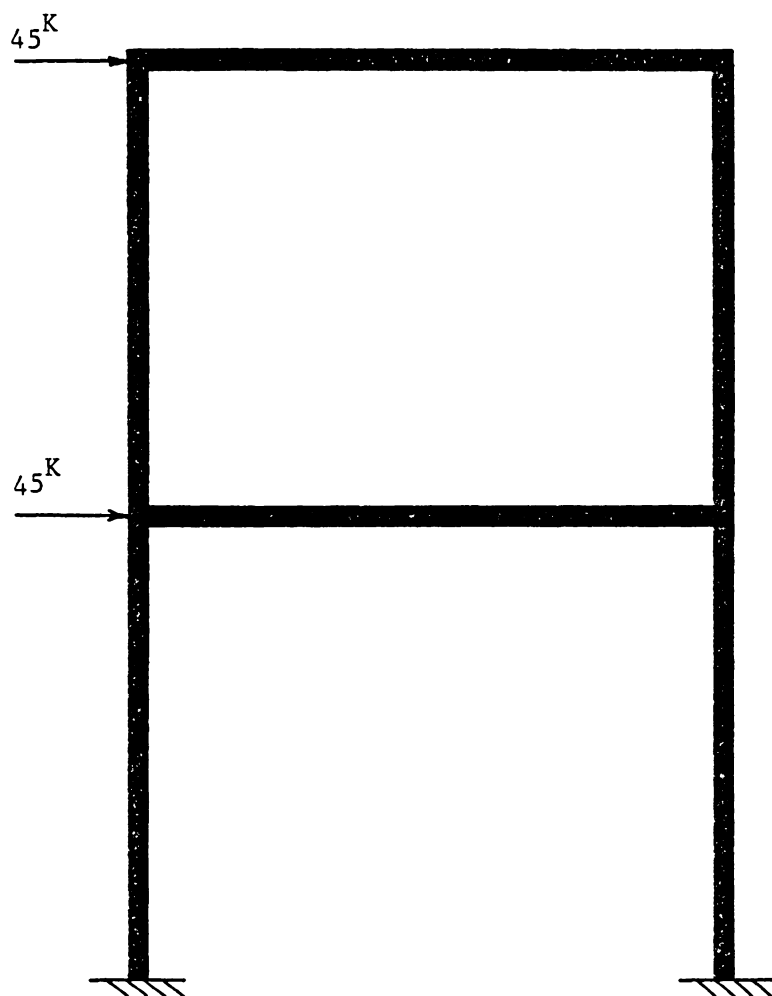


Figure 5.8 Design Example 2 - Load Condition 2

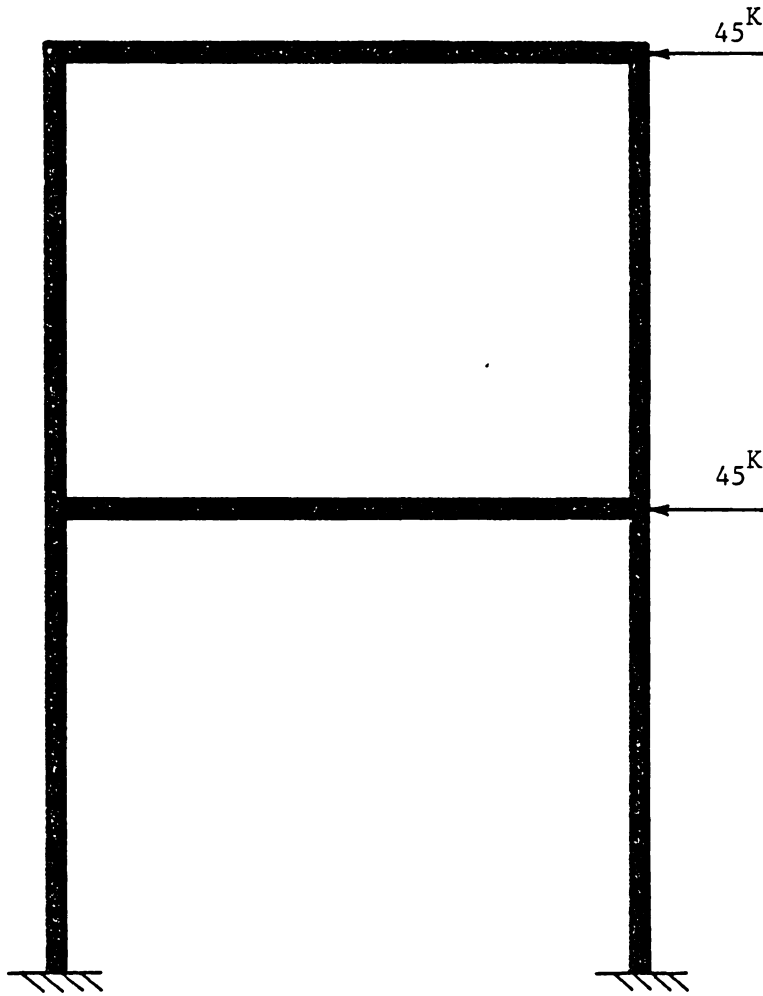


Figure 5.9 Design Example 2 - Load Condition 3

TABLE 5.3
Design Results of Example 2
Via Computer Code
(Brown and Ang Equations)

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	Weight (lb.)
1	259.2	27.8	3589.0	99.1	1419
2	94.4	15.0	1050.0	98.5	757
3	92.5	14.9	1027.0	97.0	1014
4	94.4	15.0	1050.0	98.5	757
5	213.0	24.5	2788.0	98.1	1667
6	259.2	27.8	3589.0	99.1	1419

Total Weight : 7033

* Percent Fully Stressed

TABLE 5.4
 Design Results of Example 2
 Via Computer Code
 (Median Equations)

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	Weight (lb.)
1	251.3	32.2	3035.0	99.3	1644
2	96.3	16.7	815.0	98.8	852
3	89.2	15.9	734.0	97.0	1082
4	96.3	16.7	815.0	98.8	852
5	220.2	29.5	2527.0	96.9	2008
6	251.3	32.2	3035.0	99.3	1644

Total Weight : 8082

*Percent Fully Stressed

the section moduli of the two designs vary by only 3 percent, the first structures' moment of inertia values are 9 to 28 percent higher than the I values for the second structure.

This example shows that the weight of the structure of the final design depends significantly on the dependent section property functions used. Comparing the dependent functions used in this example indicates that the more efficient structure relative to weight is obtained for functions that provide the largest I and the smallest A within the data region corresponding to a specific S value. Such functions would be defined by the upper boundary of the S-I data region (Fig. 4.3) and the lower boundary of the S-A data region (Fig. 4.4).

5.3. Design Example 3

In this example, a completely new set of beam section equations are developed for use in the design algorithm of the computer program. This is an extension of the results of the previous example and is an effort to update the equations presented by Brown and Ang [2]. The equations are derived in the same manner as Equations 4.9 through 4.11. The standard sections with cross-sectional properties along the upper boundary of the S vs I region in Figure 4.3 are used as data points for the curve-fitting algorithm. The derived equations are as follows:

$$I = 7.98S \quad (0 < S < 50\text{in}^3) \quad (5.1)$$

$$I = -191.3096 + (10.90819)S + (0.01840775)S^2 - (1.076241 \times 10^{-5})S^3 \quad (50\text{in}^3 < S < 1100\text{in}^3) \quad (5.2)$$

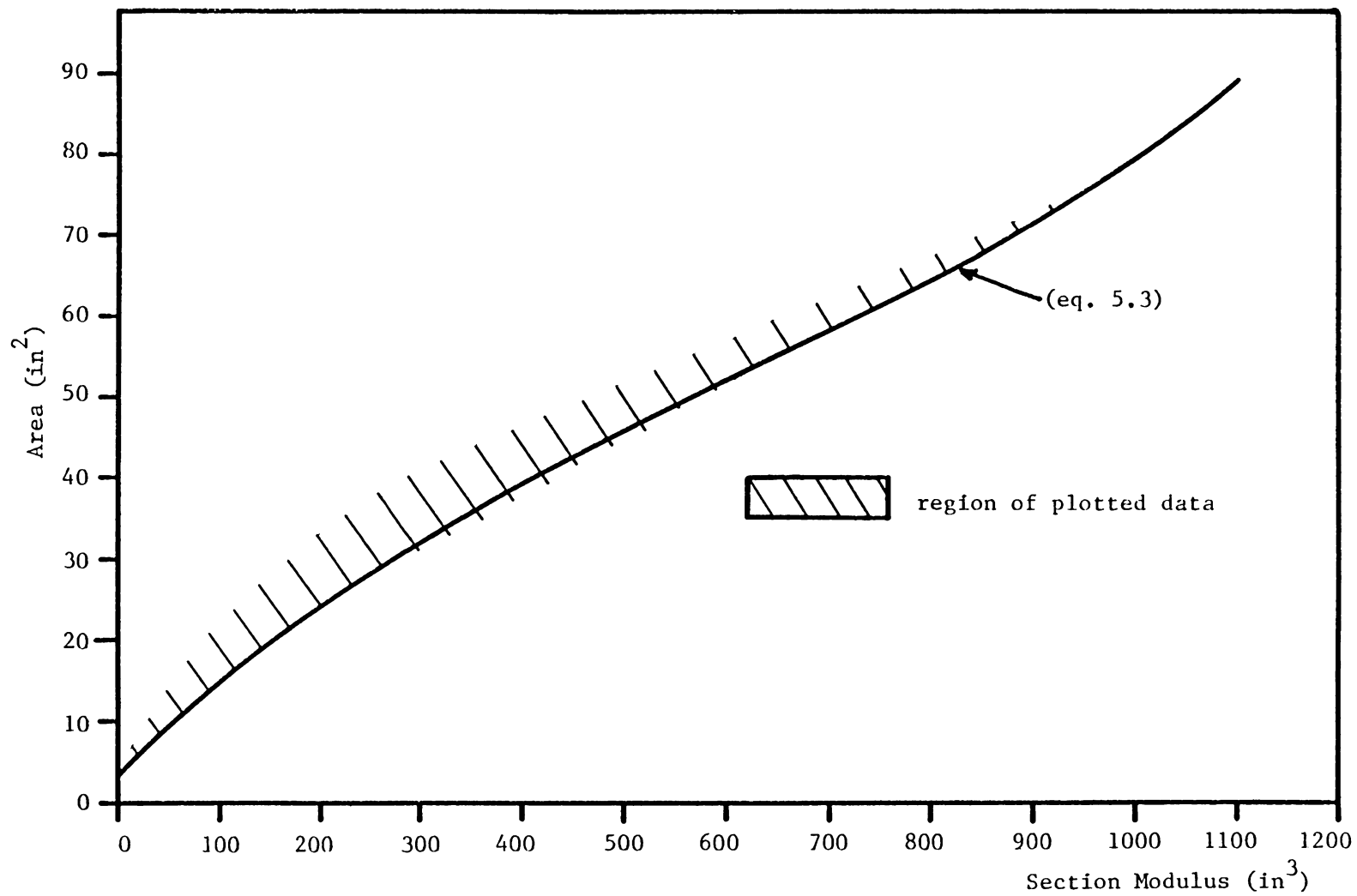


Figure 5.10 New Section Modulus vs. Area Equation for Beams

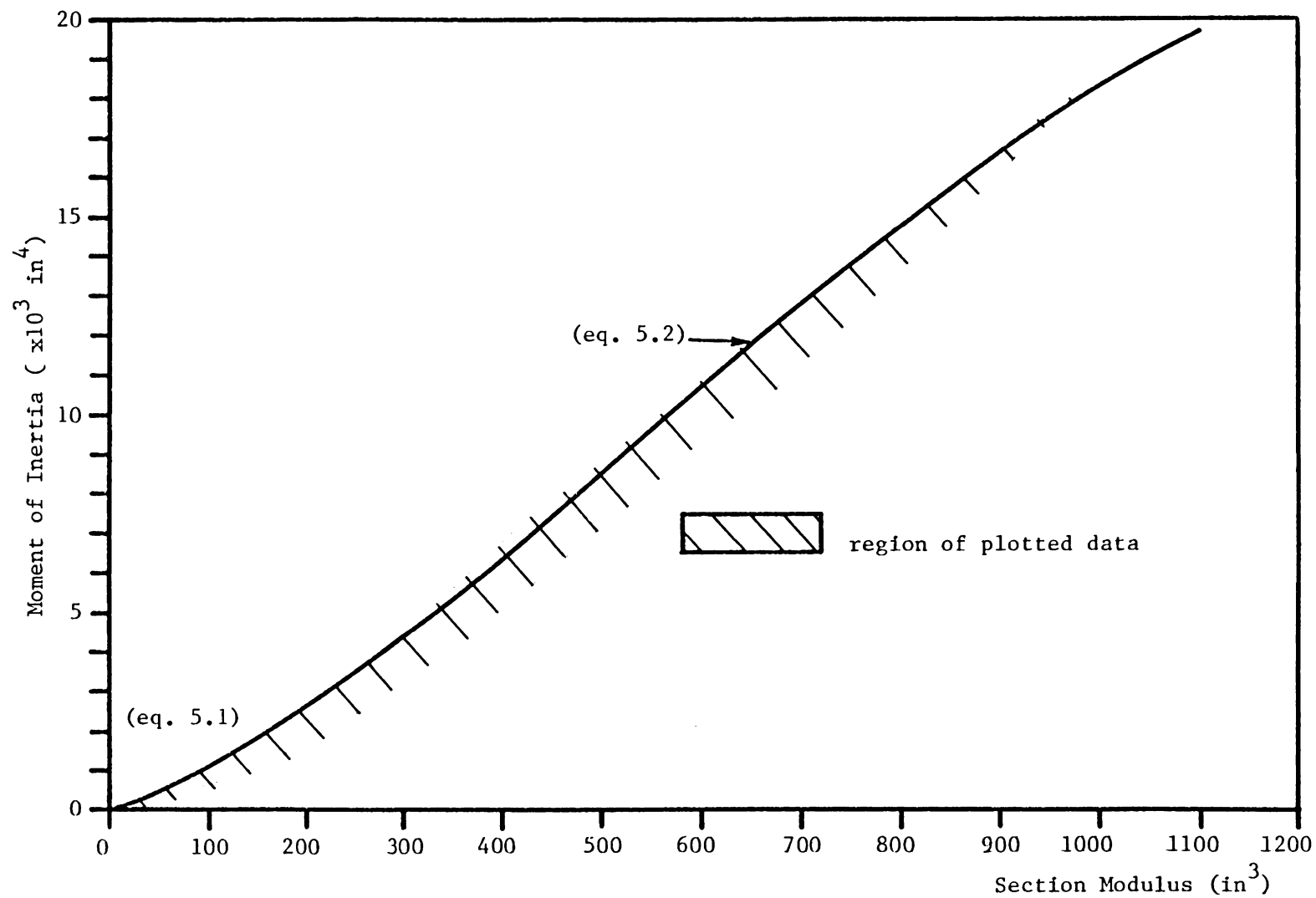


Figure 5.11 New Section Modulus vs. Moment of Inertia Equation for Beams

TABLE 5.5
Design Results of Example 3
Via Computer Code
(New Equations)

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	Weight (lb.)
1	258.5	29.0	3672.0	98.6	1480
2	96.1	14.3	1017.0	98.6	730
3	91.2	13.8	948.0	97.2	939
4	96.1	14.3	1017.0	98.6	730
5	216.5	25.5	2923.0	97.5	1735
6	258.5	29.0	3672.0	98.6	1480

Total Weight : 7094

* Percent Fully Stressed

$$A = 3.62415 + (0.119637)S - (9.708825 \times 10^{-5})S^2 + (5.341595 \times 10^{-8})S^3 \quad (0 < S < 1100 \text{ in}^3) \quad (5.3)$$

These equations are plotted in Figures 5.10 and 5.11. The structure used in this example is the same as the structure described in Design Example 2. The load conditions are also the same as the previous example. The final results are shown in Table 5.5. All of the members in this example are fully stressed. The total structural weight of 7094 pounds compares favorably with the design achieved by the B/A functions.

5.4. Conclusions

The following conclusions can be drawn with regard to the studies made:

1. The weight of the finally designed structure depends significantly on the dependent section property functions used for the development of a new trial structure. The most efficient fully stressed design, based on weight, is obtained for dependent section property functions that trace the upper limit of the S-I data region and the lower limit of the S-A data region, when all possible sections are available.
2. The weight of the structure can also be significantly altered by the allowable stress definitions. For example, the modification of beam-column design to include inelastic behavior results in a lighter design compared to the design based on elastic analysis.

CHAPTER 6

EXTENSIONS

The computer has proven to be a useful tool for both the structural designer and the structural engineer. This realization has inspired the development of numerous structural design computer programs over the last two decades. The complexity of these programs range from the simple iterative analysis-design procedures, similar to the one used in this investigation, to the sophisticated non-linear programming algorithms previously discussed in chapter two. Many of these programs provide the designer with information similar to the data produced by the program used in this investigation. However, some program packages used in industry incorporate a feature which enables the computer to select specific sections from a directory and assemble a design based on these selected sections. The computer then verifies the design by performing an analysis on the structure to insure compliance with the steel code. A procedure such as this reduces the designer's work to a minimum by eliminating the time it would take for him to search through the steel tables for suitable sections and to check the final member selections for code compliance.

The addition of such a feature to the existing program would significantly increase the capabilities of the program with respect to the overall design process in several ways. First, a feature such as this would reduce the designer's workload to such an extent that he would only have to set up the design problem in terms of

loads and structural geometry. The computer would then handle the rest of the design process. Second, the computer's selection of specific sections for a solution provides an important transition in the design process since it produces a discrete solution from a continuous one which is far more practical than a continuous solution alone. Finally, a design comprised of specific sections would allow the computer to conduct a more detailed final analysis for code compliance since section properties such as flange width and web thickness would be known for each member.

Figure 6.1 illustrates a simple procedure that would enable the existing program to perform the functions described above. In the first phase of the procedure, an allowable range of values of S and I are calculated from the final design data for each member. These parameter ranges are used as the governing criterion for member selection. A suggested method for determining the allowable parameter values could be based on a percentage variation over the entire range of each parameter and apply this to the final design values. For illustrative purposes, a variation of one percent is taken over the total range of S and I . This would approximately provide a $\pm 11.in^3$ variation in S and a $\pm 200.in^4$ variation in I . In the next step, available sections whose S and I values fall within the designated parameter range are selected from a directory of available sections and compiled into a list for each member. A discrete solution is then obtained by choosing a section for each member from the qualifying sections list. An analysis of this trial design is made to assure compliance with the code. In the interest of weight, a

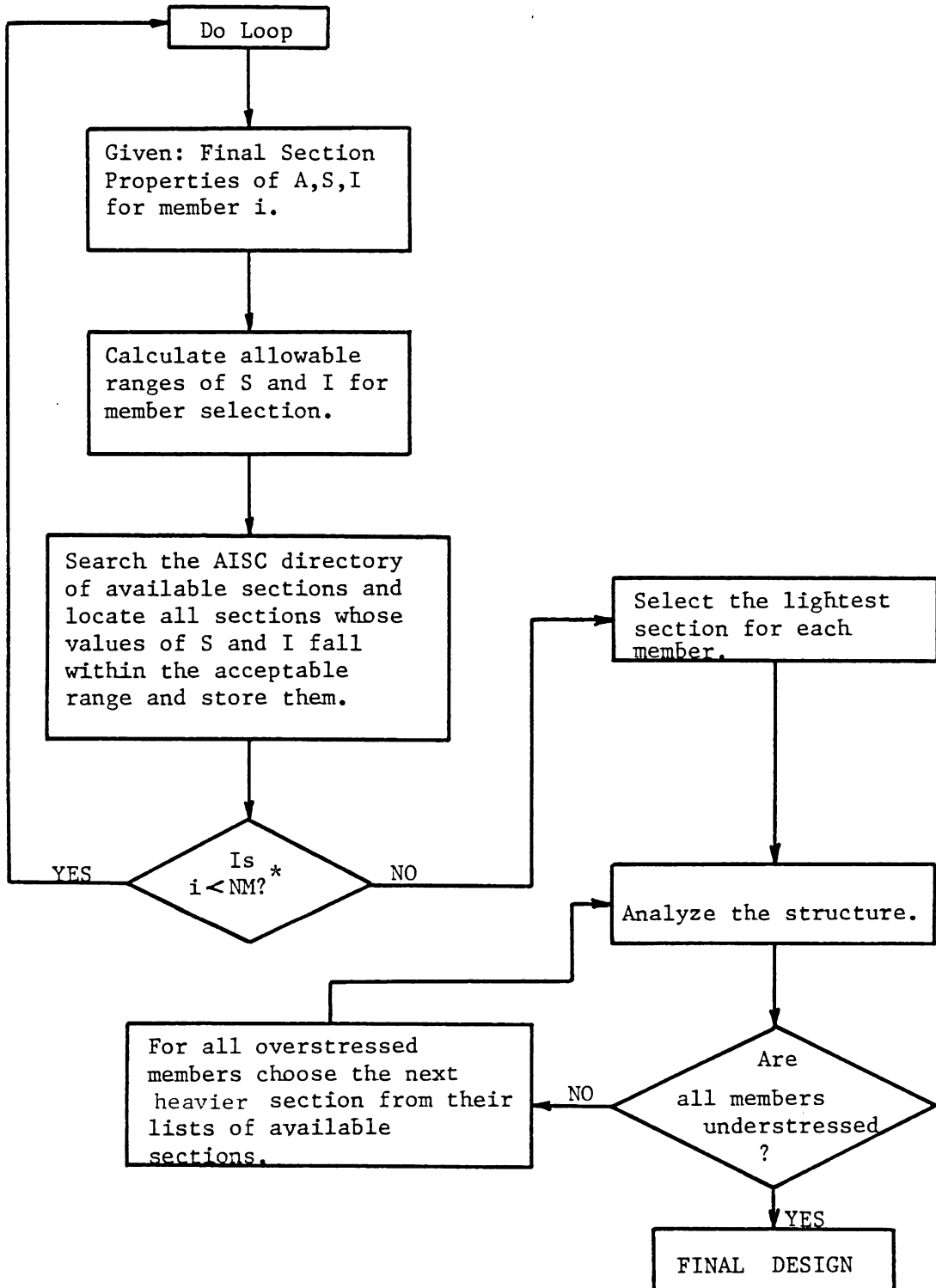


Figure 6.1 Member Selection Routine Flowchart

design composed of the lightest possible sections is checked first. If any member is found to be overstressed, the next heaviest section is taken from the section list and the structure is reanalyzed. This procedure is repeated until a final design is achieved. As an illustrative example, Table 6.1 shows the allowable parameter ranges computed from the final design data of design example 5.3. The sections that have properties within the allowable ranges for each member are also shown in Table 6.1. Table 6.2 shows a trial design, compiled from the sections of Table 6.1, and the analysis results. Not only are all of the members close to the fully stressed state, with none exceeding the allowable stress, but the total structural weight is within 2 percent of the calculated total weight in Table 5.5.

While the aforementioned procedure has only been tested on one particular design example and further studies need to be done for more general design problems before a definite conclusion can be made, indications are that this procedure can enable the present design program to perform almost all of the structural design processes. This would make the designer simply responsible for determining the geometry, boundary conditions, and applied loads.

TABLE 6.1
 Selected Sections
 Based on Computer Results
 From Design Example 5.3

Member Number		1	2	3	4	5	6
S Range	min.	247	85	80	85	205	247
	max.	269	107	102	107	227	269
I Range	min.	3472	817	748	817	2723	3472
	max.	3872	1217	1148	1217	3123	3872
Available AISC Sections		W27x102	W21x50 W18x55	W21x44 W18x50 W16x57 W21x50 W18x55	W21x50 W18x55	W27x84	W27x102

TABLE 6.2
 Selected Sections
 Based on Computer Results
 From Design Example 5.3
 and TABLE 6.1

Member Number	S (in. ³)	A (in. ²)	I (in. ⁴)	% F.S.*	WF Designation	Weight (lb.)
1	267.0	30.0	3620.0	95.6	W27x102	1530
2	94.5	14.7	984.0	99.2	W21x50	750
3	94.5	14.7	984.0	94.5	W21x50	1000
4	94.5	14.7	984.0	99.2	W21x50	750
5	213.0	24.8	2850.0	98.6	W27x84	1680
6	267.0	30.0	3620.0	95.6	W27x102	1530
Total Weight :						7240

* Percent Fully Stressed

REFERENCES

1. Arora, J. S., Haug, Jr., E. J., and Rim, K., "Optimal Design of Plane Frames", Journal of the Structural Division, ASCE, Vol. 101, No. ST10, Proc. Paper 11633, October, 1975, pp. 2063-2078.
2. Brown, D. M., and Ang, A. H.S., "Structural Optimization by Nonlinear Programming", Journal of the Structural Division, ASCE, Vol. 92, No. ST6, Proc. Paper 5016, December, 1966, pp. 319-340.
3. Clarkson, J., The Elastic Analysis of Flat Grillages, Cambridge University Press, 1965.
4. Cornell, A. C., Reinschmidt, K. F., and Brothie, J. F., "A Method for the Optimum Design of Structures", Proc., International Symposium on the Use of Computers in Structural Engineering, Newcastle, July, 1966.
5. Dayaratnam, P., and Patnaik, S., "Feasibility of Fully Stressed Design", American Institute of Aeronautics and Astronautics Journal, Vol. 7, April, 1969, pp. 773-774.
6. Disque, R. O., "Inelastic K-Factors for Column Design", AISC Engineering Journal, 2nd Quarter, 1973, pp. 33-35.
7. Forsythe, G. E., "Generation and Use of Orthogonal Polynomials for Fitting Data with a Digital Computer", J. Soc. Indust. Appl. Math., Vol. 5, No. 2, 1957, pp. 74-88.
8. Furlong, R. W., "Column Slenderness and Charts for Design", ACI Journal, Proc. Vol. 68, No. 1, January, 1971, pp. 9-18.
9. Galambos, T. V., Structural Members and Frames, Prentice-Hall, Inc., Englewood, New Jersey, 1968.
10. Gallagher, R. H., "Introduction", Optimum Structural Design, Theory and Applications, R. H. Gallagher and O. C. Zienkiewicz, Editors, John Wiley and Sons, Inc., London, England, 1973.
11. Gallagher, R. H., "Fully Stressed Design", Optimum Structural Design, Theory and Applications, R. H. Gallagher and O. C. Zienkiewicz, Editors, John Wiley and Sons, Inc., London, England, 1973.

12. Gellatly, R. A., and Gallagher, R. H., "A Procedure for Automated Minimum Weight Structural Design, Part I - Theoretical Basis; Part II - Applications", Aeronautical Quarterly, Vol. 17, 1966, pp. 216-230 and pp. 332-342.
13. Ghista, D. N., "Fully Stressed Design for Alternative Loads", Journal of the Structural Division, ASCE, Vol. 92, ST5, Paper 4952, October, 1966, pp. 237-260.
14. Johnson, J. R., et al., "Optimum Structural Design", presented at the 25th meeting of the Structures and Materials Advisory Group, Ottawa, September 25-29, 1967.
15. Johnston, B. G., Editor, Structural Stability Research Council Guide to Stability Design Criteria for Metal Structures, Third Edition, John Wiley and Sons, Inc., 1976.
16. Julian, O. G., and Lawrence, L. S., "Notes on J and L Nomograms for Determination of Effective Lengths", Unpublished, 1959.
17. Kavlie, D., Kowalik, J., and Moe, J., "Structural Optimization by Means of Nonlinear Programming", Proc., International Symposium on the Use of Computers in Structural Engineering, Newcastle, July, 1966.
18. Kavlie, D., and Moe, J., "Application of Nonlinear Programming to Optimum Grillage Design with Non-Convex Sets of Variables", International Journal for Numerical Methods in Engineering, Vol. 1, 1969, pp. 351-378.
19. Kavlie, D., and Moe, J., "Automated Design of Framed Structures", Journal of the Structural Division, ASCE, Vol. 97, No. ST1, Proc. Paper 7823, January, 1971, pp. 33-62.
20. Kicher, T. P., "Optimum Design-Minimum Weight Versus Fully Stressed", Journal of the Structural Division, ASCE, Vol. 92, No. ST6, December, 1966, pp. 265-279.
21. Kirsch, U., and Moses, F., "Formulation of Optimal Design in the Behavior Variable Space", Journal of Structural Mechanics, Vol. 4, No. 4, 1976, pp. 437-452.

22. Kuhn, H. W., and Tucker, A. W., "Nonlinear Programming", in Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1951, pp. 481-490.
23. Manual of Steel Construction, 8th Edition, American Institute of Steel Construction, Inc., New York, New York, 1978.
24. Marcal, P. V., and Gellatly, R. A., "Application of the Created Response Surface Technique to Structural Optimization", 2nd Conference on Matrix Methods in Structural Methanics, WFAB, October, 1968.
25. Melosh, R. J., and Luik, R., "Approximate Multiple Configuration Analysis and Allocation for Least Weight Structural Design", Wright-Patterson AFB, AFFDL-TR-68-156, Ohio, April, 1967.
26. Michell, A. G. M., "The Limits of Economy of Materials in Framed Structures", Phil. Mag. (series 6), Vol. 8, 1904, pp. 589-597.
27. Moses, F., "Optimum Structural Design Using Linear Programming", Journal of the Structural Division, ASCE, Vol. 90, No. ST6, Proc. Paper 4163, December, 1964, pp. 89-104.
28. Moses, F., and Onoda, S., "Minimum Weight Design of Structures with Applications to Elastic Grillages", International Journal for Numerical Methods in Engineering, Vol. 1, 1969, pp. 311-331.
29. Nakamura, Y., "Optimum Design of Framed Structures Using Linear Programming", Thesis presented to the Department of Civil Engineering at the Massachusetts Institute of Technology at Cambridge, Mass., in 1966, in partial fulfillment of the requirements for the degree of Master of Science.
30. Palmer, A. C., "Optimal Design by Dynamic Programming", Journal of the Structural Division, ASCE, Vol. 94, No. ST8, Proc. Paper 6057, August, 1968, pp. 1887-1906.
31. Prager, W., and Sheu, C., "Recent Developments in Optimal Structural Design", Applied Mechanics Review, October, 1968.

32. Razani, R., "The Behavior of the Fully Stressed Design of Structures and Its Relation to Minimum Weight Design", American Institute of Aeronautics and Astronautics Journal, Vol. 3, No. 12, December, 1965, pp. 2262-2268.
33. Reinschmidt, K. F., and Norabhoompipat, T., "Structural Optimization by Equilibrium Linear Programming", Journal of the Structural Division, ASCE, Vol. 101, No. ST4, Proc. Paper 11219, April, 1975, pp. 921-937.
34. Romstad, K. M., and Wang, C. K., "Optimum Design of Framed Structures", Journal of the Structural Division, ASCE, Vol. 94, No. ST12, Proc. Paper 6273, December, 1968, pp. 2817-2845.
35. Schmit, L., "Structural Synthesis: 1959-1969: A Decade of Progress", Recent Advances in Matrix Methods of Structural Analysis and Design, R. H. Gallagher, Editor, University of Alabama Press, 1971.
36. Schmit, L., "Structural Engineering Applications of Mathematical Programming Techniques", Symposium on Structural Optimization, Agard, Conference Proceedings No. 36, R. Gellatly, Editor, Advisory Group for Aerospace Research and Development, NATO, October, 1970.
37. Schmit, L., "Automated Design", International Science and Technology, June, 1966.
38. Schmit, L. A., and Fox, R. L., "An Integrated Approach to Structural Synthesis and Analysis", American Institute of Aeronautics and Astronautics Journal, Vol. 3, No. 6, June, 1965, pp. 1104-1112.
39. Twisdale, L. A., and Khachaturian, N., "Multistage Optimization of Structures", Journal of the Structural Division, ASCE, Vol. 101, No. ST5, Proc. Paper 11297, May, 1975, pp. 1005-1020.
40. Venkayya, V., et al., "Energy Distribution in an Optimum Structural Design", Wright-Patterson AFB, AFFDL-TR-68-156, Ohio, March, 1969.
41. Wasuitynski, Z., and Brandt, A., "The Present State of Knowledge in the Field of Optimum Design of Structures", Applied Mechanics Review, 16, 1963.

42. Young, J. W., and Christiansen, H. N., "Synthesis of a Space Truss Based on Dynamic Criteria", Journal of the Structural Division, ASCE, Vol. 92, No. ST6, December, 1966, pp. 425-442.
43. Yura, J. A., "The Effective Length of Columns in Unbraced Frames", AISC Engineering Journal, Vol. 8, No. 2, April, 1971, pp. 37-42.
44. Class notes, CE 4001, Matrix Structural Analysis, Fall Quarter, 1978.
45. Class notes, CE 4002, Matrix Structural Analysis, Winter Quarter, 1979.
46. Class notes, CE 5640, Computer-Aided Structural Design, Spring Quarter, 1979.

APPENDIX A

COMPUTER PROGRAM LISTING

```

COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
*F(40,3),MCOE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
*4,4)
COMMON/QLD/ QL(60,4)
COMMON/FOR/ FBR(40,6,4)
COMMON/CFOR/ CLOAD(40,21,4),NNP
COMMON/FSD/ ISTOP
COMMON/RES/STGMAX(40),SLR(40),FAX(40,2,4),MAXL(40)
COMMON/TORS/ G(40),XI1(40)
INTEGER GRILL
DIMENSION QLL(60),SS(60,60)
ISTOP=0
CALL DATA
CALL MEMBER
DO 1 KOUNT=1,30
CALL MACT
IF (GRILL.NE.0) CALL GSTIFF(SS)
IF (GRILL.EQ.0) CALL STIFF(SS)
DO 10 I=1,NL
DO 11 J=1,NDOF
11 QLL(J)=QL(J,I)
CALL SOLVE (SS,QLL,NDOF,MD,I)
DO 12 J=1,NDOF
12 QL(J,I)=QLL(J)
10 CONTINUE
CALL FORCE
CALL CFORCE
CALL MRES
DO 4 I=1,NM
4 WRITE(6,1001) XI(I),A(I),SX(I)

```

```

1001 FORMAT(50X,3E15.7)
      IF(ISTOP.EQ.0) GO TO 30
      1 CONTINUE
      30 STOP
      END
      SUBROUTINE DATA
      COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
      *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
      COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
      COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
      *4,4)
      COMMON/CFOR/ CLOAD(40,21,4),NNP
      COMMON/FSD/ ISTOP
      COMMON/TORS/ G(40),XI1(40)
      DATA NDCF/O/,J/O/,K/O/,NNN/O/,NNL/O/
      REAL L
      INTEGER GRILL
      READ(5,100) NM,NJ,MCOMP,NA,NL,NNP,GRILL
      100 FORMAT(8I5)
      WRITE(6,102) NM,NJ
      WRITE(6,118)NL
      118 FORMAT(10X,'NUMBER OF LOADING CONDITIONS=',I5/)
      102 FORMAT(1H1,10X,'NUMBER OF MEMBERS=',I5,5X,'NUMBER OF JOINTS=',I5/)
      WRITE(6,1000)
      1000 FORMAT(//10X,'JOINT COORDINATES',//,16X,'JOINT NUMBER',13X,'X-COOR
      *DINATE',13X,'Y-COORDINATE'//)
      DO 5 I=1,NJ
      READ(5,103) X(I),Y(I)
      5 WRITE(6,105) I,X(I),Y(I)
      103 FORMAT(2F10.3)
      105 FORMAT(16X,I5,1/X,F10.3,17X,F10.3)
      DO 10 I=1,NM

```

```

      IF(MCOMP.NE.0) GO TO 4
      READ(5,104) SX(I),E(I),G(I),XI(I)
      GO TO 10
4     READ(5,104) SX(I),A(I),XI(I),E(I)
10    CONTINUE
104   FORMAT(4F10.3)
      WRITE(6,1002)
1002  FORMAT(/,10X,'MEMBER INCIDENCE MATRIX',/,15X,'MEMBER NO.',5X,'A-
      *END',5X,'B-END'/)
      DO 15 I=1,NM
      READ(5,107) MINC(I,1),MINC(I,2)
15    WRITE(6,108) I,MINC(I,1),MINC(I,2)
107   FORMAT(2I5)
108   FORMAT(16X,I5,5X,I5,5X,I5)
      DO 20 I=1,NJ
      DO 20 J=1,3
20    JCODE(I,J)=1
12    READ(5,109) JNUM,JDIR
      IF(JNUM.EQ.0) GO TO 21
      JCODE(JNUM,JDIR)=0
      GO TO 12
21    CONTINUE
109   FORMAT(2I5)
      DO 25 I=1,NJ
      DO 25 J=1,3
      IF(JCODE(I,J).EQ.0) GO TO 25
      NDOF=NDOF+1
      JCODE(I,J)=NDOF
25    CONTINUE
      WRITE(6,1003)
1003  FORMAT(/,10X,'JOINT DISPLACEMENT MATRIX',/,17X,'JOINT NO.',6X,'
      *X-DISPL',3X,'Y-DISPL',3X,'Z-DISPL'/)

```

```

      DO 30 I=1,NJ
30  WRITE(6,110) I,JCODE(I,1),JCODE(I,2),JCODE(I,3)
110  FORMAT(16X,I5,7X,I5,5X,I5,5X,I5)
      DO 35 I=1,NM
      J=MINC(I,1)
      K=MINC(I,2)
      DO 35 LI=1,3
      MCODE(I,LI)=JCODE(J,LI)
35  MCODE(I,LI+3)=JCODE(K,LI)
      WRITE(6,1004)
1004  FORMAT(///,10X,'MEMBER CODE MATRIX',//,10X,'MEMBER NO',16X,'A-END'
      *,23X,'B-END'//)
      DO 40 I=1,NM
40  WRITE(6,111) I,(MCODE(I,J),J=1,6)
111  FORMAT(11X,I5,10X,I5,2X,I5,2X,I5,10X,I5,2X,I5,2X,I5)
      DO 45 I=1,NDCF
      DO 45 J=1,NL
45  Q(I,J)=0.
      DO 50 I=1,NL
      DO 51 J=1,NDCF
      READ(5,112) JNUM,JDIR,FORCE
112  FORMAT(2I5,F10.3)
      IF(JNUM.EQ.0) GO TO 50
      K=JCODE(JNUM,JDIR)
51  Q(K,I)=FORCE
50  CONTINUE
      WRITE(6,1005)
1005  FORMAT(//,10X,'APPLIED JOINT FORCE MATRIX'//)
      GO TO (71,72,73,74),NL
71  WRITE(6,76)
      GO TO 52
72  WRITE(6,77)

```

```

      GO TO 52
73  WRITE(6,78)
      GO TO 52
74  WRITE(6,79)
76  FORMAT(10X,'LOAD NO. 1'/)
77  FORMAT(10X,'LOAD NO. 1',5X,'LOAD NO. 2'/)
78  FORMAT(10X,'LOAD NO. 1',5X,'LOAD NO. 2', 5X,'LOAD NO. 3'/)
79  FORMAT(10X,'LOAD NO. 1',5X,'LOAD NO. 2', 5X,'LOAD NO. 3',5X,'LOAD
      *NO. 4'/)
52  DO 55 I=1,NDGF
55  WRITE(6,113)(Q(I,J),J=1,NL)
113  FORMAT(10X,4(F10.3,5X))
      READ(5,114)(MA(J),J=1,NL)
114  FORMAT(16I5)
      DO 65 J=1,NL
         >NNL=MA(J)
          IF(NNL.EQ.0) GO TO 65
          WRITE(6,117)J
117  FORMAT(//10X,'LOADING CONDITION NO.',I5)
          WRITE(6,1006)
1006  FORMAT(/,10X,'MEMBER ACTIONS'//,14X,'MEMBER',6X,'LOAD',6X,'LOAD',1
          *1X,'LOAD',9X,'A',14X,'B')
          WRITE(6,1007)
1007  FORMAT(17X,'NO.',6X,'TYPE',15X,'A-END',10X,'B-END',7X,'DISTANCE',7
          *X,'DISTANCE'/)
          DO 60 I=1>NNL
              READ(5,116)IACT(I,1,J),IACT(I,2,J),DACT(I,1,J),DACT(I,2,J),DACT(I,
              *3,J),DACT(I,4,J)
          60  WRITE(6,115)IACT(I,1,J),IACT(I,2,J),DACT(I,1,J),DACT(I,2,J),DACT(I
              *,3,J),DACT(I,4,J)
116  FORMAT(2I5,4F10.3)
115  FORMAT(15X,I5,5X,I5,10X,4(F10.3,5X))

```

```

65 CONTINUE
  WRITE(6,119) NDOF
119 FORMAT(/,10X,'NUMBER OF DEGREES OF FREEDOM=',I5)
  RETURN
  END
  SUBROUTINE MEMBER
  COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCDD
  *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
  COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
  COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
  *4,4)
  COMMON/TCRS/ G(40),XI1(40)
  REAL L
  INTEGER GRILL
  DATA IK/0/,IJ/0/,XL1/0./,XL2/0./,NMIN/0/,N/0/,MB/0/,MXB/0/,MD/0/
  DO 10 J=1,NM
    NO=0
    DO 15 II=1,6
      NO=MCODE(J,II)
      IF(NO.EQ.0) GO TO 15
    GO TO 16
15 CONTINUE
16 NMIN=NO
    NMAX=0
    DO 20 I=II,6
      N=MCODE(J,I)
      IF(N.EQ.0) GO TO 20
      IF(N.GT.NMAX) GO TO 22
    GO TO 20
22 NMAX=N
20 CONTINUE
    MB=NMAX-NMIN

```

```

      IF (MB.GT.MXB) MXB=MB
10  CONTINUE
      MD=MXB+1
      WRITE(6,121) MD
121  FORMAT(10X,'MAXIMUM HALF BAND WIDTH=',I5//)
      WRITE(6,1010)
1010 FORMAT(//10X,'GEOMETRIC MEMBER PROPERTIES',//,10X,'MEMBER NO.',9X,
      *'LENGTH',9X,'CCSINE',11X,'SINE'//)
      DO 5 I=1,NM
      IK=MINC(I,1)
      IJ=MINC(I,2)
      XL1=X(IJ)-X(IK)
      XL2=Y(IJ)-Y(IK)
      L(I)=SQRT(XL1**2+XL2**2)
      COSA(I)=XL1/L(I)
      SINA(I)=XL2/L(I)
5  WRITE(6,120) I,L(I),COSA(I),SINA(I)
120  FORMAT(15X,I5,3E15.7)
      WRITE(6,1001)
1001 FORMAT(///,10X,'ELEMENT PROPERTIES',//,10X,'MEMBER NO.',16X,'AREA'
      *,3X,'MOMENT OF INERTIA',5X,'ELASTIC MODULUS',5X,'SECTION MODULUS'//
      *)
      DO 7 I=1,NM
      IF (MCOMP.NE.0) GO TO 12
      IF (GRILL.NE.0) CALL GRDSTR(I)
      IF (GRILL.NE.0) GO TO 12
      IF (SINA(I).NE.0.) GO TO 9
      A(I)=3.62415C+.119637*SX(I)-.00009708825*SX(I)**2+.00000005341595*
      *SX(I)**3
      IF (SX(I).LT.50.) GO TO 3
      XI(I)=-191.3096+10.90819*SX(I)+.01840775*SX(I)**2-.00001076241*SX(
      *I)**3

```

```

      GO TO 7
3  XI(I)=7.98*SX(I)
      GO TO 7
9  IF(SX(I).LT.50.) GO TO 2
      XI(I)=-91.79416+6.954839*SX(I)+.0034126*SX(I)**2
      GO TO 1
2  XI(I)=-11.88349+4.132184*SX(I)+.009396914*SX(I)**2
1  IF(SX(I).GT.50.) GO TO 8
      A(I)=2.346176+.2443544*SX(I)-.00003459238*SX(I)**2
      GO TO 7
8  A(I)=.61575+.1871112*SX(I)-.00001518453*SX(I)**2
12 IF(GRILL.EQ.0) GO TO 11
      WRITE(6,106) I,A(I),XI(I),E(I),SX(I),XI1(I)
      GO TO 7
11 WRITE(6,106) I,A(I),XI(I),E(I),SX(I)
7  CONTINUE
106 FORMAT(15X,I5,6(5X,E15.7))
      DO 6 I=1,NM
          ALPHA(I)=E(I)*XI(I)/L(I)**3
          IF(GRILL.NE.0) GO TO 4
          BETA(I)=A(I)*L(I)**2/XI(I)
          GO TO 6
4  BETA(I)=XI1(I)*G(I)*L(I)**2/(XI(I)*E(I))
6  CONTINUE
      RETURN
      END
      SUBROUTINE MACT
      COMMON/DAT IN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCND
      *E(40,3),MCCDE(40,6),MA(40),Q(60,4),NM,NL,NJ,GRILL,MCOMP
      COMMON/MEM/L(40),COXA(40),SINA(40),NDOF,MD
      COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
      *4,4)

```

```

      REAL L
      DATA NN/0/,N/0/,IM/0/,ID/0/,WA/0./,WB/0./,R/0./,B/0./,C/0./,D/0./,
      *S/0./,F/0./,G/0./,H/0./,YB1/0./,YB2/0./,THTA1/0./,THTA2/0./,THETA/
      *0./,YB/0./,F5/0./,F6/0./,D40/0./,F4/0./,MAA/0/,JJ/0/,KK/0/,LN/0/,I
      *J/0/,IK/0/,IL/0/
      DO 5 K=1,NL
      DO 5 I=1,NM
      DO 5 J=1,6
5 FMA(I,J,K)=0.
      DO 10 I=1,NL
      IF(MA(I).EQ.0) GO TO 10
      WRITE(6,2000)I
2000 FORMAT(/10X,'MEMBER REACTIONS FOR LOADING CONDITION',I5/)
     >NNL=MA(I)
      WRITE(6,2001)
2001 FORMAT(10X,'MEMBER NO.',6X,'F1',13X,'F2',13X,'F3',13X,'F4',13X,'F5
      *',13X,'F6'/)
      DO 15 J=1,NNL
      IM=IACT(J,1,I)
      ID=IACT(J,2,I)
      WA=DACT(J,1,I)
      WB=DACT(J,2,I)
      R=DACT(J,3,I)
      B=DACT(J,4,I)
      C=L(IM)
      D=A(IM)
      S=E(IM)
      F=ALPHA(IM)
      G=BETA(IM)
      H=XI(IM)
      GO TO (16,17,18,19),ID
16 FMA(IM,2,I)=FMA(IM,2,I)-WA*(C-R)**2*(3.*R+(C-R))/C**3

```

```

      FMA(IM,3,I)=FMA(IM,3,I)-WA*R*(C-R)**2/C**2
      FMA(IM,5,I)=FMA(IM,5,I)-WA*R**2/C**3*(R+3.*(C-R))
      FMA(IM,6,I)=FMA(IM,6,I)+WA*R**2*(C-R)/C**2
      GO TO 14
17  FMA(IM,1,I)=FMA(IM,1,I)-WA*(C-R)/C
      FMA(IM,4,I)=FMA(IM,4,I)-WA*R/C
      GO TO 14
18  YB1=WB*C/(120.*F)*(B/C-R/C)*((B/C)**2*(15.-4.*B/C)+B/C*(R/C)*(10.-
      *3.*B/C)+(R/C)**2*(5.-2.*B/C)-(R/C)**3)
      YB2=WA*C/(120.*F)*(B/C-R/C)*((15.-B/C)*((B/C)**2+2.*B/C*R/C+3.*(R/C
      **2))-4.*(R/C)**3)
      THTA1=WB/(24.*F)*(B/C-R/C)*(3.*(B/C)**2+2.*B*R/C**2+(R/C)**2)
      THTA2=WA/(24.*F)*(B/C-R/C)*((B/C)**2+2.*B*R/C**2+3.*(R/C)**2)
      THETA=THTA1+THTA2
      YB=YB1+YB2
      F5=F*6.*(C*THETA-2.*YB)
      F6=F*2.*C*(3.*YB-2.*C*THETA)
      FMA(IM,5,I)=FMA(IM,5,I)+F5
      FMA(IM,6,I)=FMA(IM,6,I)+F6
      FMA(IM,2,I)=FMA(IM,2,I)-F5-(WA+WB)/2.*(B-R)
      FMA(IM,3,I)=FMA(IM,3,I)-F6-C*F5-(WA+WB)/2.*(B-R)*(R+(B-R)*(2.*WB+W
      *A)/(3.*(WA+WB)))
      GO TO 14
19  D40=WA/(6.*D*S)*(B-R)*(2.*R+B)
      D40=D40+WB/(6.*D*S)*(B-R)*(R+2.*B)
      F4=-F*G*D40
      FMA(IM,4,I)=FMA(IM,4,I)+F4
      FMA(IM,1,I)=FMA(IM,1,I)-F4-.5*(WA+WB)*(B-R)
14  CONTINUE
15  CONTINUE
10  CONTINUE
      CALL QLGAD

```

```

RETURN
END
SUBROUTINE STIFF(SS)
COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
*E(40,3),MCCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
*4,4)
DIMENSION INDEX(6,6),H(7),SS(NDOF,MD)
REAL L
DATA INDEX/1,2,4,-1,-2,4,2,3,5,-2,-3,5,4,5,6,-4,-5,7,-1,-2,-4,1,2,
*-4,-2,-3,-5,2,3,-5,4,5,7,-4,-5,6/,J/0/,K/0/,KB/0/,LN/0/
DO 1 I=1,NDOF
DO 1 J=1,MD
1 SS(I,J)=0.
DO 10 I=1,NM
H(1)=ALPHA(I)*(BETA(I)*COSA(I)**2+12.*SINA(I)**2)
H(2)=ALPHA(I)*COSA(I)*SINA(I)*(BETA(I)-12.)
H(3)=ALPHA(I)*(BETA(I)*SINA(I)**2+12.*COSA(I)**2)
H(4)=-ALPHA(I)*6.*L(I)*SINA(I)
H(5)=ALPHA(I)*6.*L(I)*COSA(I)
H(6)=ALPHA(I)*4.*L(I)**2
H(7)=ALPHA(I)*2.*L(I)**2
DO 15 JM=1,6
J=MCCODE(I,JM)
IF(J.EQ.0) GO TO 15
DO 20 KM=JM,6
K=MCCODE(I,KM)
IF(K.EQ.0) GO TO 20
KB=K-J+1
LN=INDEX(JM,KM)
IF(LN.GT.0) GO TO 21

```

```

        SM=-H(-LN)
        GO TO 22
21  SM=H(LN)
22  SS(J,KB)=SS(J,KB)+SM
20  CONTINUE
15  CONTINUE
10  CONTINUE
    RETURN
    END
    SUBROUTINE QLOAD
    COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
    *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
    COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
    COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
    *4,4)
    COMMON/QLD/ QL(60,4)
    DIMENSION QHT(60)
    DATA JJ/0/,KK/0/,LN/0/,IJ/0/,IK/0/,IL/0/
    DO 10 I=1,NL
    DO 20 II=1,NDOF
    QL(II,I)=0.
20  QHT(II)=0.
    DO 25 II=1,NM
    JJ=MINC(II,1)
    KK=MINC(II,2)
    M=0
    LN=KK-JJ
    DO 30 IN=JJ,KK,LN
    IJ=JCODE(IN,1)
    IK=JCODE(IN,2)
    IL=JCODE(IN,3)
    IF(GRILL.NE.0) GO TO 39

```

```

      IF(IJ.EQ.0) GO TO 31
      QHT(IJ)=QHT(IJ)+COSA(II)*FMA(II,M+1,I)-SINA(II)*FMA(II,M+2,I)
31  IF(IK.EQ.0) GO TO 32
      QHT(IK)=QHT(IK)+SINA(II)*FMA(II,M+1,I)+COSA(II)*FMA(II,M+2,I)
32  IF(IL.EQ.0) GO TO 33
      QHT(IL)=QHT(IL)+FMA(II,M+3,I)
      GO TO 33
39  IF(IJ.EQ.0) GO TO 36
      QHT(IJ)=QHT(IJ)+COSA(II)*FMA(II,M+1,I)-SINA(II)*FMA(II,M+3,I)
36  IF(IK.EQ.0) GO TO 37
      QHT(IK)=QHT(IK)+FMA(II,M+2,I)
37  IF(IL.EQ.0) GO TO 33
      QHT(IL)=QHT(IL)+SINA(II)*FMA(II,M+1,I)+COSA(II)*FMA(II,M+3,I)
33  M=3
30  CONTINUE
25  CONTINUE
      DO 35 LL=1,NDOF
      QL(LL,I)=Q(LL,I)-QHT(LL)
35  CONTINUE
10  CONTINUE
      RETURN
      END
      SUBROUTINE SOLVE (SS,P,NDOF,MD,II)
      DIMENSION SS(NDOF,MD),P(NDOF)
      IF(II.GT.1) GO TO 5
      DO 10 N=1,NDOF
      DO 15 L=2,MD
      IF(SS(N,L).EQ.0.) GO TO 15
      I=N+L-1
      C=SS(N,L)/SS(N,1)
      J=0
      DO 20 K=L,MD

```

```

      J=J+1
20  SS(I,J)=SS(I,J)-C*SS(N,K)
      SS(N,L)=C
15  CONTINUE
10  CONTINUE
      5 DO 25 N=1,NDGF
        DO 30 L=2,MD
          IF(SS(N,L).EQ.0.) GO TO 30
          I=N+L-1
          P(I)=P(I)-SS(N,L)*P(N)
30  CONTINUE
      25 P(N)=P(N)/SS(N,1)
        DO 35 M=2,NDGF
          N=NDGF+1-M
          DO 40 L=2,MD
            IF(SS(N,L).EQ.0.) GO TO 40
            K=N+L-1
            P(N)=P(N)-SS(N,L)*P(K)
40  CONTINUE
      35 CONTINUE
      RETURN
      END
      SUBROUTINE FORCE
      COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCDD
      *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
      COMMON/MEM/L(40),COSA(40),SINA(40),NDUF,MD
      COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
      *4,4)
      COMMON/QLD/ QL(60,4)
      COMMON/FOR/ FBR(40,6,4)
      DATA C/O./,S/O./,Z/O./,B/O./,RL/O./,J/O/,DL1/O./,DL2/O./,DL3/O./,D
      *L4/O./,DL5/O./,DL6/O./

```

```

      REAL L
      INTEGER GRILL
      DIMENSION D(60)
      DO 100 K=1,NL
      DO 2 I=1,NM
      DO 2 J=1,6
2 FBR(I,J,K)=0.
      DO 10 I=1,NM
      C=COSA(I)
      S=SINA(I)
      Z=ALPHA(I)
      B=BETA(I)
      RL=L(I)
      DO 15 JM=1,6
      J=MCODE(I,JM)
      IF(J.EQ.0) GO TO 16
      D(JM)=QL(J,K)
      GO TO 15
16 D(JM)=0.
15 CONTINUE
      IF(GRILL.EQ.0) GO TO 17
      DL1=C*D(1)+S*D(3)
      DL2=D(2)
      DL3=-S*D(1)+C*D(3)
      DL4=C*D(4)+S*D(6)
      DL5=D(5)
      DL6=-S*D(4)+C*D(6)
      GO TO 18
17 DL1=C*D(1)+S*D(2)
      DL2=-S*D(1)+C*D(2)
      DL3=D(3)
      DL4=C*D(4)+S*D(5)

```

```

      DL5=-S*D(4)+C*D(5)
      DL6=D(6)
18  FBR(I,1,K)=Z*B*(DL1-DL4)+FMA(I,1,K)
      FBR(I,2,K)=6.*Z*(2.*DL2+RL*DL3-2.*DL5+RL*DL6)+FMA(I,2,K)
      FBR(I,3,K)=2.*Z*(3.*RL*DL2+2.*RL**2*DL3-3.*RL*DL5+RL**2*DL6)+FMA(I
      *,3,K)
      FBR(I,4,K)=-Z*B*(DL1-DL4)+FMA(I,4,K)
      FBR(I,5,K)=-6.*Z*(2.*DL2+RL*DL3-2.*DL5+RL*DL6)+FMA(I,5,K)
      FBR(I,6,K)=Z*(6.*RL*DL2+2.*RL**2*DL3-6.*RL*DL5+4.*RL**2*DL6)+FMA(I
      *,6,K)
10  CONTINUE
100 CONTINUE
      RETURN
      END
      SUBROUTINE CFORCE
      COMMON/DATN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
      *E(40,3),MCCDE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
      COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
      COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
      *4,4)
      COMMON/CFOR/ CLOAD(40,21,4),NNP
      REAL L
      DATA>NNL/0/,NN/0/
      DO 1 I=1,NM
      DO 1 J=1,NNP
      DO 1 K=1,NL
1  CLOAD(I,J,K)=0.
      DO 100 K=1,NL
      IF(MA(K).EQ.0) GO TO 100
      NN=MA(K)
      DO 50 I=1,NN
      M=IACT(I,1,K)

```

```

      ID=I ACT(I,2,K)
      PA=D ACT(I,1,K)
      XA=D ACT(I,3,K)
      XL=L(M)
      H=XL/(NNP-1)
      GO TO (101,50,102,50),ID
101  S=XA/H
      IQ=S
      R=(S-IQ)*H
      CLOAD(M,IQ+1,K)=CLOAD(M,IQ+1,K)+PA*(1.-R/H)
      CLOAD(M,IQ+2,K)=CLOAD(M,IQ+2,K)+PA*R/H
      GO TO 50
102  XB=D ACT(I,4,K)
      PB=D ACT(I,2,K)
      SLOP=(PB-PA)/(XB-XA)
      SA=XA/H
      SB=XB/H
      IA=SA
      IB=SB
      RA=(SA-IA)*H
      RB=(SB-IB)*H
      P1=PB-SLOP*RB
      P2=PA+SLOP*(H-RA)
      IF(XL-XB) 32,32,33
32  RB=(IB-(IB-1))*H
      IB=IB-1
33  CLOAD(M,IA+1,K)=CLOAD(M,IA+1,K)+{(H-RA)**2*(PA+(P2-PA)/3.)/(2.*H)
      CLOAD(M,IA+2,K)=CLOAD(M,IA+2,K)+{(H-RA)*{PA*(H+RA)+(P2-PA)*(2.*H+RA
      *)/3.)/(2.*H)
      CLOAD(M,IB+1,K)=CLOAD(M,IB+1,K)+{(RB/H)*{P1*(H-RB/2.)+.5*(PB-P1)*(H
      *-2.*RB/3.)}
      CLOAD(M,IB+2,K)=CLOAD(M,IB+2,K)+RB**2*(P1/2.+{(PB-P1)/3.)/H

```

```

      LL=IA+2
      LN=IB
      DO 60 IL=LL,LN
      PA1=P2
      P2=PA1+SLOP*H
      CLOAD(M,IL,K)=CLOAD(M,IL,K)+H*(2.*PA1+P2)/6.
      CLOAD(M,IL+1,K)=CLOAD(M,IL+1,K)+H*(PA1+2.*P2)/6.
60  CONTINUE
50  CONTINUE
100 CONTINUE
      RETURN
      END
      SUBROUTINE MRES
      COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
      *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
      COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
      COMMON/FOR/ FBR(40,6,4)
      COMMON/CFOR/ CLOAD(40,21,4),NNP
      COMMON/RES/SIGMAX(40),SLR(40),FAX(40,2,4),MAXL(40)
      INTEGER GRILL
      REAL L
      DIMENSION SUMC(60),SUMB(60),GA(40),GB(40),PMAX(40,4),VMAX(40,4),MM
      *P(40,4),VMP(40,4),PM(40),V(40),EK(40),RX(40),EULR(40),FAC(40),SIGM
      *(40,4)
      DATA CM/.85/,CC/126.1/,PI/3.14159/
      DO 4 I=1,NJ
      SUMC(I)=0.
      SUMB(I)=0.
4  CONTINUE
      DO 3 I=1,NM
      GA(I)=0.
      MAXL(I)=0

```

```

      GB(I)=0.
      EK(I)=0.
      FAC(I)=0.
      SIGMAX(I)=0.
3  CONTINUE
      DO 1 K=1,NL
      DO 1 I=1,NM
      PMAX(I,K)=0.
      VMAX(I,K)=0.
      VMP(I,K)=0
      MMP(I,K)=0
      SIGM(I,K)=0.
1  CONTINUE
      DO 1011 K=1,NL
      DO 1011 I=1,NM
      H=L(I)/(NNP-1)
      DO 1003 J=1,NNP
      IF(J.GT.1) GO TO 101
      V(1)=FBR(I,2,K)+CLOAD(I,1,K)
      PM(1)=-FBR(I,3,K)
      GO TO 102
101  V(J)=V(J-1)+CLOAD(I,J,K)
      PM(J)=PM(J-1)+V(J-1)*H
102  IF(ABS(PM(J)).LT.PMAX(I,K)) GO TO 103
      PMAX(I,K)=ABS(PM(J))
      MMP(I,K)=J
103  IF(ABS(V(J)).LT.VMAX(I,K)) GO TO 1003
      VMAX(I,K)=ABS(V(J))
      VMP(I,K)=J
1003 CONTINUE
1011 CONTINUE
      IF(GRILL.EQ.0) GO TO 1110

```

```

      DO 1012 K=1,NL
      DO 1012 I=1,NM
      SIG=ABS(PMAX(I,K))/SX(I)
      IF(SIG.LT.SIGMAX(I)) GO TO 1012
      SIGMAX(I)=SIG/23.76
1012 CONTINUE
      IF(GRILL.NE.0) GO TO 1013
1110 DO 10 I=1,NM
      JJ=MINC(I,1)
      KK=MINC(I,2)
      IF(ABS(SINA(I)).EQ.0.) GO TO 11
      SUMC(JJ)=SUMC(JJ)+XI(I)/L(I)
      SUMC(KK)=SUMC(KK)+XI(I)/L(I)
      GO TO 10
11 SUMB(JJ)=SUMB(JJ)+XI(I)/L(I)
   SUMB(KK)=SUMB(KK)+XI(I)/L(I)
10 CONTINUE
   DO 15 I=1,NM
   JJ=MINC(I,1)
   KK=MINC(I,2)
   CALL GFACT(JCODE(JJ,1),JCODE(JJ,2),JCODE(JJ,3),GA(I))
   IF(GA(I).NE.0.) GO TO 16
   IF(SINA(I).EQ.0.) GO TO 17
   GA(I)=SUMC(JJ)/SUMB(JJ)
   GO TO 18
17 GA(I)=SUMB(JJ)/SUMC(JJ)
18 IF(GA(I).GT.10.) GA(I)=10.
   IF(GA(I).LT..5) GA(I)=.5
16 CALL GFACT(JCODE(KK,1),JCODE(KK,2),JCODE(KK,3),GB(I))
   IF(GB(I).NE.0.) GO TO 21
   IF(SINA(I).EQ.0.) GO TO 19
   GB(I)=SUMC(KK)/SUMB(KK)

```

```

      GO TO 20
19  GB(I)=SUMB(KK)/SUMC(KK)
20  IF(GB(I).GT.10.) GB(I)=10.
      IF(GB(I).LT..5) GB(I)=.5
21  CONTINUE
15  CONTINUE
      DO 35 I=1,NM
          ITER=0
          CALL FACTK(GA(I),GB(I),EK(I))
          IF(EK(I).LT.1.) EK(I)=1.
          RX(I)=SQRT(XI(I)/A(I))
          SLR(I)=EK(I)*L(I)/RX(I)
          EULR(I)=12.*PI**2*E(I)/(23.*SLR(I)**2)
          IF(SLR(I).GT.CC) GO TO 104
106  FAC(I)=36.*(1.-SLR(I)**2/(2.*CC**2))/(5./3.+3.*SLR(I)/(8.*CC)-SLR(
      *I)**3/(8.*CC**3))
          GAP=FAC(I)/EULR(I)*GA(I)
          GBP=FAC(I)/EULR(I)*GB(I)
          CALL FACTK(GAP,GBP,EKK)
          IF(EK(I)-EKK.LT..05) GO TO 105
          EK(I)=EKK
          SLR(I)=EK(I)*L(I)/RX(I)
          ITER=ITER+1
          IF(ITER.GT.20) GO TO 105
          GO TO 106
105  EK(I)=EKK
          IF(EK(I).LT.1.) EK(I)=1.
          SLR(I)=EK(I)*L(I)/RX(I)
104  IF(FAC(I).EQ.0.) FAC(I)=EULR(I)
      CONTINUE
35  CONTINUE
      DO 1000 K=1,NL

```

```

      DO 1001 I=1,NM
      IF(ABS(FBR(I,1,K)).GT.ABS(FBR(I,4,K))) GO TO 120
      FAX(I,1,K)=FBR(I,4,K)/(A(I)*FAC(I))
      FAX(I,2,K)=4.
      GO TO 121
120  FAX(I,1,K)=FBR(I,1,K)/(A(I)*FAC(I))
      FAX(I,2,K)=1.
121  CONTINUE
1001  CONTINUE
      DO 1004 I=1,NM
      IF(ABS(FAX(I,1,K)).LT..15) GO TO 122
      IF(FAX(I,2,K).EQ.4..AND.FAX(I,1,K).GT.0.) GO TO 122
      IF(FAX(I,1,K).LT.0..AND.FAX(I,2,K).EQ.1.) GO TO 122
      IF(MMP(I,K).EQ.1.OR.MMP(I,K).EQ.NNP) GO TO 123
      SIGM(I,K)=ABS(FAX(I,1,K))+CM*(ABS(PMAX(I,K))/SX(I))/(23.76*(1.-ABS
      *(FAX(I,1,K))/EULR(I)))
      GO TO 125
122  SIGM(I,K)=ABS(FAX(I,1,K))+ABS(PMAX(I,K))/(SX(I)*23.76)
      GO TO 125
123  SIGM(I,K)=ABS(FAX(I,1,K))*FAC(I)/21.6+ABS(PMAX(I,K))/(SX(I)*23.76)
125  IF(SIGM(I,K).LT.SIGMAX(I)) GO TO 126
      SIGMAX(I)=SIGM(I,K)
      MAXL(I)=K
126  CONTINUE
1004  CONTINUE
1000  CONTINUE
1013  CONTINUE
      DO 200 I=1,NM
      WRITE(6,113)SIGMAX(I)
113  FORMAT(30X,'SIGMAXX=',E15.7)
200  CONTINUE
      RETURN

```

```

END
SUBROUTINE FACTK(GA,GB,F)
GAVG=(GA+GB)/2.
IF(GAVG.GT.2.) GO TO 1
F=SQRT(1.+GAVG)*(20.-GAVG)/20.
GO TO 2
1 F=.9*SQRT(1.+GAVG)
2 CONTINUE
RETURN
END
SUBROUTINE GFACT(JC1,JC2,JC3,G1)
IF(JC1.EQ.0..AND.JC2.EQ.0..AND.JC3.EQ.0) GO TO 10
IF(JC1.EQ.0..AND.JC2.EQ.0..AND.JC3.NE.0) GO TO 20
GO TO 30
10 G1=1.
GO TO 30
20 G1=10.
30 CONTINUE
RETURN
END
SUBROUTINE FULSTR
COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
*E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
COMMON/MEM/L(40),COSA(40),SINA(40),NDOF,MD
COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
*4,4)
COMMON/RES/SIGMAX(40),SLR(40),FAX(40,2,4),MAXL(40)
COMMON/FSD/ ISTOP
COMMON/TORS/ G(40),XI1(40)
REAL L
INTEGER GRILL
K=0

```

```

      DO 10 I=1,NM
      IF(SX(I).LE.5..AND.SIGMAX(I).LE.1.) GO TO 10
      IF(SX(I).GE.1200..AND.SIGMAX(I).GE.1.) GO TO 10
      IF(ABS(SIGMAX(I)-.98).GT..02) GO TO 11
10  CONTINUE
      DO 15 I=1,NM
      15  WRITE(6,1000) I,XI(I),A(I),SX(I),SIGMAX(I)
1000  FORMAT(15X,15,5X,4(E15.7,5X))
      ISTOP=0
      GO TO 60
      11  DO 20 I=1,NM
      K=MAXL(I)
      IF(SX(I).LE.5..AND.SIGMAX(I).LE.1.) GO TO 40
      IF(SX(I).GE.1200..AND.SIGMAX(I).GE.1.) GO TO 40
      SX(I)=(1./98)*SIGMAX(I)*SX(I)
      IF(SX(I).GT.1200.) SX(I)=1200.
      IF(SX(I).LT.5.) SX(I)=5.
      IF(GRILL.NE.0) CALL GRDSTR(I)
      IF(GRILL.NE.0) GO TO 40
      IF(ABS(FAX(I,1,K)).GT..15) GO TO 30
      A(I)=3.624150+.119637*SX(I)-.00009708825*SX(I)**2+.00000005341595*
      *SX(I)**3
      IF(SX(I).LT.50.) GO TO 41
      XI(I)=-191.3096+10.90819*SX(I)+.01840775*SX(I)**2-.00001076241*SX(
      *I)**3
      GO TO 40
      41  XI(I)=7.98*SX(I)
      GO TO 40
      30  IF(SX(I).LT.50.) GO TO 31
      XI(I)=-91.79416+6.954839*SX(I)+.0034126*SX(I)**2
      GO TO 32
      31  XI(I)=-11.88349+4.132184*SX(I)+.009396914*SX(I)**2

```

```

32 CONTINUE
   IF(SX(I).GT.50.) GO TO 50
   A(I)=2.346176+.2443544*SX(I)-.00003459238*SX(I)**2
   GO TO 40
50 A(I)=.61575+.1871112*SX(I)-.00001518453*SX(I)**2
40 CONTINUE
   ALPHA(I)=E(I)*XI(I)/L(I)**3
   IF(GRILL.NE.0) GO TO 80
   BETA(I)=A(I)*L(I)**2/XI(I)
   GO TO 70
80 BETA(I)=XI(I)*G(I)*L(I)**2/(XI(I)*E(I))
70 ISTOP=1
20 CONTINUE
60 CONTINUE
   RETURN
   END
   SUBROUTINE GSTIFF(SS)
   COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCDD
   *E(40,3),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
   COMMON/HEM/L(40),COSA(40),SINA(40),NDOF,MD
   COMMON/MACIN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
   *4,4)
   DIMENSION INDEX(6,6),H(9),SS(NDOF,MD)
   REAL L
   DATA INDEX/1,-2,4,5,2,6,-2,3,7,-2,-3,7,4,7,8,6,-7,9,5,-2,6,1,2,4,2
   *, -3,-7,2,3,-7,6,7,9,4,-7,8/
   DO 1 I=1,NDOF
   DO 1 J=1,MD
1  SS(I,J)=0.
   DO 10 I=1,NM
   H(1)=ALPHA(I)*(COSA(I)**2*BETA(I)+SINA(I)**2*4.*L(I)**2)
   H(2)=ALPHA(I)*SINA(I)*6.*L(I)

```

```

H(3)=ALPHA(I)*12.
H(4)=ALPHA(I)*COSA(I)*SINA(I)*(BETA(I)-4.*L(I)**2)
H(5)=ALPHA(I)*(SINA(I)**2*2.*L(I)**2-COSA(I)**2*BETA(I))
H(6)=-ALPHA(I)*COSA(I)*SINA(I)*(BETA(I)+2.*L(I)**2)
H(7)=ALPHA(I)*COSA(I)*6.*L(I)
H(8)=ALPHA(I)*(SINA(I)**2*BETA(I)+COSA(I)**2*4.*L(I)**2)
H(9)=ALPHA(I)*(COSA(I)**2*2.*L(I)**2-SINA(I)**2*BETA(I))
DO 15 JM=1,6
J=MCODE(I,JM)
IF(J.EQ.0) GO TO 15
DO 20 KM=JM,6
K=MCODE(I,KM)
IF(K.EQ.0) GO TO 20
KB=K-J+1
LN=INDEX(JM,KM)
IF(LN.GT.0) GO TO 21
SM=-H(-LN)
GO TO 22
21 SM=H(LN)
22 SS(J,KB)=SS(J,KB)+SM
20 CONTINUE
15 CONTINUE
10 CONTINUE
RETURN
END
SUBROUTINE GRDSTR(I)
COMMON/DATIN/X(40),Y(40),A(40),E(40),XI(40),SX(40),MINC(40,2),JCOD
*E(40,2),MCODE(40,6),MA(4),Q(60,4),NM,NL,NJ,GRILL,MCOMP
COMMON/MEML/L(40),COSA(40),SINA(40),NDOF,MD
COMMON/MACTN/ FMA(40,6,4),ALPHA(40),BETA(40),IACT(60,2,4),DACT(60,
*4,4)
COMMON/TORS/ G(40),XI1(40)

```

```

COMMON/RES/SIGMAX(40),SLR(40),FAX(40,2,4),MAXL(40)
INTEGER GRILL
A(I)=3.989864+.1468865*SX(I)-.0001605061*SX(I)**2+.00000008959847*
*SX(I)**3
IF(SX(I).LT.50) GO TO 41
XI(I)=-1.07.5656+7.474257*SX(I)+.02309009*SX(I)**2-.00001222851*SX(
*I)**3
GO TO 40
41 XI(I)=-13.15975+4.499167*SX(I)+.04945434*SX(I)**2
40 XI1(I)=.6457337+.01040116*SX(I)+.00003618241*SX(I)**2
RETURN
END

```

**The vita has been removed from
the scanned document**

A STUDY OF SOME FEATURES
OF THE FULLY STRESSED DESIGN ALGORITHM

by

Kevin Haskell Traynor

(ABSTRACT)

This thesis presents the results of a study of the fully stressed design (FSD) of rigidly connected plane steel frames. The objective is to: (1) formulate the FSD algorithm consistent with the most recent AISC Manual of Steel Construction, (2) study the effect of some of the features of FSD on the final design, and (3) compare final designs with previously published results. Dependent section property functions are derived for the data points of available sections using a polynomial regression model with the section modulus as the independent design variable. Comparing the derived equations with two other published sets of equations demonstrates that the specific dependent functions used in an FSD solution significantly affect the structural weight of the final design. The most efficient sections are those that have the largest moment of inertia and smallest cross sectional area for a given section modulus. These functions are derived for the sections available. Finally, a procedure is suggested for automatic selection of a set of sections as the final step in the design.