OPTIMIZATION OF STEEL FRAMES VIA PENALTY FUNCTIONS

by

William F. Cofer

Thesis submitted to the Graduate 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:

S. M. Holzer, Chairman

R. H. Plaut

A. E. Somers, Jr.

May, 1978

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The work in this thesis was done under the direction of Dr. S. M. Holzer and I wish to express my gratitude for his guidance and encouragement.

Also, appreciation is given to Dr. R. H. Plaut and Dr. A. E. Somers for their review of this thesis and helpful suggestions.

I would also like to thank for her time and typing ability.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Objectives

The purpose of this study is to develop a program to minimize the weight of steel frames using a penalty function technique. The results will be studied in terms of reliability and cost and compared with the results of the Fully Stressed Design method.

1.2 Background

Engineers have always attempted to select the most economical solution to a given problem. With the increasing scarcity and expense of raw materials, this idea has increased in importance. Hence, in the past few years much investigation in this field has been done.

With the growing use of computers in structural design, weight optimization through nonlinear programming has become possible. The large amounts of computer execution time required, however, have made this idea feasible mainly in research.

The Fully Stressed Design method has been applied, but it has disadvantages in that there is no guarantee that a fully stressed design is optimal. Also, the only constraints available are stress constraints.

Therefore, there is a need for an economical, widely applicable, reliable nonlinear programming routine for structural optimization.

In this paper, a literature survey is given, the method is discussed mathematically and applied to a computer program, problems are solved and discussed, and conclusions are drawn.

CHAPTER 2

LITERATURE SURVEY

Mathematically, the penalty function approach was first developed by Fiacco and McCormick in 1968 [8]. Mathematicians have since discussed [22] [19] [23] [10] and improved upon the method. Lootsma, in 1972, [17] introduced a mixed penalty function which used both interior and exterior approaches while, at the same time, Osborne and Ryan [24] [29] described a hybrid penalty function where active inequality constraints were changed to equality constraints.

The penalty function technique was first applied to structural optimization by Bracken and McCormick [2] in 1968. They were concerned with the weight minimization of vertically corrugated, transverse bulkheads for oil tankers. Their design variables were the width of the flange and web of each corrugation, the depth of corrugation and the material thickness. They had constraints on section modulus to be adequate for maximum bending moment, plate thickness to be adequate for pressure requirements and corrosion while also being greater than a prescribed minimum value, and geometry such that the length of web was greater than the depth of corrugation. The problem had been solved before, and the minimum values from the penalty function method were slightly higher. The authors found that since the other solution violated a constraint, theirs was the best.

The work with oil tankers was continued by Kavlie and Moe [14] when they minimized the weight of grillages. They suggested using standard sections with only one design variable per member. As constraints, they had limits on stress from bending, axial forces, and shear, along with upper and lower limits on member plate thicknesses. They noted, that, since the problem was not convex, there was a possibility of converging to a local rather than global minimum and that a fully stressed design was not necessarily optimal. To rectify this, they suggested minimizing from different initial designs.

Kavlie and Moe also investigated the weight minimization of frames [15]. Here, the authors recommended the adoption of an extended penalty function so that infeasible initial designs could be handled. They also gave suggestions on values for the response factor, $\hat{r}$, and discussed unconstrained minimization schemes, scaling of design variables, and convergence criteria. A tanker frame was optimized, using stress constraints. They concluded that a proper choice of the initial response factor is important if the problem is not convex, and that the penalty function method is well suited for optimization of statically indeterminate structures.

Continuing with the idea of frame optimization, Felton, Nelson, and Bromowicki [7] [21] [3] wrote three consecutive papers in which built-up steel cross sections were optimized (fully stressed) such that cross sectional area was at a minimum value for a particular moment of inertia. The penalty function method was then used to optimize the frame for minimum moment of inertia. They used a

Davidon-Fletcher-Powell minimization routine and obtained smooth convergence. Bromowicki [3] noted the large amounts of execution time required, however and said that execution time increased exponentially with an increase in the number of design variables. In each of these papers were mentioned other minimization methods that were claimed to be more efficient.

Shamie and Schmit [30] optimized the weight of frames with an emphasis on buckling. They used constraints on vibrating frequency and stresses along with side constraints on member sizes. They also used the Davidon-Fletcher-Powell minimization routine and obtained convergence. For improvement, they suggested finding a way to disregard all inactive constraints and they recommended finding a more efficient unconstrained minimization routine.

DeSilva and Grant [6] minimized the weight of a three bar truss. They took advantage of the fact that the penalty function approach can also handle equality constraints, in this case, equilibrium and the stress-strain relations. Their inequality constraints were on the buckling modes and fabrication limitations. They used and compared two methods: the Heaviside penalty function technique and the Sequential Unconstrained Minimization Technique (SUMT). For unconstrained minimization, they used the methods of Rosenbrock, Powell, and Nelder-Mead. The authors concluded that SUMT was best in that it gave better results in less time and that the Powell method was the best unconstrained minimization routine.

Pickett, Rubinstein, and Nelson [26] obtained good results in their work to optimize trusses using reduced design variables. They used SUMT with the Davidon-Fletcher-Powell minimization routine because of the reliability associated with them. Ramakrishnan and Francavilla [28] also found the penalty function method to be reliable and general in its applicability in their work on shape optimization. They optimized a clamped circular plate with respect to its thickness, a pressure vessel with respect to the shape of its middle surface, and a gravity dam with respect to its cross sectional area. In all problems, they found that solutions were found in a straight forward manner with the advantage that intermediate designs were also feasible and useful.

Patnaik and Sankaran [25] minimized the weight of stiffened, cylindrical panels using the penalty function method. Their design variables were the skin thickness and the thickness, depth, and spacing of longitudinal and circumferential stiffeners. They had constraints on stresses and frequency and used the Davidon-Fletcher-Powell minimization routine. They obtained good designs, as compared to other solutions in their references. Work in three dimensions was continued by Majumder and Thornton [18] who used a mixed interior and exterior penalty function approach to minimize the weight of stiffened shells of revolution. Stress constraints were used for the interior penalty function while buckling constraints were used for the exterior. For unconstrained minimization, they used a Fletcher-Reeves routine rather than Davidon-Fletcher-Powell because it used less storage. They obtained fairly good answers except for one case in which they did not obtain a minimum value because the Fletcher-Reeves routine was not reliable. For this reason, they recommended use of the Davidon-Fletcher-Powell routine.

Moses, Fox, and Goble [20] [12] have investigated several applications of the penalty function method along with other methods. They minimized the weight of a welded box girder with constraints on stress, deflection, local buckling, lateral stiffness, and minimum plate thickness and they minimized the cross sectional area of a welded wide-flange beam with constraints from the AISC specifications and buckling. For unconstrained minimization, they suggested the use of a nongradient method, such as Powell's, only when the function does not have continuous derivatives. If the derivatives were continuous, the use of the Davidon-Fletcher-Powell method was recommended even if the derivatives had to be found numerically.

The authors concluded that the penalty function method has the advantages of being easy for the practicing engineer to understand and use and that it can be easily modified for changes in code requirements. The method is best, however, for small problems with explicit inequality constraints.

The book by Fox [10] and the book by Gallagher and Zienkiewicz [11] give excellent overviews of the penalty function method. Fox concludes that

> An appealing feature of the interior penalty function method is that, given an initial acceptable design, it produces an improving sequence of acceptable designs.

> Moreover, we approach the constraints in this sequence in such a way that they become critical only near the end.
>
> The interior penalty function method is sometimes said to "funnel" the optimum design process "down the middle," keeping the designs away from the constraint surfaces until final convergence.

Gallagher and Zienkiewicz add that the main advantages are:

> i) General applicability to widely different types of problems.
>
> ii) Ease of programming. The optimization program can be built on a modular basis so that the practical user only needs to formulate and program his own problem, while a general-purpose optimization program is used more or less as a "black box."
>
> iii) The method is robust in the sense that it works well even with approximate information. This permits one to use approximate and time-saving methods of analysis during major portions of the search.
>
> iv) By means of the extended penalty-function technique or other available techniques, the method is also able to accept infeasible starting points. This may be a quite important feature when dealing with complex design problems for which feasible starting points are not readily available.

For these reasons, the penalty function technique is gaining increasing popularity in the field of structural optimization.

CHAPTER 3

MATHEMATICAL THEORY OF THE PENALTY FUNCTION TECHNIQUE

3.1 Algorithm

The penalty function technique to solve a constrained nonlinear minimization problem is essentially a very simple concept. Using the Sequential Unconstrained Minimization Technique (SUMT), the original problem becomes converted into a sequence of unconstrained minimization problems. A "penalty function," which contains the constraints, is added to the objective function which is thereby deflected from the constraint boundaries.

For example, if the problem is:

$$\left.\begin{array}{l}\text{minimize } f(x)\\ \text{subject to } g_i(x) \geq 0,\ i=1,\ldots,m,\end{array}\right\} \qquad (1)$$

a function $\phi(x,\hat{r})$ can be formed such that

$$\phi(x,\hat{r}) = f(x) + P(x,\hat{r}) \qquad (2)$$

where $P(x,\hat{r})$ is the penalty function, $f(x)$ is the objective function, and $g_i(x)$ are the constraints. The penalty function used here is

$$P(x,\hat{r}) = \hat{r} \sum_{i=1}^{m} \frac{1}{g_i(x)} \qquad (3)$$

where $\hat{r}$ are some positive numbers used as scaling factors. The reader should notice that as any $g_i(x)$ approaches zero, $P(x,\hat{r})$ becomes large, thus deflecting the minimum from that boundary.

The SUMT algorithm is as follows:

1) Choose an initial feasible point and an initial value $\hat{r}_1$.

2) Minimize $(x,\hat{r}_1)$ as though it is unconstrained, using any technique, and calculate x_{min}.

3) Set $x_1 = x_{min}$, replace $\hat{r}_1$ by a smaller value $\hat{r}_2$ and repeat steps 2 and 3 until some convergence criterion is satisfied.

This can be illustrated in a very simple form by Fig. 1.

3.2 Development of the Interior Penalty Function Method

A detailed development of SUMT was published in 1968 by Fiacco and McCormick [8]. They defined a local minimum to be "a point at which, in a neighborhood about that point, there is no other point satisfying the constraints that gives a smaller value of the objective function." A necessary condition for a point, x*, to be a local minimum of a function h(x) is

$$\nabla h(x^*) = 0 \tag{4}$$

and a sufficient condition is that $\nabla^2 h(x^*)$ must be positive definite.

The authors used the concept of the Lagrange Multipliers technique whereby each constraint g(x) is multiplied by a certain constant, called a multiplier, and added to the objective function, f(x). The constants are specified for a constrained minimum.

For the problem in Eq. 1, the Lagrangian function is

$$\mathscr{L}(x,u) = f(x) - \sum_{i=1}^{m} u_i g_i(x) \tag{5}$$

where u_i, i=1,. . .,m are the Lagrange multipliers. Fiacco and McCormick [8] proved that these multipliers exist such that $\mathscr{L}(x,u)$ is at a local minimum, x*.

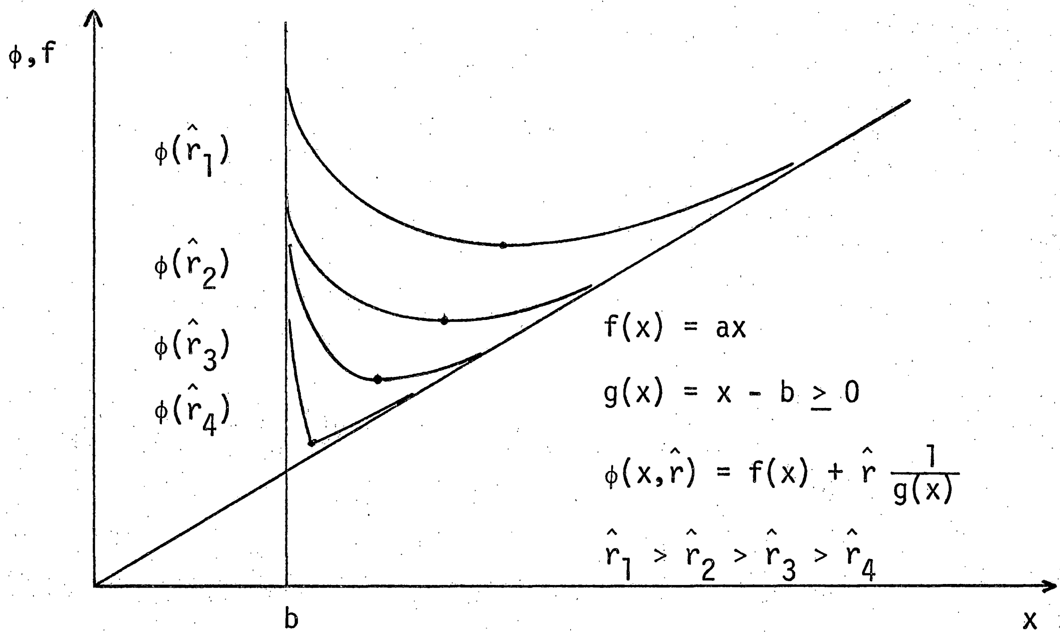


Fig. 1 - Example of Penalty Function Convergence

The authors considered a small change from the minimum point x* and called this point x. They were able to show that point x is a local minimum point of the function

$$\phi(x,\hat{r}) = f(x) + \hat{r}\sum_{i=1}^{m} \frac{1}{g_i(x)} . \qquad (6)$$

They also showed that, as $\hat{r}$ becomes very small, the point x approaches the constrained minimum point, x*.

Thus, a sequence of unconstrained minimizations of $\phi(x,\hat{r})$, with $\hat{r}$ approaching 0, will converge to the constrained minimum point.

CHAPTER 4

SPECIALIZATION OF THE SUMT ALGORITHM FOR STRUCTURAL OPTIMIZATION

4.1 Objective Function

The purpose of this study is to minimize the total weight of a structure. This is analogous to minimizing the volume of each member. Since there are only a certain number of discrete sections available for steel design, it seems logical that an approximating curve can be drawn that will relate one section property with all of the others. In 1966, Brown and Ang [4] published a relationship between section properties and moment of inertia for all wide flange "economy" sections. They are:

$$S = \begin{cases} \sqrt{60.6\ I + 84100} - 290 & 0 \le I \le 9000 \\ \dfrac{I - 8056.3}{1.876} & 9000 \le I \le 20300 \end{cases}$$

$$w = \frac{490}{144} A = \begin{cases} 1.58\ \sqrt{I} & 0 \le I \le 9000 \\ \dfrac{I + 2300}{75.3} & 9000 \le I \le 20300 \end{cases} \qquad (7)$$

$$r = \begin{cases} \left[\dfrac{I}{0.21583}\right]^{\frac{1}{4}} & 0 \le I \le 9000 \\ \dfrac{I + 174498}{12841} & 9000 \le I \le 20300 \end{cases}$$

where I is the moment of inertia, S is the section modulus, w is the weight per foot, A is the cross sectional area, and r is the radius of gyration. The reader should notice that, since $\sqrt{I}$ is proportional to w, if $\sqrt{I}$ times the member length is minimized, then the total weight of the structure will be at a minimum.

This simplifies the problem considerably because now, instead of having to work with a multitude of variables, it is possible to obtain a minimum weight section by minimizing the moment of inertia alone. Thus, the objective function to be minimized here is

$$f(I) = \sum_{i=1}^{m} \sqrt{I_i}\, L_i \tag{8}$$

where I_i and L_i are the moment of inertia and length respectively of member i.

4.2 Constraints

At this time, most steel design is done by choosing a section that will satisfy appropriate equations in the AISC Manual of Steel Construction [1]. Since this practice is widely accepted, a minimum weight design conforming to these equations should be a feasible design. Hence, they are the basis for the constraints.

This program is applicable to members under tension, compression, bending (around the major axis) and any combination of the three. Compact, wide-flange, economy sections taken from the AISC "Allowable Stress Design Selection Table" are the only sections considered for this analysis. See Appendix A for the development of the constraints.

4.3 Structural Analysis Routine

At the heart of any structural optimization program is the routine to analyze the structure with the given design variables. The analysis routine in this treatment was developed in CE 4002 and

CE 5640 and utilizes the Direct Stiffness Method to find the member forces and joint reactions. It contains a variable array allocation scheme by which the arrays for structures with varying quantities of members may be dimensioned exactly. There is a subroutine to integrate the response from member loads as well as joint loads, and the solution routine takes advantage of the bandedness of the stiffness matrix. The reader should note that this analysis routine assumes a linear response.

4.4 Extended Interior Penalty Function

One major disadvantage of the interior penalty function approach is that the starting point for a minimization must be feasible. When optimizing structures it is relatively easy to find such a point for the first iteration. When the frame is reanalyzed, however, the subsequent change in member forces almost assures that the design is no longer feasible. In order to avoid this problem, Kavlie and Moe suggested the extended penalty function [15]. With this idea, the penalty function is

$$\begin{aligned} &\frac{1}{g_i(x)} \text{ for } g_i(x) \geq \varepsilon \\ &\frac{2\varepsilon - g_i(x)}{\varepsilon^2} \text{ for } g_i(x) < \varepsilon \end{aligned} \qquad (9)$$

This can be illustrated by Fig. 2.

If ε is selected too large, however, the minimum point may be affected. Consequently, Cassis and Schmit [5] have developed a

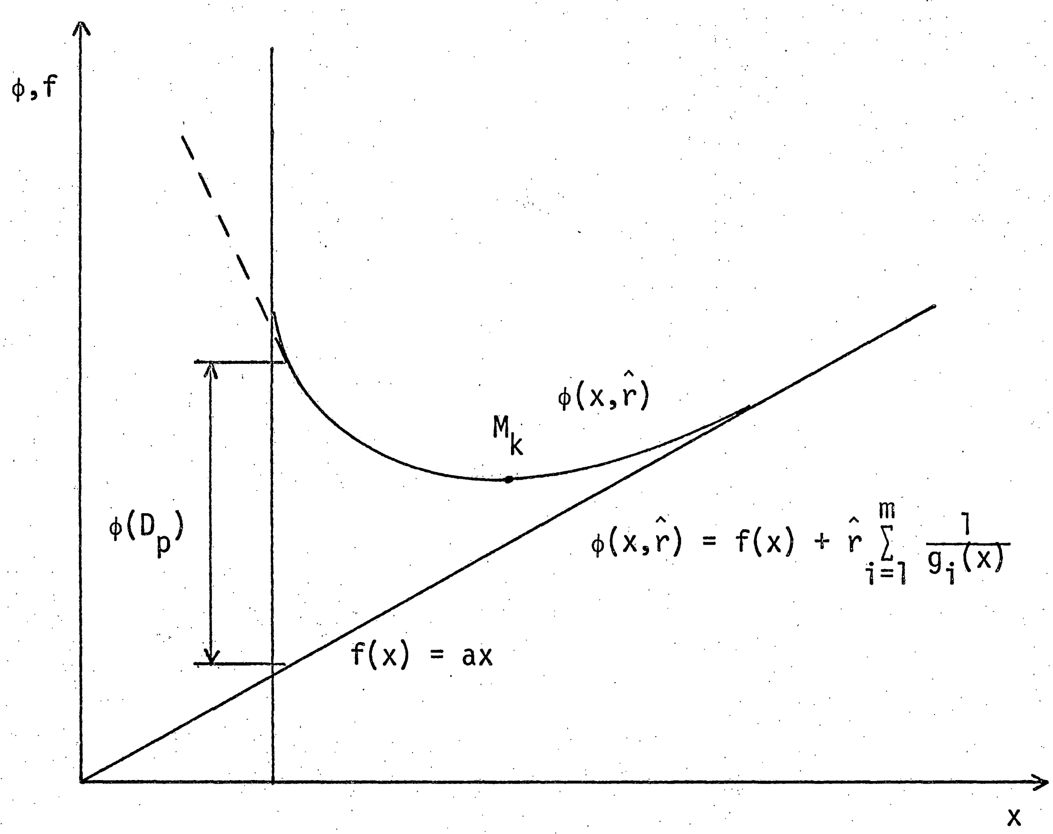


Fig. 2 - Example of an Extended Penalty Function

rational method for determining the best ε. If D_p is the initial point in the minimization process,

$$\varepsilon = \frac{\hat{r}}{\phi(D_p)} . \tag{10}$$

4.5 Extrapolation

Since efficiency is critical to any minimization routine, methods have been developed to speed the convergence process. Fiacco and McCormick [8] have shown that in a SUMT analysis the unconstrained minimum points form a unique, smooth curve. This can be seen for one design variable in Fig. 3. If the formula of this curve, or one closely approximating it, can be found, then a very good starting point for the next minimization would be the extrapolation to the next minimum value.

Fox [10] suggests a parabolic curve, but a more recent technique was suggested by Lund and is described by Gallagher and Zienkiewicz [11]. With his method,

$$x^*(\hat{r}_{i+1}) = \frac{(\hat{r}_{i-1})^{\beta} x^*(\hat{r}_i) - (\hat{r}_i)^{\beta} x^*(\hat{r}_{i-1})}{(\hat{r}_{i-1})^{\beta} - (\hat{r}_i)^{\beta}} \tag{11}$$

where β is a constant. The procedure is

1) Choose β such that $0 < \beta < 1$. Lund suggests $\beta = 0.4$.
2) Compute $x^*(\hat{r}_{i+1})$ by Eq. 11. Evaluate the constraints to see if $x^*(\hat{r}_{i+1})$ is feasible. If it is, it may be used for the next minimization. If $x^*(\hat{r}_{i+1})$ is not feasible, increase β by 0.1 and repeat until a feasible $x^*(\hat{r}_{i+1})$ is

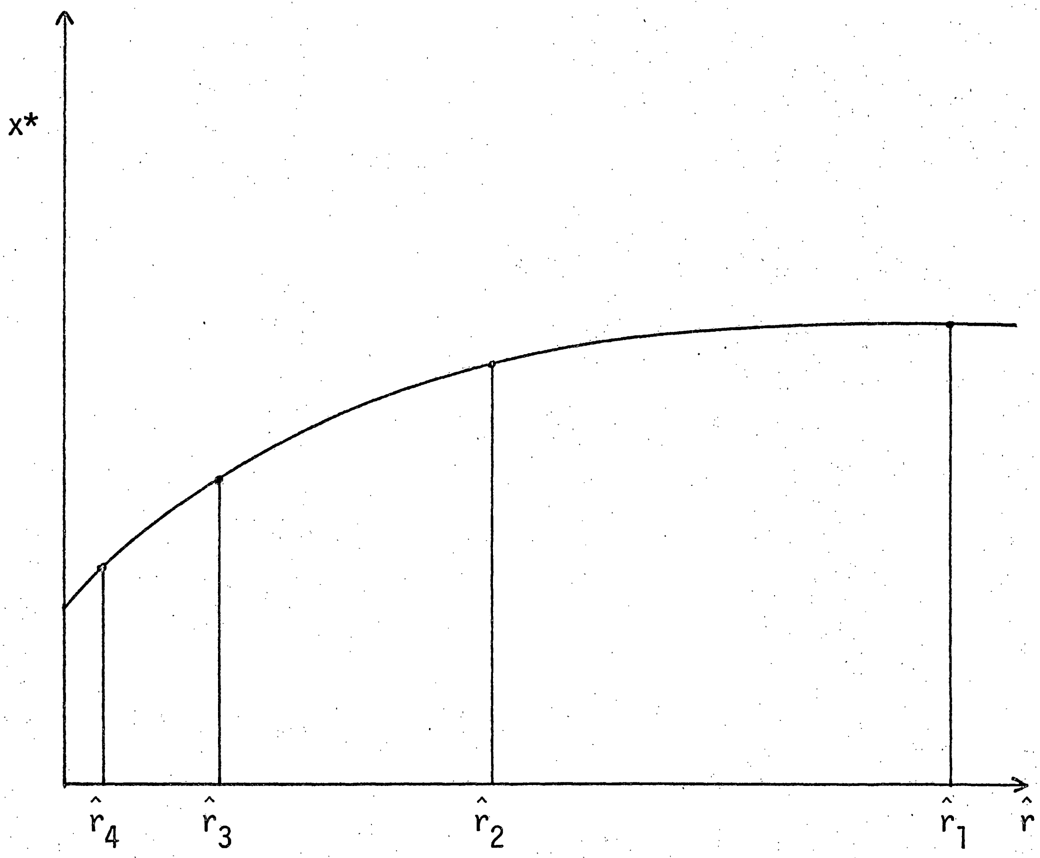


Fig. 3 - Example of a Sequence of Unconstrained Minimum Points

found. If one has not been found and β has reached 1.0, disregard the extrapolation and use $x^*(\hat{r}_i)$ for the next minimization. This method worked well, as can be seen in Table 1.

4.6 Convergence

There are two methods for ending this program. Either the problem will converge or it will reach the limit on the number of iterations allowed. This program will not allow more than 10 iterations of the minimization scheme.

The convergence formula is

$$\varepsilon = \frac{f(x_i) - f(x_{i-1})}{f(x_i)} \tag{12}$$

If $\varepsilon \leq 0.01$, the program will stop. For the unconstrained minimization, ε was set to 0.01 also, and a limit of 100 iterations was imposed.

4.7 Unconstrained Minimization Routines

In this research, two unconstrained minimization routines were used and compared. The first was FMFP and it can be found in the Scientific Subroutine Package at the Virginia Tech Computing Center. It uses a method by Davidon, Fletcher, and Powell [9]. The other routine is called VA13A and it is a member of the Harwell Subroutine Library. It uses a method by Powell [27] and it was published more recently than FMFP.

Table 1 - Example of Extrapolation

	Predicted Mimimum			Actual Mimimum		
i	1	2	3	1	2	3
0	--	--	--	17300.00	17300.00	17300.00
1	--	--	--	1061.51	628.37	304.68
2	--	--	--	915.49	465.49	141.80
3	--	--	--	556.61	193.25	77.77
4	504.92	154.04	68.54	500.90	162.07	60.20
5	482.21	151.60	54.30	483.49	152.64	54.64
6	475.44	148.29	52.46	478.31	149.01	52.99
7	477.02	148.11	52.57	477.01	148.35	52.57

4.8 Other Considerations

There are times when the subroutine to provide the function and gradient values is given negative design variables by the unconstrained minimization routine. Since, in some constraints, square roots are required, a method had to be found by which the design variables are kept positive without affecting the minimization.

Using the same approach as that used for the extended penalty function, the following formula is used:

$$\begin{aligned} I_i &= I_i & I_i &\geq 0 \\ I_i &= \frac{\varepsilon_2^2}{2\varepsilon_2 - I_i} & I_i &< 0 \end{aligned} \tag{13}$$

where I_i is the moment of inertia of member i and ε_2 is an arbitrary constant. In this case, $\varepsilon_2 = 10$ was used.

In order to help the minimization run smoothly, proper choices of response factors, $\hat{r}$, are necessary. Kavlie and Moe [15] suggest an initial $\hat{r}$ value chosen such that the value of the penalty term is approximately half that of the objective function. They also recommend that subsequent $\hat{r}$ values be multiplied by 1/20. These ideas worked well when the FMFP subroutine was used. However, with the VA13A subroutine, a factor of 1/10 was found to work better.

CHAPTER 5

PROBLEM SOLUTIONS

5.1 Program Modifications

The execution time was examined in all problems and efforts were made to make it as small as possible. Therefore, certain observations resulted.

The program was first run with the DFMFP routine using double precision, but single precision provided essentially the same answers with a 90.0% reduction in execution time.

The VA13A routine was tried and it was found to be more likely to diverge than FMFP. For this reason, the scaling factor, $\hat{r}$, was reduced by 1/10 for each minimization rather than 1/20 as before. Even with this change, however, the execution time decreased an additional 3.2%. The VA13A routine was found to require slightly more storage, however, with the array area increasing from 2640 bytes for FMFP to 2660 bytes.

Decreasing the accuracy to which the VA13A routine would minimize did not appreciably affect the results, but caused a further decrease in execution time. Starting from an accuracy of 0.000001, decreasing to an accuracy of 0.01 caused a reduction in time of 1.9%. It should be noted that an accuracy of 0.1 caused the solution to diverge.

Hence, through these revisions with only an insignificant change in the solution, the execution time went from 105.07 sec. to

5.10 sec. for one specific problem.

The first objective funtion used was the sum of the member moments of inertia. In an effort to more closely approximate the actual total weight of the structure, however, the objective function was changed to the sum of the square roots of the moments of inertia, multiplied by the respective member lengths. This corresponds to Eq. 7. There were no changes in the solutions. In a problem with radically different member lengths, however, the solution with the new objective function would be a more accurate minimization of total weight.

5.2 Results

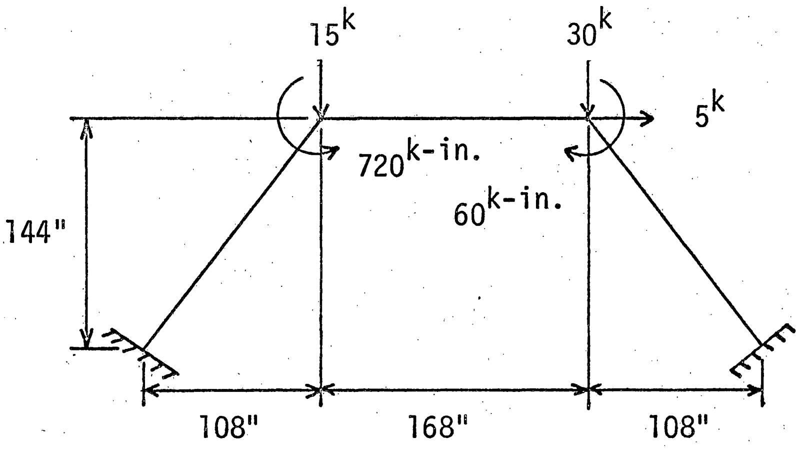
Several problems were solved using this program and are described in Fig. 4 and Fig. 5. The results can be seen in Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7.

All problems were solved using three different initial designs, and none of the solutions were affected. This can be seen in Table 2 for one of the problems. Table 2 also shows a Fully Stressed Design solution for comparison. It required 1.92 sec. of computer execution time as compared to 8.18 sec. for the Penalty Function method.

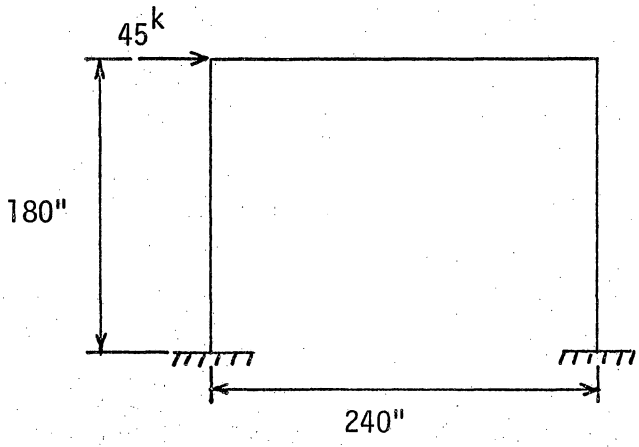
To show that the solution obtained is the minimum weight design within the prescribed constraints, the design was checked by hand. See Table 8.

5.3 Discussion of Results

Since the constraints active at the solution point were the

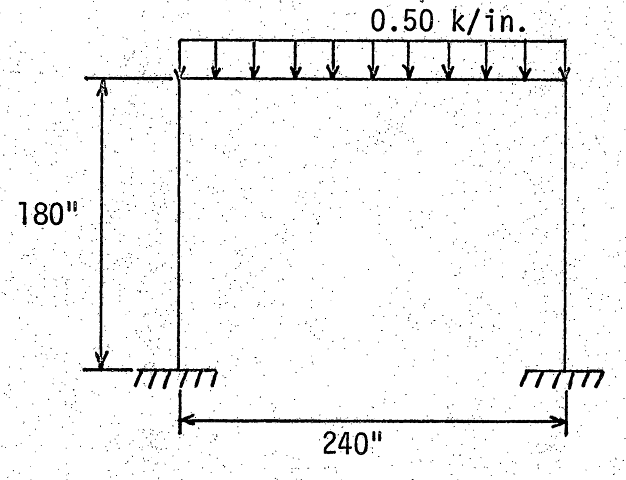


Problem 1

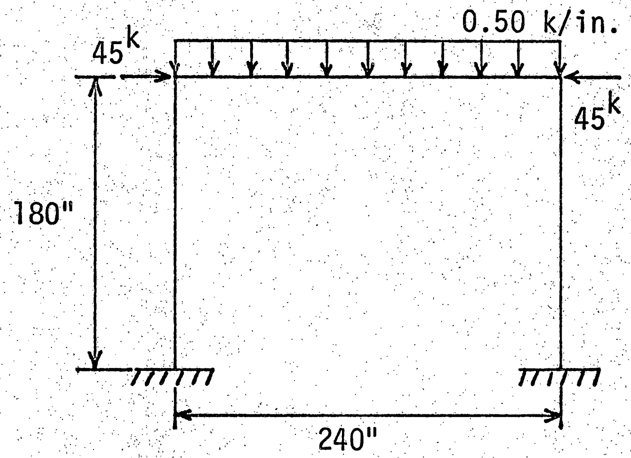


Problem 2

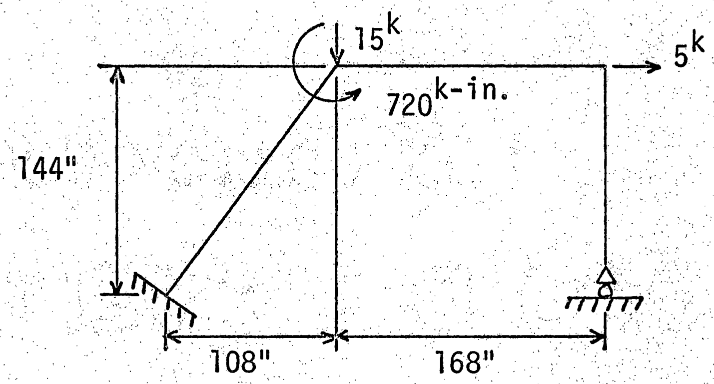
Fig. 4 - Example Problems



Problem 3



Problem 4



Problem 5

Fig. 5 - Example Problems

Table 2 - Solution Sequence, Problem 1

	Initial Design	Revisions						Final Design
		1	2	3	4	5	6	
Trial 1								
Member 1								
Axial Force (K)	--	21.30C	22.04C	22.29C	22.42C	22.51C	22.56C	--
Moment (K-in.)	--	327.45	264.56	228.08	207.31	194.90	187.45	--
Moment of Inertia (in.^{4})	18900.00	161.73	134.06	117.33	107.79	102.38	98.47	98.47
Member 2								
Axial Force (K)	--	10.35C	11.60C	11.95C	12.15C	12.27C	12.34C	--
Moment (K-in.)	--	392.55	455.44	491.91	512.69	525.10	532.55	--
Moment of Inertia (in.^{4})	18900.00	176.00	205.75	222.64	232.09	237.77	241.02	241.02
Member 3								
Axial Force (K)	--	30.12C	30.89C	31.05C	31.15C	31.21C	31.25C	--
Moment (K-in.)	--	315.54	249.74	222.39	206.08	195.96	189.74	--
Moment of Inertia (in.^{4})	18900.00	169.10	138.67	125.95	117.99	113.15	109.91	109.91

Table 2 - Solution Sequence, Problem 1 (continued)

Trial 2	Initial Design	Revisions 1	2	3	4	5	6	Final Design
Member 1								
Axial Force (K)	--	21.79C	22.19C	22.37C	22.48C	22.54C	--	--
Moment (K-in.)	--	287.98	241.52	215.11	199.61	190.03	--	--
Moment of Inertia (in.^{4})	1540.00	144.16	123.46	111.51	104.22	99.89	--	99.89
Member 2								
Axial Force (K)	--	11.20C	11.82C	12.07C	12.22C	12.31C	--	--
Moment (K-in.)	--	432.02	478.48	504.89	520.39	529.97	--	--
Moment of Inertia (in.^{4})	1540.00	194.68	216.36	228.74	235.91	240.28	--	240.28
Member 3								
Axial Force (K)	--	30.65C	30.99C	31.11C	31.19C	31.24C	--	--
Moment (K-in.)	--	272.05	232.77	212.32	199.82	191.92	--	--
Moment of Inertia (in.^{4})	1540.00	148.89	130.79	121.04	114.97	111.55	--	111.55

Table 2 - Solution Sequence, Problem 1 (continued)

	Initial Design	Revisions 1	2	3	4	5	6	Final Design
Trial 3								
Member 1								
Axial Force (K)	--	22.04C	22.28C	22.42C	22.50C	22.56C	--	--
Moment (K-in.)	--	267.66	229.34	207.94	195.48	187.68	--	--
Moment of Inertia ($in.^4$)	52.00	135.05	117.83	108.16	102.49	98.94	--	98.94
Member 2								
Axial Force (K)	--	11.64C	11.94C	12.14C	12.26C	12.34C	--	--
Moment (K-in.)	--	452.34	490.66	512.06	524.52	532.32	--	--
Moment of Inertia ($in.^4$)	52.00	204.11	222.02	231.63	237.52	241.38	--	241.38
Member 3								
Axial Force (K)	--	30.92C	31.05C	31.15C	31.21C	31.25C	--	--
Moment (K-in.)	--	249.73	223.27	206.57	196.45	189.93	--	--
Moment of Inertia ($in.^4$)	52.00	138.56	126.17	118.45	113.29	110.56	--	110.56
FSD								
Member 1	1540.00	163.35	142.91	128.86	123.69	121.28	120.02	119.36
Member 2	1540.00	179.93	207.70	220.86	225.90	228.98	230.62	231.48
Member 3	1540.00	189.75	152.81	146.48	141.95	139.89	138.81	138.24

Table 3 - Solution Sequence, Problem 2

	Initial Design	Revisions 1	Revisions 2	Revisions 3	Final Design
Member 1					
Axial Force (K)	--	13.74T	12.72T	12.29T	--
Moment (K-in.)	--	2428.16	2573.41	2647.41	--
Moment of Inertia ($in.^4$)	1540.00	1196.58	1270.31	1273.17	1273.17
Member 2					
Axial Force (K)	--	22.29C	22.15C	22.02C	--
Moment (K-in.)	--	1660.36	1539.30	1489.50	--
Moment of Inertia ($in.^4$)	1540.00	817.43	751.08	752.75	752.75
Member 3					
Axial Force (K)	--	13.74C	12.72C	12.29C	--
Moment (K-in.)	--	2374.23	2473.71	2503.31	--
Moment of Inertia ($in.^4$)	1540.00	1167.73	1215.25	1228.15	1228.15

Execution Time = 2.93 sec.

Table 4 - Solution Sequence, Problem 3

	Initial Design	Revisions 1	2	3	Final Design
Member 1					
Axial Force (K)	--	60.00C	60.00C	60.00C	--
Moment (K-in.)	--	1737.99	1820.87	1816.19	--
Moment of Inertia ($in.^4$)	1540.00	992.76	1037.84	1034.48	1034.48
Member 2					
Axial Force (K)	--	14.41C	15.10C	15.07C	--
Moment (K-in.)	--	1737.99	1820.87	1816.46	--
Moment of Inertia ($in.^4$)	1540.00	830.64	877.46	871.18	871.18
Member 3					
Axial Force (K)	--	60.00C	60.00C	60.00C	--
Moment (K-in.)	--	1737.99	1820.86	1816.46	--
Moment of Inertia ($in.^4$)	1540.00	992.75	1038.58	1034.65	1034.65

Execution Time = 4.44 sec.

Table 5 - Solution Sequence, Problem 4

	Initial Design	Revisions 1	Revisions 2	Final Design
Member 1				
Axial Force (K)	--	60.00C	60.00C	--
Moment (K-in.)	--	1714.87	1723.17	--
Moment of Inertia ($in.^4$)	1540.00	979.10	981.54	981.54
Member 2				
Axial Force (K)	--	58.98C	59.11C	--
Moment (K-in.)	--	1714.87	1723.17	--
Moment of Inertia ($in.^4$)	1540.00	975.30	978.29	978.29
Member 3				
Axial Force (K)	--	60.00C	60.00C	--
Moment (K-in.)	--	1714.87	1723.18	--
Moment of Inertia ($in.^4$)	1540.00	979.13	981.94	981.94

Execution Time = 2.96 sec.

Table 6 - Solution Sequence, Problem 5

	Initial Design	Revisions 1	2	3	4	5	Final Design
Member 1							
Axial Force (K)	--	7.39C	7.78C	7.96C	8.06C	8.13C	--
Moment (K-in.)	--	1064.62	1197.52	1261.26	1297.12	1319.21	--
Moment of Inertia ($in.^4$)	802.00	475.65	539.15	570.47	588.66	597.79	597.79
Member 1							
Axial Force (K)	--	5.00T	5.00T	5.00T	5.00T	5.00T	--
Moment (K-in.)	--	338.03	257.12	218.34	196.50	183.06	--
Moment of Inertia ($in.^4$)	706.00	144.77	110.28	93.97	84.85	78.69	78.69
Member 3							
Axial Force (K)	--	2.01C	1.53C	1.30C	1.17C	1.09C	--
Moment (K-in.)	--	0.00	0.00	0.00	0.00	0.00	--
Moment of Inertia ($in.^4$)	802.00	52.37	52.44	52.48	52.48	52.11	52.11

Execution Time = 5.25 sec.

Table 7 - Steel Sections Chosen

Problem	Member	Section	Moment of Inertia (in.^{4})
1	1	W12 x 16.5	105
	2	W14 x 26	244
	3	W12 x 19	130
2	1	W24 x 55	1340
	2	W21 x 44	843
	3	W24 x 55	1340
3	1	W21 x 55	1140
	2	W21 x 49	971
	3	W21 x 55	1140
4	1	W21 x 55	1140
	2	W21 x 55	1140
	3	W21 x 55	1140
5	1	W18 x 40	612
	2	W12 x 14	88
	3	W10 x 11.5	52

Table 8 - Solution Check, Problem 1

	Member 1	Member 2	Member 3
Axial Force, P(K)	$22.54^{K}C$	$12.31^{K}C$	$31.24^{K}C$
Moment, M(K-in.)	190.03	529.97	191.92
Moment of Inertia, I(in^4)	99.89	240.28	111.55
Section Modulus, S(in^3)	10.3	24.1	11.4
Cross Sectional Area, A(in^2)	4.6	7.2	4.9
Radius of Gyration, r(in)	4.6	5.8	4.8
Member Length, L(in)	180.00	168.00	180.00
Axial Stress, $f_a = \frac{P}{A}$	4.9	1.71	6.38
Bending Stress, $f_b = \frac{M}{S}$	18.45	21.99	16.84
$C_c = \frac{2\pi^2 E}{F_y}$	128.26	128.26	128.26
$\frac{KL}{r}$ (K = 1 assumption)	39.13	28.97	37.50
Allowable Axial Stress: $F_a = \dfrac{\left[1 - \frac{(\frac{KL}{r})^2}{2C_c^2}\right] F_y}{\frac{5}{3} + \frac{3(\frac{KL}{r})}{8C_c} - \frac{(\frac{KL}{r})^3}{8C_c^3}}$	19.31	20.047	19.43
Allowable Bending Stress: $F_b = 0.66\ F_y$	24	24	24

Table 8 - Solution Check, Problem 1 (continued)

	Member 1	Member 2	Member 3
$\frac{f_a}{F_a}$	0.25>0.15	0.08<0.15	0.33>0.15
$\frac{f_a}{F_a} + \frac{f_b}{F_b}$	--	1.002	--
$F_e' = \frac{12\pi^2 E}{23(\frac{KL}{r})^2}$	100.89	--	109.85
$C_m = 0.85$ (Assumed)			
$\frac{f_a}{F_a} + \frac{C_m f_b}{(1 - \frac{f_a}{F_e'})F_b}$	0.9405	--	0.9616
$\frac{f_a}{0.6F_a} + \frac{f_b}{F_b}$	0.996	--	0.997

same as those used in the Fully Stressed Design routine (FSD), the answers should have been the same for both methods. The FSD solution, however, was 8.2% higher than that of the penalty function method, with an error of as much as 25% for specific members.

If one studies the results and values of the constraints at the solution, it is apparent that the FSD design is not fully stressed. Therefore, the difference is caused by the FSD routine.

The penalty function method took approximately four times more execution time than the FSD method. This would be the most extreme difference because all members were loaded in compression. When that is the case, the constraints are much more complicated and they take more time to compute. Also, it should be noted that the penalty function design progressed further toward convergence.

The penalty function method requires less execution time if the initial design point is close to the solution point. If it is too close, though, there is a chance of getting a premature convergence. When this happens, the unconstrained minimization routine is used only once, since the solution is within the convergence criteria. The same convergence criteria is used for the main program, so these values will be printed as the final solution. Therefore, to be assured of an accurate solution, at least two unconstrained minimizations must be made with a subsequent reanalysis.

Occasionally, when the $\hat{r}$ value is small, the VA13A routine will diverge. To counter this, a statement has been inserted to limit to 100 the number of times the function and gradient values are called. When this happens, the minimum value from the previous step will be returned. The program will continue and a final solution will be obtained.

CHAPTER 6

CONCLUSION

It has been shown that the extended interior penalty function approach, as applied in this paper, does give optimum designs for the given constraints, in a reasonable amount of execution time. The method's advantages are:

i) Any initial design, feasible or infeasible, may be chosen.

ii) It is simple to use and interpret.

iii) Any change in the constraints may be made with the addition or deletion of only a few computer statements.

iv) The efficiency will improve if a more efficient unconstrained minimization routine is found. The new routine would be easy to implement.

v) The constraints include all AISC requirements and there are no restrictions on loading conditions.

A disadvantage is that quite often the solution for a particular member will converge from below, with intermediate designs being infeasible. Consequently, choosing heavier rather than lighter steel sections is recommended.

Compared with the Fully Stressed Design method, the penalty function approach gives better answers since any restrictions can be implemented as constraints. It does take more computer time, however, with time increasing exponentially with an increase in design variables [3]. Consequently, the method might be best suited for relatively small structures and designs that would be used more than once.

Kirsch and Shamir [16] discovered that much computer time can be saved when optimizing large structures by optimizing a series of substructures. Further research could be done using this program on larger structures, possibly incorporating this idea.

BIBLIOGRAPHY

[1] American Institute of Steel Construction, Manual of Steel Construction, 7th edition, 1973.

[2] Bracken, J. and McCormick, G. P., Selected Applications of Nonlinear Programming, Wiley, New York, 1968, pp. 50 - 57.

[3] Bromowicki, A. J. and Felton, L. P., "Optimum Design of Continuous Thin Walled Beams," International Journal for Numerical Methods in Engineering, Vol. 9, No. 3, 1975, pp. 711 - 720.

[4] 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, Dec. 1966, pp. 319 - 340.

[5] Cassis, J. H. and Schmit, L. A. Jr., "On Implementation of the Extended Interior Penalty Function," International Journal for Numerical Methods in Engineering, Vol. 10, No. 1, 1976, pp. 3 - 23.

[6] DeSilva, B. M. E. and Grant, G. N. C., "Comparison of Some Penalty Function Based Optimization Procedures for the Synthesis of a Planar Truss," International Journal for Numerical Methods in Engineering, Vol. 7, No. 2, 1973, pp. 155 - 173.

[7] Felton, L. P. and Nelson, R. B., "Optimized Components in Frame Synthesis," AIAA Journal, Vol. 9, 1971, pp. 1027 - 1031.

[8] Fiacco, A. V. and McCormick, G. P., Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968.

[9] Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," Computer Journal, Vol. 6, iss. 2, 1963, pp. 163 - 168.

[10] Fox, R. L., Optimization Methods for Engineering Design, Addison-Wesley Publishing Co., 1971, pp. 137 - 150.

[11] Gallagher, R. H. and Zienkiewicz, O. C., Optimum Structural Design, Theory and Applications, Wiley, 1973, pp. 143 - 177.

[12] Goble, G. C. and Moses, F., "Practical Applications of Structural Optimization," Journal of the Structural Division, ASCE, Vol. 101, No. ST4, Proc. Paper 11267, Apr. 1975, pp. 635 - 648.

[13] Johnston, B. G. and Lin, F. J., Basic Steel Design, Prentice-Hall, Englewood Cliffs, N. J., 1974.

[14] 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, No. 4, Oct. - Dec., 1969, pp. 351 - 378.

[15] Kavlie, D. and Moe, J., "Automated Design of Frame Structures," Journal of the Structural Division, ASCE, Vol. 97, No. ST1, Jan., 1971, pp. 33 - 62.

[16] Kirsch, U., Reris, M. and Shamir, U., "Optimum Design by Partitioning into Substructures," Journal of the Structural Division, ASCE, Vol. 98, No. ST1, Proc. Paper 8645, Jan., 1972, pp. 249 - 267.

[17] Lootsma, F. A., "A Survey of Methods for Solving Constrained Minimization Problems Via Unconstrained Minimization," Numerical Methods for Nonlinear Optimization (Ed. Lootsma, F. A.) Academic Press, 1972.

[18] Majumder, D. K. and Thornton, W. K., "Design of Efficient Stiffened Shells of Revolution," International Journal for Numerical Methods in Engineering, Vol. 10, No. 3, 1976, pp. 535 - 549.

[19] Moe, J., "Fundamentals of Optimization," Computers and Structures, Vol. 4, No. 1, Jan., 1974, pp. 95 - 113.

[20] Moses, F., Fox, R. L., and Goble, G., "Mathematical Programming Applications in Structural Design," Computer-Aided Engineering (Ed. Gladwell, G. L. M.), S. M. Study No. 5, Univ. of Waterloo Press, 1971, pp. 379 - 393.

[21] Nelson, R. B. and Felton, L. P., "Thin-Walled Beams in Frame Synthesis," AIAA Journal, Vol. 10, No. 12, Dec. 1972, pp. 1565 - 1569.

[22] Osborne, M. R., "On Penalty and Barrier Functions in Mathematical Programming," Optimization (Ed. Anderssen, R. S., Jennings, L. S., and Ryan, D. M.), Univ. of Queensland Press, 1972, pp. 106 - 115.

[23] Osborne, M. R. and Ryan, D. M., "On Penalty Function Methods for Nonlinear Programming Problems," Journal of Mathematical Analysis and Applications, Vol. 31, No. 3, Sept. 1970, pp. 559 - 578.

[24] Osborne, M. R. and Ryan, D. M. "A Hybrid Algorithm for Nonlinear Programming," Numerical Methods for Nonlinear Optimization, (Ed. Lootsma, F. A.), Academic Press, 1972.

[25] Patnaik, S. N. and Sankaran, G. V., "Optimum Design of Stiffened Cylindrical Panels with Constraint on the Frequency in the Presence of Initial Stresses," International Journal for Numerical Methods in Engineering, Vol. 10, No. 2, 1976, pp. 283 - 299.

[26] Pickett, R. M. Rubinstein, M. F., and Nelson, R. B., "Automated Structural Synthesis using a Reduced Number of Design Coordinates," AIAA Journal, Vol. 11, No. 4, April, 1973, pp. 489 - 494.

[27] Powell, M. J. D., Report C. S. S. 15, A. E. R. E. Harwell, 1975.

[28] Ramakrishnan, C. V. and Francavilla, A., "Structural Shape Optimization using Penalty Functions," Journal of Structural Mechanics, Vol. 3, No. 4, 1974 - 1975, pp. 403 - 422.

[29] Ryan, D. M. "Penalty and Barrier Functions," Numerical Methods for Constrained Optimization (Ed. Gill, P. E. and Murray, W.), Academic Press, 1974.

[30] Shamie, J. and Schmit, Jr., L. A., "Frame Optimization Including Frequency Constraints," Journal of the Structural Division, ASCE, Vol. 101, No. ST1, Proc. Paper 11075, Jan., 1975, pp. 283 - 293.

APPENDIX A

CONSTRAINTS USING AISC NOTATION

Condition	Formula	Constraint
Moment of Inertia should not go below that of the smallest or above that of the largest section available in the "allowable Stress Design Section Table"	$52 \leq I \leq 20{,}300$	$GG1 = \dfrac{1}{\frac{I}{52} - 1}$ $GG2 = \dfrac{1}{\frac{20{,}300}{I} - 1}$
$\frac{KL}{r}$ should not go above 200. K is assumed to be 1.0 to be conservative	$\frac{KL}{r} \leq 200$	$GG3 = \dfrac{1}{\frac{200}{\frac{KL}{r}} - 1}$

Interaction Formulas:

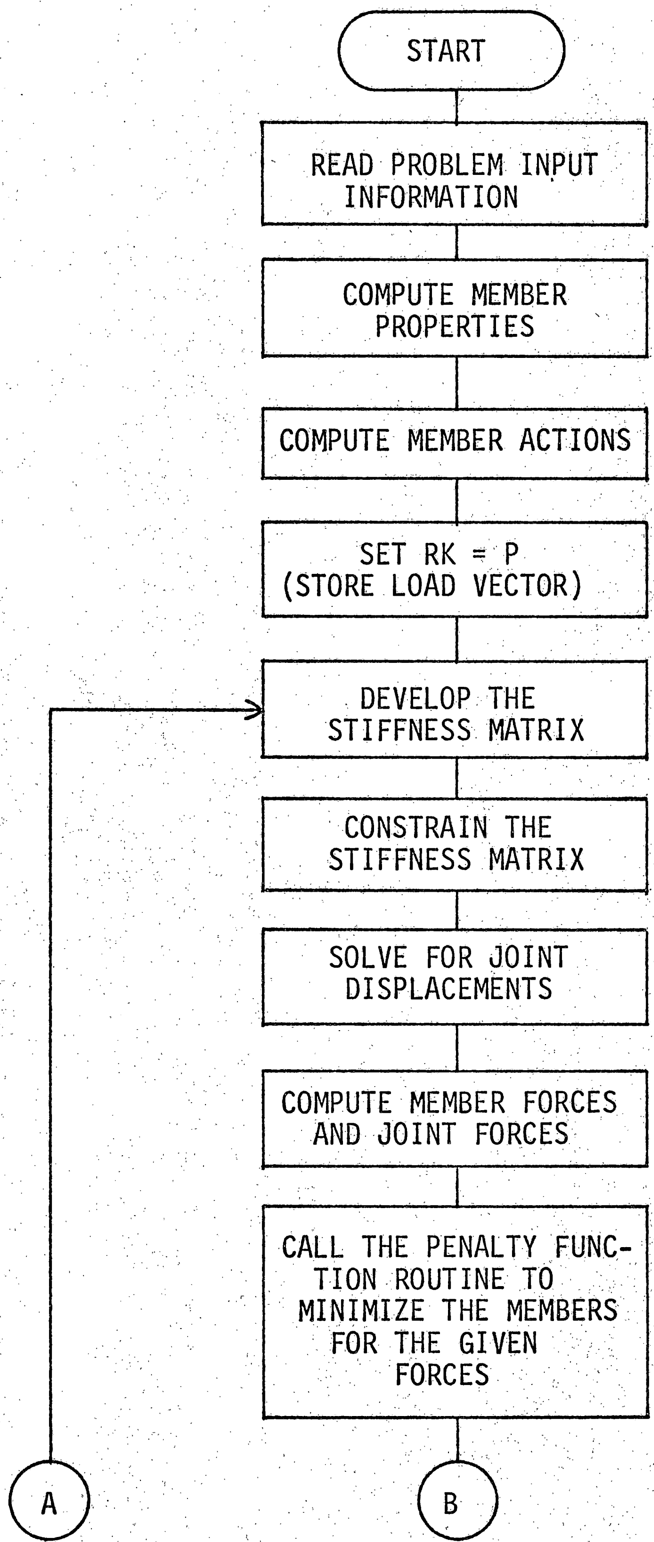
The allowable axial stress F_a and the allowable bending stress, F_b must be found for each particular case.	$C_c = \dfrac{2\pi^2 E}{F_y}$	
For Compression, 1) If $\frac{KL}{r} < C_c$	$F_a = \dfrac{\left[1 - \dfrac{(\frac{KL}{r})^2}{2C_c^2}\right] F_y}{\frac{5}{3} + \dfrac{3(\frac{KL}{r})}{8C_c} - \dfrac{(\frac{KL}{r})^3}{8C_c^3}}$	
2) If $\frac{KL}{r} \geq C_c$	$F_a = \dfrac{12\pi^2 E}{23(\frac{KL}{r})^2}$	
For Tension,	$F_a = 0.60\ F_y$	
In bending, all sections are assumed to be compact	$F_b = 0.66\ F_y$	

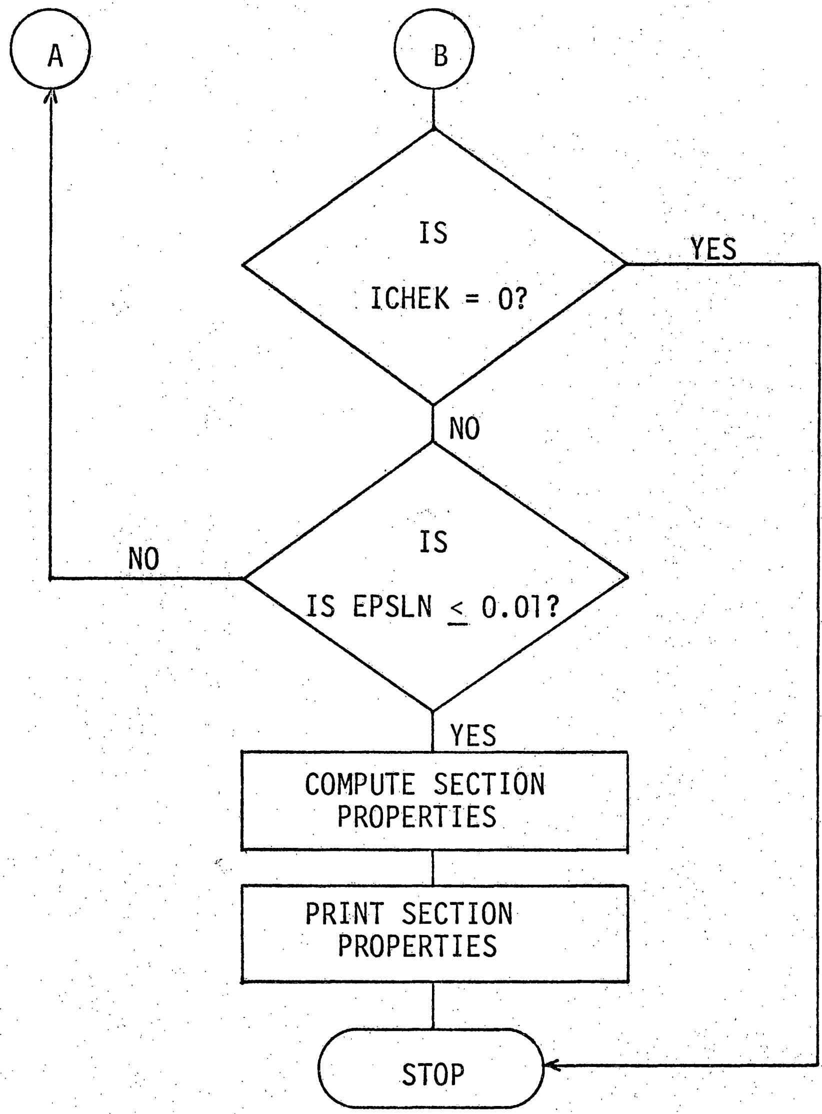
Condition	Formula	Constraint
The bending stress from the loads, f_b, and the axial stress from the loads, f_a, must be found	$F_b = \frac{M}{S}$ $f_a = \frac{P}{A}$	
There are three interaction formulas:		
If there is compression, and $\frac{f_a}{F_a} > 0.15$;	$F'_e = \frac{12\pi^2 E}{23(\frac{KL}{r})^2}$	
$C_m = 0.85$ to be conservative	$\frac{f_a}{F_a} + \frac{C_m f_b}{(1 - \frac{f_a}{F'_e})\ F_b} \leq 1$	$GG4 = \frac{1}{1 - [\frac{f_a}{F_a} + \frac{C_m F_b}{(1-\frac{f_a}{Fe})F_b}]}$
	and $\frac{f_a}{0.6F_y} + \frac{f_b}{F_b} \leq 1$	$GG5 = \frac{1}{1 - [\frac{f_a}{0.6F_y} + \frac{f_b}{F_b}]}$
If there is tension or if $\frac{f_a}{F_a} \leq 0.15$	$\frac{f_a}{F_a} + \frac{f_b}{F_b} \leq 1$	$GG6 = \frac{1}{1 - [\frac{f_a}{F_a} + \frac{f_b}{F_b}]}$

APPENDIX B

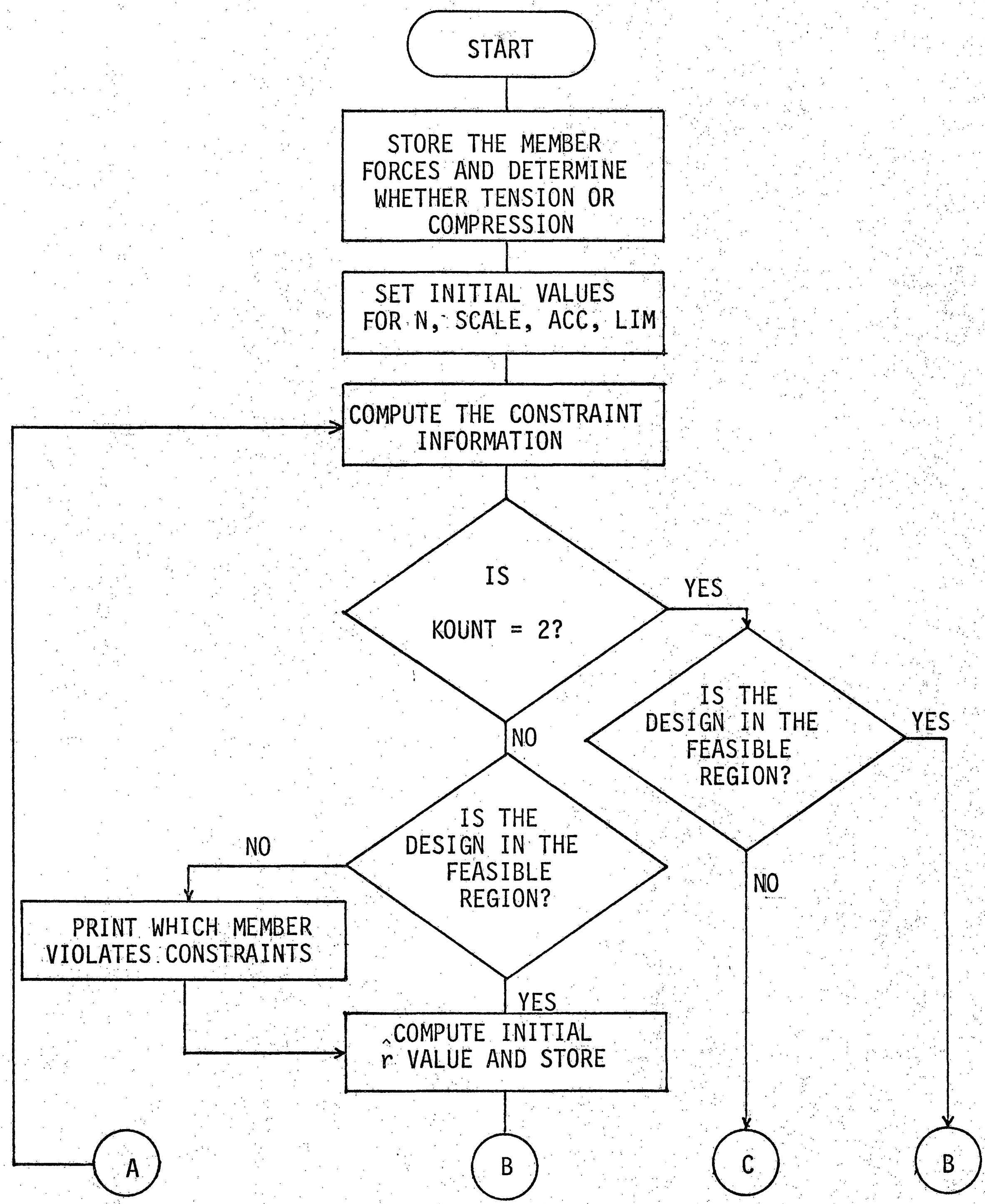
FLOW CHARTS

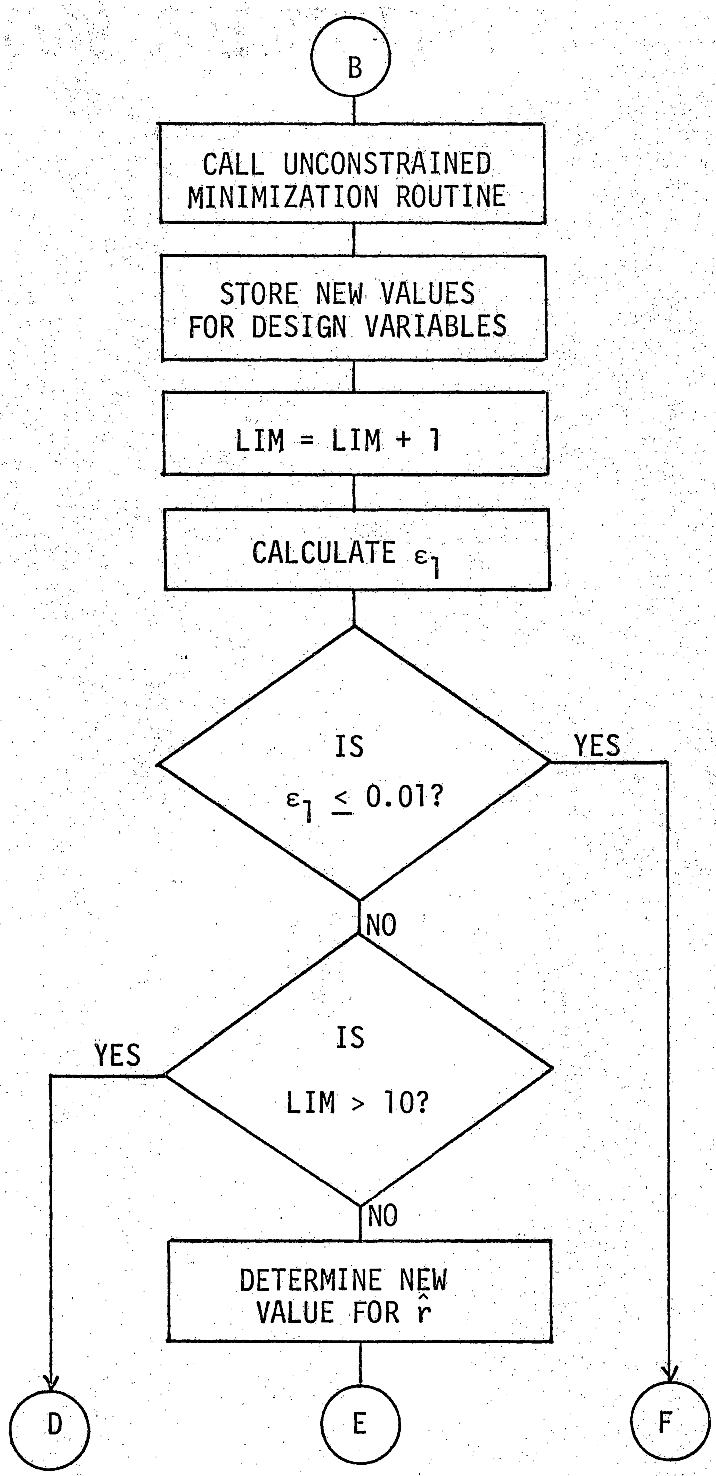
Main Routine

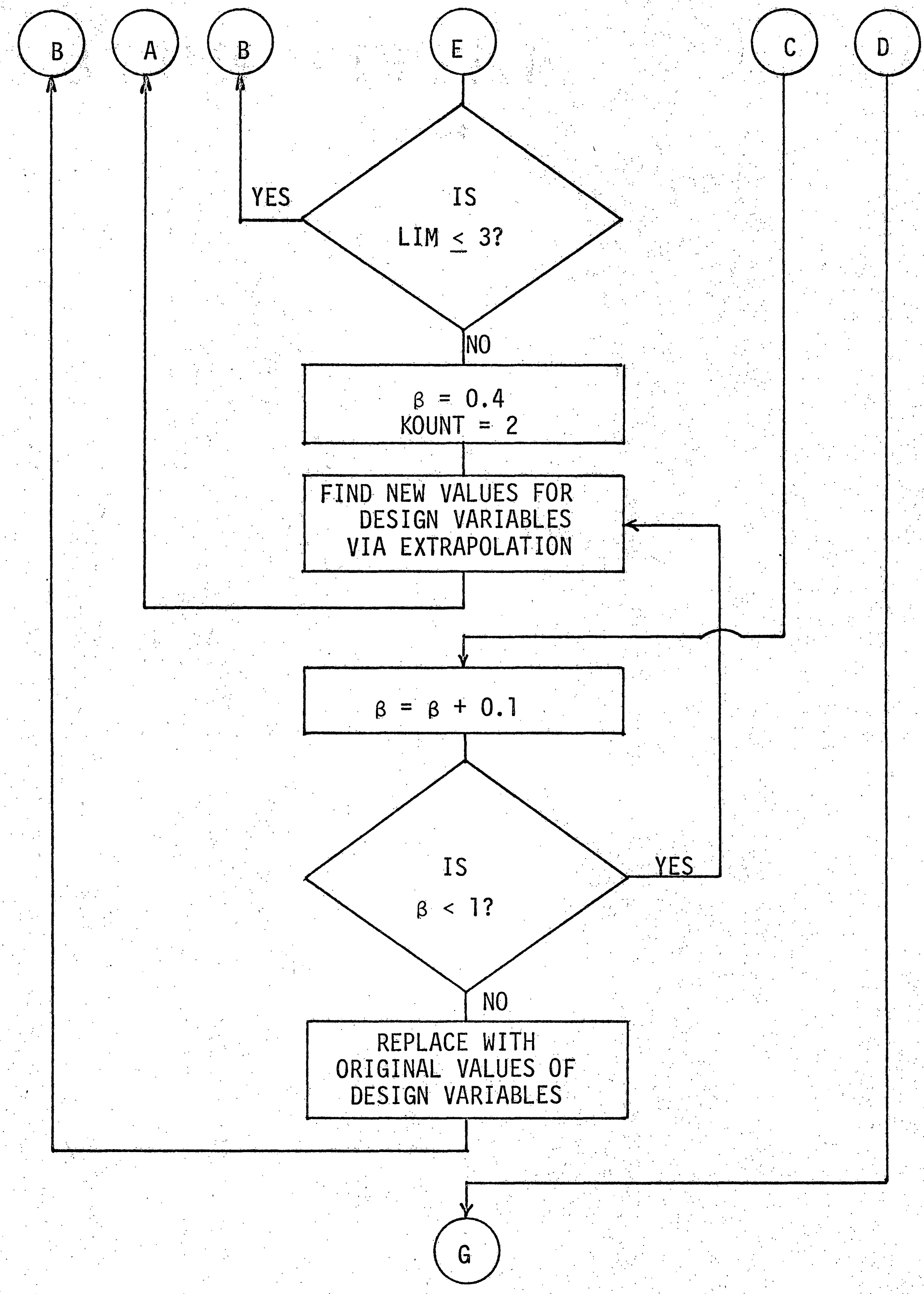


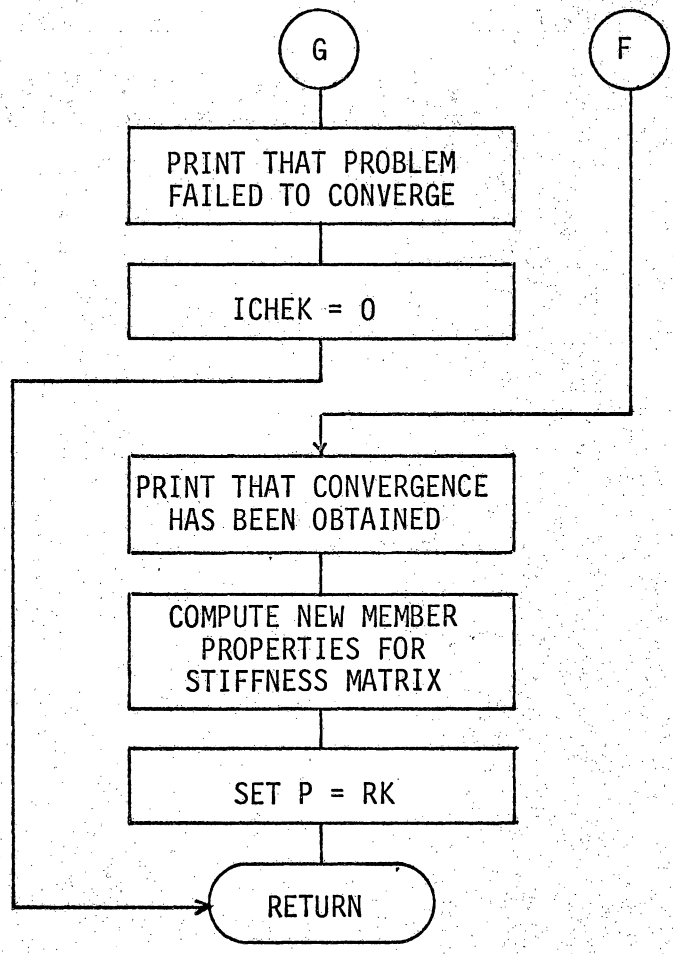

A
B
IS
ICHEK = 0?
YES
NO
NO
IS
IS EPSLN ≤ 0.01?
YES
COMPUTE SECTION PROPERTIES
PRINT SECTION PROPERTIES
STOP

Penalty Function Routine

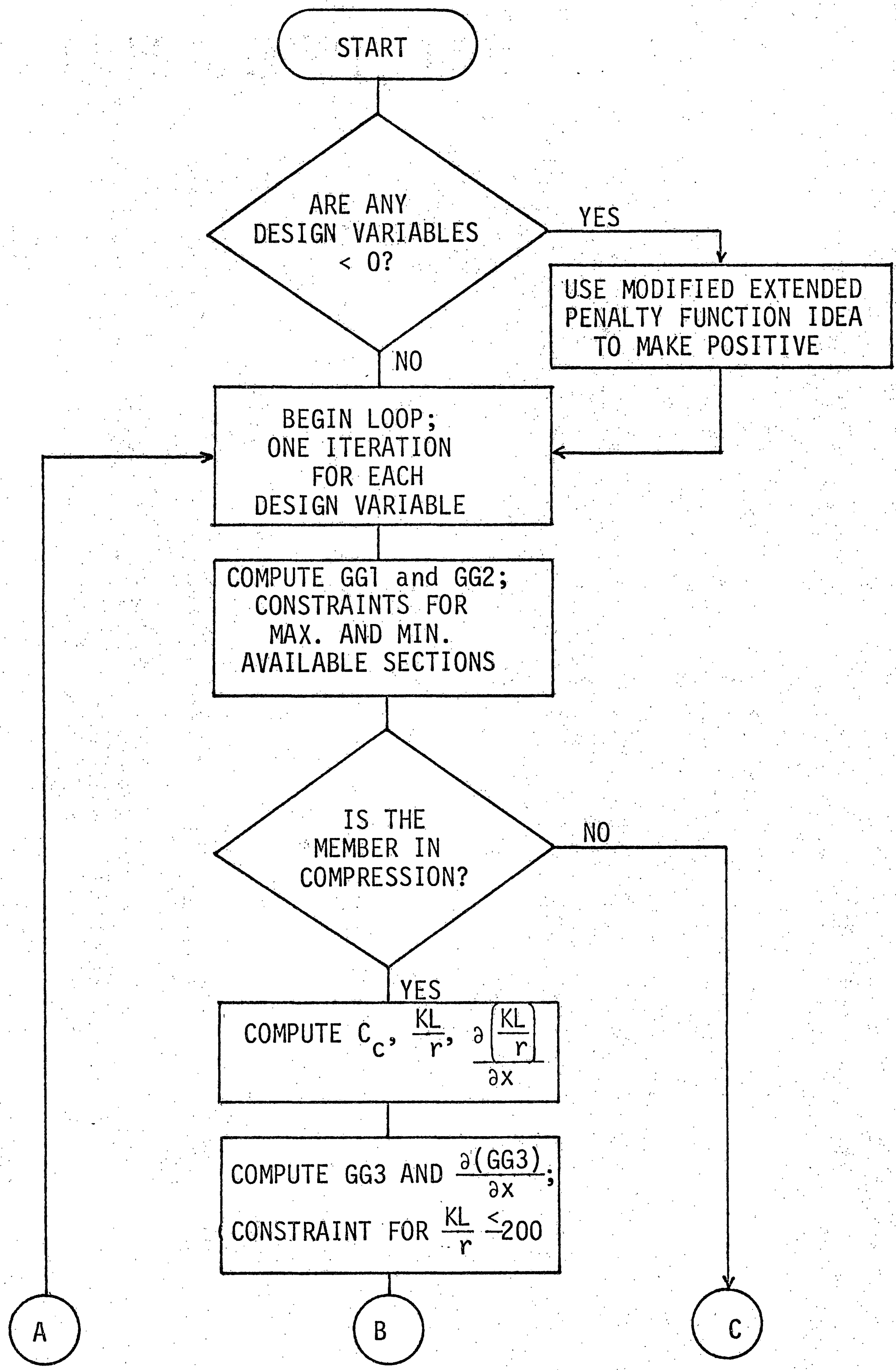


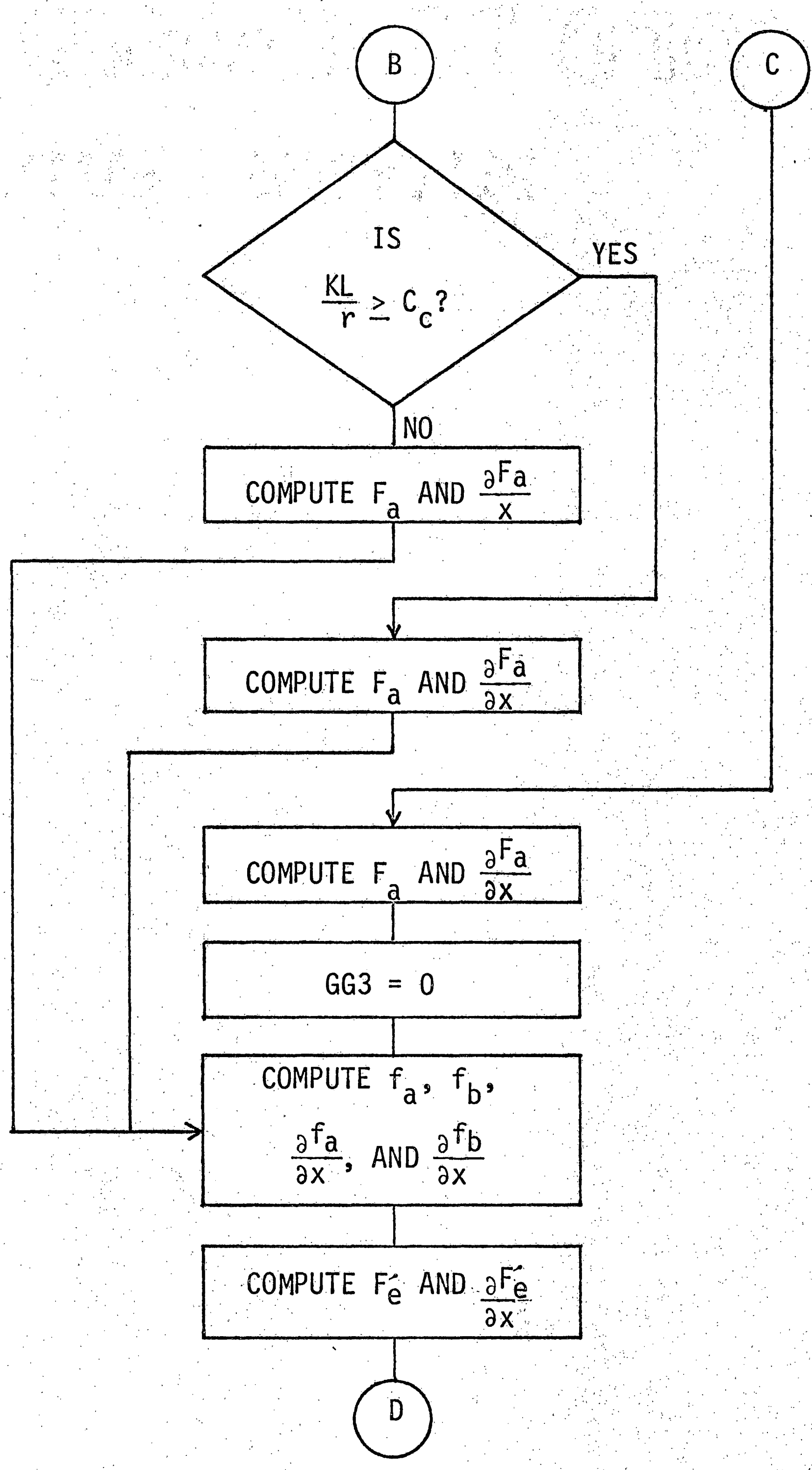

B
CALL UNCONSTRAINED MINIMIZATION ROUTINE
STORE NEW VALUES FOR DESIGN VARIABLES
LIM = LIM + 1
CALCULATE ε_1
IS $\varepsilon_1 \leq 0.01$?
YES
NO
IS LIM > 10?
YES
NO
DETERMINE NEW VALUE FOR $\hat{r}$
D
E
F

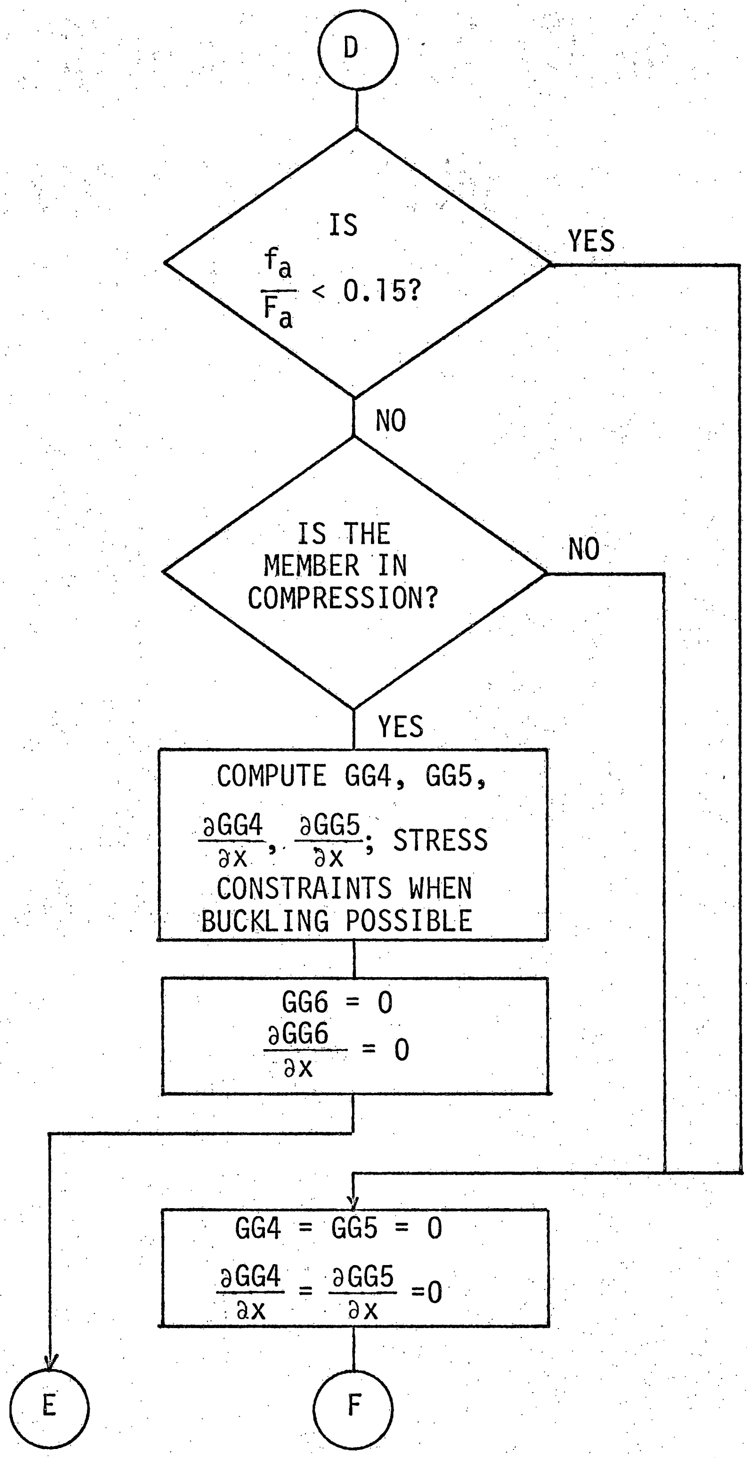

B
A
B
E
C
D
YES
IS
LIM ≤ 3?
NO
β = 0.4
KOUNT = 2
FIND NEW VALUES FOR
DESIGN VARIABLES
VIA EXTRAPOLATION
β = β + 0.1
IS
β < 1?
YES
NO
REPLACE WITH
ORIGINAL VALUES OF
DESIGN VARIABLES
G

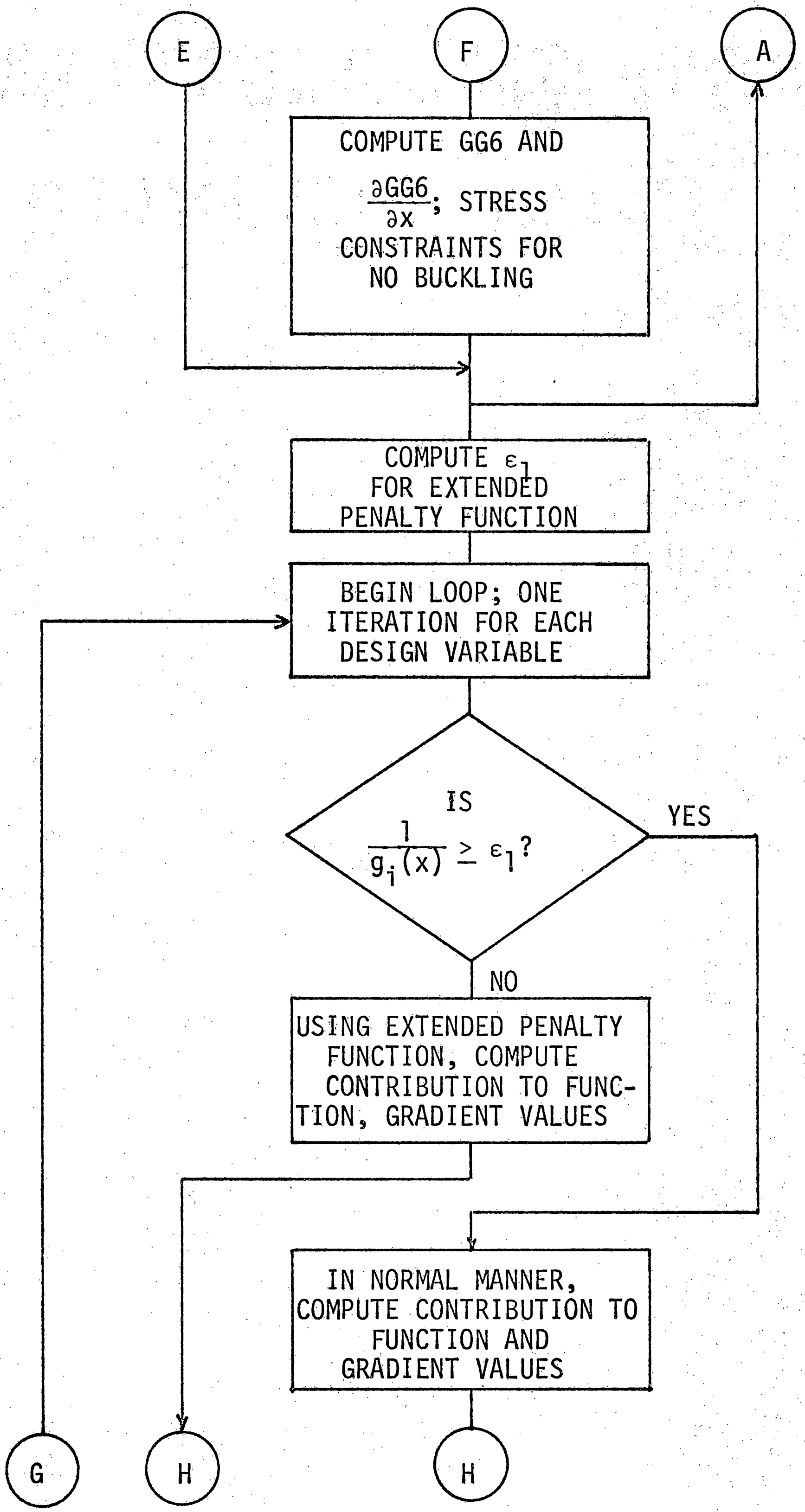

G
F
PRINT THAT PROBLEM FAILED TO CONVERGE
ICHEK = 0
PRINT THAT CONVERGENCE HAS BEEN OBTAINED
COMPUTE NEW MEMBER PROPERTIES FOR STIFFNESS MATRIX
SET P = RK
RETURN

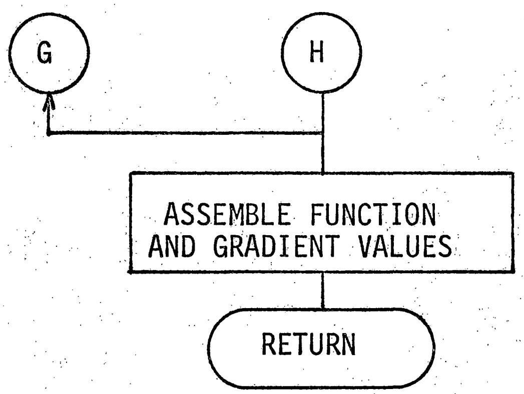
Function and Gradient Value Routine




B
C
IS
$\frac{KL}{r} \geq C_c$?
YES
NO
COMPUTE F_a AND $\frac{\partial F_a}{x}$
COMPUTE F_a AND $\frac{\partial F_a}{\partial x}$
COMPUTE F_a AND $\frac{\partial F_a}{\partial x}$
GG3 = 0
COMPUTE f_a, f_b, $\frac{\partial f_a}{\partial x}$, AND $\frac{\partial f_b}{\partial x}$
COMPUTE F_e' AND $\frac{\partial F_e'}{\partial x}$
D


D
IS $\frac{f_a}{F_a} < 0.15$?
YES
NO
IS THE MEMBER IN COMPRESSION?
NO
YES
COMPUTE GG4, GG5, $\frac{\partial GG4}{\partial x}$, $\frac{\partial GG5}{\partial x}$; STRESS CONSTRAINTS WHEN BUCKLING POSSIBLE
GG6 = 0
$\frac{\partial GG6}{\partial x} = 0$
GG4 = GG5 = 0
$\frac{\partial GG4}{\partial x} = \frac{\partial GG5}{\partial x} = 0$
E
F


E
F
A
COMPUTE GG6 AND $\frac{\partial GG6}{\partial x}$; STRESS CONSTRAINTS FOR NO BUCKLING
COMPUTE ε_1 FOR EXTENDED PENALTY FUNCTION
BEGIN LOOP; ONE ITERATION FOR EACH DESIGN VARIABLE
IS $\frac{1}{g_i(x)} \geq \varepsilon_1$?
YES
NO
USING EXTENDED PENALTY FUNCTION, COMPUTE CONTRIBUTION TO FUNCTION, GRADIENT VALUES
IN NORMAL MANNER, COMPUTE CONTRIBUTION TO FUNCTION AND GRADIENT VALUES
G
H
H


G
H
ASSEMBLE FUNCTION AND GRADIENT VALUES
RETURN

APPENDIX C

COMPUTER PROGRAM LISTING

```
C
C  THIS PROGRAM ANALYZES PLANE FRAMES AND THEN USES THE EX-
C  TENDED INTERIOR PENALTY FUNCTION METHOD TO OBTAIN A MINIMUM WEIGHT
C  DESIGN FOR STEEL, WIDE FLANGE MEMBERS.
C
C
C     WRITTEN BY: W. F. COFER
C     NOTE THAT UNITS ARE INCHES, KIPS, AND RADIANS.
C     VARIABLE           MEANING
C     INC             MEMBER INCIDENCE MATRIX
C     MBAND              SEMI-BANDWIDTH
C     NJ              NUMBER OF JOINTS
C     NM              NUMBER OF MEMBERS
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION S(1500), N(75)
      WRITE (6,3)
      READ 4, NM,NJ
      WRITE (6,5) NM,NJ
      WRITE (6,6)
      DO 1 I=1,NM
      READ 4, INC(I,1),INC(I,2)
      WRITE (6,7) I,INC(I,1),INC(I,2)
1     CONTINUE
C
C     CALCULATE SEMI-BANDWIDTH (MBAND).
C
      MDIF=0
      DO 2 I=1,NM
      J=INC(I,1)
      K=INC(I,2)
      IDIFO=J-K
```

```
      IDIF=IABS(IDIFO)                                                  DUMY  33
      IF (MDIF.LT.IDIF) MDIF=IDIF                                       DUMY  34
    2 CONTINUE                                                          DUMY  35
      MBAND=3*(MDIF+1)                                                  DUMY  36
      NJ3=NJ*3                                                          DUMY  37
C                                                                       DUMY  38
C     DETERMINE STORAGE LOCATIONS IN THE MAIN ARRAY.                    DUMY  39
C                                                                       DUMY  40
      N1=1                                                              DUMY  41
      N2=1+NJ3                                                          DUMY  42
      N3=N2+NJ*2                                                        DUMY  43
      N4=N3+NM                                                          DUMY  44
      N5=N4+NM                                                          DUMY  45
      N6=N5+NM                                                          DUMY  46
      N7=N6+NM                                                          DUMY  47
      N8=N7+NM                                                          DUMY  48
      N9=N8+NM                                                          DUMY  49
      N10=N9+NM                                                         DUMY  50
      N11=N10+NM                                                        DUMY  51
      N12=1                                                             DUMY  52
      N13=N11+MBAND*NJ3                                                 DUMY  53
      N14=N13+NJ3                                                       DUMY  54
      N15=N14+NM*6                                                      DUMY  55
      N16=N15+NM*6                                                      DUMY  56
      N17=N16+NM                                                        DUMY  57
      N18=N17+NM*10                                                     DUMY  58
      N20=N12+NJ3                                                       DUMY  59
      N21=N18+NM                                                        DUMY  60
      N22=N21+NM                                                        DUMY  61
      N23=N22+NM                                                        DUMY  62
      N24=N23+NM                                                        DUMY  63
      N25=N24+NM                                                        DUMY  64
```

```
      N26=N25+NM                                                        DUMY
      N27=N26+NM                                                        DUMY
      N28=N27+NM                                                        DUMY
      N29=N28+NM                                                        DUMY
      N30=N29+NM                                                        DUMY
      N31=N30+NM                                                        DUMY
      N32=N31+NM                                                        DUMY
      N33=N32+NM                                                        DUMY
      N34=N33+NM                                                        DUMY
      N35=N34+NM                                                        DUMY
      N36=N35+NM                                                        DUMY
      N37=N36+NM                                                        DUMY
      N38=N37+NM                                                        DUMY
      CALL MAIN1 (S(N1),S(N2),S(N3),S(N4),S(N5),S(N6),S(N7),S(N8),S(N9),DUMY
     1S(N10),S(N11),N(N12),S(N13),S(N14),S(N15),S(N16),S(N17),S(N18),N(NDUMY
     220),S(N21),S(N22),S(N23),S(N24),S(N25),S(N26),S(N27),S(N28),S(N29)DUMY
     3,S(N30),S(N31),S(N32),S(N33),S(N34),S(N35),S(N36),S(N37),S(N38))  DUMY
      STOP                                                              DUMY
C                                                                       DUMY
3     FORMAT (1H1,39HUNITS ARE IN INCHES, KIPS, AND RADIANS.)           DUMY
4     FORMAT (T1,I2,T3,I2)                                              DUMY
5     FORMAT (1H0,18HTHIS STRUCTURE HAS,I3,12H MEMBERS AND,I3,8H JOINTS.DUMY
     1,/)                                                               DUMY
6     FORMAT (1H0,10X,6HMEMBER,10X,10HFROM JOINT,9X,8HTO JOINT,/)       DUMY
7     FORMAT (1H ,11X,I3,15X,I3,13X,I3)                                 DUMY
      END                                                               DUMY
```

```
      SUBROUTINE MAIN1 (P,X,XI,E,A,XL,C,S,ALPH,BETA,AK,ICON,RK,FKI,FTIL,
     1SCALE,XI1,G,IND,AXIAL,BEND,W1,W2,W3,W4,W5,W6,GG1,GG2,GG3,GG4,GG5,G
     2G6,DG3,DG4,DG5,DG6)
C
C     VARIABLE          MEANING
C     A,AI             MEMBER CROSS-SECTIONAL AREA
C     AS              SECTION MODULUS
C     AW              WEB AREA
C     E               YOUNG'S MODULUS
C     FY              YIELD STRENGTH OF THE MATERIAL
C     ICON             CONSTRAINT MATRIX; 1 MEANS CONSTRAINED, 0 MEANS
C                     UNCONSTRAINED.
C     LIM1             COUNTER TO LIMIT THE NUMBER OF ITERATIONS.
C     RK              DUMMY ARRAY FOR STORING THE LOAD VECTOR FOR THE NEXT
C                     ANALYSIS.
C     STORI             ARRAY TO STORE THE MOMENTS OF INERTIA FROM EACH
C                     OPTIMIZATION.
C     X               JOINT COORDINATE MATRIX
C     XI              MEMBER MOMENT OF INERTIA
C     XR              RADIUS OF GYRATION
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION P(NJ3), X(NJ,2), XI(NM), E(NM), A(NM), XL(NM), C(NM), S(
     1NM), ALPH(NM), BETA(NM), AK(NJ3,MBAND), ICON(NJ3), RK(NJ3), FKI(NM
     2,6)
      DIMENSION XI1(NM,10), FTIL(NM,6), SCALE(NM), G(NM), IND(NM), AXIAL
     1(NM), BEND(NM), W1(NM), W2(NM), W3(NM), W4(NM), W5(NM), W6(NM), GG
     21(NM), GG2(NM), GG3(NM), GG4(NM), GG5(NM), GG6(NM), DG3(NM), DG4(N
     3M), DG5(NM), DG6(NM)
C
C     THE INPUT SUBPROGRAM IS USED TO READ AND PRINT ALL DATA.
C
```

```
      CALL INPUT (X,A,XI,E,ICON,P)
C
C     THE MPROP SUBPROGRAM COMPUTES THE MEMBER PROPERTIES SUCH AS LENGTH
C     COSINE, SINE, ALPHA, AND BETA.
C
      CALL MPROP (X,C,S,E,XI,A,XL,ALPH,BETA)
C
C     THE MACT SUBROUTINE COMPUTES MEMBER ACTIONS.
C
      CALL MACT (XL,E,A,XI,C,S,P,ICON,FTIL,RK)
      DO 1 I=1,NJ3
      RK(I)=P(I)
1     CONTINUE
      LIM1=1
      DO 2 I=1,NM
      STORI(1,I)=XI(I)
2     CONTINUE
3     CONTINUE
C
C     THE STIFF SUBPROGRAM GENERATES THE SYSTEM STIFFNESS MATRIX IN GLOBAL
C     COORDINATES.  IT TAKES ADVANTAGE OF BANDEDNESS AND IS STORED WITH THE
C     DIAGONAL IN THE FIRST COLUMN.
C
      CALL STIFF (C,S,ALPH,BETA,XL,AK)
C
C     THE CONSTR SUBPROGRAM INTRODUCES CONSTRAINTS INTO THE SYSTEM STIFFNESS
C     MATRIX.  THE SOLVE SUBPROGRAM SOLVES FOR THE JOINT DISPLACEMENTS.
C
      CALL CONSTR (ICON,AK)
      CALL SOLVE (AK,P)
C
C     THE FORCE SUBPROGRAM TAKES THE DISPLACEMENTS AND SOLVES FOR THE MEMBER
```

```
C     AND JOINT FORCES.
C
      CALL FORCE (P,C,S,ALPH,BETA,XL,FKI,FTIL)
C
C     THE PENFUN SUBROUTINE OPTIMIZES THE MEMBERS FOR THE GIVEN LOADS.
C
      CALL PENFUN (XI,XL,E,FKI,ICHEK,RK,P,ALPH,BETA,XI1,G,IND,AXIAL,BEND
     1,W1,W2,W3,W4,W5,W6,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6)
      LIM1=LIM1+1
C
C     ICHEK=0 MEANS THAT THE SOLUTION FROM PENFUN DID NOT CONVERGE.
C
      IF (ICHEK.EQ.0) GO TO 10
      DO 4 I=1,NM
      STORI(LIM1,I)=XI(I)
4     CONTINUE
C
C     CHECK FOR CONVERGENCE.
C
      SUMEP1=0.
      SUMEP2=0.
      DO 5 I=1,NM
      SUMEP1=SUMEP1+XI(I)
      SUMEP2=SUMEP2+STORI(LIM1-1,I)
5     CONTINUE
      EPS=ABS((SUMEP1-SUMEP2)/SUMEP1)
      IF (EPS.GT.0.01) GO TO 9
      WRITE (6,11)
      WRITE (6,12)
C
C     COMPUTE MEMBER PROPERTIES.
C
```

```
      DO 8 I=1,NM
      XII=XI(I)
      XLI=XL(I)
      EI=E(I)
      IF (XII.GT.9000.) GO TO 6
      AI=0.464*SQRT(XII)
      XR=(XII/0.21583)**0.25
      AS=SQRT(60.6*XII+84100.)-290.
      GO TO 7
6     AI=0.0039*(XII+2300.)
      XR=(XII+174498.)/12841.
      AS=(XII-8056.3)/1.876
7     V=ABS(FKI(I,5))
      IF (ABS(FKI(I,6)).GT.V) V=ABS(FKI(I,6))
      AW=V/(0.4*FY)
      WRITE (6,13) I,XII,AS,AI,XR,AW
8     CONTINUE
      GO TO 10
9     IF (LIM1.LT.21) GO TO 3
10    RETURN
C
11    FORMAT (1H0,10X,52HCONVERGENCE HAS NOW BEEN OBTAINED FOR THE STRUC
     1TURE.)
12    FORMAT (1H0,10X,6HMEMBER,10X,1HI,7X,1HS,7X,1HA,7X,1HR,7X,2HAW)
13    FORMAT (1H0,12X,I1,9X,F7.2,3X,F5.1,2X,F4.1,5X,F3.1,4X,F4.1)
      END
```

```
      SUBROUTINE INPUT (X,A,XI,E,ICON,P)
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION X(NJ,2), A(NM), XI(NM), E(NM), ICON(NJ3), P(NJ3)
      WRITE (6,5)
      DO 1 I=1,NJ
      READ 6, X(I,1),X(I,2)
      WRITE (6,7) I,X(I,1),X(I,2)
    1 CONTINUE
      WRITE (6,8)
      DO 2 I=1,NM
      READ 9, A(I),XI(I),E(I)
      WRITE (6,10) I,A(I),XI(I),E(I)
    2 CONTINUE
      WRITE (6,14)
      I1=0
      DO 3 I=1,NJ3,3
      READ 11, ICON(I),ICON(I+1),ICON(I+2)
      READ 12, P(I),P(I+1),P(I+2)
      I1=I1+1
      WRITE (6,13) I1,ICON(I),ICON(I+1),ICON(I+2)
    3 CONTINUE
      WRITE (6,14)
      I1=0
      DO 4 I=1,NJ3,3
      I1=I1+1
      WRITE (6,15) I1,P(I),P(I+1),P(I+2)
    4 CONTINUE
      READ 16, FY
      WRITE (6,17) FY
      RETURN
C
    5 FORMAT (1H0,10X,5HJOINT,10X,11H1-DIRECTION,8X,11H2-DIRECTION,/)
```

```
6     FORMAT (T1,F9.4,T10,F9.4)
7     FORMAT (1H ,11X,I3,12X,F8.3,12X,F8.3)
8     FORMAT (1H-,10X,6HMEMBER,7X,1HA,10X,1HI,10X,1HE,/)
9     FORMAT (3F9.3)
10    FORMAT (1H ,11X,I3,4X,F7.2,5X,F7.2,5X,F7.1)
11    FORMAT (T1,I1,T5,I1,T10,I1)
12    FORMAT (T1,F7.3,T8,F7.3,T15,F7.3)
13    FORMAT (1H ,10X,I3,10X,I1,10X,I1,10X,I1)
14    FORMAT (1H-,10X,5HJOINT,8X,1H1,10X,1H2,10X,1H3,/)
15    FORMAT (1H ,10X,I3,7X,F6.1,7X,F6.1,7X,F6.1)
16    FORMAT (T1,F5.2)
17    FORMAT (1H0,51HTHE MATERIAL IN THIS PROBLEM HAS A YIELD STRESS OF
     1,F4.1,5H KSI.)
      END
```

```
      SUBROUTINE MPROP (X,C,S,E,XI,A,XL,ALPH,BETA)
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION X(NJ,2), C(NM), S(NM), E(NM), XI(NM), A(NM), XL(NM), ALP
     1H(NM), BETA(NM)
C
C     VARIABLE       MEANING
C         J,K        DUMMY
C        DX1         LENGTH OF MEMBER IN 1-DIRECTION
C        DX2         LENGTH OF MEMBER IN 2-DIRECTION
C       XLEN         LENGTH
C        XL          LENGTH, TO BE STORED
C       C,S          COSINE, SINE
C     ALPH, BETA     OBVIOUS
C
      DO 1 I=1,NM
      J=INC(I,1)
      K=INC(I,2)
      DX1=X(K,1)-X(J,1)
      DX2=X(K,2)-X(J,2)
      XLEN=SQRT(DX1**2+DX2**2)
      XL(I)=XLEN
      C(I)=DX1/XLEN
      S(I)=DX2/XLEN
      ALPH(I)=E(I)*XI(I)/XLEN**3
      BETA(I)=A(I)*XLEN**2/XI(I)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MACT (AL,E,A,XI,C,S,P,ICON,FTIL,RK)
C
C     VARIABLE           MEANING
C     FTIL              FIXED END FORCES
C     LDTYP             LOAD TYPE
C     MNO              MEMBER NUMBER
C     W1-W4             PARAMETERS DEFINED AS FOLLOWS:
C
C     LOAD TYPE          PARAMETER        MEANING
C      1                 W1              MULTIPLE OF LENGTH OF MEMBER AT WHICH
C                                        STARTS.
C                        W2              MULTIPLE OF LENGTH OF MEMBER AT WHICH
C                                        ENDS.
C                        W3              LOAD (K/IN.)
C     2,3                W1              LOAD (K)
C                        W2              DISTANCE FROM A END OF MEMBER.
C      4                 W1              INCREASE IN TEMPERATURE (DEG. F)
C                        W2              ALPHA
C      5                 W1              MULTIPLE OF LENGTH OF MEMBER AT WHICH
C                                        STARTS.
C                        W2              MULTIPLE OF LENGTH OF MEMBER AT WHICH
C                                        ENDS.
C                        W3              INITIAL LOAD (K/IN.)
C                        W4              FINAL LOAD (K/IN.)
C      6                 W1              EXCESS LENGTH
C      7                 W1              INCREASE IN TEMPERATURE (DEG. F)
C                        W2              ALPHA
C                        W3              DEPTH OF SECTION
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION AL(NM), E(NM), A(NM), XI(NM), C(NM), S(NM), P(NJ3), ICON
     1(NJ3), FTIL(NM,6)
```

```
      DIMENSION RK(NJ3)
C
C     THE PURPOSE OF THE SUBROUTINE IS TO COMPUTE MEMBER END FORCES DUE
C     MEMBER LOADS AND TEMPERATURE CHANGES.
C
      DIMENSION PJTIL(10,3), PTIL(30)
      KAP=0
      DO 1 I=1,10
      DO 1 J=1,3
      PJTIL(I,J)=0.
1     CONTINUE
      DO 2 I=1,NM
      DO 2 J=1,6
      FTIL(I,J)=0.
2     CONTINUE
3     READ (5,25) MNO,LDTYP,W1,W2,W3,W4
      IF (LDTYP.LE.0) GO TO 11
      KAP=KAP+1
      GO TO (4,5,6,7,8,9,10), LDTYP
C
C     LOAD TYPE 1 UNIFORM LOAD
C
4     A1=W2**2/2.
      A2=W2**3/3.
      A3=W2**4/4.
      A4=W1**2/2.
      A5=W1**3/3.
      A6=W1**4/4.
      FTIL(MNO,2)=FTIL(MNO,2)-(W2-A2*3.+A3*2.-W1+A5*3.-A6*2.)*W3*AL(MNO)
      FTIL(MNO,3)=FTIL(MNO,3)-(A1-A2*2.+A3-A4+A5*2.-A6)*W3*AL(MNO)**2
      FTIL(MNO,5)=FTIL(MNO,5)-(A2*3.-A3*2.-A5*3.+A6*2.)*W3*AL(MNO)
      FTIL(MNO,6)=FTIL(MNO,6)+(A2-A3-A5+A6)*W3*AL(MNO)**2
```

```
      WRITE (6,17) MNO,W3,W1,W2
      GO TO 3
C
C     LOAD TYPE 2 SINGLE CONCENTRATED LOAD
C
5     B=AL(MNO)-W2
      ALI=AL(MNO)
      FTIL(MNO,2)=FTIL(MNO,2)-W1*B**2/(ALI**3)*(3.*W2+B)
      FTIL(MNO,3)=FTIL(MNO,3)-W1*W2*B**2/(ALI**2)
      FTIL(MNO,5)=FTIL(MNO,5)-W1*W2**2*(W2+3.*B)/(ALI**3)
      FTIL(MNO,6)=FTIL(MNO,6)+W1*B*W2**2/(ALI**2)
      WRITE (6,18) MNO,W1,W2
      GO TO 3
C
C     LOAD TYPE 3 AXIAL LOAD
C
6     FTIL(MNO,1)=FTIL(MNO,1)-W1*(AL(MNO)-W2)/AL(MNO)
      FTIL(MNO,4)=FTIL(MNO,4)-W1*W2/AL(MNO)
      WRITE (6,19) MNO,W1,W2
      GO TO 3
C
C     LOAD TYPE 4 UNIFORM INCREASE IN TEMPERATURE
C
7     FTIL(MNO,1)=FTIL(MNO,1)+A(MNO)*E(MNO)*W1*W2
      FTIL(MNO,4)=FTIL(MNO,4)-A(MNO)*E(MNO)*W1*W2
      WRITE (6,20) MNO,W1,W2
      GO TO 3
C
C     LOAD TYPE 5 LINEAR LOAD
C
8     XM=(W4-W3)/(W2-W1)
      Q1=-XM*W1+W3
```

```
      A1=W2**2/2.
      A2=W2**3/3.
      A3=W2**4/4.
      A4=W1**2/2.
      A5=W1**3/3.
      A6=W1**4/4.
      A7=W2**5/5.
      A8=W1**5/5.
      B1=W2-A2*3.+A3*2.-W1+A5*3.-A6*2.
      B2=A1-A3*3.+A7*2.-A4+A6*3.-A8*2.
      FTIL(MNO,2)=FTIL(MNO,2)-(Q1*B1+XM*B2)*AL(MNO)
      B1=A1-A2*2.+A3-A4+A5*2.-A6
      B2=A2-A3*2.+A7-A5+A6*2.-A8
      FTIL(MNO,3)=FTIL(MNO,3)-(Q1*B1+XM*B2)*AL(MNO)**2
      B1=A2*3.-A3*2.-A5*3.+A6*2.
      B2=A3*3.-A7*2.-A6*3.+A8*2.
      FTIL(MNO,5)=FTIL(MNO,5)-(Q1*B1+XM*B2)*AL(MNO)
      B1=A2-A3-A5+A6
      B2=A3-A7-A6+A8
      FTIL(MNO,6)=FTIL(MNO,6)+(Q1*B1+XM*B2)*AL(MNO)**2
      WRITE (6,21) MNO,W3,W1,W4,W2
      GO TO 3
C
C     LOAD TYPE 6 FABRICATION ERROR--- + MEANS MEMBER TOO LONG
C
9     FTIL(MNO,1)=FTIL(MNO,1)+A(MNO)*W1*E(MNO)/AL(MNO)
      FTIL(MNO,4)=FTIL(MNO,4)-A(MNO)*W1*E(MNO)/AL(MNO)
      WRITE (6,22) MNO,W1
      GO TO 3
C
C     LOAD TYPE 7 LINEAR VARIATION IN TEMPERATURE
C
```

```
10    FTIL(MNO,3)=FTIL(MNO,3)+XI(MNO)*E(MNO)*W1*W2/W3
      FTIL(MNO,6)=FTIL(MNO,6)-XI(MNO)*E(MNO)*W1*W2/W3
      WRITE (6,23) MNO,W1,W2
      WRITE (6,24) W3
      GO TO 3
11    IF (KAP.EQ.0) GO TO 16
      WRITE (6,26)
      DO 12 I=1,NM
12    WRITE (6,27) I,(FTIL(I,J),J=1,6)
      DO 13 I=1,NM
      J1=INC(I,1)
      J2=INC(I,2)
      CI=C(I)
      SI=S(I)
      GTILA1=CI*FTIL(I,1)-SI*FTIL(I,2)
      GTILA2=SI*FTIL(I,1)+CI*FTIL(I,2)
      GTILA3=FTIL(I,3)
      GTILB1=CI*FTIL(I,4)-SI*FTIL(I,5)
      GTILB2=SI*FTIL(I,4)+CI*FTIL(I,5)
      GTILB3=FTIL(I,6)
      PJTIL(J1,1)=PJTIL(J1,1)+GTILA1
      PJTIL(J1,2)=PJTIL(J1,2)+GTILA2
      PJTIL(J1,3)=PJTIL(J1,3)+GTILA3
      PJTIL(J2,1)=PJTIL(J2,1)+GTILB1
      PJTIL(J2,2)=PJTIL(J2,2)+GTILB2
13    PJTIL(J2,3)=PJTIL(J2,3)+GTILB3
      DO 14 I=1,NJ
      L1=1-ICON(3*I-2)
      L2=1-ICON(3*I-1)
      L3=1-ICON(3*I)
      PTIL(3*I-2)=PJTIL(I,1)*L1
      PTIL(3*I-1)=PJTIL(I,2)*L2
```

```
14    PTIL(3*I)=PJTIL(I,3)*L3
      DO 15 K=1,NJ3
      P(K)=P(K)-PTIL(K)
15    CONTINUE
16    RETURN
C
17    FORMAT (/20X,7HMEMBER ,I2,35H IS SUBJECTED TO A UNIFORM LOAD OF ,F
     18.5,10H KIPS/INCH/20X,9HSTARTING ,F5.2,14H L AND ENDING ,F5.2,32H
     2L FROM THE A END OF THE MEMBER.)
18    FORMAT (/20X,7HMEMBER ,I2,38H IS SUBJECTED TO A TRANSVERSE LOAD OF
     1 ,F8.5,5H KIPS,F9.5,32H INCHES FROM THE A END OF MEMBER)
19    FORMAT (/20X,7HMEMBER ,I2,33H IS SUBJECTED TO AN AXIAL LOAD OF,F8.
     15,5HKIPS ,F9.5,32H INCHES FROM THE A END OF MEMBER)
20    FORMAT (/20X,7HMEMBER ,I2,53H IS SUBJECTED TO A UNIFORM INCREASE I
     1N TEMPERATURE OF,F7.4,11H DEGREES F./45X,36HCOEFFICIENT OF THERMAL
     2 EXPANSION IS ,F10.8)
21    FORMAT (/20X,7HMEMBER ,I2,34H IS SUBJECTED TO A LINEAR LOAD OF ,F8
     1.5,10H KIPS/INCH/20X,9HSTARTING ,F5.2,52H L FROM THE A END OF THE
     2MEMBER AND VARYING LINEARLY,/20X,4H TO ,F8.5,25H KIPS/INCH AT A DI
     3STANCE ,F5.2,32H L FROM THE A END OF THE MEMBER.)
22    FORMAT (/20X,7HMEMBER ,I2,4H IS ,F8.5,41H INCHES TOO LONG DUE TO F
     1ABRICATION ERROR)
23    FORMAT (20X,7HMEMBER ,I2,54H IS SUBJECTED TO A LINEAR VARIATION IN
     1 TEMPERATURE OF ,F7.4,11H DEGREES F./45X,36HCOEFFICIENT OF THERMAL
     2 EXPANSION IS ,F10.8)
24    FORMAT (45X,20HDEPTH OF SECTION IS ,F9.6,8H INCHES.)
25    FORMAT (2I2,1X,4F9.3)
26    FORMAT (//40X,31HTILDE FORCE MATRIX (TRANSPOSED)/)
27    FORMAT (8X,7HMEMBER ,I3,6(5X,F12.5)/)
      END
```

```
      SUBROUTINE STIFF (C,S,ALPH,BETA,XL,AK)
C
C     VARIABLE          MEANING
C     A1-A6           GENERAL COEFFICIENTS OF THE STIFFNESS MATRIX.
C     AK              SYSTEM STIFFNESS MATRIX
C
C     J1-J3           DEFINE THE LOCATION OF EACH ELEMENT IN THE SYSTEM STIFFNESS
C     K1-K3           MATRIX.
C     L1-L5
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION C(NM), S(NM), ALPH(NM), BETA(NM), XL(NM), AK(NJ3,MBAND)
C
C     INITIALIZE AK TO 0.
C
      DO 1 J=1,NJ3
      DO 1 K=1,MBAND
1     AK(J,K)=0.
      DO 2 I=1,NM
C
C     COMPUTE THE GENERAL COEFFICIENTS.
C
      CI=C(I)
      SI=S(I)
      AI=ALPH(I)
      BI=BETA(I)
      XLI=XL(I)
      A1=(BI*CI**2+12.*SI**2)*AI
      A2=((BI-12.)*CI*SI)*AI
      A3=(BI*SI**2+12.*CI**2)*AI
      A4=(-6.*XLI*SI)*AI
      A5=(6.*XLI*CI)*AI
```

```
      A6=(4.*XLI**2)*AI
C
C     DETERMINE THE LOCATIONS.
C
      J=INC(I,1)
      K=INC(I,2)
      J1=3*J-2
      J2=3*J-1
      J3=3*J
      K1=3*K-2
      K2=3*K-1
      K3=3*K
      L=3*(K-J)-2
      L1=L+1
      L2=L+2
      L3=L+3
      L4=L+4
      L5=L+5
C
C     ASSEMBLE THE SYSTEM STIFFNESS MATRIX.
C
      AK(J1,1)=AK(J1,1)+A1
      AK(J1,2)=AK(J1,2)+A2
      AK(J1,3)=AK(J1,3)+A4
      AK(J1,L3)=AK(J1,L3)-A1
      AK(J1,L4)=AK(J1,L4)-A2
      AK(J1,L5)=AK(J1,L5)+A4
      AK(J2,1)=AK(J2,1)+A3
      AK(J2,2)=AK(J2,2)+A5
      AK(J2,L2)=AK(J2,L2)-A2
      AK(J2,L3)=AK(J2,L3)-A3
      AK(J2,L4)=AK(J2,L4)+A5
```

```
      AK(J3,1)=AK(J3,1)+A6
      AK(J3,L1)=AK(J3,L1)-A4
      AK(J3,L2)=AK(J3,L2)-A5
      AK(J3,L3)=AK(J3,L3)+A6/2.
      AK(K1,1)=AK(K1,1)+A1
      AK(K1,2)=AK(K1,2)+A2
      AK(K1,3)=AK(K1,3)-A4
      AK(K2,1)=AK(K2,1)+A3
      AK(K2,2)=AK(K2,2)-A5
      AK(K3,1)=AK(K3,1)+A6
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONSTR (ICON,AK)
C
C     IF A JOINT IS CONSTRAINED, THE ELEMENT IN THE FIRST COLUMN IS MADE 1,
C     THE REST OF THE ELEMENTS IN THAT ROW AND UP THE DIAGONAL ARE MADE 0.
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION ICON(NJ3), AK(NJ3,MBAND)
      DO 2 II=1,NJ3
      IF (ICON(II).EQ.0) GO TO 2
      AK(II,1)=1.
      DO 1 L=2,MBAND
      AK(II,L)=0.
      M=II+1-L
      IF (M.LT.1) GO TO 1
      AK(M,L)=0.
1     CONTINUE
2     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOLVE (AK,P)
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION AK(NJ3,MBAND), P(NJ3)
      WRITE (6,6)
      NRS=NJ3-1
      DO 2 N=1,NRS
      M=N-1
      MR=NJ3-M
      IF (MBAND.LT.MR) MR=MBAND
      PIVOT=AK(N,1)
      DO 2 L=2,MR
      CP=AK(N,L)/PIVOT
      I=M+L
      J=0
      DO 1 K=L,MR
      J=J+1
    1 AK(I,J)=AK(I,J)-CP*AK(N,K)
    2 AK(N,L)=CP
      DO 3 N=1,NRS
      M=N-1
      MR=NJ3-M
      IF (MBAND.LT.MR) MR=MBAND
      CP=P(N)
      P(N)=CP/AK(N,1)
      DO 3 L=2,MR
      I=M+L
    3 P(I)=P(I)-AK(N,L)*CP
      P(NJ3)=P(NJ3)/AK(NJ3,1)
      DO 4 I=1,NRS
      N=NJ3-I
      M=N-1
      MR=NJ3-M
```

```
      IF (MBAND.LT.MR) MR=MBAND
      DO 4 K=2,MR
      L=M+K
4     P(N)=P(N)-AK(N,K)*P(L)
      I1=0
      DO 5 I=1,NJ3,3
      I1=I1+1
      WRITE (6,7) I1,P(I),P(I+1),P(I+2)
5     CONTINUE
      RETURN
C
6     FORMAT (1H-,10X,19HJOINT DISPLACEMENTS,/,/,10X,5HJOINT,8X,1H1,10X,
     11H2,10X,1H3,/)
7     FORMAT (1H ,9X,I3,6X,F8.5,2X,F8.5,2X,F8.5)
      END
```

```
      SUBROUTINE FORCE (U,C,S,ALPH,BETA,XL,FKI,FTIL)
C
C     VARIABLE          MEANING
C      F               MEMBER FORCES IN LOCAL COORDINATES.
C     FA,FB             MEMBER FORCES IN GLOBAL COORDINATES.
C     FKI              ARRAY TO STORE MEMBER FORCES.
C     V1-V6             DISPLACEMENTS IN LOCAL COORDINATES.
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      DIMENSION FKI(NM,6)
      DIMENSION U(NJ3), C(NM), S(NM), ALPH(NM), BETA(NM), XL(NM)
      DIMENSION F(6), P(10,3), FA(3), FB(3)
      DIMENSION FTIL(NM,6)
      DO 1 II=1,NJ
      DO 1 KK=1,3
    1 P(II,KK)=0.
C
C     JOINT DISPLACEMENTS IN LOCAL COORDINATES ARE FOUND FIRST.
C
      WRITE (6,7)
      DO 5 L=1,NM
      J=INC(L,1)
      K=INC(L,2)
      J1=3*(J-1)+1
      K1=3*(K-1)+1
      CI=C(L)
      SI=S(L)
      AI=ALPH(L)
      BI=BETA(L)
      XLI=XL(L)
      V1=CI*U(J1)+SI*U(J1+1)
      V2=-SI*U(J1)+CI*U(J1+1)
```

```
      V3=U(J1+2)
      V4=CI*U(K1)+SI*U(K1+1)
      V5=-SI*U(K1)+CI*U(K1+1)
      V6=U(K1+2)
C
C     MEMBER FORCES ARE NOW FOUND.
C
      F(1)=AI*BI*(V1-V4)
      F(2)=6.*AI*(2.*V2+XLI*V3-2.*V5+XLI*V6)
      F(3)=2.*AI*(3.*XLI*V2+2.*XLI**2*V3-3.*XLI*V5+XLI**2*V6)
      F(4)=-F(1)+FTIL(L,4)
      F(5)=-F(2)+FTIL(L,5)
      F(6)=XLI*F(2)-F(3)+FTIL(L,6)
      F(1)=F(1)+FTIL(L,1)
      F(2)=F(2)+FTIL(L,2)
      F(3)=F(3)+FTIL(L,3)
      FKI(L,1)=F(1)
      FKI(L,2)=F(3)
      FKI(L,3)=F(6)
      FKI(L,4)=F(4)
      FKI(L,5)=F(2)
      FKI(L,6)=F(5)
      DO 3 M=1,6
      IF (M.EQ.3) GO TO 2
      WRITE (6,8) L,M,F(M)
      GO TO 3
2     WRITE (6,9) L,M,F(M)
3     CONTINUE
      WRITE (6,10)
C
C     MEMBER FORCES ARE PUT INTO GLOBAL COORDINATES.
C
```

```
      FA(1)=CI*F(1)-SI*F(2)
      FA(2)=SI*F(1)+CI*F(2)
      FA(3)=F(3)
      FB(1)=CI*F(4)-SI*F(5)
      FB(2)=SI*F(4)+CI*F(5)
      FB(3)=F(6)
C
C     JOINT FORCES ARE FOUND BY SUMMING APPROPRIATE MEMBER FORCES.
C
      DO 4 N=1,3
      P(J,N)=P(J,N)+FA(N)
      P(K,N)=P(K,N)+FB(N)
4     CONTINUE
5     CONTINUE
      WRITE (6,11)
      DO 6 I=1,NJ
      WRITE (6,12) P(I,1),I,P(I,2),P(I,3)
6     CONTINUE
      RETURN
C
7     FORMAT (1H-,10X,13HMEMBER FORCES,/)
8     FORMAT (1H ,10X,2HF(,I1,1H,,I1,1H),5X,F10.5)
9     FORMAT (1H ,10X,2HF(,I1,1H,,I1,1H),2X,1H=,2X,F10.5)
10    FORMAT (1H-)
11    FORMAT (1H-,10X,12HJOINT FORCES,/)
12    FORMAT (1H ,17X,F10.5,/,11X,2HP(,I1,4H)=  ,F10.5,/,18X,F10.5,/)
      END
```

```
      SUBROUTINE PENFUN (X,XL,E,FKI,ICHEK,RK,P,ALPH,BETA,XI1,G,IND,AXIAL
     1,BEND,W1,W2,W3,W4,W5,W6,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6)
C
C     VARIABLE            MEANING
C     ACC             ACCURACY TO WHICH VA13A WILL MINIMIZE.
C     AKLR             KL/R
C     AXIAL            ARRAY OF AXIAL LOADS ON EACH MEMBER.
C     BEND            ARRAY OF MOMENTS ON EACH MEMBER.
C     BET             BETA IN EXTRAPOLATION FORMULA.
C     CC              C  IN AISC MANUAL FORMULA 1.5-1
C                      C
C     CM              SEE FORMULA 1.6-1A IN AISC MANUAL
C     EPSLN            USED IN DETERMINING CONVERGENCE
C     FA              ALLOWABLE AXIAL STRESS
C     FA1             ACTUAL AXIAL STRESS
C     FA2             UNDER CERTAIN CONDITIONS, FA2=FA1
C     FB              ALLOWABLE STRESS FOR MOMENT
C     FB1             ACTUAL BENDING STRESS
C     FS              FOR CONVENIENCE, DENOMINATOR OF FORMULA 1.5-1 IN AISC MANUAL
C     FV              SUM OF MOMENTS OF INERTIA OF MEMBERS
C     GG1-GG6           CONSTRAINTS 1 THROUGH 6
C     IND             IND=1 MEANS TENSION
C                     IND=2 MEANS COMPRESSION
C     KOUNT            KOUNT=1 MEANS THE FIRST ITERATION
C                     KOUNT=2 MEANS PART OF AN EXTRAPOLATION LOOP.
C     LIM             COUNTER TO CONTROL THE NUMBER OF ITERATIONS.
C     MARK,FIVAL          USED IN FORMULATION OF EXTENDED PENALTY FUNCTION
C     R               ARRAY OF RESPONSE FACTORS
C     XI1             ARRAY TO STORE SOLUTION FROM EACH ITERATION
C
      COMMON STORI(10,10),INC(10,2),NM,NJ,NJ3,MBAND,FY,LIM1
      EXTERNAL FUNCT
```

```
      DIMENSION W(115), XI1(NM,10), R(10), X(NM), G(NM), XL(NM), E(NM),
     1FKI(NM,6), IND(NM), AXIAL(NM), BEND(NM), W1(NM), W2(NM), W3(NM), W
     24(NM), W5(NM), W6(NM), RK(NJ3), P(NJ3), ALPH(NM), BETA(NM), SCALE(
     310)
      DIMENSION GG1(NM), GG2(NM), GG3(NM), GG4(NM), GG5(NM), GG6(NM), DG
     13(NM), DG4(NM), DG5(NM), DG6(NM)
      N=NM
      DO 1 I=1,10
    1 SCALE(I)=50.
      ACC=0.01
      LIM=1
      DO 2 I=1,NM
      XI1(I,1)=X(I)
    2 CONTINUE
C
C     THE FORCES IN EACH MEMBER ARE NOW FOUND.
C
      DO 4 I=1,NM
      IF (FKI(I,1).LE.0.) IND(I)=1
      IF (FKI(I,1).GT.0.) IND(I)=2
      AXIAL(I)=ABS(FKI(I,1))
      BEND(I)=ABS(FKI(I,2))
      IF (ABS(FKI(I,3)).GT.BEND(I)) BEND(I)=ABS(FKI(I,3))
      IF (IND(I).EQ.2) GO TO 3
      WRITE (6,45) I,AXIAL(I),BEND(I)
      GO TO 4
    3 WRITE (6,46) I,AXIAL(I),BEND(I)
    4 CONTINUE
      KOUNT=1
    5 CONTINUE
C
C     CHECK THE FEASIBILITY OF THE INITIAL POINT.
```

```
C
      I=1
      FB=0.6666667*FY
6     XI=X(I)
      EI=E(I)
      INDI=IND(I)
      XLI=XL(I)
      BI=BEND(I)
      AI=AXIAL(I)
      IF (XI.EQ.52.) XI=52.01
      IF (XI.EQ.20300.) XI=20299.99
      GGG1=XI/52.-1.
      GGG2=20300./XI-1.
      AKLR=1.
      IF (INDI.EQ.1) GO TO 10
      CC=SQRT(118.4352*EI/FY)
      IF (XI.GT.9000.) GO TO 7
      AKLR=XLI/(XI/0.21583)**0.25
      GO TO 8
7     AKLR=XLI*12841./(XI+174498.)
8     GGG3=200./AKLR-1.
      IF (AKLR.GE.CC) GO TO 9
      FS=1.66667+(0.375*AKLR)/CC-AKLR**3/(8.*CC**3)
      FA=(1.-AKLR**2/(2.*CC**2))*FY/FS
      GO TO 11
9     FA=5.14936*EI/AKLR**2
      GO TO 11
10    FA=0.6*FY
      GGG3=0.
11    IF (XI.GT.9000.) GO TO 12
      FB1=BI/(SQRT(60.6*XI+84100.)-290.)
      FA1=AI/(SQRT(XI)*0.464)
```

```
      GO TO 13
12    FB1=1.876*BI/(XI-8056.3)
      FA1=AI/(0.0039*(XI+2300.))
13    FE=5.14936*(EI/AKLR**2)
      IF ((FA1/FA).LT.0.15) GO TO 14
      FA2=0.
      CM=1.
      IF (INDI.EQ.2) FA2=FA1
      IF (INDI.EQ.2) CM=0.85
      GGG4=1.-(FA1/FA+CM*FB1/((1.-FA2/FE)*FB))
      GGG5=1.-(FA1/(0.6*FY)+FB1/FB)
      GGG6=0.
      GO TO 15
14    GGG4=0.
      GGG5=0.
      GGG6=1.-(FA1/FA+FB1/FB)
15    CONTINUE
      IF (KOUNT.EQ.2) GO TO 16
      IF (GGG1.LT.0.) GO TO 17
      IF (GGG2.LT.0.) GO TO 17
      IF (GGG3.LT.0.) GO TO 17
      IF (GGG4.LT.0.) GO TO 17
      IF (GGG5.LT.0.) GO TO 17
      IF (GGG6.LT.0.) GO TO 17
      GO TO 18
16    IF (GGG1.LT.0.) GO TO 36
      IF (GGG2.LT.0.) GO TO 36
      IF (GGG3.LT.0.) GO TO 36
      IF (GGG4.LT.0.) GO TO 36
      IF (GGG5.LT.0.) GO TO 36
      IF (GGG6.LT.0.) GO TO 36
      GO TO 24
```

```
17    WRITE (6,47) I
18    W1(I)=1./GGG1
      W2(I)=1./GGG2
      IF (GGG3.NE.0.) GO TO 19
      W3(I)=0.
      GO TO 20
19    W3(I)=1./GGG3
20    IF (GGG4.NE.0.) GO TO 21
      W4(I)=0.
      W5(I)=0.
      GO TO 22
21    W4(I)=1./GGG4
      W5(I)=1./GGG5
22    IF (GGG6.NE.0.) GO TO 23
      W6(I)=0.
      GO TO 24
23    W6(I)=1./GGG6
24    I=I+1
      IF (I.LT.NM+1) GO TO 6
      IF (KOUNT.EQ.2) GO TO 28
      FVSUM=0.
      DO 25 I=1,NM
      FVSUM=FVSUM+XL(I)*SQRT(X(I))
25    CONTINUE
      FV=FVSUM
C
C     CALCULATE THE INITIAL R VALUE.
C
      WSUM=0.
      DO 26 I=1,NM
      WSUM=WSUM+W1(I)+W2(I)+W3(I)+W4(I)+W5(I)+W6(I)
26    CONTINUE
```

```
      R(LIM)=(0.5*FV)/WSUM
      IF (R(LIM).LT.0.) R(LIM)=50.
      L=LIM-1
      WRITE (6,48)
      DO 27 I=1,NM
      WRITE (6,49) I,X(I)
27    CONTINUE
      WRITE (6,50) FV
28    CONTINUE
      RR=R(LIM)
      MARK=0
      FIVSUM=0.
      DO 29 I=1,NM
      FIVSUM=FIVSUM+XL(I)*SQRT(X(I))
29    CONTINUE
      FIVAL=FIVSUM
C
C     CALL THE MINIMIZATION ROUTINE.
C
      CALL VA13A (FUNCT,N,X,F,G,SCALE,ACC,W,RR,IND,XL,E,BEND,AXIAL,FY,FI
     1VAL,MARK,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6,NM)
      LIM=LIM+1
C
C     STORE THE NEW SOLUTION.
C
      DO 30 I=1,NM
      XI1(I,LIM)=X(I)
30    CONTINUE
      FVSUM=0.
      DO 31 I=1,NM
      FVSUM=FVSUM+XL(I)*SQRT(X(I))
31    CONTINUE
```

```
      FV=FVSUM
C
C     CHECK CONVERGENCE.
C
      SUMXI=0.
      DO 32 I=1,NM
      SUMXI=SUMXI+XL(I)*SQRT(XI1(I,LIM-1))
32    CONTINUE
      EPSLN=(FV-SUMXI)/FV
      IF (EPSLN.LT.0.) EPSLN=-EPSLN
      L=LIM-1
      WRITE (6,51) L
      DO 33 I=1,NM
      WRITE (6,49) I,X(I)
33    CONTINUE
      WRITE (6,52) RR,EPSLN
      WRITE (6,50) FV
      IF (EPSLN.LE.0.01) GO TO 39
      IF (LIM.GT.10) GO TO 38
C
C     REVISE R VALUE.
C
      R(LIM)=R(LIM-1)/10.
      IF (LIM.LE.3) GO TO 28
C
C     PERFORM EXTRAPOLATION.
C
      BET=0.4
      KOUNT=2
34    CONTINUE
      DO 35 I=1,NM
      X(I)=((R(LIM-2)**BET)*XI1(I,LIM)-(R(LIM-1)**BET)*XI1(I,LIM-1))/(R(
```

```
     1LIM-2)**BET-R(LIM-1)**BET)
35    CONTINUE
      GO TO 5
36    BET=BET+0.1
      IF (BET.LT.1.) GO TO 34
      DO 37 I=1,NM
      X(I)=XI1(I,LIM)
37    CONTINUE
      GO TO 28
38    WRITE (6,53)
      ICHEK=0
      GO TO 44
39    WRITE (6,54)
      ICHEK=1
C
C     REVISE STIFFNESS CHARACTERISTICS.
C
      DO 42 I=1,NM
      XI=X(I)
      XLI=XL(I)
      EI=E(I)
      IF (XI.GT.9000.) GO TO 40
      A=0.464*SQRT(XI)
      GO TO 41
40    A=0.0039*(XI+2300.)
41    ALPH(I)=EI*XI/XLI**3
      BETA(I)=A*XLI**2/XI
42    CONTINUE
C
C     REDEFINE THE LOAD VECTOR.
C
      DO 43 I=1,NJ3
```

```
      P(I)=RK(I)
43    CONTINUE
44    RETURN
C
45    FORMAT (1H0,10X,17HFORCES IN MEMBER ,I1,1H:,//,15X,13HAXIAL FORCE=
     1 ,F10.5,9H, TENSION,/,15X,8HMOMENT= ,F10.5)
46    FORMAT (1H0,10X,17HFORCES IN MEMBER ,I1,1H:,//,15X,13HAXIAL FORCE=
     1 ,F10.5,13H, COMPRESSION,/,15X,8HMOMENT= ,F10.5)
47    FORMAT (1H0,10X,14HMEMBER NUMBER ,I1,35H IS NOT WITHIN THE FEASIBL
     1E REGION.)
48    FORMAT (1H0,10X,15HINITIAL DESIGN:)
49    FORMAT (1H0,10X,2HI(,I2,2H)=,2X,F8.2)
50    FORMAT (1H0,10X,25HOBJECTIVE FUNCTION VALUE=,2X,F9.4)
51    FORMAT (1H0,10X,8HREVISION,2X,I2,1H:)
52    FORMAT (1H0,10X,15HSCALING FACTOR=,2X,F9.4,//,11X,8HEPSILON=,2X,F6
     1.3)
53    FORMAT (1H0,10X,35HFAILED TO CONVERGE AFTER 10 TRIALS.)
54    FORMAT (1H0,10X,34HCONVERGENCE HAS NOW BEEN OBTAINED.)
      END
```

```
      SUBROUTINE VA13A (FUNCT,N,X,F,G,SCALE,ACC,W,RR,IND,XL,E,BEND,AXIAL
     1,FY,FIVAL,MARK,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6,NM)
      COMMON /VA13B/ IPRINT,LP,MAXFUN,MODE,NFUN
      EXTERNAL FUNCT
      DIMENSION X(1), G(1), SCALE(1), W(1)
      DIMENSION XL(NM), E(NM), BEND(NM), AXIAL(NM), IND(NM), GG1(NM), GG
     12(NM), GG3(NM), GG4(NM), GG5(NM), GG6(NM), DG3(NM), DG4(NM), DG5(N
     2M), DG6(NM)
      ND=1+(N*(N+1))/2
      NW=ND+N
      NXA=NW+N
      NGA=NXA+N
      NXB=NGA+N
      NGB=NXB+N
      CALL VA13C (FUNCT,N,X,F,G,SCALE,ACC,W,W(ND),W(NW),W(NXA),W(NGA),W(
     1NXB),W(NGB),RR,IND,XL,E,BEND,AXIAL,FY,FIVAL,MARK,GG1,GG2,GG3,GG4,G
     2G5,GG6,DG3,DG4,DG5,DG6,NM)
      RETURN
      END
      BLOCK DATA
      COMMON /VA13B/ IPRINT,LP,MAXFUN,MODE,NFUN
      END
      SUBROUTINE VA13C (FUNCT,N,X,F,G,SCALE,ACC,H,D,W,XA,GA,XB,GB,RR,IND
     1,XL,E,BEND,AXIAL,FY,FIVAL,MARK,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5
     2,DG6,NM)
      COMMON /VA13B/ IPRINT,LP,MAXFUN,MODE,NFUN
      DIMENSION X(1), G(1), SCALE(1), H(1), D(1), W(1), XA(1), GA(1), XB
     1(1),EGB(1)
      DIMENSION XL(NM), E(NM), BEND(NM), AXIAL(NM), IND(NM), GG1(NM), GG
     12(NM), GG3(NM), GG4(NM), GG5(NM), GG6(NM), DG3(NM), DG4(NM), DG5(N
     2M),6DG6(NM)
      ISTOP=0
```

```
C     BEGIN THE PRINTING FROM THE SUBROUTINE
      IF (IPRINT.EQ.0) GO TO 1
      WRITE (LP,30)
C     CALCULATE THE INITIAL FUNCTION VALUE
1     CALL FUNCT (N,X,F,G,RR,IND,XL,E,BEND,AXIAL,FY,FIVAL,MARK,PEN,NM,GG
     11,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6)
      NFUN=1
      ITR=0
      NP=N+1
C     SET THE HESSIAN TO A DIAGONAL MATRIX DEPENDING ON SCALE(.)
      IF (MODE.GE.2) GO TO 6
2     C=0.
      DO 3 I=1,N
3     C=AMAX1(C,ABS(G(I)*SCALE(I)))
      IF (C.LE.0.) C=1.
      K=(N*NP)/2
      DO 4 I=1,K
4     H(I)=0.
      K=1
      DO 5 I=1,N
      H(K)=0.01*C/SCALE(I)**2
5     K=K+NP-I
      GO TO 10
C     FACTORIZE THE GIVEN HESSIAN MATRIX
6     IF (MODE.GE.3) GO TO 8
      CALL MC11B (H,N,K)
      IF (K.GE.N) GO TO 10
7     WRITE (LP,31)
      GO TO 2
C     CHECK THAT THE GIVEN DIAGONAL ELEMENTS ARE POSITIVE
8     K=1
      DO 9 I=1,N
```

```
      IF (H(K).LE.0.) GO TO 7
9     K=K+NP-I
C     SET SOME VARIABLES FOR THE FIRST ITERATION
10    DFF=0.
      IPRA=IABS(IPRINT)
      IP=IABS(IPRA-1)
11    FA=F
      ISFV=1
      DO 12 I=1,N
      XA(I)=X(I)
12    GA(I)=G(I)
C     BEGIN THE ITERATION BY GIVING THE REQUIRED PRINTING
13    IP=IP+1
      IF (IP.NE.IPRA) GO TO 14
      IP=0
      WRITE (LP,32) ITR,NFUN
      WRITE (LP,33) FA
      IF (IPRINT.LE.0) GO TO 14
      WRITE (LP,34) (XA(I),I=1,N)
      WRITE (LP,35) (GA(I),I=1,N)
14    ITR=ITR+1
C     CALCULATE THE SEARCH DIRECTION OF THE ITERATION
      DO 15 I=1,N
15    D(I)=-GA(I)
      CALL MC11E (H,N,D,W,N)
C     CALCULATE A LOWER BOUND ON THE STEP-LENGTH
C     AND THE INITIAL DIRECTIONAL DERIVATIVE
      C=0.
      DGA=0.
      DO 16 I=1,N
      C=AMAX1(C,ABS(D(I)/SCALE(I)))
16    DGA=DGA+GA(I)*D(I)
```

```
C     TEST IF THE SEARCH DIRECTION IS DOWNHILL
      IF (DGA.GE.0.) GO TO 24
C     SET THE INITIAL STEP-LENGTH OF THE LINE SEARCH
      STMIN=0.
      STEPBD=0.
      STEPLB=ACC/C
      FMIN=FA
      GMIN=DGA
      STEP=1.
      IF (DFF.LE.0.) STEP=AMIN1(STEP,1./C)
      IF (DFF.GT.0.) STEP=AMIN1(STEP,(DFF+DFF)/(-DGA))
17    C=STMIN+STEP
C     TEST WHETHER FUNC HAS BEEN CALLED MAXFUN TIMES
      IF (NFUN.EQ.MAXFUN) GO TO 25
      NFUN=NFUN+1
C     CALCULATE ANOTHER FUNCTION VALUE AND GRADIENT
      DO 18 I=1,N
18    XB(I)=XA(I)+C*D(I)
      ISTOP=ISTOP+1
      IF (ISTOP.GT.100) GO TO 26
      CALL FUNCT (N,XB,FB,GB,RR,IND,XL,E,BEND,AXIAL,FY,FIVAL,MARK,PEN,NM
     1,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6)
C     STORE THIS FUNCTION VALUE IF IT IS THE SMALLEST SO FAR
      ISFV=MIN0(2,ISFV)
      IF (FB.GT.F) GO TO 22
      IF (FB.LT.F) GO TO 20
      GL1=0.
      GL2=0.
      DO 19 I=1,N
      GL1=GL1+(SCALE(I)*G(I))**2
19    GL2=GL2+(SCALE(I)*GB(I))**2
      IF (GL2.GE.GL1) GO TO 22
```

```
20    ISFV=3
      F=FB
      DO 21 I=1,N
      X(I)=XB(I)
21    G(I)=GB(I)
C     CALCULATE THE DIRECTIONAL DERIVATIVE AT THE NEW POINT
22    DGB=0.
      DO 23 I=1,N
23    DGB=DGB+GB(I)*D(I)
C     BRANCH IF WE HAVE FOUND A NEW LOWER BOUND ON THE STEP-LENGTH
      IF (FB-FA.LE.0.1*C*DGA) GO TO 28
C     FINISH THE ITERATION IF THE CURRENT STEP IS STEPLB
      IF (STEP.GT.STEPLB) GO TO 27
24    IF (ISFV.GE.2) GO TO 11
C     AT THIS STAGE THE WHOLE CALCULATION IS COMPLETE
25    IF (IPRINT.EQ.0) GO TO 26
      WRITE (LP,36)
      WRITE (LP,32) ITR,NFUN
      WRITE (LP,33) F
      WRITE (LP,34) (X(I),I=1,N)
      WRITE (LP,35) (G(I),I=1,N)
26    RETURN
C     CALCULATE A NEW STEP-LENGTH BY CUBIC INTERPOLATION
27    STEPBD=STEP
      C=GMIN+DGB-3.*(FB-FMIN)/STEP
      ZAP=C+GMIN-SQRT(C*C-GMIN*DGB)
      IF (ZAP.EQ.0.) GO TO 26
      C=GMIN/ZAP
      STEP=STEP*AMAX1(0.1,C)
      GO TO 17
C     SET THE NEW BOUNDS ON THE STEP-LENGTH
28    STEPBD=STEPBD-STEP
```

```
      STMIN=C
      FMIN=FB
      GMIN=DGB
C     CALCULATE A NEW STEP-LENGTH BY EXTRAPOLATION
      STEP=9.*STMIN
      IF (STEPBD.GT.2.E-78) STEP=0.5*STEPBD
      C=DGA+3.*DGB-4.*(FB-FA)/STMIN
      IF (C.GT.0.) STEP=AMIN1(STEP,STMIN*AMAX1(1.,-DGB/C))
      IF (DGB.LT.0.7*DGA) GO TO 17
C     TEST FOR CONVERGENCE OF THE ITERATIONS
      ISFV=4-ISFV
      IF (STMIN+STEP.LE.STEPLB) GO TO 24
C     REVISE THE SECOND DERIVATIVE MATRIX
      IR=-N
      DO 29 I=1,N
      XA(I)=XB(I)
      XB(I)=GA(I)
      D(I)=GB(I)-GA(I)
   29 GA(I)=GB(I)
      CALL MC11A (H,N,XB,1./DGA,W,IR,1,0.)
      IR=-IR
      CALL MC11A (H,N,D,1./(STMIN*(DGB-DGA)),D,IR,0,0.)
C     BRANCH IF THE RANK OF THE NEW MATRIX IS DEFICIENT
      IF (IR.LT.N) GO TO 25
C     BEGIN ANOTHER ITERATION
      DFF=FA-FB
      FA=FB
      GO TO 13
C
   30 FORMAT (15H1ENTRY TO VA13A)
   31 FORMAT (/5X,50HBECAUSE THE HESSIAN GIVEN TO VA13A IS NOT POS DEF,/
     15X,50HIT HAS BEEN REPLACED BY A POSITIVE DIAGONAL MATRIX)
```

```
32    FORMAT (/5X,11HITERATION =,I5,5X,11HFUNCTIONS =,I5)
33    FORMAT (5X,3HF =,E15.7)
34    FORMAT (5X,6HX(.) =,(7E15.7))
35    FORMAT (5X,6HG(.) =,(7E15.7))
36    FORMAT (/5X,37HTHE RESULTS FROM VA13A ARE AS FOLLOWS)
      END
      SUBROUTINE MC11A (A,N,Z,SIG,W,IR,MK,EPS)
      DIMENSION A(1), Z(1), W(1)
C     UPDATE FACTORS GIVEN IN A BY   SIG*Z*ZTRANSPOSE
      IF (N.GT.1) GO TO 1
      A(1)=A(1)+SIG*Z(1)**2
      IR=1
      IF (A(1).GT.0.) RETURN
      A(1)=0.
      IR=0
      RETURN
1     CONTINUE
      NP=N+1
      IF (SIG.GT.0.) GO TO 14
      IF (SIG.EQ.0..OR.IR.EQ.0) RETURN
      TI=1./SIG
      IJ=1
      IF (MK.EQ.0) GO TO 3
      DO 2 I=1,N
      IF (A(IJ).NE.0.) TI=TI+W(I)**2/A(IJ)
2     IJ=IJ+NP-I
      GO TO 9
3     CONTINUE
      DO 4 I=1,N
4     W(I)=Z(I)
      DO 8 I=1,N
      IP=I+1
```

```
      V=W(I)
      IF (A(IJ).GT.0.) GO TO 5
      W(I)=0.
      IJ=IJ+NP-I
      GO TO 8
5     CONTINUE
      TI=TI+V**2/A(IJ)
      IF (I.EQ.N) GO TO 7
      DO 6 J=IP,N
      IJ=IJ+1
6     W(J)=W(J)-V*A(IJ)
7     IJ=IJ+1
8     CONTINUE
9     CONTINUE
      IF (IR.LE.0) GO TO 10
      IF (TI.GT.0.) GO TO 11
      IF (MK-1) 14,14,12
10    TI=0.
      IR=-IR-1
      GO TO 12
11    TI=EPS/SIG
      IF (EPS.EQ.0.) IR=IR-1
12    CONTINUE
      MM=1
      TIM=TI
      DO 13 I=1,N
      J=NP-I
      IJ=IJ-I
      IF (A(IJ).NE.0.) TIM=TI-W(J)**2/A(IJ)
      W(J)=TI
13    TI=TIM
      GO TO 15
```

```
14    CONTINUE
      MM=0
      TIM=1./SIG
15    CONTINUE
      IJ=1
      DO 26 I=1,N
      IP=I+1
      V=Z(I)
      IF (A(IJ).GT.0.) GO TO 18
      IF (IR.GT.0.OR.SIG.LT.0..OR.V.EQ.0.) GO TO 17
      IR=1-IR
      A(IJ)=V**2/TIM
      IF (I.EQ.N) RETURN
      DO 16 J=IP,N
      IJ=IJ+1
16    A(IJ)=Z(J)/V
      RETURN
17    CONTINUE
      TI=TIM
      IJ=IJ+NP-I
      GO TO 26
18    CONTINUE
      AL=V/A(IJ)
      IF (MM) 19,19,20
19    TI=TIM+V*AL
      GO TO 21
20    TI=W(I)
21    CONTINUE
      R=TI/TIM
      A(IJ)=A(IJ)*R
      IF (R.EQ.0.) GO TO 27
      IF (I.EQ.N) GO TO 27
```

```
      B=AL/TI
      IF (R.GT.4.) GO TO 23
      DO 22 J=IP,N
      IJ=IJ+1
      Z(J)=Z(J)-V*A(IJ)
22    A(IJ)=A(IJ)+B*Z(J)
      GO TO 25
23    GM=TIM/TI
      DO 24 J=IP,N
      IJ=IJ+1
      Y=A(IJ)
      A(IJ)=B*Z(J)+Y*GM
24    Z(J)=Z(J)-V*Y
25    CONTINUE
      TIM=TI
      IJ=IJ+1
26    CONTINUE
27    CONTINUE
      IF (IR.LT.0) IR=-IR
      RETURN
C     FACTORIZE A MATRIX GIVEN IN A
      ENTRY MC11B(A,N,IR)
      IR=N
      IF (N.GT.1) GO TO 28
      IF (A(1).GT.0.) RETURN
      A(1)=0.
      IR=0
      RETURN
28    CONTINUE
      NP=N+1
      II=1
      DO 32 I=2,N
```

```
      AA=A(II)
      NI=II+NP-I
      IF (AA.GT.0.) GO TO 29
      A(II)=0.
      IR=IR-1
      II=NI+1
      GO TO 32
29    CONTINUE
      IP=II+1
      II=NI+1
      JK=II
      DO 31 IJ=IP,NI
      V=A(IJ)/AA
      DO 30 IK=IJ,NI
      A(JK)=A(JK)-A(IK)*V
30    JK=JK+1
31    A(IJ)=V
32    CONTINUE
      IF (A(II).GT.0.) RETURN
      A(II)=0.
      IR=IR-1
      RETURN
C     MULTIPLY OUT THE FACTORS GIVEN IN A
      ENTRY MC11C(A,N)
      IF (N.EQ.1) RETURN
      NP=N+1
      II=N*NP/2
      DO 37 NIP=2,N
      JK=II
      NI=II-1
      II=II-NIP
      AA=A(II)
```

```
      IP=II+1
      IF (AA.GT.0.) GO TO 34
      DO 33 IJ=IP,NI
33    A(IJ)=0.
      GO TO 37
34    CONTINUE
      DO 36 IJ=IP,NI
      V=A(IJ)*AA
      DO 35 IK=IJ,NI
      A(JK)=A(JK)+A(IK)*V
35    JK=JK+1
36    A(IJ)=V
37    CONTINUE
      RETURN
C     MULTIPLY A VECTOR Z BY THE FACTORS GIVEN IN A
      ENTRY MC11D(A,N,Z,W)
      IF (N.GT.1) GO TO 38
      Z(1)=Z(1)*A(1)
      W(1)=Z(1)
      RETURN
38    CONTINUE
      NP=N+1
      II=1
      N1=N-1
      DO 41 I=1,N1
      Y=Z(I)
      IF (A(II).EQ.0.) GO TO 40
      IJ=II
      IP=I+1
      DO 39 J=IP,N
      IJ=IJ+1
39    Y=Y+Z(J)*A(IJ)
```

```
40    Z(I)=Y*A(II)
      W(I)=Z(I)
41    II=II+NP-I
      Z(N)=Z(N)*A(II)
      W(N)=Z(N)
      DO 43 K=1,N1
      I=N-K
      II=II-NP+I
      IF (Z(I).EQ.0.) GO TO 43
      IP=I+1
      IJ=II
      Y=Z(I)
      DO 42 J=IP,N
      IJ=IJ+1
42    Z(J)=Z(J)+A(IJ)*Z(I)
43    CONTINUE
      RETURN
C     MULTIPLY A VECTOR Z BY THE INVERSE OF THE FACTORS GIVEN IN A
      ENTRY MC11E(A,N,Z,W,IR)
      IF (IR.LT.N) RETURN
      W(1)=Z(1)
      IF (N.GT.1) GO TO 44
      Z(1)=Z(1)/A(1)
      RETURN
44    CONTINUE
      DO 46 I=2,N
      IJ=I
      I1=I-1
      V=Z(I)
      DO 45 J=1,I1
      V=V-A(IJ)*Z(J)
45    IJ=IJ+N-J
```

```
      W(I)=V
46    Z(I)=V
      Z(N)=Z(N)/A(IJ)
      NP=N+1
      DO 48 NIP=2,N
      I=NP-NIP
      II=IJ-NIP
      V=Z(I)/A(II)
      IP=I+1
      IJ=II
      DO 47 J=IP,N
      II=II+1
47    V=V-A(II)*Z(J)
48    Z(I)=V
      RETURN
C     COMPUTE THE INVERSE MATRIX FROM FACTORS GIVEN IN A
      ENTRY MC11F(A,N,IR)
      IF (IR.LT.N) RETURN
      A(1)=1./A(1)
      IF (N.EQ.1) RETURN
      NP=N+1
      N1=N-1
      II=2
      DO 53 I=2,N
      A(II)=-A(II)
      IJ=II+1
      IF (I.EQ.N) GO TO 51
      DO 50 J=I,N1
      IK=II
      JK=IJ
      V=A(IJ)
      DO 49 K=I,J
```

```
      JK=JK+NP-K
      V=V+A(IK)*A(JK)
49    IK=IK+1
      A(IJ)=-V
50    IJ=IJ+1
51    CONTINUE
      A(IJ)=1./A(IJ)
      II=IJ+1
      AA=A(IJ)
      IJ=I
      IP=I+1
      NI=N-I
      DO 53 J=2,I
      V=A(IJ)*AA
      IK=IJ
      K=IJ-IP+J
      I1=IJ-1
      NIP=NI+IJ
      DO 52 JK=K,I1
      A(JK)=A(JK)+V*A(IK)
52    IK=IK+NIP-JK
      A(IJ)=V
53    IJ=IJ+NP-J
      RETURN
      END
```

```
      SUBROUTINE FUNCT (N,ARG,VAL,GRAD,RR,IND,XL,E,BEND,AXIAL,FY,FIVAL,M
     1ARK,PEN,NM,GG1,GG2,GG3,GG4,GG5,GG6,DG3,DG4,DG5,DG6)
C
C     VARIABLE         MEANING
C     DG1-DG6          DERIVATIVES USED IN DEFINITION OF GRADIENTS
C     GRAD             GRADIENT VALUE
C     VAL              FUNCTION VALUE
C
      DIMENSION ARG(NM), GRAD(NM), GG1(NM), GG2(NM), GG3(NM), GG4(NM), G
     1G5(NM), GG6(NM), XL(NM), E(NM), BEND(NM), AXIAL(NM), DG3(NM), DG4(
     2NM), DG5(NM), DG6(NM), IND(NM)
      DIMENSION GRSUM(10)
      MARK=MARK+1
C
C     MAKE SURE NO ARGUMENTS ARE NEGATIVE.
C
      EPS2=10.
      DO 1 I=1,NM
      IF (ARG(I).LT.0.) ARG(I)=EPS2**2/(2.*EPS2-ARG(I))
    1 CONTINUE
      FB=0.6666667*FY
      DO 10 I=1,NM
      XI=ARG(I)
      EI=E(I)
      INDI=IND(I)
      XLI=XL(I)
      BI=BEND(I)
      AI=AXIAL(I)
      IF (XI.EQ.52.) XI=52.01
      IF (XI.EQ.20300.) XI=20299.99
      GG1(I)=1./(XI/52.-1.)
      GG2(I)=1./(20300./XI-1.)
```

```
      AKLR=1.
      IF (INDI.EQ.1) GO TO 5
      IF (XI.GT.9000.) GO TO 2
      AKLR=XLI/(XI/0.21583)**0.25
      DAKLR=-XLI/(0.86332*(XI/0.21583)**1.25)
      GO TO 3
2     AKLR=XLI*12841./(XI+174498.)
      DAKLR=-XLI*12841./(XI+174498.)**2
3     GG3(I)=1./(200./AKLR-1.)
      DG3(I)=-200.*DAKLR/AKLR**2
      CC=SQRT(118.4352*EI/FY)
      IF (AKLR.GE.CC) GO TO 4
      FS=1.66667+(0.375*AKLR)/CC-(AKLR**3)/(8.*CC**3)
      DFS=DAKLR*((0.375/CC)-(0.375/CC**3*AKLR**2))
      FA=(1.-(AKLR**2)/(2.*CC**2))*FY/FS
      DFA=(-FY*DFS/FS**2)-((4.*CC**2*FS*AKLR*DAKLR*FY-2.*AKLR**2*FY*CC**
     12*DFS)/(2.*CC**2*FS)**2)
      GO TO 6
4     FA=5.14936*EI/AKLR**2
      DFA=-10.29872*EI*DAKLR/AKLR**3
      GO TO 6
5     FA=0.6*FY
      DFA=0.
      GG3(I)=0.
6     IF (XI.GT.9000.) GO TO 7
      FORMLB=SQRT(60.6*XI+84100.)-290.
      IF (FORMLB.EQ.0.) FORMLB=0.0000001
      FB1=BI/FORMLB
      FORMLC=SQRT(60.6*XI+84100.)*FORMLB**2
      IF (FORMLC.EQ.0.) FORMLC=0.0000001
      DFB1=-30.3*BI/FORMLC
      FA1=AI/(SQRT(XI)*0.464)
```

```
      DFA1=-1.079*AI/XI**1.5
      GO TO 8
    7 FB1=1.876*BI/(XI-8056.3)
      DFB1=-1.876*BI/(XI-8056.3)**2
      FA1=AI/(0.0039*(XI+2300.))
      DFA1=-AI*0.0039/(0.0039*XI+8.97)**2
    8 IF ((FA1/FA).LT.0.15) GO TO 9
      IF (INDI.EQ.1) GO TO 9
      FE=5.14936*(EI/AKLR**2)
      DFE=-10.29872*EI*DAKLR/AKLR**3
      FA2=FA1
      DFA2=DFA1
      CM=0.85
      GG4(I)=1./(1.-(FA1/FA+CM*FB1/((1.-FA2/FE)*FB)))
      FORMLD=FA**2
      FORMLE=FE**2
      FORMLF=(FB-FB*FA2/FE)**2
      IF (FORMLD.EQ.0.) FORMLD=0.0000001
      IF (FORMLE.EQ.0.) FORMLE=0.0000001
      IF (FORMLF.EQ.0.) FORMLF=0.0000001
      DG4(I)=-((FA*DFA1-FA1*DFA)/FORMLD+(CM*DFB1*(1.-FA2/FE)*FB-CM*FB1*(
     1-(FE*FB*DFA2-FB*FA2*DFE)/FORMLE))/FORMLF)
      GG5(I)=1./(1.-(FA1/(0.6*FY)+FB1/FB))
      DG5(I)=-(DFA1/(0.6*FY)+DFB1/FB)
      GG6(I)=0.
      DG6(I)=0.
      GO TO 10
    9 GG4(I)=0.
      DG4(I)=0.
      GG5(I)=0.
      DG5(I)=0.
      FORMLA=FA1/FA+FB1/FB
```

```
      IF (FORMLA.EQ.1.) FORMLA=0.9999998
      GG6(I)=1./(1.-FORMLA)
      DG6(I)=-(DFA1/FA+DFB1/FB)
10    CONTINUE
C
C     ASSEMBLE FUNCTION AND GRADIENT VALUES.
C
      IF (MARK.NE.1) GO TO 12
      PENS=0.
      DO 11 I=1,NM
      PENS=PENS+GG1(I)+GG2(I)+GG3(I)+GG4(I)+GG5(I)+GG6(I)
11    CONTINUE
      PEN=RR*PENS
12    EPS1=RR/(FIVAL+PEN)
      PENSUM=0.
      DO 24 I=1,NM
      GRSUM(I)=0.
      IF ((1./GG1(I)).GE.EPS1) GO TO 13
      PENSUM=PENSUM+(2.*EPS1-(1./GG1(I)))/EPS1**2
      GRSUM(I)=GRSUM(I)-0.01923/EPS1**2
      GO TO 14
13    PENSUM=PENSUM+GG1(I)
      GRSUM(I)=GRSUM(I)-0.01923*GG1(I)**2
14    IF ((1./GG2(I)).GE.EPS1) GO TO 15
      PENSUM=PENSUM+(2.*EPS1-(1./GG2(I)))/EPS1**2
      GRSUM(I)=GRSUM(I)+20300./((EPS1**2)*(ARG(I)**2))
      GO TO 16
15    PENSUM=PENSUM+GG2(I)
      GRSUM(I)=GRSUM(I)+(20300./ARG(I)**2)*GG2(I)**2
16    IF (GG3(I).EQ.0.) GO TO 18
      IF ((1./GG3(I)).GE.EPS1) GO TO 17
      PENSUM=PENSUM+(2.*EPS1-(1./GG3(I)))/EPS1**2
```

```
      GRSUM(I)=GRSUM(I)-DG3(I)/EPS1**2
      GO TO 18
17    PENSUM=PENSUM+GG3(I)
      GRSUM(I)=GRSUM(I)-DG3(I)*GG3(I)**2
18    IF (GG4(I).EQ.0.) GO TO 22
      IF ((1./GG4(I)).GE.EPS1) GO TO 19
      PENSUM=PENSUM+(2.*EPS1-(1./GG4(I)))/EPS1**2
      GRSUM(I)=GRSUM(I)-DG4(I)/EPS1**2
      GO TO 20
19    PENSUM=PENSUM+GG4(I)
      GRSUM(I)=GRSUM(I)-DG4(I)*GG4(I)**2
20    IF ((1./GG5(I)).GE.EPS1) GO TO 21
      PENSUM=PENSUM+(2.*EPS1-(1./GG5(I)))/EPS1**2
      GRSUM(I)=GRSUM(I)-DG5(I)/EPS1**2
      GO TO 22
21    PENSUM=PENSUM+GG5(I)
      GRSUM(I)=GRSUM(I)-DG5(I)*GG5(I)**2
22    IF (GG6(I).EQ.0.) GO TO 24
      IF ((1./GG6(I)).GE.EPS1) GO TO 23
      PENSUM=PENSUM+(2.*EPS1-(1./GG6(I)))/EPS1**2
      GRSUM(I)=GRSUM(I)-DG6(I)/EPS1**2
      GO TO 24
23    PENSUM=PENSUM+GG6(I)
      GRSUM(I)=GRSUM(I)-DG6(I)*GG6(I)**2
24    CONTINUE
      FV=0.
      DO 25 I=1,NM
      FV=FV+SQRT(ARG(I))*XL(I)
25    CONTINUE
      VAL=FV+PENSUM*RR
      DO 26 I=1,NM
      GRAD(I)=XL(I)/(2.*SQRT(ARG(I)))+GRSUM(I)*RR
```

```
26    CONTINUE
      RETURN
      END
```

The vita has been removed from
the scanned document

OPTIMIZATION OF STEEL FRAMES VIA

PENALTY FUNCTIONS

by

William F. Cofer

(ABSTRACT)

The purpose of this study is to develop a computer program to minimize the weight of steel frames using a penalty function technique.

The program is developed and several example problems are analyzed. All structures are designed in accordance with the AISC code. One of the problems is also analyzed by the Fully Stressed Design method, and the penalty function solution has less weight, but takes more computer time. The penalty function solution is checked by hand and found to be a reasonable design.

In this investigation, the penalty function method is found to be simple to use, reliable, versatile, and fairly economical for small structures.