PRACTICAL OPTIMIZATION OF STEEL HIGHWAY
BRIDGE BEAMS

MAY 1971 - NUMBER 8

BY

R.H. BUSEK

JHRP

JOINT HIGHWAY RESEARCH PROJECT

PURDUE UNIVERSITY AND
INDIANA STATE HIGHWAY COMMISSION

Final Report

PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS

TO: J. F. McLaughlin, Director
Joint Highway Research Project

May 11, 1971

FROM: H. L. Michael, Associate Director
Joint Highway Research Project

File No.: 7-4-17

Project No.: C-36-56Q

The attached Final Report titled "Practical Optimization of Steel Highway Bridge Beams" is submitted for acceptance as fulfilling the objectives of the Plan of Study "Optimum Design of Continuous Highway Bridge Girders" approved by the Advisory Board on February 17, 1970. Mr. Robert H. Busek, Graduate Assistant in Research on our staff, authored the report and conducted the research under the direction of Professors J. T. Gaunt and A. D. M. Lewis.

A program for optimization of a rolled section highway bridge girder is presented with either minimization of weight or cost. Either composite or non-composite design theory can be used. Two published design examples were solved by the computer and the results are reported and compared to the published results.

The research was financed from JHRP funds and is submitted with a request that comments on the report are encouraged. A report of application of the results by the ISHC would certainly be welcome.

Respectfully submitted,

Harold L. Michael

Harold L. Michael
Associate Director

HLM:ms

cc: F. L. Ashbaucher
W. L. Dolch
W. H. Goetz
W. L. Grecco
M. J. Gutzwiller
G. K. Hallock

M. E. Harr
R. H. Harrell
M. L. Hayes
E. M. Mikhail
R. D. Miles
J. W. Miller

C. F. Scholer
M. B. Scott
W. T. Spencer
N. W. Steinkamp
H. R. J. Walsh
K. B. Woods
E. J. Yoder

Final Report

PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS

by

Robert H. Busek
Graduate Assistant in Research

Joint Highway Research Project

File: 7-4-17
Project: C-36-56Q

Purdue University
Lafayette, Indiana
May 11, 1971

Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

ACKNOWLEDGMENTS

The author expresses his appreciation and thanks to Professor John T. Gaunt and Professor Albert D. M. Lewis, who were co-chairmen of his advisory committee, for their help in selecting this topic and for their guidance, leadership and comments throughout the course of the research project.

The financial support of the Joint Highway Research Project is gratefully acknowledged. The guidance and help of Professor Harold L. Michael, the Associate Director of the Project, is appreciated.

The confidence, expressed in him by Neil H. Bettigole, is acknowledged by the author. Many thanks go to Linda and Frances for their support, help and friendship during the past two years at Purdue.

To his family, who, although they remained many miles away, always expressed their love and concern, as well as an interest in the research, the author is eternally grateful.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES.	vi
LIST OF SYMBOLS.	vii
ABSTRACT	x
CHAPTER I - INTRODUCTION.	1
1.1 Structural Optimization.	1
1.2 Background Review.	3
1.3 Scope of the Investigation.	5
CHAPTER II - GIRDER ANALYSIS	7
2.1 Design Loads	7
2.2 Analysis Theory	10
2.3 Analysis Program	19
2.4 Load Subroutine	27
CHAPTER III - GIRDER DESIGN.	32
3.1 Girder Design Theory.	32
3.2 Fatigue Design.	38
3.3 Design Details.	41
3.4 Design Program.	45
CHAPTER IV - PROBLEM OPTIMIZATION.	54
4.1 Optimization Theory	54
4.2 Objective Function	61
4.3 Constraints.	63
CHAPTER V - COMPUTER PROGRAM	69
5.1 Description of the MAIN Program	69
5.2 Sample Problems	78

	Page
CHAPTER VI - SUMMARY AND CONCLUSIONS.	85
6.1 Summary	85
6.2 Recommendations for Further Research	86
BIBLIOGRAPHY.	88
APPENDICES	
APPENDIX A - SUBROUTINE DESCRIPTIONS.	91
APPENDIX B - SELECTED PROGRAM NOMENCLATURE.	95
APPENDIX C - PROGRAM DATA	108
READIN Subroutine	108
Method of Data Input	113
Input Data Samples	123
APPENDIX D - COMPUTER PROGRAM LISTING	127

LIST OF TABLES

Table		Page
2.1	AASHO Load Designations and Parameters.	8
3.1	Values of the Modular Ratio	36
3.2	Fatigue Constants.	40
5.1	Composite Section Properties	79
5.2	Unit Costs	79
5.3	Costs and Weights for Problem No. 1.	80
5.4	Costs and Weights for Problem No. 2.	82

Appendix
Table

C-1	Input Data for Sample Data Problem No. 1	124
C-2	Input Data for Sample Data Problem No. 2	126

LIST OF FIGURES

Figure		Page
2.1	Determination of Compatibility Equations. . . .	15
2.2	Sample Girder for Equation Development	18
2.3	Analysis Program Flowchart	20
3.1	Standard Composite Girder Cross Section	35
3.2	Facia Girder Composite Cross Section	35
3.3	Design Program Flowchart	46
5.1	Flowchart for the MAIN Program	70
5.2	Optimum Design for Problem No. 1	81
5.3	Optimum Design for Problem No. 2	83

LIST OF SYMBOLS

$a(x)$	=	the influence line function
a_{ij}	=	the influence coefficient at point i
A_{pl}	=	the area of a cover plate
A_r	=	the area of the longitudinal steel reinforcement
A_{web}	=	the area of the steel section web
b	=	the width of the steel section flange
c	=	the distance to the extreme steel fiber
E_c	=	the modulus of elasticity of concrete
E_s	=	the modulus of elasticity of steel
f	=	the longitudinal flexural stress
f_{all}	=	the allowable flexural stress
f'_c	=	the 28-day concrete strength
f_{ij}	=	the flexibility coefficient
F_a	=	the allowable column stress for the stiffeners
F_b	=	the allowable bending stress
F_r	=	the allowable stress reduced for fatigue
F_u	=	the minimum tensile strength of steel
F_v	=	the allowable weld metal stress
F_y	=	the yield point stress of steel
$g(x)$	=	the distributed load function
I_i	=	the moment of inertia of element i

I_{hmc}	=	the moment of inertia of the high modulus concrete section
I_{lmc}	=	the moment of inertia of the low modulus concrete section
I_{st}	=	the moment of inertia of the steel section
ℓ	=	the unsupported length of the compression flange
ℓ_i	=	the length of the subelement i
$L_{cp_{min}}$	=	the minimum cover plate length
L_w	=	the length of a weld
M_{DL}	=	the moment due to the dead load
M_i	=	the moment at section i
M_{LL}	=	the moment due to the live load
M_{SLL}	=	the moment due to the superimposed live load
M_{tot}	=	the total moment at a given section
n	=	the modular ratio for concrete
n_c	=	the number of shear connectors per section
P_j	=	the concentrated load at point j
Q	=	the statical moment of the area above or below a section, taken about the neutral axis
R	=	the algebraic ratio of the minimum stress to the maximum stress
S_r	=	the range of horizontal shear per inch
S_u	=	the ultimate strength of a shear connector
t	=	the required weld size
$t_{cp_{max}}$	=	the maximum cover plate thickness
t_f	=	the flange thickness of the rolled section

t_w	=	the web thickness of the rolled section
v	=	the unit shear stress
V	=	the external vertical shear
V_r	=	the range of external vertical shear due to the live load plus impact
w	=	the magnitude of a uniformly distributed load
W_{pl}	=	the width of a cover plate
y_{lmc}	=	the extreme fiber distance for the low modulus composite section
y_{hmc}	=	the extreme fiber distance for the high modulus composite section
y_{st}	=	the extreme fiber distance for the steel section
Z_r	=	the allowable shear per connector

ABSTRACT

Busek, Robert Henry. M.S.C.E., Purdue University, June 1971. PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS. Major Professors: J. T. Gaunt and A. D. M. Lewis.

A program for the optimization of a rolled section highway bridge girder is presented. The girder is designed by minimizing either the weight or the cost. The cost function used is a representative model of the total cost of the girder, including both material and fabrication costs. The function presented is a sample and can be altered if a different cost function is more appropriate.

The program produces the optimum design of a rolled section girder, using either composite or noncomposite design theory. The required cover plates are determined, using both static and fatigue stress conditions. The fatigue stresses are based on 500,000 cycles of stress. If a design is composite, 7/8-in. shear connector studs are provided in the positive moment region. No connectors are placed in the negative moment region. All design theory, analysis theory, and optimization constraints are based on the 1969 AASHTO bridge specifications. The method of influence lines is used in the analysis.

The exhaustive search technique is used as the main method of optimization. The cover plate thicknesses are

determined by using the method of interval halving, and the locations of the cover plate splices are determined by using dynamic programming.

Computer solutions for two published design examples were obtained and compared to the published results.

CHAPTER I

INTRODUCTION

1.1 Structural Optimization

Optimization is the process of obtaining "the best" solution to a problem based on a given criterion. Structural optimization, therefore, is the process of obtaining the "optimum" solution to a structural problem. This optimum is obtained by using analysis and design theory and an optimum-seeking method suited to the problem.

Structural optimization has only recently been approached systematically. One of the reasons for this is that the computer is a relatively new tool for the engineer. When doing an optimization, there are a large number of calculations which must be completed. Performance of these calculations by hand would be very time consuming. The computer gives the engineer a quick and fairly inexpensive method of calculating quantities and making decisions based on these calculations.

Another reason for the recent prominence of structural optimization is the rising cost of construction. Since the costs of material and labor have risen steadily over the past years, and since the trend for the future seems to be in the same direction, the engineer must place more emphasis

on the economics of design. By using optimization procedures, the designer may actually reduce construction costs by minimizing the total cost of the structure.

The formulation of the optimization problem depends on the type of structure being designed. For any problem, the design variables, the design constraints and the criterion (or the objective function) of the optimization must be defined. It is the values of the design parameters which are desired so that the value of the objective function is a minimum and the constraints are not violated.

The design parameters required depend on the type of optimization problem and on the theoretical equations used. The specification used to control the design specifies the necessary constraints for the problem. In most problems, both equality and inequality constraints are used.

The objective function can be any desirable measure of the effectiveness of the design. Some of the measures which have been used are weight, cost, cost-benefit ratios and reliability. For the highway bridge girder, minimum weight has been used for many years, because of the relationship which exists between weight and cost. In this optimization problem, however, the cost function used is more realistic because it includes the fabrication costs.

The general optimization problem can be expressed in the following mathematical form:

$$\begin{aligned} &\text{Find the variables } x_i \quad i = 1, 2, \dots, m \\ &\text{which optimize } G(x_i) \end{aligned} \quad 1.1$$

$$\text{where} \quad g(x_p) = 0 \quad p = 1, 2, \dots, n \quad 1.2$$

$$h(x_k) \geq 0 \quad k = n, \dots, m \quad 1.3$$

$$x_j \geq 0 \quad j = 1, 2, \dots, m \quad 1.4$$

The variable vector, x , must be determined so that the equality constraints (Equation 1.2) and the inequality constraints (Equations 1.3 and 1.4) are satisfied and the objective function, $G(x)$, is optimized. The G function can be either maximized or minimized depending on the type of problem.

The optimum-seeking methods which are available to the structural designer are numerous. Many of the methods are general in nature and can be used for any type of optimization. There are, however, some specific methods which are best suited for particular types or formulations of problems.

1.2 Background Review

The methods of optimization can be divided into direct and indirect methods. The indirect methods are mathematical methods which can use the gradient of the algebraic form of the objective function to solve a given problem. Direct methods are ones which make trial solutions in some organized manner. Each solution is compared to the present optimum and a decision is then made as to the next design

step. Each of the types has its own advantages and disadvantages when used in a particular structural optimization problem.

Structural optimization problems have been approached in the past as minimum weight design problems. Drucker and Shield [1 and 2] have presented some of the general minimum weight design theory. Faulkes [3] and Heyman [4] have used this criterion in optimizing frames, and Krishnan and Shetty [5 and 6] and Haug [7] have used it in elastic beam design problems. Hahn [8] extended the work to include plastic design of beams and simple structures. The assumption made in these minimum weight problems is that the cost of the girder is proportional to its weight.

Computer design methods have been developed for highway bridge design. Sturman, Albertson, Cornell and Roesset [9] developed a program called BRIDGE which can be used to make comparative studies of different types of bridges. The system can also completely design a particular structure, but it makes no attempt to optimize the design. Since the complete optimization of a highway bridge is a long and complicated problem, the approach so far has been to optimize the individual parts of the bridge system.

The optimization of girders has also been approached by using the minimum weight method. Edward [10] and Holt and Heithecker [11] have attempted to optimize the girder by minimizing the cross-sectional area. This in turn minimizes the weight of the structure.

The total cost optimization has been developed for both building and highway girders. Annamalai [12] uses the back-track programming method for welded plate girders, developed by Lewis [13], to optimize the design of building plate girders based on the AISC 1969 specifications. Okuba [14] solves the girder problem by using a linear programming method.

The work by Razani, Goble and DeSantis [15, 16, and 17] on the optimum design of plate girders has led to the development of a computer-aided design system. Goble and DeSantis [18 and 19], with the aid of the Ohio Department of Transportation, developed a program referred to as Girder Automated Design - I or GAD-I. The program completely designs the constant depth plate girder and specifies all of the details of the design, including the flange thicknesses, the flange splice locations and the locations of the transverse and longitudinal stiffeners if they are required.

1.3 Scope of the Investigation

Since the GAD-I program has been developed, it has been possible to design the optimum welded plate girder for a highway bridge. This girder, however, is not necessarily the actual optimum design for the problem. There is a possibility that either a composite or noncomposite rolled-section girder may be more economical than the plate girder. This is particularly true if the spans of the girder are relatively short.

In order to determine the true optimum design, a computer-aided design system for the rolled-section bridge girder has been developed. With this program, a given problem can be designed with a plate girder and a rolled-section girder and the true optimum can be chosen directly.

In the present study, the American Association of State Highway Officials bridge code [20] is used to control the design of a highway bridge girder. The cross section of the girder is limited to the 36-in. and 33-in. wide-flange rolled sections. The design can be either composite or noncomposite and can have cover plates if they are required. The problem is limited to a maximum of four continuous spans designed for all types of static and fatigue highway loads. The fatigue conditions are based on 500,000 cycles of stress. If the design is composite, then 7/8-in. diameter shear connector studs are provided at the proper spacing. The method of exhaustive search is used in the program because of the limited number of rolled sections of proper size. Each of the eighteen possibilities is designed, using the optimum-seeking methods of interval halving and dynamic programming to determine the required cover plates.

CHAPTER II

GIRDER ANALYSIS

2.1 Design Loads

The first problem that confronts the designer of a particular structure is to determine the required design-load conditions for that structure. These conditions usually vary with the size, purpose and location of the structure. The actual conditions are often designated by design specifications and can be found in design codes.

The design of a highway bridge is controlled by the American Association of State Highway Officials code [20]. This code, known as the AASHTO code, states that all of the girders in a bridge structure must be designed for the largest truck load and the corresponding equivalent lane load which the bridge must sustain. The code designates the loads by the size of the truck being considered, i.e., HS20. Once the required designation is chosen, the actual loading parameters can be determined. These parameters include the axle loads, the axle spacings, the equivalent lane loads and the corresponding concentrated loads. Table 2.1 contains a summary of the load parameters for all of the AASHTO truck designations.

Table 2.1: AASHTO Load Designations and Parameters

Des.	Truck Loadings						Equivalent Lane Loading		
	Axle Loads			Axle Spacings			Uniform	Concentrated	
	1	2	3	1 - 2	Min 2 - 3	Max 2 - 3		Moment	Shear
H10	4 ^k	16 ^k		14'			.32 ^k /'	9.0 ^k	13.0 ^k
H15	6 ^k	24 ^k		14'			.43 ^k /'	13.5 ^k	19.5 ^k
H20	8 ^k	32 ^k		14'			.64 ^k /'	18.0 ^k	26.0 ^k
HS15	6 ^k	24 ^k	24 ^k	14'	14'	30'	.48 ^k /'	13.5 ^k	19.5 ^k
HS20	8 ^k	32 ^k	32 ^k	14'	14'	30'	.64 ^k /'	18.0 ^k	26.0 ^k

If the bridge is being designed as part of the Interstate Highway System, the girders must also be designed for a specified interstate loading condition. This loading condition is based on the dimensions and weight of the largest military vehicle which would use the structure during a national emergency. The loading consists of two, 24-kip concentrated loads with an axle spacing of 4 ft. The effect of this loading must be calculated in order to determine if it governs the girder design.

Another possible girder loading condition is produced by pedestrian pathways. If the bridge is supplied with sidewalks, then the sidewalk loads must be calculated. According to the AASHTO code, the following set of formulas is used in calculating the sidewalk load:

$$P = 85 \quad \text{for } L_S \leq 25 \text{ ft}$$

$$P = 60 \quad \text{for } 25 \text{ ft} < L_S \leq 100 \text{ ft}$$

$$P = \left(30 - \frac{3000}{L_S} \right) \times \left(\frac{55-W}{50} \right) \quad \text{for } L_S > 100 \text{ ft} \quad 2.1$$

where P = the load in pounds per square foot.

L_S = the loaded length of sidewalk in feet.

W = the width of sidewalk in feet.

The total sidewalk load is distributed over all of the girders in the structure. Although the moments due to this loading condition are usually small, they are included to provide a complete analysis.

Besides the above loads, which are considered to be the live loads for the structure, there are also dead loads which must be considered. Dead loads include the weight of the steel girder, the weight of the deck slab and the weight of the bridge railing, light fixtures, and similar loads placed on the constructed bridge. If the design of the girder is noncomposite, then all of these loads are combined in one dead load condition. If the design is composite, however, the loads are placed into two groups because they act on different composite sections. The first group is considered the dead load and includes the weight of the slab and the steel girder. The second group is the superimposed dead load (sometimes referred to as the long-term live load). This group contains dead loads applied to the composite girder after the concrete has reached its maximum strength

capacity. An example of a superimposed dead load is the weight of the bridge railings and the light fixtures.

2.2 Analysis Theory

Once the loads which act on the structure have been defined, the stresses caused by these loads must be determined. The design loads must be placed on the structure so that maximum design conditions are developed at the critical points in the structure. The determination of these design values constitutes a complete analysis of the structure being designed.

Since the loading conditions, as shown above, for the highway bridge girder are complex in nature, the method of analysis which is most efficient is the method of influence lines. This method enables the structure to be loaded with a large number of loading conditions and the controlling maximum to be chosen from these conditions.

An influence line is a graph showing the variation in a particular function at a point with respect to the position of a unit load. The ordinates of the influence line are referred to as the influence coefficients. The ordinate at point j for the function F at point i , usually called the influence coefficient a_{ij} , is defined as the value F_i due to a unit load at point j [21]. With this definition various theorems can be developed pertaining to the use of influence lines. Norris and Wilbur [22] describe the following four basic influence line theorems:

Theorem 1 - The maximum value of a function due to a single concentrated load is found by placing the load over the maximum influence coordinate.

Theorem 2 - The value of a function due to a single concentrated load equals the product of the load and the influence coefficient at the load location.

Theorem 2 is stated mathematically as

$$F_i = P_j \times a_{ij} \quad 2.2$$

From Theorem 1, the maximum value of F_i , $F_{i_{\max}}$, is found when the single concentrated load is placed at the point where a_{ij} is the maximum. By using superposition, the value of the function F due to a series of concentrated loads can be found from Equation 2.3.

$$F_i = \sum_{j=1}^m P_j \times a_{ij} \quad 2.3$$

where m is the number of loads.

Theorem 3 - The maximum value due to a distributed load occurs when the structure is loaded over those portions which have influence coefficients with the same sign as the character of the function desired.

Theorem 4 - The value of the function due to a distributed load is the product of the load

magnitude and the net area under the portion of the influence line loaded.

Theorem 4 is stated mathematically as

$$F = \int_r^s g(x) \cdot a(x) dx \quad 2.4$$

where r and s are the load limits.

The derivation of Equation 2.4 can be found in the structural analysis texts by Norris and Wilbur [22] and by Shedd and Vawter [23].

There are two simplifications that can be made to Equation 2.4 when dealing with highway structures. In all cases, the distributed load, $g(x)$, can be replaced by a constant representing the uniformly distributed loading condition. The uniform load is usually represented by w . The second simplification deals with the influence line function, $a(x)$. Since this function is usually fairly complex in nature and too difficult to obtain, the coefficients are determined at a number of discrete points and the values in between are found using linear interpolation. The value of the function F , due to a uniform loading condition, over a piecewise continuous influence line, can be determined by using Equation 2.5.

$$F_i = \sum_{k=1}^n \frac{w}{2} \left(a_{i,k} + a_{i,k+1} \right) \ell_k \quad 2.5$$

where n is the number of elements.

With the theorems and the equations expressed above, any effect of a loading condition can be obtained, provided the influence coefficients can be calculated. The influence line is developed by computing the required influence coefficients for each position of a unit concentrated load. The process is quite simple for statically determinate girders, but becomes much more complicated for indeterminate or continuous girders.

For the statically determinate problem, all that is required to solve for any influence coefficient are the equilibrium equations. The unit load is placed on the structure and the reactions are determined using the equations shown as Equation 2.6:

$$\sum M_{\text{left end}} = 0$$

2.6

$$\sum F_{\text{vertical}} = 0$$

Once the reactions are found, the influence coefficient for any function can be determined.

If the girder being designed is statically indeterminate, then equations of compatibility are required. One compatibility equation is needed for each redundant reaction. These equations refer to the requirement that all the displacements throughout the structure are consistent. Willems and Lucas [24] note that the compatibility conditions require that the displacements at supports must be externally

satisfied. In the particular case of unyielding supports, the total displacement must be equal to zero.

The compatibility equations can be developed with reference to Figure 2.1. The structure is made statically determinate by removing the interior supports (Figure 2.1b). With the determinate structure defined, the deflections at the interior support points, due to the external loads on the simple girder, can be determined. Next the flexibility coefficients are determined (Figure 2.1c and d) and the equations can be formed:

$$R_2 f_{22} + R_3 f_{32} - D_2 = 0 \quad 2.7a$$

$$R_2 f_{23} + R_3 f_{33} - D_3 = 0$$

In matrix form, these equations may be written

$$\begin{bmatrix} f_{22} & f_{32} \\ f_{23} & f_{33} \end{bmatrix} \begin{bmatrix} R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} D_2 \\ D_3 \end{bmatrix} \quad 2.7b$$

$$\text{or} \quad [F] [R] = [D] \quad 2.8$$

The F matrix is called the flexibility matrix and is independent of the external loads on the structure. The coefficients of this matrix represent the deflections at the interior support locations due to unit loads at the interior support locations. These deflections can be found by using the moment-area theorems.

2.1 a: Indeterminate Structure

2.1 b: Determinate Structure

2.1 c: Unit Load at Point 2

2.1 d: Unit Load at Point 3

Figure 2.1: Determination of Compatibility Equations

The moment-area theorem used to determine the deflection is called the second moment-area theorem. The theorem states that the deflection at point 1, relative to a tangent at point 2, is equal to the moment of the area under the bending moment diagram, between points 1 and 2, about point 1 divided by the flexural rigidity of the girder. Pippard and Baker [25] give the following mathematical form of the theorem:

$$D = \frac{1}{EI} \int_1^2 Mx dx \quad 2.9$$

If the moment of inertia varies along the girder and the moment diagram is given as a piecewise continuous curve, varying linearly between the analysis points, then Equation 2.9 takes the following form:

$$D = \frac{1}{E} \sum_{i=1}^n \frac{1}{2} (M_i + M_{i+1}) \frac{l_i}{l_i} x_i \quad 2.10$$

where x_i is the distance from the center of the element to the center of moments.

Once the required deflections are determined and the compatibility equations, 2.7a, are defined, the values of the redundant reactions can be determined. With these reactions and the equilibrium equations, 2.6, the remaining reactions can be calculated.

In order to develop the influence line, the unit load must be located at each point along the structure. This means that there would be NA different matrix equations of

the type shown in Equation 2.8, where NA is the number of analysis points. Since the flexibility matrix, however, does not change with the location of the external load, the only part of the equation that will vary is the D matrix. For this reason it is possible to set up one matrix of NA columns for all the positions of the unit load. Also by adding two more rows, which represent the equilibrium equations, all of the reaction influence coefficients can be determined at the same time. Shown in Figure 2.2 is a three span girder with seven analysis points. The matrix equation required to solve for the reaction influence coefficients is

$$[CC] [R] = [RIL] \quad 2.11$$

where

$$[CC] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & L_1 & L_2 & L_3 \\ 0 & d_{33} & d_{53} & 0 \\ 0 & d_{35} & d_{55} & 0 \end{bmatrix}$$

and

$$[RIL] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \end{bmatrix}$$

This equation is the same as the one which appears in the analysis development by Goble and DeSantis [18].

The solution to this matrix equation can be approached in a number of ways. The most efficient method is one in which the final solution for the reaction influence lines

Figure 2.2: Sample Girder for Equation Development

is found in the RIL matrix. By using a Gaussian search technique, the CC matrix is reduced to an identity matrix by simple row and column operations common to matrix algebra. By repeating the same operations on the RIL matrix, the reaction influence coefficients are determined and stored in the RIL matrix. A complete description of the Gaussian search technique employed in the solution is given by Goble and DeSantis [19].

With the solution for the reaction influence coefficient complete, all of the remaining influence lines for the complete analysis can be calculated by simple statics. The influence lines can then be loaded and all the design conditions for the problem can be determined. With these conditions, the girder can be designed and optimized.

All of the theorems and equations discussed above are used to determine the design reactions, moments and shears.

After a final design has been chosen, the AASHTO code requires that the deflections be calculated and compared to certain allowable live-load deflections. In order to calculate the maximum deflections, the deflection influence coefficients must be determined. The amount of calculations, however, can be reduced by employing Maxwell's law of reciprocal deflections. The law says that, on a girder, the deflection at point 1 due to a unit load at point 2 is equal to the deflection at point 2 due to a unit load at point 1. This enables the deflection influence coefficients for a point j to be determined by calculating all of the deflections along the girder due to a unit load at point j . The maximum deflections can then be calculated by using the influence coefficients and the design loads. These maximums can then be compared to the allowable deflections specified by the AASHTO code.

2.3 Analysis Program

The analysis portion of the computer program uses the theory described in Section 2.2 to develop the values required by the design portion of the program. A basic flowchart for the analysis program is shown in Figure 2.3. The step numbers, used in the following program description, refer to the numbers in parentheses on the flowchart.

Figure 2.3: Analysis Program Flowchart

Figure 2.3: (Cont'd.)

Figure 2.3: (Cont'd.)

Step 1

The analysis program is started by calling the ANAL subroutine. This subroutine calls on the other subroutines in the proper order required to produce a set of usable design conditions.

Step 2

The REAC subroutine is called on to solve for the reaction influence lines. The simple-span girder, defined as the problem girder with the interior supports removed, is loaded with a unit load at each analysis point, and the bending moments are determined. These simple-span bending moments are transferred to the EQSET subroutine.

Step 3

The EQSET subroutine is used to calculate the simple-span deflections at the interior support points. The value of the tangent at the left end (TALE) is determined by using Equation 2.10 and is then used in the following equation to determine the required deflections:

$$d_i = \text{TALE} \times \text{COOR}(I) - \text{WA}(I) \quad 2.12$$

Step 4

The RIL matrix is prepared by using the deflections obtained in step 3 and the equilibrium equations. The RIL matrix is the right-hand-side matrix of Equation 2.11.

Step 5

The coefficient matrix (CC) is prepared by using the appropriate values of the deflections found in step 3 and the coefficients of the equilibrium equations. This CC matrix corresponds to the matrix CC in Equation 2.11.

Step 6

The SOLVE subroutine is called to determine the solution to Equation 2.11. By using a Gaussian search technique, the CC matrix is reduced to an identity matrix and the reaction influence coefficients are determined and stored in the RIL matrix.

Step 7

Steps 2 thru 6 are repeated using a different moment of inertia for each subelement. By doing this, the different influence coefficients for the steel section, the low modulus concrete section, and the high modulus concrete section are determined. Each set of coefficients is stored in its proper location in the RIL matrix. Control is now passed back to the ANAL subroutine where the reaction influence coefficients are printed if IPT10 is greater than 2.

Step 8

The reaction influence line is loaded by calling on the LOAD subroutine and the values of the ten (10) loading

conditions are returned from this subroutine. Since the LOAD subroutine is used extensively in the analysis, it is described in Section 2.4.

Step 9

The design reactions (DESREA) are now determined. The dead-load reaction is set equal to STORE(1) and the super-imposed dead-load reaction is set equal to STORE(10). The positive and negative live-load design reactions are equal to the maximum live-load reactions possible. These maximums are obtained by combining the sidewalk live load with the other individual live loads of the same sign until the largest value of the sums is found. If the sidewalk live load is zero, then the positive and negative design reactions are respectively equal to the largest positive and negative truck load, lane load or interstate load.

Step 10

Steps 8 and 9 are repeated for each support along the girder. The maximum number of repetitions is NS. The DESREA array is now printed out if the LPT10 parameter is greater than zero.

Step 11

The reaction influence coefficients are used to determine the moment influence line at a particular point. This influence line is then loaded by the LOAD subroutine and the ten loading conditions are determined. The design moments

moments (DESMOM) are determined in the same manner as the design reactions in step 9.

Step 12

Step 11 is repeated for each analysis point along the structure and the DESMOM array is printed if IPT10 is greater than 0. The maximum number of repetitions is NA.

Step 13

The reaction influence coefficients are used to determine the shear influence line and this line is loaded by the LOAD subroutine to calculate the values of the ten loading conditions. These conditions are used to calculate the design shears (DESSH) in the same manner as described in step 9.

Step 14

Step 13 is repeated for the locations immediately to the left and to the right of each analysis point. The maximum number of repetitions is $2NA-2$. (The minus 2 is necessary because there is only one repetition made for the first and the last analysis points.) The DESSH array is printed if IPT10 is greater than 0.

Step 15

The analysis of the girder is complete and the control is returned to the main program for design.

2.4 Load Subroutine

The load subroutine is used to load any given influence line with the complex loadings required in a highway girder analysis. It should be noted that a given influence line actually consists of three different influence lines, if the design is composite. There is one line for the steel section, one for the low modulus concrete section, and one for the high modulus concrete section. All three lines are calculated in the ANAL subroutine and are transferred to the LOAD subroutine at the same time.

Step 1

The steel section influence line is set into the RO array and the properties of this line are determined by the ILPROP subroutine. The properties which are required by the LOAD routine are the positive and negative areas and the locations of the maximum and minimum ordinates.

Step 2

The unit dead load is calculated. Using the area properties of the influence line, the dead-load loading is found by using Equation 2.5 with the load extending over the entire girder. The dead-load loading is then stored in STORE(1).

Step 3

If the girder is noncomposite, then the superimposed dead-load loading (STORE(10)) is set equal to zero. If the design is composite, then the low modulus concrete section influence line is set into the RO array and the properties are found. The superimposed dead load is determined and the values of STORE(10) is calculated using Equation 2.5 with the load extending over the entire girder.

Step 4

If the design is noncomposite, then the live-load determination can be started immediately. If the design is composite, however, the high modulus concrete section influence line is set into the RO matrix and the properties are determined by the ILPROP subroutine.

Step 5

The IMPACT subroutine is called on to determine the impact factor required by the AASHO code. This factor (HT) is determined from the following formula:

$$HT = 1.0 + \frac{50}{DING - 125} \quad 2.13$$

The code places an upper limit of 1.3 on the impact factor.

Step 6

The positive lane load is found by loading the positive area with the equivalent lane load and by placing the required concentrated load over the maximum ordinate. The

load is increased by the impact factor and stored in STORE(2). The negative lane load is obtained by loading the negative area with the equivalent lane load and by placing the required concentrated load over the minimum ordinate. The AASHTO code also requires a second concentrated load placed in one other span as to produce the maximum negative moment. The negative lane load is then increased by the impact factor and stored in STORE(3).

Step 7

The sidewalk live load is found by multiplying the uniform sidewalk load by the positive and negative areas respectively. The positive sidewalk load is placed in STORE(8) and the negative sidewalk load is placed in STORE(9).

Step 8

If the interstate loading is desired, the influence line is loaded with the two interstate concentrated loads which are spaced 4 ft apart. There are three loading conditions considered in the interstate loading. The first two have one load on the maximum ordinate and the second 4 ft to the right or 4 ft to the left. The third condition has the two loads centered about the maximum ordinate. The total maximum condition is increased by the impact factor and stored in STORE(4). To find the negative interstate load, the same three conditions are repeated using the

minimum ordinate as the reference. The negative interstate load is stored in STORE(5).

Step 9

The truck load analysis begins by moving the truck from left to right. A complete analysis consists of placing each of the truck's axles over the point in question and calculating the loads due to the combinations of the remaining axles.

Step 10

The first step is to place the rear axle on the analysis point and to use the maximum spacing for the front axles. The values of the influence line are determined by using the interpolation subroutine (ILINT). The resultant load is found by using Equation 2.3. This load is compared to the current maximums. (The positive maximum is found in POS and the negative maximum is found in SEG.) The calculation is repeated using the minimum spacings for the front axles.

Step 11

Next the front axle is placed on the point in question and the second axle is located the specified fixed distance away. The effect of these two axles is calculated and compared with 40 percent of the current maximums. It is assumed that if the effect of the front axles is less than 40 percent of the maximums, then the effect of the rear

axle can be neglected. If the effect of the rear axle must be calculated, the rear axle is placed at the minimum distance, at the maximum distance, and at each analysis point in between. Goble and DeSantis [18 and 19] refer to this as "wiggling the rear axles." Each total effect is compared to the current maximums and stored in POS and SEG if required.

Step 12

The final loading for a particular point is to place the second axle on the point and the first axle the specified fixed distance ahead. The effect of these two loads is calculated and compared in the same manner as in step 11. If necessary, the rear axle is wiggled and the total effects are compared to the current maximums.

Step 13

The truck load is repeated for the truck moving from right to left. The final maximums are increased by the impact factor and the positive load is stored in STORE(6) and the negative load is stored in STORE(7).

Step 14

The STORE array is now returned to the ANAL subroutine where it is used to determine the design loading conditions.

CHAPTER III

GIRDER DESIGN

3.1 Girder Design Theory

With the design conditions defined at each analysis point, the cross section can now be proportioned to carry the necessary design stresses. If the design is noncomposite, then the simple, pure bending theory can be used to determine the stresses. The equation which expresses this pure bending theory is known as the flexure formula and is shown as Equation 3.1:

$$f = M_{\text{tot}} \times \frac{c}{I} \quad 3.1$$

or

$$f = M_{\text{tot}}/S$$

where S = the section modulus of the section and
is equal to I/c .

A derivation of this formula is presented by Byars and Snyder [26] and many other mechanics textbooks.

The section used in a noncomposite design is symmetrical about the horizontal axis. This allows the top and bottom fiber stresses of the steel section to be equal. The concrete in a noncomposite design is assumed to carry no longitudinal stresses because there is no provision for the

transfer of shear between the slab and the beam. If provision is made for this transfer, then the design becomes composite.

The design of a composite bridge girder takes advantage of the high compressive strength available in the concrete slab. Some of the advantages of composite action expressed by Viest, Fountain, and Singleton [27] and by McCormac [28] are:

1. A savings in steel costs is realized,
2. The beam depth can be reduced,
3. A larger percentage of steel is in tension,
4. The deflections are reduced due to the increase in the beam stiffness,
5. A composite section can withstand a greater overload than a noncomposite design.

The one disadvantage of the composite action is the cost of providing the shear connectors. This disadvantage, however, only takes precedence over the advantages in short, lightly loaded structures.

In designing a composite girder by the AASHTO code, the properties of the steel beam and slab are established on the basis of the moment of inertia of the composite section. Bresler, Lin and Scalzi [29] give the following two basic assumptions used in the analysis and design of composite beams for highway girders:

1. The slab is connected throughout the length of the girder,

2. The stress and the strain are linear across the depth of the member.

The AASHTO code, unlike the American Institute of Steel Construction specification [30], does not allow any slippage for the shear connectors.

The composite design section shown in Figure 3.1 is composed of a rolled section, an effective slab section, a bottom cover plate, and a haunch. The section could also be designed without the bottom plate. It is also possible for the slab to be on one side of the girder. This type of section would be used for a fascia girder and is shown in Figure 3.2.

The effective slab section is defined as a section of the bridge deck having an effective slab width and a thickness equal to the structural depth of the slab. The AASHTO code defines the effective slab width of the section in Figure 3.1 as not exceeding any of the following:

1. One-fourth of the girder span,
2. The distance center-to-center of girders,
3. Twelve times the slab thickness.

If the composite section is similar to the fascia girder shown in Figure 3.2, then the effective slab width should not exceed any of the following:

1. One-twelfth of the span length,
2. One-half the center-to-center distance to the adjacent girder,
3. Six times the slab thickness.

Figure 3.1: Standard Composite Girder Cross Section

Figure 3.2: Facia Girder Composite Cross Section

The section properties of the composite section are determined by the transformed area method. This method assumes that the bond between the concrete and steel is strong enough so that the strains in the concrete and steel at the junction are equal. Due to this assumption, the relationship between the stresses at a given distance from the neutral axis is

$$f_c = \frac{1}{n} f_s \quad 3.2$$

where

$$n = E_s/E_c$$

The factor n is called the modular ratio. The area of the concrete slab is transformed into an equivalent area of steel by dividing the effective slab width by the modular ratio. The values of the modular ratio specified by the AASHO code are shown in Table 3.1.

Table 3.1: Values of the Modular Ratio

Concrete Strength	n
2000 - 2400	15
2500 - 2900	12
3000 - 3900	10
4000 - 4900	8
5000 or more	6

The actual form of the flexure formula used in a composite design contains three terms instead of just one as in Equation 3.1:

$$f = \frac{M_{DL} y_s}{I_{st}} + \frac{M_{SLL} y_{lmc}}{I_{lmc}} + \frac{M_{LL} y_{hmc}}{I_{hmc}} \quad 3.3$$

The first term in Equation 3.3 represents the stress at a point due to the dead load of the structure. This dead load is carried by the steel section alone. The properties of the steel section are independent of the slab dimensions.

The second term in the equation represents the stress due to the superimposed dead load. This load must, according to the AASHTO code, be carried by the low modulus composite section. The properties of this section are calculated by using a concrete area transformed by a modular ratio equal to $3n$. The increased ratio is used to account for creep effects.

The third term in the equation represents the stress due to the live load on the structure. The live load, since it is placed on the structure after the concrete has reached its maximum strength, acts on the high modulus composite section. The properties of this section are based on a concrete area transformed with a modular ratio of n .

By using Equation 3.1 for a noncomposite design and Equation 3.3 for a composite design, the steel girder cross section can be proportioned so that the resulting flexural stresses are within the allowables required by the design

code. If the design is composite, then the compressive stress in the extreme concrete fiber must also be within the accepted code allowable.

With the girder designed for flexural stresses, the shear stresses must be determined. The horizontal and vertical shear stresses at any point in the cross section can be found from Equation 3.4:

$$v = \frac{VQ}{bI} \quad 3.4$$

where b = the width of the section where the shear stress is desired.

Byars and Snyder [26] derive this formula and also demonstrate that, for an I-shaped section, the shear stress across the web can be assumed to be uniform. Since the stress is uniform, the AASHO code states that all of the external vertical shear is carried by the steel girder web and can be calculated using Equation 3.5:

$$v = V_{\text{tot}}/A_{\text{web}} \quad 3.5$$

For this calculation, the area of the web is equal to the product of the overall depth of the rolled section and the web thickness.

3.2 Fatigue Design

In a highway bridge girder there is a fluctuating stress level at every point because of the alternating loading and unloading of the structure due to the live

loads. This continuous fluctuation causes dislocations in the crystalline structure of the steel, and eventually causes cracks and other localized failures. This type of failure is referred to as a fatigue failure.

The fatigue failures in a structure may occur at loads which are well below the allowable static loading conditions, because the stress is continually fluctuating over a particular range. The larger the fluctuation, the worse the condition. The fatigue condition becomes extremely critical when the stress alternates between tension and compression.

The number of fluctuations, or cycles, for which a girder can be designed varies from 100,000 cycles to 2,000,000 cycles. The exact number of cycles used depends on the use and location of the bridge.

The method used in fatigue design is to reduce the allowable stresses at critical points. The AASHO code provides the following formulas for this fatigue reduction:

$$F_r = \frac{k_1 f_{ro}}{1 - k_2 R} \quad 3.6$$

and

$$F_r = \frac{.55 F_y}{1 - \left(\frac{.55 F_y}{k_1 f_{ro}} - 1 \right) R} \quad 3.7$$

where

$$k_1 = 1.0 + \alpha \left(\frac{F_u}{58000} - 1.0 \right)$$

and R = the algebraic ratio of the minimum stress to the maximum stress.

The k_2 , f_{ro} , and α are the fatigue constants. These fatigue constants have been determined primarily from fatigue tests. A description of some of the tests and the results are given in the U. S. Steel bridge manual [31].

The values of the fatigue constants depend on the number of cycles of stress, the category, type and location of material and the yield point of the base metal. The values required for a particular fatigue calculation can be found in the AASHO code. The constants for 500,000 cycles of stress are given in Table 3.2. These are the values which are used in the design program.

Table 3.2: Fatigue Constants

AASHO Cate- gory	Type and Location of Metal	Type of Maximum Stress	Equation	500,000 cycles		
				f_{ro}	α	k_2
A	Base Metal	Tension Compr.	3.6	20500	.78	.55
			3.7	13300	.78	--
D	Metal Adjacent to Butt Weld	Tension Compr.	3.6	17200	.23	.62
			3.7	10600	.23	--
F	Metal Adjacent to Fillet Weld	Tension Compr.	3.6	12000	0.0	1.0
			3.6	12000	0.0	1.0
G	Weld Metal	Shear	3.6	10800	.36	.55

In the design of a rolled section, fatigue is a major factor in the cover plate design. The ends of the cover plates are determined by using fatigue calculations for the base metal adjacent to a fillet weld (Category F). The allowable stresses in this case are greatly reduced, thus requiring that the plates be cut off in regions of low moments. The remaining categories are checked in the design, but they usually do not govern the design of the girder.

3.3 Design Details

After the cover plates required for a particular rolled section are determined, the details of the design must be calculated. These design details include the determination of the weld sizes for the cover plates, the design of the required bearing stiffeners and the calculation of the shear connector spacing for a composite girder. In the optimization problem, each of these details must be completed for each design because they add a substantial cost to the girder.

The welds which must be determined for each cover plate include the end welds and the seal welds. The end welds are designed to carry the maximum flexural stress developed at a fixed distance from the ends of the plate. The code sets this distance as 1.5 times the width of the cover plate. The required stress is determined using either Equation 3.1 or 3.3. When the stress in the plate is known, Equation 3.8 is used to determine the force in the plate:

$$\text{Force} = f_{\text{tot}} \times A_{\text{pl}} \quad 3.8$$

When the force in the plate is known, Blodgett [31] shows that the size of the required fillet weld is

$$t = \frac{\text{Force}}{.707 L_w f_{\text{all}}} \quad 3.9$$

In the above equation the L_w is the total length of the weld which, according to the AASHTO code, is four times the plate width. The allowable weld metal stress (f_{all}) is the lesser value of the allowable static shear stress and the allowable fatigue stress, calculated using Equation 3.6 and the constants listed under category G from Table 3.2. The fillet weld equation then becomes

$$t = \frac{\text{Force}}{2.828 W_{\text{pl}} f_{\text{all}}} \quad 3.10$$

The required weld size is usually given in sixteenths of an inch. The code requires that a minimum size be used depending on the thicknesses of the plates being joined.

The design of the seal weld for a cover plate is basically similar to that of the end weld except that shear stress is used instead of flexural stress. Equation 3.4 is used to determine the maximum shear stress developed along the cover plate and then Equation 3.11 is used to determine the size of the seal weld:

$$t = \frac{1}{2} \frac{v}{(.707) f_{\text{all}}} = \frac{v}{1.414 f_{\text{all}}} \quad 3.11$$

The 2 in the denominator is present because there is a seal weld on each side of the cover plate. The code specifies that the seal weld must extend over the entire length of the plate in order to prevent corrosion and must have a minimum thickness of 5/16 in.

The bearing stiffeners are required to transfer the large web shear stresses, at the reactions, to prevent web crippling. In most of the wide-flange rolled sections, bearing stiffeners are not required. AASHTO requires bearing stiffeners in those girders which have shear stresses in excess of 75 percent of the allowable shear stress. If stiffeners are required, they are designed as columns. The column section used is composed of the two stiffener plates and a portion of the web which is 18 times the web thickness in length. The fillet welds which connect the stiffener to the girder web are designed similar to the seal welds for the cover plates.

The shear connectors for a composite girder are designed to transfer the horizontal shear from the slab to the steel section. These shear connectors are placed transversely across the flange at either regular or variable spacings. The connectors are designed for fatigue as presented by Slutter and Fisher [33] and checked for ultimate strength. The spacing at any location can be found as follows:

$$\text{Spacing} = \frac{Z_r n_c}{S_r} \quad 5.12$$

where S_r = the range of horizontal shear per inch. The allowable shear per connector depends on the number of fatigue cycles and the type and the size of the connector used. The horizontal shear per inch is found by using an equation similar to 3.4:

$$S_r = V_r Q/I \quad 3.13$$

where V_r = the range of vertical shear due to live load plus impact.

The moment of inertia used in Equation 3.13 is the transformed moment of inertia in the positive region. If shear connectors are placed in the negative moment region, then the moment of inertia, for the negative region, is that of the steel girder and the longitudinal slab reinforcement.

In order to satisfy the ultimate strength requirements for the connectors, a specified number of connectors must be placed between the points of maximum moment and the points of zero dead-load moment. Equation 3.14 is used to determine the number of connectors required for the ultimate strength:

$$N = \frac{P}{.85 S_u} \quad 3.14$$

where P = the lesser value of the force in the steel girder and the concrete slab.

The ultimate strength of a shear connector depends on the size and the type of connector.

The final shear connector requirement which must be satisfied is an additional number of connectors to develop the slab stress around the points of contraflexure. The number of connectors required is

$$N_c = \frac{A_r f_r}{Z_r} \quad 3.15$$

where A_r = the area of the steel reinforcement,
and f_r = the range of live load stress in the
reinforcement (may be taken as 10,000 psi).

These additional connectors are needed only if the negative moment region is noncomposite.

3.4 Design Program

The design portion of the computer program operates on each wide-flange rolled section and completely designs the girder for the given problem. A complete design consists of the determination of the cover plates and their required welds, the design of the bearing stiffeners and the calculation of the shear connector spacing.

The design program is started by determining the lightest section which will carry the loads without cover plates. This section is completely determined and its cost becomes the base cost for the optimization problem. The program now enters a loop which designs all of the remaining rolled sections. A flowchart for this loop is shown in Figure 3.3.

Figure 5.3: Design Program Flowchart

Figure 3.3: (Cont'd.)

Figure 3.3: (Cont'd.)

Step 1

The rolled section being designed is transferred to the THDES subroutine where the thicknesses of the cover plates required for each subelement are determined.

Step 2

The THDES subroutine first determines the maximum moment in the subelement. With this moment determined, the rolled section is tried without plates. The individual stresses are determined and checked against the allowable static and fatigue base metal stresses. If these stresses are not violated, then no cover plates are required for the element.

Step 3

If a cover plate is required at a particular subelement, the width of the plate is determined based on the flange width of the rolled section. The thickness of the plate is determined by using interval halving. The interval halving technique uses only thicknesses of sixteenths of an inch. The thickness is varied until the allowable stress condition is satisfied. The thickness and width of the required plate are stored in the proper array.

Step 4

Steps 2 and 3 are repeated for each of the subelements along the girder. With all the plate sizes determined, the control is passed back to the main program.

Step 5

The properties of each subelement are calculated by using the SFMIC subroutine and these properties are transferred to the CPDES routine.

Step 6

The CPDES subroutine uses the fatigue theory discussed in Section 3.2 to determine the actual location of the ends of the required cover plates. The routine is used to determine all of the necessary conditions required to determine the cutoff point for each plate. The actual point is located by the TERMIN subroutine.

Step 7

The TERMIN subroutine uses an increasing step size optimization technique to determine the ends of the plates in the most efficient manner. The end of the plate is defined as the point along the girder where the stress does not violate the fatigue allowable stress. Once this point is determined, control is passed back to the CPDES routine.

Step 8

Steps 6 and 7 are repeated until the fatigue limits of each cover plate on the top and bottom of the girder are determined. With the limits found, the final cover plates for the top and bottom of the girder are determined in the FLOPT subroutine.

Step 9

The FLOPT routine uses the cost array and the cover plate arrangement to determine the optimum location of the cover plate splices. A dynamic programming method is used to accomplish this optimization. Upon completion of this optimization, the COVPL array is determined in its final form and control is returned to the main program.

Step 10

The CPWDES subroutine is now used to design all of the cover plate welds. The end welds for a particular plate are designed for both ends and the larger thickness is retained as the design. If the plate is butt welded to other plates at both ends, then the end weld thickness is set to zero. The seal welds are designed by the subroutine using Equation 3.11. After these calculations for each cover plate along the girder are completed, the control is returned to the main program.

Step 11

The STIFF routine is called to design the bearing stiffeners for each support. If the support does not require a stiffener, then the values of the BEAR array are set equal to zero. If a stiffener is required, the width and thickness are calculated such that the allowable bearing and the allowable compressive stresses are not violated. The welds which hold the stiffeners in place are calculated and the control is returned to the main program.

Step 12

The CONDES routine is used to determine the spacing of the shear connectors. The shear connector design is based on the use of 7/8-in. shear connector studs. The required spacing at each analysis point is determined and rounded off to the nearest lower multiple of three inches. The spacing is corrected for the ultimate strength discussed in Section 3.3 and the CONSP array is filled with the required spacing information. The routine sets the least number of spaces at a particular value at three. Upon completion of the spacing design, the control is returned to the main program.

Step 13

Once the spacing for the connectors is determined, the design of a rolled section is complete. The OBJECT subroutine is now called on to determine the weight and, if necessary, the cost of the given design.

Step 14

The cost or weight of the design is compared to the current optimum by the KEEP routine. If the girder is an improved design, the elements are stored in the proper arrays. If the girder is not an improved design, then the old optimum is retained.

Step 15

Steps 1 to 14 are repeated for each of the remaining rolled sections. After the last section is designed, the program takes the current optimum and continues into the remaining optimization phases of the program.

CHAPTER IV

PROBLEM OPTIMIZATION

4.1 Optimization Theory

Optimization, as stated by Beveridge and Schechter [34], "is the collective process of finding a set of conditions required to achieve the best result for a given situation." In other words, optimization is the process of obtaining a set of design parameters which either maximizes or minimizes a particular function. The function being optimized is referred to as the objective function. This function may be either mathematically simple or complex, depending on the nature of the problem. The actual form of this objective function usually determines the most appropriate method of optimization.

The design of a physical system ideally contains three steps. These steps, according to Wilde and Beightler [35], are:

1. Determination of the interaction of the system variables,
2. Development of a simple measure of effectiveness,
3. Determination, development, and solution of the most effective optimum-seeking method.

The first step is handled by developing a set of constraints for the particular problem being solved. These constraints can be imposed by design codes, tradition, or the designer. This step usually requires the greatest amount of effort.

The second step is the actual development of the objective function. The formulation of the function can range from being simple to being so complex it is almost impossible. The degree of difficulty in describing the function depends on the type of problem, the type of effectiveness being measured, and the degree of accuracy desired. The less the degree of accuracy of the objective function, the less accurate will be the developed optimum solution.

The third step leads the designer to many different types of optimization procedures. The advantages of each method vary with the problem being solved. It is the designer's responsibility to know the limitations, the advantages and the disadvantages of each method as they relate to the particular problem being solved.

In the design program for the optimization of a rolled-section highway bridge girder, there are actually two different optimum-seeking methods employed. The method of exhaustive search is used as the basic design method and dynamic programming is used to determine the locations of the cover plate splices. The thicknesses of the cover plates are determined by using interval halving to solve the governing equation.

The exhaustive search technique, sometimes called exhaustive enumeration or brute force method, simply evaluates the objective function for each of the problem possibilities and picks the optimum solution directly. This method can be very tedious if the number of design possibilities is large. It is therefore used only when the overall design space for the problem is fairly limited in size.

Since the wide-flange sections used for highway girders include only the 36- and 33-in. sections, there are only 18 sections to be considered. The value of the objective function is determined for each wide-flange section after it has been completely designed, including the cover plates, the bearing stiffeners and the shear connectors. The design with the minimum value of the objective function is then chosen as the optimum design.

By setting up the wide-flange sections in descending weight order, the number of executions in the exhaustive search can be reduced. Starting with the heaviest, and proceeding in the descending weight order, each section is tried without cover plates at the critical moment location, until the lightest girder which is acceptable is determined. By assuming that the cost of the shear connectors and stiffeners for a uniform section are independent of the section used, all of the sections heavier than the one which has been determined can be eliminated from the search. This

means that a number of the rolled sections need not be designed completely and a resulting savings in computation costs is realized.

Interval halving is a technique which can be used to solve equations when the simple, direct methods are inappropriate. The method uses a bounded interval which is continuously reduced in size until a certain tolerance is obtained. By reducing the interval size by a factor of one-half with each calculation, the total number of necessary calculations is greatly reduced and there is a resulting savings in calculating costs. The two necessary conditions for the interval halving technique are:

1. The function must be continuously increasing or decreasing.
2. The initial boundaries must be specified.

The first condition enables the method to completely eliminate one half of the interval with each calculation, and the second condition enables the first calculation value to be specified.

The method begins with the specified interval and calculates all the necessary information for the interval midpoint. If all of the constraints are satisfied for this midpoint value, then the value becomes the new upper boundary. This is possible because the function is continuously increasing or decreasing and the higher values of the variables will also satisfy the constraints, but will cause a

larger cast. If the constraints are not satisfied, then the midpoint value becomes the lower boundary. The new interval is used in the same manner as the initial interval. The entire procedure is repeated until the remaining interval is less than the tolerance desired. Either the upper or the lower bound can then be used as the optimum solution. In most problems the average of the two bounds is considered as the solution, provided that the final interval is small enough.

In determining the cover plate thicknesses for each subelement, there are many calculations required, including the section properties for each plate size, the bending stress at the point in question and the allowable stresses for the fatigue and static conditions. Since, for each moment condition and rolled section there is only one plate size which will give a stress equal to the allowable stress, the function for the thickness can be considered to be continuously decreasing. The initial boundary conditions can be specified by using the requirements of the AASHTO code. With the two conditions for interval halving satisfied, the method can be used for the determination of the cover plate thicknesses.

In order to use interval halving, however, the method must be adjusted so that the thicknesses will be determined in values of sixteenth-inch increments. This is accomplished by adjusting the initial interval size and by making the

boundaries integers. Each halving process then produces an integer which corresponds to the number of sixteenths of an inch in the plate thickness. The calculations are completed for the thickness and the interval is reduced. The process is continued until the final interval size is 1.0. The optimum plate size is then stored in the proper array location.

Dynamic programming is a method used in sequential system optimization problems. It can be applied to situations in which many decisions are required, as long as the decisions made at later stages do not affect the performance of earlier stages. A mathematical formulation of the dynamic programming method can be found in the optimization texts by Wilde and Beightler [35], Pierre [36] or Denn [37]. All of the mathematical formulations are based on Bellman's [38] principle of optimality.

Dynamic programming is a method of decomposition which divides the given problem into a set number of individual problems. In determining the optimum condition for section i , the only quantities which are used are the values of the variables in sections $i+1$ and i , and the value of the objective function for the system up to and including section $i-1$. The objective function is determined for all of the possible combinations of the variables in section i and $i+1$. The values of the objective function are then added to the value of the function for the problem through

section $i-1$. The optimum variable can then be directly chosen and placed in an appropriate array location defined by the section number i and the variable value in section $i+1$.

The method begins by calculating the variable combinations for all of the sections and their adjacent sections in the problem sequence. When the last section is reached, the final values of the objective function are calculated. By comparing these values, the optimum for the last section is determined. Then, by working backwards, all of the variables in each section can be determined. This process leads to an optimization of the entire problem.

The dynamic programming method is used to determine the final cover plate combination for a given rolled section. The actual problem is decomposed into sections corresponding to the cover plates which have been defined by the fatigue stress conditions. The variable of each section is the plate thickness, and the function being minimized is the total cost of the cover plate. By varying the thickness of each section and determining the total cost of the plate by the procedure described above, the optimum location of the cover plate splices can be located. This method is similar to the flange smoothing method used by Goble and Razani [17]. DeSantis and Goble [18] illustrate this method for the optimization of the flanges for a welded plate girder.

4.2 Objective Function

The term objective function refers to the means by which the program evaluates the merit of a particular design. For the rolled-section highway girder design, the function can be either the total weight or the total cost of the girder. The designer specifies which method of optimization is desired by the value used for IPT6. If IPT6 has a value of 0, then the optimization is based on cost. The equation for the total girder cost is

$$\begin{aligned}
 \text{Total Cost} = & \text{Cost of the rolled section} + \\
 & \text{Cost of the cover plates} + \\
 & \text{Cost of the shear connectors} + \quad 4.1 \\
 & \text{Cost of the bearing stiffeners} + \\
 & \text{Cost of the welding.}
 \end{aligned}$$

If the value of IPT6 is 1, then the optimization is based on the weight of the girder. The equation for the total girder weight is

$$\begin{aligned}
 \text{Total Weight} = & \text{Weight of the rolled section} + \\
 & \text{Weight of the cover plates} + \\
 & \text{Weight of the bearing stiffeners} + \quad 4.2 \\
 & \text{Weight of the shear connectors.}
 \end{aligned}$$

The weight function first determines the volume of steel used in each of the individual portions of Equation 4.2. A unit weight of 490 pounds per cubic foot is used to convert

the volumes into weights. These weights are added together to get the total weight.

If the cost of a girder is desired, then the weights are calculated as above and then converted into costs. The costs of the rolled section, the cover plates, and the bearing stiffeners are determined by multiplying each of the weights by their respective unit costs. These unit costs are expressed in dollars per pound.

The cost of the shear connectors consists of two separate elements. The first element is the material cost of the connectors and is based on the total weight of the connectors. The cost is calculated by multiplying the total connector weight by the unit cost. This unit cost is also expressed in dollars per pound. The second element is the installation cost and is based on the total number of connectors. This cost is calculated by multiplying the total number of connectors by the installation cost per connector.

The welding costs are determined for all the welds on the girder. The welds which are considered are the cover plate butt welds, the cover plate seal welds, and the welds used to connect the stiffeners to the girder. The weld cost can be found for any weld using the following equation:

$$C_{\text{weld}} = C_1 + C_2(L) + C_3(V) \quad 4.5$$

where C_1 is the fixed cost of the weld, $C_2(L)$ is the variable cost which is a function of the weld length and $C_3(V)$ is the variable cost which is a function of the weld volume.

The fixed cost of a weld accounts for all of the costs involved in the set-up time, the inspection, and the other fixed items of cost. The variable cost, based on the weld length, reflects the costs of x-ray inspection and joint preparation.

The variable cost, based on volume, reflects the actual cost of the weld material. The volume of a weld depends on the size and shape of the weld. For a fillet weld, the volume is based on a weld cross section which is triangular. For a butt weld, the volume is based on the cross section of a double vee butt weld. Each of these welds is the standard weld used in highway girder design.

The objective function for this problem is similar to the one found in the GAD-1 program developed by Goble and DeSantis [19]. This is done so that final comparisons between rolled-section girders and welded plate girders can be made. These comparisons can only be accurate if the objective functions for both types of design are similar in form and content.

4.3 Constraints

The problem constraints are the limitations placed on various parts of the design in order to limit the total design space. There are two types of constraints in an

optimization problem. One type of constraint, referred to as a side constraint, is imposed on the design by functional limitations. The second type, called a behavioral constraint, is imparted to the design by a specific specification.

A functional, or side, constraint used in the girder design problem is the total number of rolled sections designed. Since the design of a highway bridge girder is usually limited to the 36- and 33-in. wide-flange sections, the limited number of design possibilities actually becomes a side constraint for the problem.

The allowable width of the cover plates is also controlled by a functional constraint. The plates are limited to sizes of whole inches and must be smaller than the width of the rolled-section flange. The reason for this is to make the welding process more efficient. With this constraint, the widths of the cover plates are limited to 10, 14, or 15 in.

The behavioral constraints for the highway girder design are specified in the American Association of State Highway Officials bridge design code [20]. The AASHTO code specifies all of the design criteria, which must be converted into the problem constraints, for the design of a rolled-section girder. These behavioral constraints are divided into the following four groups:

1. Static stress constraints,
2. Fatigue stress constraints,

3. Cover plate size constraints,
4. Secondary design constraints.

The static stress constraints are based on the required allowable stresses with no regard to fatigue considerations. There are actually five different static stress conditions which must be satisfied before a design can be considered acceptable. Four of the conditions consider the bending stresses in the girder and the fifth considers the shear stress in the girder web. The five constraints are:

1. The allowable flexural tension stress in the extreme fibers of rolled shapes or built-up sections is $0.55 F_y$ where F_y is the yield point stress.
2. The allowable flexural compression stress in the extreme fibers of rolled shapes and built-up sections is $0.55 F_y$ when the compression flange is continuously supported by being embedded in concrete.
3. The allowable flexural compression stress in the extreme fibers of rolled shapes or built-up sections is determined by Equation 4.4:

$$F_b = 0.55 F_y \left[1.0 - \frac{3F_y (\ell/b)^2}{\pi^2 E} \right] \quad 4.4$$

This constraint reduces the allowable compressive stress in order to prevent the buckling of the compressive flange.

4. The allowable compression stress in the extreme fibers of the concrete slab is $0.4 f'_c$ where f'_c is the 28-day concrete strength.
5. The allowable shear stress in the girder web is $0.33 F_y$.

The fatigue stress constraints reduce the allowable stresses because of the fluctuations in stress caused by the repeated applications of the live load. The fatigue stress conditions are governed by Equations 3.6 and 3.7. The fatigue theory used in the design program is described in Section 3.2.

The maximum permissible thickness of a cover plate is governed by Equation 4.5:

$$t_{cp_{max}} = 1.5 t_f \quad 4.5$$

Equation 4.6 governs the length of the cover plate:

$$L_{cp_{min}} = 2 D + 3 \quad 4.6$$

where D is the depth of the rolled section in feet.

Equations 4.5 and 4.6 set the requirements for a cover plate on a given rolled section and make up the cover plate size constraints.

The final group of design constraints control the secondary details of the girder design. The details which

are controlled by this group are the shear connector spacings, the bearing stiffener design, and the weld design.

If the design is composite, then the following four constraints control the spacing of the shear connectors:

1. The shear connectors are spaced so that the horizontal shear is transferred from the slab to the steel girder. This spacing is designed according to fatigue stress conditions.
2. The design of the shear connector spacing is checked for ultimate strength.
3. Extra connectors must be provided at the points of contraflexure.
4. The maximum permissible spacing is 24 in.

The equations and the theory used in the design of these shear connector constraints can be found in Section 3.3.

The design of the bearing stiffeners is controlled by the following constraints:

1. Stiffeners are required at a support if the shear stress in the web at that support is greater than 75 percent of the allowable shear for girder webs.
2. The minimum thickness for a stiffener is given by

$$t_{\min} = \frac{b'}{12} \sqrt{\frac{I_y}{33000}} \quad 4.7$$

where b' is the width of the stiffener.

3. The bearing stress between the stiffener and the rolled-section flanges must be less than $0.80 F_y$.
4. The compressive stress in the stiffener must be less than the allowable column stress determined from Equation 4.8:

$$F_a = \frac{0.55F_y}{1.25} \left[1.0 - \frac{\left(\frac{0.75L'}{r} \right)^2 F_y}{4\pi^2 E} \right] \quad 4.8$$

where L'/r is the slenderness ratio of the stiffener and a section of the web which is $18t_w$ inches wide.

The final secondary design constraint is used to control the weld design. The allowable static shear stress in the weld material (F_v) is specified by Equation 4.9:

$$\begin{aligned} \text{If } F_y &\leq 36 \text{ ksi, } F_v = 12.4 \text{ ksi.} \\ \text{If } F_y &> 36 \text{ ksi, } F_v = 14.7 \text{ ksi.} \end{aligned} \quad 4.9$$

The allowable shear stress for the weld material must be reduced for fatigue by using Equation 3.6 and the constants under category G in Table 3.2.

CHAPTER V

COMPUTER PROGRAM

5.1 Description of the MAIN Program

The computer program for the design of rolled-section girders consists of two parts. The analysis portion is used to determine the design conditions for the problem and the design portion is used to determine the optimum combination of rolled section and cover plates. The entire program is composed of a MAIN program and thirty-four subroutines.

The purpose of the MAIN program is to call on the subroutines in the correct sequence to produce the optimum design for the particular girder. The following description is a step-by-step summary of the MAIN program. A flowchart for this program is presented in Figure 5.1.

Step 1

The rolled section information required for the design program is read from data cards. All the information for the eighteen 36- and 35-in. wide-flange sections is already on cards supplied with the program.

Step 2

The READIN subroutine is called to read all of the data for the design problem. A detailed description of READIN is presented in Appendix C.

Figure 5.1: Flowchart for the MAIN Program

Figure 5.1: (Cont'd.)

Figure 5.1: (Cont'd.)

Figure 5.1: (Cont'd.)

Figure 5.1: (Cont'd.)

Step 3

The minimum tensile strength used in the fatigue calculations is determined from the yield point stress of the material used in the problem. This tensile strength is a measure of the minimum ultimate strength of the steel.

Step 4

The ANAL subroutine is called to analyze the given problem and to determine all of the design reactions, the design moments, and the design shears. These conditions are used to determine the optimum girder for the problem.

Step 5

The minimum weight and the minimum cost variables are initialized at very high values.

Step 6

The UNIFST subroutine is used to design the uniform section girder. This girder is the lightest rolled section which satisfies all of the design constraints without the use of cover plates. This section then becomes the initial design.

Step 7

A design loop, which completely determines the girders composed of the sections lighter than the one used as a uniform section, is begun. This loop is described in Section 3.4 and a flowchart is presented in Figure 3.3.

Step 8

After the design loop is completed, the optimum design for the cycle is stored in the optimum storage arrays. This design is then placed into the working arrays of the program and all of the section properties are computed using the SFMIC subroutine.

Step 9

The new unit dead weight of the girder is determined and the problem is analyzed again, using the new distribution of the moments of inertia. If the problem is a continuous girder, then the ANAL subroutine is called to do the complete reanalysis. If the problem is a simple-span girder, the dead-load design conditions are changed directly in the MAIN program. This is the only change required because the moments in a simple-span girder are independent of the cross-section properties.

Step 10

If IPT10 is greater than 0, all of the design conditions are printed out.

Step 11

Using the new design conditions, the stiffeners and connector spacing for the optimum design are recalculated. All of the stresses in the girder are determined by the STRCHK subroutine and a new value of the objective function

is determined. All of the information regarding this design is then printed if IPT10 is greater than 0.

Step 12

If the design cycle parameter, IPT7, equals 2 and the maximum number of cycles has not been reached, the counter is increased by one and control is returned to step 5. Steps 5 thru 11 are repeated until the specified maximum number of cycles is reached. Control is now passed to step 15.

Step 13

If IPT7 equals 1, the total moment at the critical section is determined and the percent change from the previous analysis is computed. If this percentage is less than the specified percentage, control is passed to step 15. Otherwise, control is returned to step 5, where steps 5 thru 11 are repeated until the percentage is less than that specified.

Step 14

If IPT7 equals 0, no redesign cycles are required and control is passed to step 15.

Step 15

Using the final conditions, the rolled section used in the optimum design is completely recalculated using the sequence of steps described in Section 3.4.

Step 16

The required deflections are calculated, using the moment of inertia distribution of the final design. These deflections are calculated by the DEFLEC subroutine.

Step 17

All of the information pertaining to the final design is printed in the output. This includes the section properties, the required cover plates, the required stiffeners, the shear connector spacing and all of the deflections. The cost or weight of the design is also presented, depending on the value of IPT6.

Step 18

Control is returned to Step 2 where the program is either terminated or the data for a new design problem are read.

5.2 Sample Problems

In order to show the usefulness of the computer program, two sample girders have been designed. Both problems are presented in the U. S. Steel Manual [31]. The costs of the manual design and program design are compared below.

The first problem is a two-span girder having equal span lengths of 70 ft. The design is based on a composite section having the properties shown in Table 5.1. The design loading is the AASHO HS20 loading.

Table 5.1 Composite Section Properties

Property	Value
SLABWD	84.00 in
SLABTH	7.00 in
FPC	3000.00 psi
MR	8.00
AS	4.34 sq in
HAUNCH	1.875 in

Table 5.2 Unit Costs

Item	unit	Cost in Dollars
Rolled Section	1b	0.15
Cover Plates	1b	0.12
Stiffeners	1b	0.12
Connectors	1b	0.00
Connector Installation	ea	0.75
Fixed Weld Cost	ea	20.00
Weld Material Cost	cu in	2.00
Weld Length Cost	in	0.0

The optimization of the girder is based on the total cost. The unit costs used in the optimization procedure are shown in Table 5.2

The final design produced by the computer program is shown in Figure 5.2. The itemized costs and weights of both the computer design and the manual design are given in Table 5.3.

Table 5.3: Costs and Weights for Problem No. 1

Item	U. S. Steel Manual		Computer Program	
	Weight	Cost	Weight	Cost
Rolled Section	18900.0	\$2835.00	22400.0	\$3360.00
Cover Plates	2361.6	\$283.40	243.1	\$28.16
Stiffeners	---	---	---	---
Connectors	202.2	\$193.50	181.0	\$174.00
Welds	---	\$628.29	---	\$206.90
Totals	21463.8	\$3940.19	22824.1	\$3769.06

Table 5.3 shows that the optimization program produced a girder which is approximately 6.0 percent heavier than the girder found in the manual. The optimum design, however, costs 5.0 percent less than the manual design. Most of the cost reduction can be found in the cover plate cost. The total cost of the plates, including both material and fabrication costs, for the optimum girder is \$235.06. The

Notes: Use 4 connectors in each row.
 Use $5/16"$ end welds.
 All seal welds are $5/16"$.

Figure 5.2: Optimum Design for Problem No. 1

same costs for the manual design give a total of \$911.69 or approximately four times as much as the program solution. This cost difference offsets the increased cost of the rolled section used in the optimum design.

The second example is a four-span composite girder. The outer spans are 70 ft and the interior spans are 90 ft. The composite section properties, the loading conditions and the cost parameters are the same as for the first problem. The program design and the itemized costs and weights are shown in Figure 5.3 and Table 5.4 respectively.

Table 5.4: Costs and Weights for Problem No. 2

Item	U. S. Steel Manual		Computer Program	
	Weight	Cost	Weight	Cost
Rolled Section	43200.0	\$6480.00	51200.0	\$7670.00
Cover Plate	9855.0	\$1182.60	3509.6	\$420.93
Stiffeners	---	---	---	---
Connectors	475.0	\$458.00	348.0	\$336.50
Welds	---	\$1598.44	---	\$1251.68
Totals	53520.0	\$9719.04	55057.6	\$9679.11

Table 5.4 shows that the computer solution is approximately 2.9 percent heavier and only 0.5 percent cheaper than the manual design. These figures show that there is a slight saving in the computer solution. The small amount

Notes: Use 4 connectors in each row.
 All seal welds are 5/16".
 Use 5/16" and 1/4" end welds.

Figure 5.3: Optimum Design for Problem No. 2

indicates that the manual solution approaches, for these cost figures, the optimum design. The similar costs show that the computer not only approaches the optimum solution, but also produces a practical design.

The computer program is approximately 4000 cards in length and was developed using the CDC 6500 computer. The compilation time for the program is about 48 seconds. The execution times for the two sample problems were 60 and 40 seconds respectively.

The $10 \times 1/4$ plates, shown in Figure 5.3, violate width-to-thickness ratios of the AASHO code. The program is being modified such that the cover plates conform to the code requirements.

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

The cost optimization of a highway bridge girder, composed of a rolled section acting compositely with the bridge deck, has been formulated and programmed. An objective function, based on the material and the fabrication costs, has been developed and used in the examples. The design of the girder is controlled by the 1969 AASHO code.

Due to the complex loadings required on a highway bridge, the method of influence lines is used in the analysis. The analysis portion of the program produces the design conditions for the problem, which are used to design and optimize the required girder.

Since the number of possible designs is fairly limited, the program uses the exhaustive search technique. The techniques of interval halving and dynamic programming are also used within the design program. Interval halving is used to determine the thicknesses of the cover plates for the subelements and dynamic programming is used to determine the optimum location of the cover plate splices.

The computer program developed will solve a simple-span problem as well as a continuous problem of up to four spans. The girder can be either composite or noncomposite and is designed for static and fatigue loads. All the design details are developed and then listed in the output. No attempt is made to design the girder for deflection, but the total deflections for the final design are tabulated. The engineer has the option to optimize the design based on either the total weight or the total cost.

6.2 Recommendations for Further Research

1. Beams with different steel strengths may be considered.
2. Varying the rolled section at points of zero dead-load moment may be considered.
3. The effect of field splices on the optimization problem could be established.
4. A more accurate objective function could be developed, provided that cost information could be obtained from steel fabricators.
5. A complete optimization study of rolled-section girders and plate girders could be made using the program in conjunction with GAD-I.
6. The effect of the girder spacing on the optimum solution could be determined.

7. The effect of composite action in the negative moment region on the optimum solution could be studied.
8. The effect of a variable slab thickness on the problem optimization could be determined.
9. An optimization of the bridge deck and stringer portion of a highway bridge could be developed.
10. An optimization study of the various types of highway overpass structures, such as plate girders, rolled-section girders, reinforced and prestressed concrete girders and concrete slab bridges, could be made with the ultimate outlook on optimizing the entire bridge design project.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Drucker, D. C. and R. T. Shield: "Bounds in Minimum Weight Design," Quarterly of Applied Mathematics, 1957.
2. Drucker, D. C. and R. T. Shield: "Design for Minimum Weight," Proceedings on the 9th International Congress of Applied Mechanics, Brussels, 1956.
3. Faulkes, J.: "The Minimum Weight Design of Structural Frames," Proceedings of the Royal Society, London, 1954.
4. Heyman, J.: "On the Absolute Minimum Weight Design of Framed Structures," Quarterly Journal of Mechanics and Applied Mathematics, No. 12, 1959.
5. Krishman, S. and K. V. Shetty: "On the Optimum Design of an I-Section Beam," Journal of Aerospace Science, No. 26, 1959.
6. Krishman, S. and K. V. Shetty: "A Method of Minimum Weight Design for Thin-Walled Beams," Structural Engineer, No. 5, 1961.
7. Haug, E.: Minimum Weight Design of Beams with Inequality Constraints of Stress and Deflection, Ph.D. Thesis, 1966, Kansas State University.
8. Hahn, E. W.: Minimum Weight Elastic and Plastic Design of Beams and Simple Structures, Ph.D. Thesis, June 1969, Purdue University.
9. Sturman, G., Albertson, L., Cornell, C. and J. Roesset: "Computer-Aided Bridge Design," ASCE Journal of the Structural Division, Vol. 92, No. 6, Dec. 1966.
10. Edward, C. H. and L. H. Gary: "Minimum Weight Proportions for Steel Girders," ASCE Journal of the Structural Division, Vol. 95, Oct. 1969.
11. Holt, E. C. and G. L. Heithecker: "Minimum Weight Properties for Steel Girders," ASCE Journal of the Structural Division, Vol. 95, No. 10, Oct. 1969.

12. Annamalai, N.: Cost Optimization of Welded Plate Girders, Ph.D. Thesis, Purdue University, 1970.
13. Lewis, A. D. M.: "Backtrack Programming in Welded Girder Design," Proceedings of the Share-ACM-IEEE Design Automation Workshop, July 15-18, 1968, Washington, D. C.
14. Okuba, S.: Optimum Design of Composite Plate Girder Superstructures, CE Thesis, 1965, M.I.T.
15. Razani, R.: The Iterative Smoothing Method and Its Application to Minimum Cost Design of Highway Bridge Girders, Ph.D. Thesis, 1965, Case Institute of Technology.
16. Razani, R. and G. Goble: "Optimum Design of Constant Depth Plate Girders," ASCE Journal of the Structural Division, Vol. 92, April 1966.
17. Goble, G. and P. V. DeSantis: "Optimum Design of Mixed Steel Composite Girders," ASCE Journal of the Structural Division, Vol. 92, Dec. 1966.
18. Goble, G. and P. V. DeSantis: Girder Automated Design-I, Vol. 1 - User's Manual, Oct. 1968.
19. Goble, G. and P. V. DeSantis: Girder Automated Design-I, Vol. 2 - Maintenance Manual, Oct. 1968.
20. American Association of State Highway Officials, Standard Specifications for Highway Bridges, Tenth Edition, 1969.
21. Gaylord, E. H. and C. N. Gaylord: Structural Engineering Handbook, McGraw-Hill Book Company, New York, 1968.
22. Norris, C. H. and J. B. Wilbur: Elementary Structural Analysis, Second Edition, McGraw-Hill Book Company, New York, 1960.
23. Shedd, T. C. and J. K. Vawter: Theory of Simple Structures, J. Wiley and Sons, Inc., New York, 1941.
24. Willems, N. and W. M. Lucas: Matrix Analysis for Structural Engineers, Prentice-Hall, Inc., New Jersey, 1968.
25. Pippard, A. J. S. and J. Baker: The Analysis of Engineering Structures, Fourth Edition, American Elsevier Publishing Co., New York, 1968.

26. Byars, E. F. and R. D. Snyder: Engineering Mechanics of Deformable Bodies, International Textbook Co., Pennsylvania, 1963.
27. Viest, I. M., Fountain, R. S. and R. C. Singleton: Composite Construction in Steel and Concrete, McGraw-Hill Book Co., New York, 1958.
28. McCormac, J. C.: Structural Steel Design, International Textbook Co., Pennsylvania, 1965.
29. Bresler, B., Lin, T. Y. and J. B. Scalzi: Design of Steel Structures, Second Edition, John Wiley and Sons, New York, 1968.
30. American Institute of Steel Construction: Manual of Steel Construction, Seventh Edition, 1970.
31. United States Steel Company: Highway Structures Design Handbook, Volumes 1 and 2, 1965.
32. Blodgett, O. W.: Design of Welded Structures, The James F. Lincoln Arc Welding Foundation, Ohio, 1966.
33. Slutter, R. G. and J. W. Fisher: "Fatigue Strength of Shear Connectors," Highway Research Record, No. 147, Highway Research Board, Washington, D. C., 1966.
34. Beveridge, G. S. G. and R. S. Schechter: Optimization: Theory and Practice, McGraw-Hill Book Company, New York, 1970.
35. Wilde, D. J. and C. S. Beightler: Foundations of Optimization, Prentice-Hall, Inc., New York, 1967.
36. Pierre, D. A.: Optimization Theory with Applications, J. Wiley and Sons, Inc., New York, 1969.
37. Denn, M. M.: Optimization by Variational Methods, McGraw-Hill Book Company, New York, 1969.
38. Bellman, R.: Dynamic Programming, Princeton University Press, New Jersey, 1957.

APPENDICES

APPENDIX A

APPENDIX A

SUBROUTINE DESCRIPTIONS

The subroutines used in the program are listed and described in the following list. The listing is presented in alphabetical order.

- ALLOW - returns the allowable bending stress for any location along the top and the bottom of the girder, disregarding all fatigue conditions.
- ANAL - determines the design conditions (reactions, moments and shears) for a particular problem.
- BASFAT - returns the allowable base metal fatigue stress at a given point on the girder.
- CONDES - designs the shear connectors for a given girder.
- CPDES - designs the cover plates for a given rolled section.
- CPWDES - designs the required end and seal welds for each cover plate required on the girder.
- DEFLEC - determines the required deflections for the final girder design.

- EQSET - calculates the simple beam deflections, at the interior support points, which are used in calculating the influence lines for the reactions.
- FLOPT - uses the optimization method of dynamic programming to determine the final cover plate arrangement along a girder.
- ICALC - determines the moments of inertia and the distances to the lower extreme fibers for any rolled section and cover plate combination. It is also used to find the similar properties for the composite sections by treating the slab as a top cover plate.
- ILINT - returns the interpolated value of any point along an influence line using straight line interpolation.
- ILPROP - determines the various required properties of a given influence line including the crossover points, the positive and negative areas and the locations of the maximum and minimum ordinates.
- IMPACT - determines the impact factor for any particular analysis function.
- KEEP - decides whether or not to retain the given design as the present optimum.

- LOAD - places all the required loads on the influence line in such a manner as to return the maximum loading conditions required.
- MAIN - calls on the remaining subroutines in the correct order to produce the optimum design.
- OBJECT - contains the function being minimized during the optimization of the design problem.
- PRCON - prints the connector design for a given girder.
- PRCOV - prints the cover plate information for a girder.
- PRDEF - prints the deflections for the final girder design.
- PRSEC - prints the section table for a given girder.
- PRSTIF - prints the bearing stiffener information for a given girder design.
- PRSTR - prints the stress table for a given girder design.
- REAC - determines the reaction influence lines for a continuous girder using the moment-area theorems.

- READIN - reads in all the pertinent data required for a particular design problem.
- SEMIC - calculates the properties for each subelement along the girder.
- SORT - places the required cover plate thicknesses in ascending order so they can be used in the FLOPT subroutine.
- STIFF - completely designs the bearing stiffeners for a given girder if they are required.
- STRCHK - determines the stresses on a given girder and checks them against the allowable stresses.
- STRMAX - determines the maximum design stress conditions.
- TERMIN - determines the locations of the ends of the required cover plates based on the fatigue stress conditions.
- THDES - calculates the required cover plate thicknesses for each subelement along the girder.
- TOPT - determines the required minimum weld thickness for a fillet weld.
- UNIFST - designs the girder which has a uniform steel section composed of a rolled, wide-flange section without cover plates.

APPENDIX B

APPENDIX B
SELECTED PROGRAM NOMENCLATURE

The following alphabetic list contains some of the more pertinent variables used in the program. The list basically contains all the variables which are interchanged between the program subroutines. The numbers in the parentheses immediately following the variable name give the maximum dimensions of that array within the program.

ANEG	-	the negative area under a given influence line.
APOS	-	the positive area under a given influence line.
BEAR (5,3)	-	the bearing stiffener array.
(I,1)	-	width of stiffener at support I.
(I,2)	-	thickness of stiffener at support I.
(I,3)	-	size of connecting weld for stiffener at support I.
CC (5,5)	-	the coefficient array for the reaction influence line calculations.
CH (80)	-	the distance from the neutral axis to the lower extreme fiber within a subelement for the high modulus composite section.

- CL (80) - the distance from the neutral axis to the lower extreme fiber within a subelement for the low modulus composite section.
- CMIN - the cost of the present optimum design.
- CONSP(16,3,4) - the shear connector design array.
- (J,1,I) - spacing of the J group of connectors in span I.
- (J,2,I) - the starting coordinate of the J group of connectors in span I.
- (J,3,I) - the final coordinate of the J group of connectors in span I.
- COOR (81) - the coordinates of the analysis points.
- COST (9) - the unit cost array.
- (1) - cost of the rolled sections.
- (2) - cost of the cover plates.
- (3) - cost of the bearing stiffeners.
- (4) - cost of the shear connectors.
- (5) - cost of the connector installation.
- (6) - fixed cost of a weld.
- (7) - cost of the weld material.
- (8) - cost of the weld length.
- COVPL(12,6,2) - the cover plate design array.
- (J,I,I) - the starting coordinate of Jth plate.

- (J,2,I) - the final coordinate of Jth plate.
- (J,3,I) - the thickness of Jth plate.
- (J,4,I) - the width of the Jth plate.
- (J,5,I) - the size of the end welds for the
Jth plate.
- (J,6,I) - the size of the seal weld for the
Jth plate.

Note: For the COVPL array, I equals 1 for the cover plates on the top of the girder and I equals 2 for the cover plates on the bottom of the girder.

- CS (80) - the distance from the neutral axis to the lower extreme fiber within a subelement for the steel section alone.
- DAF (18) - the depth divided by the flange area for the rolled sections used in the design program.
- DEPTH(18) - the depth of the rolled sections used in the design program.
- DESDEF(20,4) - the design girder deflections.
 - (1,1) - the deflection at IDEF(I) due to the dead load.
 - (1,2) - the deflection at IDEF(I) due to the positive live load.

- (I,3) - the deflection at IDEF(I) due to the superimposed dead load.
- (I,4) - the deflection at IDEF(I) due to the negative live load.
- DESMOM(81,4) - the design moment array.
 - (I,1) - the moment at I due to the dead load.
 - (I,2) - the moment at I due to the positive live load.
 - (I,3) - the moment at I due to the superimposed dead load.
 - (I,4) - the moment at I due to the negative live load.
- DESREA(5,4) - the design reaction array.
 - (I,1) - the reaction at support I due to the dead load.
 - (I,2) - the reaction at support I due to the positive live load.
 - (I,3) - the reaction at support I due to the superimposed dead load.
 - (I,4) - the reaction at support I due to the negative live load.
- DESSH(162,4) - the design shear array.
 - (I,1) - the shear at point I due to the dead load.

- (I,2) - the shear at point I due to the positive live load.
- (I,3) - the shear at point I due to the superimposed dead load.
- (I,4) - the shear at point I due to the negative live load.

Note: For analysis point I, J equals $2I - 1$ just to the right of I and J equals $2I - 2$ just to the left of I.

- DIA - the diameter of the shear connector used in the program.
- DING - the impact length for a given influence line.
- DIST - the distance a cover plate is cut off based on fatigue stress considerations.
- EM - the modulus of elasticity which is set at 29,000 ksi in the program.
- EPS (4) - the number of elements per span.
- FC - the allowable concrete compressive stress.
- FLTH (18) - the thickness of the flange for the rolled sections used in the design program.
- FLWID (18) - the width of the flange for the rolled sections used in the design program.

- FPC - the 28-day strength of the concrete.
- FU - the minimum tensile strength of the steel.
- FV - the allowable weld metal stress.
- GDL - the weight of the steel girder.
- HAUNCH - the value of the concrete haunch for the composite design.
- HT - the calculated impact factor.
- IDEF (30) - the analysis points at which the deflections are calculated.
- IHCON (80) - the moment of inertia of a subelement for the high modulus composite section.
- ILCON (80) - the moment of inertia of a subelement for the low modulus composite section.
- IOSP (5) - the index of the support points.
- IPT1 - the subelement input option.
- IPT2 - the loading type input option.
- IPT3 - the initial cross section input option.
- IPT4 - the units input option.
- IPT5 - the loading components or designation input option.
- IPT6 - the optimization input option.
- IPT7 - the design cycle input option.

IPT8	-	the material input option.
IPT9	-	the deflection input option.
IPT10	-	the output option.
ISAVE (80)	-	the rolled section number for each subelement of the present optimum design girder.
IST	-	the identification number of the rolled section being designed.
ISTEEL(80)	-	the moment of inertia of a subelement for the steel section.
IX (18)	-	the moment of inertia about the x-x axis for the rolled sections used in the design program.
IY (18)	-	the moment of inertia about the y-y axis for the rolled sections used in the program.
JPT3	-	the composite section variable.
Note: If the design is composite then JPT3 is less than 9 and if the design is noncomposite then JPT3 is greater than 9.		
KOP (10)	-	the location of the crossover points for a given influence line.
LDIS	-	the live load designation.
LENGTH	-	the total length of the girder.

- LPCM (10) - the boundary points of the negative moment regions along the girder.
- MAXOL - the point which is the maximum ordinate of a given influence line.
- MINOL - the point which is the minimum ordinate of a given influence line.
- MN - the number of spans along the design problem girder.
- MR - the modular ratio.
- NA - the total number of analysis points along the girder.
- NAME (18) - the AISC name of the rolled sections used in the design program.
- NCS - the number of shear connectors per row.
- NCY - the design cycle counter.
- NCYC - the maximum number of design cycles required.
- NDEF - the total number of points at which the deflections are required.
- NE - the total number of subelements.
- NOS (80) - the NSECT number of the rolled section in each subelement.
- NS - the total number of supports.
- NSECT (18) - the identification number of the rolled sections used in the design program.

- OI - the impact option.
- OIL - the interstate loading option.
- PERCEN - the allowable percentage change
in the critical design moment.
- PLATH (80,2) - the plate thicknesses for each
subelement.
(I,1) - the thickness of the top cover
plate in subelement I.
(I,2) - the thickness of the bottom plate
in subelement I.
- PLL - the live load factor.
- PLTSAV(80,2) - the PLATH array for the current
optimum design.
- PLWID (80,2) - the plate width for each subelement.
(I,1) - the width of the top cover plate
in subelement I.
(I,2) - the width of the bottom cover
plate in subelement I.
- PLWSAV(80,2) - the PLWID array for the current
optimum design.
- PROP (3) - the steel properties used in the
design.
(1) - the allowable bending stress.
(2) - the allowable shear stress.
(3) - the steel yield point stress.
- RIL (5,81,3) - the reaction influence line array.

- (I,J,1) - the reaction at support I, with
the unit load at point J, for the
steel section.
- (I,J,2) - the reaction at support I, with
the unit load at point J, for the
low modulus composite section.
- (I,J,3) - the reaction at support I, with
the unit load at point J, for the
high modulus composite section.
- RATIO - the ratio of the minimum stress
to the maximum stress used in the
fatigue calculations.
- SAREA (18) - the area of the rolled sections
used in the design program.
- SAVE (12,6,2)- the cover plate array (COVPL) for
the current optimum design.
- SDL - the superimposed dead load.
- SLABA - the area of the concrete slab
which acts compositely with the
steel section.
- SLABTH - the thickness of the concrete slab.
- SLABWD - the effective width of the concrete
slab.
- SLABWT - the weight of the concrete deck
which is carried by the girder.
- SLL - the sidewalk live load.

- SMAX - the maximum stress at a point used to determine RATIO.
- SMIN - the minimum stress at a point used to determine RATIO.
- STORE (10) - the loadings for a given influence line.
- STRESS(81,4,2)- the stress array.
- (I,1,1) - the bottom steel stress at point I.
- (I,2,1) - the top steel stress at point I.
- (I,3,1) - the concrete stress at point I.
- (I,4,1) - the shear stress at point I.
- (I,1,2) - the bottom stress indicator.
- (I,2,2) - the top stress indicator.
- (I,3,2) - the concrete stress indicator.
- (I,4,2) - the shear stress indicator.
- Note: If the indicator is 1.0, the stress is greater than the allowable and if the indicator is 0.0, the stress is less than the allowable.
- SUBLEN(80) - the subelement length array.
- TCOST - the total cost of a girder design.
- TDS - the total design stress at a point.
- TITLE (36) - the problem title array.
- TRLO (10) - the truck load components.
- (1) - the first axle load.
- (2) - the second axle load.

- (3) - the third axle load.
 - (4) - the spacing between the first and second axles.
 - (5) - the minimum spacing between the second and the third axles.
 - (6) - the maximum spacing between the second and third axles.
 - (7) - the superimposed dead load.
 - (8) - the weight of the steel girder.
 - (9) - the impact option.
 - (10) - the weight of the slab carried by the steel girder.
- TSWID1 - the transformed slab width for the low modulus composite section.
- TSWID2 - the transformed slab width for the high modulus composite section.
- UNLO (9) - the uniform load components.
- (1) - the equivalent uniform live load.
 - (2) - the concentrated load for moment.
 - (3) - the concentrated load for shear.
 - (4) - the superimposed dead load.
 - (5) - the weight of the steel girder.
 - (6) - the interstate concentrated load.
 - (7) - the sidewalk live load.
 - (8) - the impact option.
 - (9) - the weight of the slab carried by the steel girder.

- WEBTH (18) - the thickness of the webs for the rolled sections used in the design program.
- WMIN - the weight of the present optimum design.
- WTOT - the total weight of a given girder design.

APPENDIX C

APPENDIX C

PROGRAM DATA

READIN Subroutine

The READIN subroutine is used to provide the program with all of the necessary data to solve a given problem. The routine does not, however, read the data cards which pertain to the section properties of the wide-flange rolled sections. These cards are read by the main program so that, if multiple problems are being solved with one run of the program, the section properties are read only once. The order of operations used in the READIN subroutine are described below. After the description, the method of data input is explained and a few samples of input data are presented.

Step 1

All of the arrays involved in the subroutine are zeroed and prepared for the given problem.

Step 2

The title cards are read and placed in the TITLE array. Two cards must be present for the title description. If a title is not desired or if the title is only on one card, then blank cards must appear in this position.

Step 3

The input parameter card, containing the input parameters IPT1 to IPT10, is read. These parameters are defined in Appendix B.

Step 4

The subroutine prints out some specified heading information if the output parameter, IPT10, dictates this operation.

Step 5

The lengths of the individual spans are read and used to determine the total length of the girder and the number of supports.

Step 6

Depending on the value of the subelement input option, IPT1, the subelement information is either read from cards or determined by the program. The quantities included in the subelement information are the number of elements per span, the analysis point coordinates, the support point indices, and the lengths of the subelements. If the design is continuous and composite, the limits of the negative moment regions are tentatively defined as the two elements on either side of the interior supports. These limits are then stored in the LPCM array.

Step 7

The data determined in steps 5 and 6 are printed out and converted to inches if required. This conversion is necessary because the program operates in the units of kips and inches.

Step 8

The steel table information for the wide-flange rolled sections used in the program is printed if the output parameter, IPT10, indicates that this information is desired.

Step 9

If the design is composite, then the composite section data are read and printed. The composite action parameter, JPT3, is calculated. If the design is noncomposite, step 9 is omitted.

Step 10

Depending on the value of IPT3, the initial cross section for each subelement along the girder is either read from cards or determined by the subroutine. With the sections specified for each subelement, SEMIC is called to evaluate the properties for each element. These properties are then printed in tabular form if required by the output parameter.

Step 11

The required loading conditions desired in the problem are read from cards. The form of the input depends on the

value of the loading parameter, IPT2, and the value of the loading components parameter, IPT5. The TRLO and the UNLO arrays are converted to the units of inches, if necessary, and are then printed in the output.

Step 12

The properties of the steel used in the problem are either read or determined by the routine. If the steel used has the ASTM designation of A36, then the PROP array is completely prepared by the subroutine. For all other steels, the yield point stress is read from a data card and the properties are calculated. The value of the modulus of elasticity for the steel is set at 29,000 ksi.

Step 13

The allowable compression stress for the concrete is computed using the 28-day concrete strength read in with the composite section data. All of the steel and concrete allowable stress properties are then listed in the output.

Step 14

If the optimization is based on cost, the unit cost data card is read. This card is not read if the optimization is based on weight. The COST array is then added to the output.

Step 15

The method of terminating the design program is either read or determined by the routine. The method which is

employed depends on the value of the design cycle parameter, IPT7. This termination method is then described in the output.

Step 16

The points at which the deflections are required are either read or determined by the subroutine. The form of this input depends on the value of the deflection parameter, IPT9. The points at which the deflections will be calculated are then listed in the output.

This completes the READIN subroutine. Once the data have been completely determined, the control of the program is returned to the main program and the design and optimization of the desired problem is started.

Method of Data Input

The method of data input for this program resembles the method used in the plate girder design program, GAD-I. It should be noted, however, that although the methods are similar, the data cards cannot be interchanged. The reason is that some of the data have a different meaning and purpose in this program. The designer should note these differences and take the appropriate precautions when working with both programs. The following is a description of the method of data input.

I. Card Set 1 - Title Cards

- A. The card set consists of two cards with the first 72 columns used on each card.

Format (18A4/18A4)

II. Card Set 2 - Input Parameter Card

- A. The card set consists of one card with the following integer information:

- 1. IPT1 - Subelement Information

- 0 - 10 elements per span.

- 1 - Read in the coordinates.

- 2 - Read in the number of elements per span.

- 2. IPT2 - Type of Loading Required

- 0 - Lane, Interstate, Sidewalk and Truck Loading.

- 1 - Lane, Interstate and Sidewalk Loading.
- 2 - Truck Loading.
- 3. IPT3 - Initial Girder Section
 - 0 - W33×118 used with composite design.
 - 1 - Read the uniform section with composite design.
 - 2 - Read the section for each sub-element with composite design.
 - 10 - W33×118 used with noncomposite design.
 - 11 - Read the uniform section with noncomposite design.
 - 12 - Read the section for each sub-element with noncomposite design.
- 4. IPT4 - Units of Data
 - 0 - Units of inches.
 - 1 - Units of feet.
- 5. IPT5 - Loading Components/Designation
 - 0 - Read in the load factor and the truck designation.
 - 1 - Read in all the load components.
- 6. IPT6 - Type of Optimization
 - 0 - Optimize based on cost.
 - 1 - Optimize based on weight.

7. IPT7 - Design Cycles
 - 0 - Use only one cycle of design.
 - 1 - Base the design on the percentage change in the critical moment.
 - 2 - Read in the maximum number of design cycles.
8. IPT8 - Steel Used in the Design
 - 0 - Use A36 steel.
 - 1 - Read the required yield point stress.
9. IPT9 - Deflections Required
 - 0 - Determine the deflections at the span centerlines.
 - 1 - Determine the deflections at the 0.2, 0.5, and 0.8 points in each span.
 - 2 - Read in the analysis points at which the deflections are required.
10. IPT10 - Output Parameter
 - 0 - Data and the final solution.
 - 1 - Level 0 plus each cycle solution.
 - 2 - Level 1 plus each girder design.
 - 3 - Level 2 plus additional output.
 - 4 - Debugging output.

Format (1015)

III. Card Set 3 - Span Length Card

- A. The card set consists of one card which contains the lengths of the spans.

Format (4F10.4)

IV. Card Set 4 - Subelement Information

- A. If IPT1 = 0, card set 4 is not required.
- B. If IPT1 = 1, the card set consists of the following cards:

1. One card containing the total number of analysis points.

Format (I10)

2. The necessary number of cards, each containing seven coordinates, to completely define the total number of analysis points.

Format (7F10.2)

- C. If IPT1 = 2, read the number of elements for each span.

Format (4I10)

V. Card Set 5 - Composite Section Requirements

- A. If IPT3 = 10, 11, or 12, card set 5 is not required.
- B. If IPT3 = 0, 1, or 3, read the following composite section requirements from one card:
1. Effective width of the slab, in.

2. Thickness of slab, in.
3. Modular Ratio as a floating point number.
4. 28-day concrete strength, psi.
5. Slab steel area, sq. in.
6. Value of the haunch, in.

Format (5F10.2,F10.4)

VI. Card Set 6 - Initial Cross Sections

- A. If IPT3 = 0 or 10, card set 6 is not required.
- B. If IPT3 = 1 or 11, the card set consists of one card containing the following values:
 1. ID number of the rolled section.
 2. Width of top cover plate, in.
 3. Thickness of top cover plate, in.
 4. Width of the bottom cover plate, in.
 5. Thickness of the bottom cover plate, in.

Format (I10,4F10.4)

- C. If IPT3 = 2 or 12, the card sets consist of one card for each subelement. The following values are contained on each card:

1. ID number of the rolled section.
2. Width of top cover plate, in.
3. Thickness of top cover plate, in.
4. Width of the bottom plate, in.

5. Thickness of the bottom plate, in.

Format (110,4F10.4)

VII. Card Set 7 - Load Information

A. If IPT5 = 0, the card set consists of one card containing the following values:

1. Load designation.
 - a. Use one of the five (5) standard AASHO designations, i.e., H10, H15, H20, HS15 or HS20.
 - b. This input must be right justified in the first 4 columns.
2. Load factor in decimal form.
3. Weight of the slab carried by the girder.
4. The superimposed dead load acting on the low modulus concrete section.
5. Dead weight of the girder.
6. Sidewalk live load.
7. Impact option.
 - a. Use 0.0 if impact is included.
 - b. Use 1.0 if impact is to be excluded from the design.
8. Interstate loading option.
 - a. Use 0.0 if loading is to be considered.

b. Use 1.0 if loading is excluded.

Format (A4,F6.3,6F10.4)

B. If $IPT5 = 1$ and $IPT2 = 0$, the card set consists of two cards. The first card contains the truck loading components (TRLO) and the second card contains the uniform load components (UNLO).

1. Truck load components.

- a. First axle load in kips.
- b. Second axle load in kips.
- c. Third axle load in kips.
- d. Spacing of the first two axles.
- e. Minimum spacing between the rear axles.
- f. Maximum spacing between the rear axles.
- g. Superimposed dead load.
- h. Girder dead weight.
- i. Impact option.
- j. Weight of slab.

Format (10F7.3)

2. Uniform load components.

- a. Uniform lane load.
- b. Concentrated load for moment.
- c. Concentrated load for shear.
- d. Superimposed dead load.

- e. Girder dead weight.
- f. Interstate concentrated load.
- g. Sidewalk live load.
- h. Impact option.
- i. Weight of slab.

Format (9F8.4)

- C. If $IPT5 = 1$ and $IPT2 = 1$, the card set consists of one card containing the lane loading components shown above.
- D. If $IPT5 = 1$ and $IPT2 = 2$, the card set consists of one card containing the truck loading components shown above.

VIII. Card Set 8 - Material Properties

- A. If $IPT8 = 0$, card set 8 is not required.
- B. If $IPT8 = 1$, the card set consists of one card containing the yield point stress of the steel used in the problem. Up to 4 percent overstress can be programmed by increasing the yield point stress by the desired overstress percentage.

IX. Card Set 9 - Optimization Information

- A. If $IPT6 = 0$, the card set consists of one card containing the following values:
 - 1. Rolled section cost, \$/lb.
 - 2. Cover plate cost, \$/lb.
 - 3. Bearing stiffener cost, \$/lb.

4. Shear connector cost, \$/lb.
5. Connector installation cost, \$/ea.
6. Fixed weld cost, \$/weld.
7. Weld material cost, \$/cu. in.
8. Weld length cost, \$/in.

Format (9F8.2)

- B. If IPT6 = 1, card set 9 is not required.

X. Card Set 10 - Design Cycle Information

- A. If IPT7 = 0, card set 10 is not required.
- B. If IPT7 = 1, the card set consists of one card containing the following values:
 1. Maximum number of cycles.
 2. Allowable percentage change in the critical moment.

Format (I10,F10.2)

- C. If IPT7 = 2, the card set consists of one card containing the number of design cycles desired.

Format (I10) -

XI. Card Set 11 - Deflection Information

- A. If IPT9 = 0, card set 11 is not required.
- B. If IPT9 = 1, card set 11 is not required.
- C. If IPT9 = 2, the card set consists of one card containing the following:
 1. Number of points at which the deflections are required. (Max. = 20)

2. Analysis point numbers of the
deflection points.

Format (I10,20I3)

Note: When preparing the data, all integer values should be right justified and all floating point numbers should be typed according to their respective formats. All floating point fields are large enough to include a punched decimal point and this is recommended to avoid confusion.

Input Data Samples

Problem No. 1

Design a girder with three spans, each having a length of 70 ft. Design the girder for HS20 loading with a load factor of one axle per girder. Also design for impact loading and interstate loading. Use A36 steel in the design, and base the design on ten elements per span.

Design the girder compositely with the deck slab. Use concrete with a 28-day strength of 3,000 psi and a modular ratio of 8 for the bridge deck. The composite section properties include an effective slab width of 84 in., a slab thickness of 7 in., a slab steel area of 1.94 sq. in., and a haunch depth of 1.5 in.

The optimization is based on cost. The unit cost per pound of steel for all steel elements is 10 cents. The fixed weld cost is \$30.00. The weld material costs \$3.00 per cubic inch of material and there is no charge per length of weld. The installation cost for the shear connectors is 50 cents.

Design the girder until the change in the critical moment is less than one-half percent. Determine the deflections at the centerline of each span and use the debugging output option.

The input data for this problem are shown in Table C-1.

Table C-1: Input Data for Sample Data Problem No. 1

Card Set	Card No.	DATA CARDS									
1	1	THREE SPAN TEST PROBLEM									
	2	BRIDGE NO. 1									
2	3	0	0	0	1	0	0	1	0	0	4
3	4	70.00	70.00	70.00	70.00						
5	5	84.00	7.00	8.00	3000.00	1.94	1.5				
7	6	HS20	1.0	.73	.20	.15	0.0	0.0	0.0	0.0	0.0
9	7	.10	.10	.10	.10	.50	30.00	3.00	0.0		
10	8		5	.5							

Problem No. 2

Design a noncomposite girder for a 70 ft simple-span bridge. Design the girder in A36 steel and for an H20 AASHO loading. Use a load factor of 75 percent and do not include impact in the design. Read in the coordinates of fifteen analysis points.

Optimize the girder design based on weight and use a total of three design cycles. Determine the deflections at all points and print the solution.

The input data for this problem are shown in Table C-2.

Table C-2: Input Data for Sample Data Problem No. 2

Card Set		Card No.	DATA CARDS															
1		1	SIMPLE SPAN TEST PROBLEM															
		2	BRIDGE NO. 4															
2		3	1	0	10	1	0	1	2	0	2	0	MARCH 23, 1971					
3		4	70.00															
4		5	15															
		6	0.0	5.0	10.0	15.0	20.0	25.0	30.0									
		7	35.0	40.0	45.0	50.0	55.0	60.0	65.0									
		8	70.0															
7		9	H20	.75	.75	.20	.15	0.0	1.0	0.0								
10		10	3															
11		11	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

APPENDIX D


```

000124      IR = IST * I                                MAIN 57
000126      IF (IR .EQ. 19) GO TO 1080                MAIN 58
C          BEGIN THE DESIGN LOOP FOR THE LIGHTER SECTIONS    MAIN 59
000130      DO 1020 JJ = IR, 1R                        MAIN 60
000131      IST = JJ                                      MAIN 61
000132      DO 1030 K = 1, NR                          MAIN 62
C          DETERMINE THE SUPPLEMENT PLATE THICKNESSES        MAIN 64
000134      1030 NOS(K) = IST                          MAIN 63
000140      CALL THUES                                    MAIN 65
000141      IF (ISWIT .EQ. 1) GO TO 1020                MAIN 66
000143      CALL SEMIC                                    MAIN 67
000144      IF (IPT10 .GT. 2) CALL PRSEC(NE,NOS,NAME,PLWID,PLATH,ISTEEL,CS, MAIN 68
      IILCON,CL,IHCON,CH,COOR,SURLEN,0,NCY)            MAIN 69
C          DESIGN THE COVER PLATES                          MAIN 70
000165      CALL CPDES                                    MAIN 71
000166      CALL SEMIC                                    MAIN 72
C          DESIGN THE WELDS                                MAIN 73
000167      CALL CPWDES                                MAIN 74
C          DESIGN THE STIFFENERS                            MAIN 75
000170      CALL STIFF(IOSP,NS,DESREA,NOS,FLTH,FLWID,WERTH,DEPTH,PROP,EM, MAIN 76
      IREAR)                                           MAIN 77
C          DESIGN THE SHEAR CONNECTOR SPACING              MAIN 78
000203      CALL CONDES                                MAIN 79
C          DETERMINE THE WEIGHT AND COST                   MAIN 80
000204      CALL OBJECT                                MAIN 81
C          STORE THE DESIGN IF IT IS THE PRESENT OPTIMUM    MAIN 82
000205      CALL KEEP(NE,NOS,PLATH,PLWID,COVPL,CMIN,WMIN,IPTAI) MAIN 83
000215      IF (IPT10 .LT. 1) GO TO 1020              MAIN 84
000220      CALL PRSEC(NE,NOS,NAME,PLWID,PLATH,ISTEEL,CS,IILCON,CL,IHCON,CH, MAIN 85
      ICON,SURLEN,0,NCY)                               MAIN 86
000236      CALL PROCV(COVPL)                          MAIN 87
000240      WRITE (6,700)                               MAIN 88
000244      700 FORMAT (1H1)                             MAIN 89
000244      CALL PRSTIF(NS,REAR)                         MAIN 90
000245      CALL PRCON(NS,CONSP,NCS)                     MAIN 91
000251      WRITE (6,1518) WTOT,TCOST                  MAIN 92
000261      1518 FORMAT (1H0,1X,15HTOTAL WEIGHT = ,F15.5,5X,15HTOTAL COST = , MAIN 93
      1F15.2)                                           MAIN 94
000261      IF (IPT6 .GT. .5) GO TO 6090              MAIN 95
000265      WRITE (6,1542) CMIN                         MAIN 96
000272      1542 FORMAT (1H0,30X,7HCMIN = ,F12.2)     MAIN 97
000272      GO TO 1020                                  MAIN 98
000273      6090 WRITE (6,7542) WMIN                     MAIN 99
000301      7542 FORMAT (1H0,30X,7HWMIN = ,F12.2)     MAIN 100
000301      1020 CONTINUE                             MAIN 101
000303      1040 CONTINUE                             MAIN 102
C          PLACE THE OPTIMUM DESIGN IN THE WORKING ARRAYS    MAIN 103
000303      DO 1 I = 1, NR                             MAIN 104
000305      1 NOS(I) = ISAVE(I)                         MAIN 105
000311      IST = NOS(I)                                MAIN 106
000313      DO 2 I = 1, NR                             MAIN 107
000314      DO 2 J = 1, 2                               MAIN 108
000315      PLATH(I,J) = PLTHSAV(I,J)                 MAIN 109
000322      2 PLWID(I,J) = PLWCSAV(I,J)               MAIN 110
000331      DO 4 I = 1, 12                             MAIN 111
000333      DO 4 J = 1, 6                               MAIN 112
000334      DO 4 K = 1, 2                               MAIN 113
000335      3 COVPL(I,J,K) = CSAVE(I,J,K)              MAIN 114

```

```

000355      CALL SFMIC                                MAIN 115
000356      IF (MN.EQ. 1) GO TO 100                     MAIN 116
C      DETERMINE THE NEW STEEL DEAD WEIGHT          MAIN 117
000360      IF (IPT2.LT. 2) UNLO(5) = WMIN / (LENGTH * 1000.) MAIN 118
000366      IF (IPT2.NE. 1) TRLO(8) = WMIN / (LENGTH * 1000.) MAIN 119
C      REANALYZE THE STRUCTURE IF IT IS CONTINUOUS MAIN 120
000373      CALL ANAL                                    MAIN 121
000374      GO TO 200                                    MAIN 122
000375  100 CONTINUE                                    MAIN 123
C      REANALYZE THE SIMPLE SPAN PROBLEM           MAIN 124
000375      IF (IPT3.LT. 9) GO TO 40                   MAIN 125
000400      DLO = TRLO(7) + TRLO(8) + TRLO(10)         MAIN 126
000403      IF (IPT2.EQ. 2) GO TO 41                   MAIN 127
000405      IF (DLO.LT. UNLO(6) + UNLO(5) + UNLO(9)) DLO = UNLO(6) + UNLO(5) + UNLO(9) MAIN 128
000414      GO TO 41                                     MAIN 129
000415      40 DLO = TRLO(8) + TRLO(10)                 MAIN 130
000417      IF (IPT2.EQ. 2) GO TO 41                   MAIN 131
000421      IF (DLO.LT. UNLO(5) + UNLO(9)) DLO = UNLO(5) + UNLO(9) MAIN 132
000426      41 IF (IPT2.LT. 2) UNLO(5) = WMIN / (LENGTH * 1000.) MAIN 133
000434      IF (IPT2.NE. 1) TRLO(8) = WMIN / (LENGTH * 1000.) MAIN 134
000441      IF (IPT3.LT. 9) GO TO 42                   MAIN 135
000444      DLN = TRLO(7) + TRLO(9) + TRLO(10)         MAIN 136
000447      IF (IPT2.EQ. 2) GO TO 43                   MAIN 137
000451      IF (DLN.LT. UNLO(6) + UNLO(5) + UNLO(9)) DLN = UNLO(6) + UNLO(5) + UNLO(9) MAIN 138
000460      GO TO 43                                     MAIN 139
000461      42 DLN = TRLO(8) + TRLO(10)                 MAIN 140
000463      IF (IPT2.EQ. 2) GO TO 43                   MAIN 141
000465      IF (DLN.LT. UNLO(5) + UNLO(9)) DLN = UNLO(5) + UNLO(9) MAIN 142
000472      43 F1 = DLN / DLO                          MAIN 143
000474      DO 44 I = 1, NA                           MAIN 144
000476      44 DESM(I,1) = DESMOD(I,1) * F1           MAIN 145
000503      DESREA(1,1) = DESREA(1,1) * F1           MAIN 146
000504      DESREA(2,1) = DESREA(2,1) * F1           MAIN 147
000505      IR = 2 * NA - 2                             MAIN 148
000507      DO 45 I = 1, IR                             MAIN 149
000511      45 DESSH(I,1) = DESSH(I,1) * F1           MAIN 150
000516      IF (IPT10.EQ. 0) GO TO 200                 MAIN 151
000517      WRITE (6,70)                                  MAIN 152
000523      FORMAT (1H1,10XDESIGN REACTIONS...//)     MAIN 153
000523      DO 71 I = 1, AS                             MAIN 154
000525      71 WRITE (6,72) 1,DESREA(1,1),DESREA(1,2),DESREA(1,3),DESREA(1,4) MAIN 155
000545      72 FORMAT (15,4(F15.4,3X))                 MAIN 156
000545      WRITE (6,73)                                  MAIN 157
000550      73 FORMAT (1H0,10XDESIGN MOMENTS...//)     MAIN 158
000550      DO 74 I = 1, NA                             MAIN 159
000552      74 WRITE (6,72) 1,DESMOD(I,1),DESMOD(I,2),DESMOD(I,3),DESMOD(I,4) MAIN 160
000572      WRITE (6,75)                                  MAIN 161
000575      75 FORMAT (1H0,10XDESIGN SHEARS...//)     MAIN 162
000575      KJ = 2 * NA - 2                             MAIN 163
000577      DO 76 I = 1, KJ                             MAIN 164
C      DETERMINE ALL OF THE STRESSES                MAIN 165
000601      76 WRITE (6,72) 1,DESSH(I,1),DESSH(I,2),DESSH(I,3),DESSH(I,4) MAIN 166
000621      200 CALL STRCHN                             MAIN 167
C      DESIGN THE STIFFENERS                        MAIN 168
000622      CALL STIFF (ISPR,1,DESREA,NOS,FLTH,FI*10,FLTH,PERTH,POOD,FM, MAIN 169
      IPRX)                                           MAIN 170
C      DESIGN THE SHEAR CONNECTION SPACING          MAIN 171
000635      CALL CONDES                                 MAIN 172

```

	C	DETERMINE THE WEIGHT AND COST	MAIN 173
000636		CALL OBJECT	MAIN 174
000637		IF (IPT10 .EQ. 0) GO TO 601	MAIN 175
000640		CALL PRSECFNF(NOS,NAME,PLWID,PLATH,ISTEEL,CS,ICCON,CI,THCON,CH, ICOU,SUBLEN,1,NCY)	MAIN 176
000657		CALL PROCV(COVP1)	MAIN 177
000661		CALL PRSTR(STRESS,NA)	MAIN 178
000663		CALL PRSTIF(NS,HEAR)	MAIN 179
000665		CALL PRCON(NS,CONSP,NCS)	MAIN 180
000670		IF (IPT6 .GT. .5) GO TO 10	MAIN 181
000674		WRITE (6,1) ICOST	MAIN 182
000701	11	FORMAT (1H0,10X,25HTHE COST OF THE GIRDER = ,F12.2)	MAIN 183
000701		GO TO 16	MAIN 184
000702	10	WRITE (6,15) WLOT	MAIN 185
000710	15	FORMAT (1H0,10X,27HTHE WEIGHT OF THE GIRDER = ,F12.2)	MAIN 186
000710	16	CONTINUE	MAIN 187
000710	601	CONTINUE	MAIN 188
	C	DETERMINE THE NEXT DESIGN CYCLE	MAIN 189
000710		IF (IPT7 .NE. 2) GO TO 20	MAIN 190
000712		IF (NCY .EQ. NCYC) GO TO 25	MAIN 191
000714		NCY = NCY + 1	MAIN 192
000715		GO TO 80	MAIN 193
000716	20	IF (IPT7 .EQ. 0) GO TO 25	MAIN 194
000717		O1 = DESMOM(ITT,1) + DESMOM(ITT,3) + DESMOM(ITT,4)	MAIN 195
000723		IF (MN .EQ. 1) O1 = DESMOM(ITT,1)	MAIN 196
000727		IF (MN .EQ. 1 .AND. JPT3.GT.9) O1 = DESMOM(ITT,1) + DESMOM(ITT,2)	MAIN 197
000741		P1 = ABS((SMFEC - O1) / O1)	MAIN 198
000746		IF (P1 .LT. PERFCN/100.) GO TO 25	MAIN 199
000750		IF (NCY .EQ. NCYC) GO TO 25	MAIN 200
000752		NCY = NCY + 1	MAIN 201
000753		GO TO 80	MAIN 202
000754	25	ISI = NOS(ITT)	MAIN 203
	C	DETERMINE THE SHELFMENT PLATE THICKNESSES FOR THE FINAL DESIGN	MAIN 204
000756		CALL THDES	MAIN 205
000757		CALL SEMIC	MAIN 206
	C	DESIGN THE FINAL COVER PLATES	MAIN 207
000760		CALL CPDES	MAIN 208
000761		CALL SEMIC	MAIN 209
	C	DESIGN THE FINAL WELDS	MAIN 210
000762		CALL CPWDES	MAIN 211
	C	DESIGN THE FINAL STIFFENERS	MAIN 212
000763		CALL STIFF(IOSP,NS,DESREA,NOS,FLTH,FLWID,FWTH,DEPTH,PROP,FM, IBEAR)	MAIN 213
	C	DESIGN THE FINAL SHEAR CONNECTOR SPACING	MAIN 214
000776		CALL CONDES	MAIN 215
	C	DETERMINE THE FINAL WEIGHT AND COST	MAIN 216
000777		CALL OBJECT	MAIN 217
	C	CALCULATE THE FINAL STRESSES	MAIN 218
001000		CALL STRCHK	MAIN 219
	C	CALCULATE THE REQUIRED DEFLECTIONS	MAIN 220
001001		CALL DEFLC	MAIN 221
001002		CALL PRSECFNF(NOS,NAME,PLWID,PLATH,ISTEEL,CS,ICCON,CI,THCON,CH, ICOU,SUBLEN,2,NCY)	MAIN 222
001021		CALL PROCV(COVP1)	MAIN 223
001023		CALL PRSTR(STRESS,NA)	MAIN 224
001025		CALL PRSTIF(NS,HEAR)	MAIN 225
001027		CALL PRCON(NS,CONSP,NCS)	MAIN 226
001032		IF (IPT6 .GT. .5) GO TO 28	MAIN 227

001036	WRITE (6,11) TCONST	MAIN 231
001043	GO TO 29	MAIN 232
001044	28 WRITE (6,15) WTOT	MAIN 233
001052	29 CONTINUE	MAIN 234
001052	WRITE (6,21)	MAIN 235
001056	21 FORMAT (1H1)	MAIN 236
001056	CALL PRODEF (NDEF,DEFSDEF,NS,SPAN,IOFF,IPT2,IPN 0(7))	MAIN 237
	C RETURN TO THE READIN STEP FOR A NEW PROBLEM	MAIN 238
001066	GO TO 900	MAIN 239
001067	END	MAIN 240

PROGRAM LENGTH INCLUDING I/O BUFFERS

004006

UNUSED COMPILER SPACE

036500

	FUNCTION ALLOW(DM,FLANGE,LPCM,IOSP,IA,NS,PROP,COORD,DESMOM,SURLEN, ALL 1
	LEM,KEY) ALL 2
	C DETERMINES THE ALLOWABLE STRESS AT A SURFLEMENT ALL 3
000017	DIMENSION LPCM(10),IOSP(5),PROP(3),COORD(4),SURLEN(4), ALL 4
	IDESMOM(4,4) ALL 5
000017	IF (DM .LT. -.01) GO TO 617 ALL 6
000021	ALLOW = PROP(1) ALL 7
000023	RETURN ALL 8
000024	617 DO 520 I = 1, 10, 2 ALL 9
000026	II = I ALL 10
000027	IF (IA .GE. LPCM(I) .AND. IA .LE. LPCM(I+1)) GO TO 621 ALL 11
000040	620 CONTINUE ALL 12
000042	621 IR = LPCM(II) ALL 13
000045	IC = LPCM(II+1) ALL 14
000046	DO 530 I = 1, NS ALL 15
000050	IF (IOSP(I) .GE. IR .AND. IOSP(I) .LE. IC) GO TO 631 ALL 16
000062	GO TO 630 ALL 17
000062	631 ID = IOSP(I) ALL 18
000065	GO TO 635 ALL 19
000065	630 CONTINUE ALL 20
000070	635 IF (IA .LT. ID) GO TO 640 ALL 21
000072	DUS = COOR(IC)-COORD(ID)-(DESMOM(IC,1)*SURLEN/IC)/ ALL 22
	1(DESMOM(IC+1,1)-DESMOM(IC,1)) ALL 23
000106	GO TO 641 ALL 24
000107	640 DUS = COOR(ID)-COORD(IR)-(DESMOM(IR,1)*SURLEN/IR)/ ALL 25
	1(DESMOM(IR+1,1)-DESMOM(IR,1)) ALL 26
000125	641 IF (DUS/FLANGE .GE. 216./SQRT(PROP(3))) GO TO 642 ALL 27
000141	RATIO = DUS/FLANGE ALL 28
000142	GO TO 643 ALL 29
000142	642 RATIO = 216./SQRT(PROP(3)) ALL 30
000150	643 PAS = PROP(1) * .1671791711 * RATIO * RATIO * PROP(3)**2/PM ALL 31
000157	IF (KEY .EQ. 1) GO TO 650 ALL 32
000165	ALLOW = PAS ALL 33
000166	RETURN ALL 34
000167	650 IF (1./2**NS .LE. PROP(3)) GO TO 644 ALL 35
000173	ALLOW = 1./2 * PAS ALL 36

000175	GO TO 645	ALLO	37
000175	644 ALL D = PROP(1)	ALLO	38
000177	645 CONTINUE	ALLO	39
000177	RETURN	ALLO	40
000201	END	ALLO	41

SUBPROGRAM LENGTH

000235

UNUSED COMPILER SPACE

042300

	SUBROUTINE ANAL	ANAL	1
	CALCULATES THE DESIGN REACTIONS, MOMENTS, AND SHEARS	ANAL	2
	FOR A GIVEN GIRDER	ANAL	3
000002	COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WENTH(18),	ANAL	4
	IPAF(18),IX(18),IY(18),NSFCT(18),NOS(18),ISTEEL(18),CS(18),	ANAL	5
	PILC(18),CL(18),IHCN(18),CH(18),PLATH(18),PI(18),NF,	ANAL	6
	SLASH(18),SLANTH,MR,CORR(18),RIL(5,18,1),CC(5,5),FM,SPAN(4),TOSP(5),	ANAL	7
	4THO(18),NA,LENGTH,RO(18),SHLEF(18),NS,UNIC(18),TRLO(18),PROP(3),	ANAL	8
	SFC,STRESS(18,4,2),DEFSHM(18,4),DESFRA(5,4),DESSH(162,4)	ANAL	9
000002	COMMON/ONE/JPT3,IPT10	ANAL	10
000002	COMMON/TWO/PCMH	ANAL	11
000002	COMMON/THREE/HAUNCH	ANAL	12
000002	COMMON/FOUR/IPT2	ANAL	13
000002	REAL IX,IY,ISTEEL,IHCN,IHCN,MR,LENGTH	ANAL	14
000002	DIMENSION IITL(18),EPS(4),IIL(18,3),STORE(10),IPCM(10)	ANAL	15
000002	INTEGER TYPE	ANAL	16
000002	INTEGER EPS,IITL	ANAL	17
	C ZERO OUT THE DESIGN ARRAYS	ANAL	18
000002	DO 300 J = 1, 4	ANAL	19
000004	DO 301 I = 1, 18	ANAL	20
000005	501 DESHM(I,J) = 0.0	ANAL	21
000013	DO 302 I = 1, 5	ANAL	22
000014	502 DESFRA(I,J) = 0.0	ANAL	23
000021	DO 303 I = 1, 162	ANAL	24
000022	503 DESSH(I,J) = 0.0	ANAL	25
000030	500 CONTINUE	ANAL	26
000032	MR = NS - 1	ANAL	27
000034	IF (MR.NE.1) GO TO 3000	ANAL	28
	C DETERMINE THE REACTION INFLUENCE LINES FOR THE SIMPLE SPAN	ANAL	29
000035	DO 3001 I = 1, MR	ANAL	30
000036	RIL(1,I,1) = (LENGTH - CORR(1)) / LENGTH	ANAL	31
000043	3001 RIL(2,I,1) = 1.0 - RIL(1,I,1)	ANAL	32
000052	DO 3002 I = 2, 3	ANAL	33
000053	DO 3002 J = 1, MR	ANAL	34
000054	RIL(1,J,I) = RIL(1,I,1)	ANAL	35
000062	3002 RIL(2,J,I) = RIL(2,I,1)	ANAL	36
000073	IF (IPTH.EQ.1) GO TO 1002	ANAL	37
000075	GO TO 3003	ANAL	38
	C CALL ON READ TO DETERMINE THE CONTINUOUS REACTION LINES	ANAL	39
000076	3000 CALL READ	ANAL	40
	C PRINTOUT THE REACTION INFLUENCE LINE IF REQUIRED	ANAL	41

```

000077      IF (IPTIO .LT. 3) GO TO 1002          ANAL 42
000102      IF (JPT3 .GE. 9) GO TO 1003          ANAL 43
000105      WRITE (6,1010)                      ANAL 44
000110 1010 FORMAT (1H1,4FX,23HREACTION INFLUENCE LINE//) ANAL 45
000110      DO 1040 K = 1, 3                      ANAL 46
000112      IF (K .EQ. 1) WRITE (6,1050)        ANAL 47
000117      IF (K .EQ. 2) WRITE (6,1051)        ANAL 48
000125      IF (K .EQ. 3) WRITE (6,1052)        ANAL 49
000133 1050 FORMAT (1H0,3FX,29HREACTION LINE FOR STEEL ALONE//) ANAL 50
000133 1051 FORMAT (1H1,3FX,46HREACTION LINE FOR LOW MODULUS COMPOSITE ACTION/ANAL 51
1/) ANAL 52
000133 1052 FORMAT (1H1,3FX,47HREACTION LINE FOR HIGH MODULUS COMPOSITE ACTION/ANAL 53
1/) ANAL 54
000133      DO 1041 J = 1, NA                   ANAL 55
000135 1041 WRITE (6,1053) (RIL(I,J,K), I = 1, NS) ANAL 56
000155 1053 FORMAT (31X,5(F12.4,2X))            ANAL 57
000155 1040 CONTINUE                           ANAL 58
000157      GO TO 1002                          ANAL 59
000157 1003 WRITE (6,1010)                     ANAL 60
000163      WRITE (6,1050)                      ANAL 61
000167      DO 2060 I = 1, NS                   ANAL 62
000171      RIL(I,J,2) = RIL(I,J,1)            ANAL 63
000176 2060 RIL(I,J,3) = RIL(I,J,1)           ANAL 64
000202      DO 1060 J = 1, NA                   ANAL 65
000203 1060 WRITE (6,1053) (RIL(I,1,1), I = 1, NS) ANAL 66
000221      WRITE (6,1054)                      ANAL 67
000224 1054 FORMAT (1H0,2FX,43HIF THERE IS NO COMPOSITE ACTION IN THE DESIGN.) ANAL 68
000224      GO TO 1002                          ANAL 69
000225 3003 WRITE (6,1054)                      ANAL 70
000231 3004 FORMAT (1H1,2FX,39HREACTION INFLUENCE LINE FOR SIMPLE BEAM//) ANAL 71
000231      DO 3005 I = 1, NA                   ANAL 72
000233 3005 WRITE (6,1053) (RIL(I,1,1), J = 1, NS) ANAL 73
000251      IF (JPT3 .GE. 9) WRITE (6,1054)    ANAL 74
C          PRINT OUT THE LOADING DESCRIPTION ANAL 75
000257 1002 IF (IPTIO .LT. 3) WRITE (6,6000)    ANAL 76
000265 6000 FORMAT (1H1)                       ANAL 77
000265      IF (IPTIO .GE. 3) WRITE (6,1080)    ANAL 78
000274 1080 FORMAT (1H1,10X,34HDESCRIPTION OF ANALYSIS RESULTS...//11X,39HCOLUMN ANAL 79
1X, 1 = DEAD LOAD ON STEEL SECTION//11X,39HCOLUMN 2 = POSITIVE LANE ANAL 80
2 LOAD//11X,39HCOLUMN 3 = NEGATIVE LANE LOAD//11X,39HCOLUMN 4 = POS ANAL 81
3ITIVE INTERSTATE LOADING//11X,39HCOLUMN 5 = NEGATIVE INTERSTATE LOANAL 82
4ADING//11X,39HCOLUMN 6 = POSITIVE TRUCK LOAD//11X,39HCOLUMN 7 = NEANAL 83
5GATIVE TRUCK LOAD//11X,39HCOLUMN 8 = POSITIVE SIDEWALK LOAD//11X, ANAL 84
639HCOLUMN 9 = NEGATIVE SIDEWALK LOAD//11X,69HCOLUMN 10 = SUPERIMPOSED ANAL 85
7SEUD DEAD LOAD ON LOW MODULUS CONCRETE SECTION//) ANAL 86
000274      IF (IPTIO .NE. 9) WRITE (6,6001)    ANAL 87
000301 6001 FORMAT (1H0,10X,34HDESCRIPTION OF DESIGN LOADS...//11X, ANAL 88
135HCOLUMN 1 = LOADING DUE TO DEAD LOAD//11X,49HCOLUMN 2 = LOADING DUE ANAL 89
2OR POSITIVE LIVE LOAD//11X,49HCOLUMN 3 = LOADING DUE TO SUPERIMPOSED ANAL 90
3D DEAD LOAD//11X,49HCOLUMN 4 = LOADING FOR NEGATIVE LIVE LOAD//) ANAL 91
C          LOAD THE REACTION INFLUENCE LINES ANAL 92
000301      IF (IPTIO .GE. 3) WRITE (6,1090)    ANAL 93
000310 1090 FORMAT (1H0,22HREACTION LOADINGS.....//) ANAL 94
000310      DO 1100 I = 1, NS                   ANAL 95
000312      NR = I                                  ANAL 96
000313      DO 1101 K = 1, NA                   ANAL 97
000315      DO 1101 J = 1, 3                   ANAL 98
000315 1101 RIL(K,J) = RIL(I,K,J)            ANAL 99

```

```

C      CALL THE LOAD SUBROUTINE
000333 CALL LOAD(TIL,TRLO,SURLEN,NA,IP12,LENGTH,COORD,NP,4HDEAC,NS, ANAL 100
      ILOS,SPAN,STORE) ANAL 101
C      PRINT THE REACTION LOADINGS IF REQUIRED ANAL 102
000351 IF (IP110 .GE. 3) WRITE(6,1103) I,(STORE(J),JJ=1,10) ANAL 103
000370 1103 FORMAT (15,10F12.3) ANAL 104
C      DETERMINE THE DESIGN REACTIONS ANAL 105
000370 DESREA(1,1) = STORE(1) ANAL 106
000372 DESREA(1,3) = STORE(10) ANAL 107
000374 PR = 0.0 ANAL 108
000375 TR = 0.0 ANAL 109
000376 DO 1100 KK = 2, 7 ANAL 110
000377 IF (STORE(KK) .GE. 0.0) GO TO 8100 ANAL 111
000401 RR = STORE(9) + STORE(KK) ANAL 112
000403 IF (ABS(RR) .LT. TR) GO TO 1100 ANAL 113
000406 TR = ABS(RR) ANAL 114
000407 DESREA(1,4) = RR ANAL 115
000410 GO TO 1100 ANAL 116
000411 8100 RR = STORE(8) + STORE(KK) ANAL 117
000414 IF (ABS(RR) .LT. PR) GO TO 1100 ANAL 118
000417 PR = ABS(RR) ANAL 119
000420 DESREA(1,2) = PR ANAL 120
000421 1100 CONTINUE ANAL 121
C      PRINT THE DESIGN REACTIONS IF REQUIRED ANAL 122
000426 IF (IP110 .EQ. 0) GO TO 100 ANAL 123
000427 WRITE (6,1115) ANAL 124
000432 1115 FORMAT (1H0,14HDESIGN REACTIONS...//) ANAL 125
000432 DO 1118 I = 1, NS ANAL 126
000434 1118 WRITE (6,1120) I,DESREA(I,1),DESREA(I,2),DESREA(I,3),DESREA(I,4) ANAL 127
000454 1120 FORMAT (15,4(F15.4,3X)) ANAL 128
000454 IF (OPT3 .GT. 9) WRITE(6,6010) ANAL 129
000462 6010 FORMAT (10X,57HTHE THIRD COLUMN IS ZERO BECAUSE DESIGN IS NON-COMPANAL 130
      IOSILE.) ANAL 131
C      CONSTRUCT THE MOMENT INFLUENCE LINES ANAL 132
000462 100 IF (IP110 .GE. 3) WRITE (6,1143) ANAL 133
000471 1143 FORMAT (1H0,14HMMOMENT LOADINGS...//) ANAL 134
000471 DO 1130 I = 1, 3 ANAL 135
000473 TIL(I,1) = 0.0 ANAL 136
000476 1130 TIL(NA,1) = 0.0 ANAL 137
000502 DO 1140 I = 2, NF ANAL 138
000504 DO 1150 K = 1, 3 ANAL 139
000505 DO 1135 J = 2, NF ANAL 140
000506 WP = 0.0 ANAL 141
000507 SP = 0.0 ANAL 142
000510 IF (J .LT. I) SP = COOR(I) - COOR(J) ANAL 143
000514 JJ = 1 ANAL 144
000515 1141 KK = IOSP(JJ) ANAL 145
000517 IF (I .GT. KK) WP = WP + TIL(JJ,J,K) * (COOR(I) - COOR(KK)) ANAL 146
000532 IF (KK .GT. I) GO TO 1142 ANAL 147
000534 JJ = JJ + 1 ANAL 148
000537 GO TO 1141 ANAL 149
000540 1142 TIL(J,K) = WP - SP ANAL 150
000545 1135 CONTINUE ANAL 151
000547 1150 CONTINUE ANAL 152
C      CALL ON THE LOAD SUBROUTINE ANAL 153
000551 CALL LOAD(TIL,TRLO,INLEN,SURLEN,NA,IP12,LENGTH,COORD,I,4HDMT,NS, ANAL 154
      ILOS,SPAN,STORE) ANAL 155
C      PRINT THE MOMENT LOADINGS IF REQUIRED ANAL 156

```

```

000566 IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(I)), JJJ = 1, 10) ANAL 158
C DETERMINE THE DESIGN MOMENTS ANAL 159
000605 DESMOM(I,1) = STORE(1) ANAL 160
000607 DESMOM(I,3) = STORE(10) ANAL 161
000611 PR = 0.0 ANAL 162
000612 TR = 0.0 ANAL 163
000613 DO 5140 KK = 2, 7 ANAL 164
000614 KR = 2 * (KK / 2) ANAL 165
000616 IF (ABS(STORE(KK)) .LT. .00001 .AND. KR .EQ. KK) RR = STORE(9) + STORE(KK) ANAL 166
000632 IF (ABS(STORE(KK)) .LT. .00001 .AND. KR .NE. KK) PR = STORE(9) + STORE(KK) ANAL 167
000647 IF (STORE(KK) .GT. 0.0) RR = STORE(9) + STORE(KK) ANAL 168
000653 IF (STORE(KK) .LT. 0.0) RR = STORE(9) + STORE(KK) ANAL 169
000657 IF (KR .GE. 0.0) GO TO 5240 ANAL 170
000661 IF (ABS(KR) .LT. 1R) GO TO 5140 ANAL 171
000664 TR = ABS(KR) ANAL 172
000665 DESMOM(I,4) = RR ANAL 173
000666 GO TO 5140 ANAL 174
000667 5240 IF (ABS(KR) .LT. PR) GO TO 5140 ANAL 175
000673 PR = ABS(KR) ANAL 176
000674 DESMOM(I,2) = PR ANAL 177
000675 5140 CONTINUE ANAL 178
000677 1140 CONTINUE ANAL 179
C PRINT THE DESIGN MOMENTS IF REQUIRED ANAL 180
000702 IF (IPT10 .EQ. 0) GO TO 101 ANAL 181
000703 WRITE (6,1156) ANAL 182
000706 1156 FORMAT (1H0,17HDESIGN MOMENTS...//) ANAL 183
000706 DO 1157 I = 1, NA ANAL 184
000710 1157 WRITE (6,1120) I, DESMOM(I,1), DESMOM(I,2), DESMOM(I,3), DESMOM(I,4) ANAL 185
000730 IF (JPT3 .GT. 9) WRITE (6,6010) ANAL 186
C DETERMINE THE NEW LPCM VECTOR AND CORRECT THE I VALUES ANAL 187
000736 101 CONTINUE ANAL 188
000736 IF (MN .EQ. 1) GO TO 1173 ANAL 189
000740 J = 1 ANAL 190
000741 DO 1160 I = 1, NA ANAL 191
000743 NJ = I + 1 ANAL 192
000745 IF (NJ .GT. NA) GO TO 1170 ANAL 193
000750 SUM1 = DESMOM(I,1) ANAL 194
000751 SUM2 = DESMOM(NJ,1) ANAL 195
000752 IF (SUM1 .LT. 0.0 .AND. SUM2 .GT. 0.0) GO TO 1161 ANAL 196
000761 IF (SUM2 .LT. 0.0 .AND. SUM1 .GT. 0.0) GO TO 1165 ANAL 197
000770 GO TO 1160 ANAL 198
000770 1161 LPCM(J) = I ANAL 199
000772 J = J + 1 ANAL 200
000774 GO TO 1160 ANAL 201
000774 1162 LPCM(J) = NJ ANAL 202
000776 J = J + 1 ANAL 203
001000 1160 CONTINUE ANAL 204
001003 1170 IF (JPT3 .GE. 10) GO TO 1173 ANAL 205
001006 CALL SFMIC ANAL 206
001007 1173 CONTINUE ANAL 207
C CONSTRUCT AND LOAD THE SHEAR INFLUENCE LINES TO THE ANAL 208
C LEFT AND RIGHT OF THE ANALYSIS POINTS ANAL 209
001007 IF (IPT10 .GE. 3) WRITE (6,1300) ANAL 210
001016 1300 FORMAT (1H0,19HSHEAR LOADINGS...//) ANAL 211
001016 MN = NS - 1 ANAL 212
001020 DO 1201 I = 1, NA ANAL 213
001022 KL = 0 ANAL 214
001023 IF (I .NE. 1) GO TO 1206 ANAL 215

```

	C	CONSTRUCT THE LINE FOR POINT 1	ANAL 216
001025		K1 = 1	ANAL 217
001026		DO 1202 K = 1, 3	ANAL 218
001027		TIL(1,K) = -1.0	ANAL 219
001032		DO 1203 J = 2, NA	ANAL 220
001034	1203	TIL(J,K) = -RIL(1,1,K)	ANAL 221
001046	1202	CONTINUE	ANAL 222
	C	LOAD THE RIGHT POINT WITH THE LOAD SUBROUTINE	ANAL 223
001050		CALL LOAD(TIL,TRIO,INLO,SURLFN,NA,IP12,LENGTH,COOR,1,4HSHER,NS, 110SP,SPAN,STORE)	ANAL 224
	C	PRINT THE SHEAR LOADINGS IF REQUIRED	ANAL 225
001066		IF (IP10 .GE. 3) WRITE (6,1103) I, (STORE(I,JJ), JJ) = 1, 101	ANAL 226
001105		GO TO 1220	ANAL 227
001106	1205	IF (I .NE. NA) GO TO 1210	ANAL 228
	C	CONSTRUCT THE LINE FOR POINT NA	ANAL 229
001110		SP = 1.0	ANAL 230
001112		K1 = 2 * NA - 2	ANAL 231
001114		DO 1207 K = 1, 3	ANAL 232
001115		TIL(NA,K) = 1.0	ANAL 233
001121		DO 1208 J = 1, NF	ANAL 234
001122		WP = 0.0	ANAL 235
001123		DO 1209 JJ = 1, MN	ANAL 236
001125	1209	WP = WP + RIL(JJ,J,K)	ANAL 237
001135	1208	TIL(J,K) = SP - WP	ANAL 238
001144	1207	CONTINUE	ANAL 239
	C	LOAD THE LINE WITH THE LOAD SUBROUTINE	ANAL 240
001146		CALL LOAD(TIL,TRIO,INLO,SURLFN,NA,IP12,LENGTH,COOR,NA,4HSHER,NS, 110SP,SPAN,STORE)	ANAL 241
	C	PRINT THE SHEAR LOADINGS IF REQUIRED	ANAL 242
001164		IF (IP10 .GE. 3) WRITE (6,1103) I, (STORE(I,JJ), JJ) = 1, 101	ANAL 243
001203		GO TO 1220	ANAL 244
001204	1210	KL = 1	ANAL 245
001205		K1 = 2 * 1 - 2	ANAL 246
	C	CONSTRUCT THE LINE TO THE LEFT OF THE POINT	ANAL 247
001207		DO 1214 K = 1, 3	ANAL 248
001211		DO 1215 J = 1, NA	ANAL 249
001212		WP = 0.0	ANAL 250
001213		SP = 0.0	ANAL 251
001214		IF (J .LT. 1) SP = 1.0	ANAL 252
001220		JJ = 1	ANAL 253
001221	1217	KK = IONE(JJ)	ANAL 254
001223		IF (I .EQ. KK .AND. 1 .EQ. J) GO TO 2217	ANAL 255
001232		IF (I .GT. KK) WP = WP + RIL(JJ,J,K)	ANAL 256
001242		IF (K .GT. 1) GO TO 1216	ANAL 257
001246		JJ = JJ + 1	ANAL 258
001247		GO TO 1217	ANAL 259
001250	1216	TIL(J,K) = SP - WP	ANAL 260
001255		GO TO 1215	ANAL 261
001256	2217	TIL(J,K) = +1.00	ANAL 262
001262	1215	CONTINUE	ANAL 263
001265	1214	CONTINUE	ANAL 264
	C	LOAD THE LINE WITH THE LOAD SUBROUTINE	ANAL 265
001267		CALL LOAD(TIL,TRIO,INLO,SURLFN,NA,IP12,LENGTH,COOR,1,4HSHER,NS, 110SP,SPAN,STORE)	ANAL 266
	C	PRINT THE SHEAR LOADINGS IF REQUIRED	ANAL 267
001304		IF (IP10 .GE. 3) WRITE (6,1103) I, (STORE(I,JJ), JJ) = 1, 101	ANAL 268
001323		GO TO 1220	ANAL 269
	C	DETERMINE THE DESIGN SHEARS	ANAL 270
			ANAL 271
			ANAL 272
			ANAL 273


```

001324 1220 DESSH(KJ,1) = STORE(1) ANAL 274
001326 DESSH(KJ,3) = STORE(10) ANAL 275
001330 PR = 0.0 ANAL 276
001331 TR = 0.0 ANAL 277
001332 DO 1230 KK = 2, 7 ANAL 278
001333 KR = 2 * (KK / 2) ANAL 279
001335 IF (ABS(STORE(KK)) .LT. .00001 .AND. KR .EQ. KK) RR = STORE(8) + STORE(KK) ANAL 280
001351 IF (ABS(STORE(KK)) .LT. .00001 .AND. KR .NE. KK) RR = STORE(9) + STORE(KK) ANAL 281
001366 IF (STORE(KK) .GT. 0.0) RR = STORE(8) + STORE(KK) ANAL 282
001372 IF (STORE(KK) .LT. 0.0) RR = STORE(9) + STORE(KK) ANAL 283
001376 IF (RR .GE. 0.0) GO TO 6230 ANAL 284
001400 IF (ABS(RR) .LT. 12) GO TO 1230 ANAL 285
001403 TR = ABS(RR) ANAL 286
001404 DESSH(K,4) = RR ANAL 287
001405 GO TO 1230 ANAL 288
001406 6230 IF (ABS(RR) .LT. PR) GO TO 1230 ANAL 289
001412 PR = ABS(RR) ANAL 290
001413 DESSH(KJ,2) = RR ANAL 291
001414 1230 CONTINUE ANAL 292
001416 IF (TR .EQ. 0) GO TO 1201 ANAL 293
C CONSTRUCT THE LINE TO THE RIGHT OF THE POINT ANAL 294
001417 KJ = 2 * J - 1 ANAL 295
001421 KL = 0 ANAL 296
001422 IF (KN .EQ. 1) GO TO 3100 ANAL 297
001423 DO 1250 J = 2, MN ANAL 298
001425 IA = J ANAL 299
001426 IF (I .EQ. 105P(J)) GO TO 1255 ANAL 300
001430 1250 CONTINUE ANAL 301
001432 3100 DO 1251 K = 1, 3 ANAL 302
001434 1251 TIL(1,K) = TIL(1,K) + 1.0 ANAL 303
C LOAD THE LINE WITH THE LOAD SUBROUTINE ANAL 304
001443 CALL LOAD(TIL,1,0, INLO,5,INLEN,NA,IPT2,LENGTH,COORD,I,4,HSUER,NS, ANAL 305
1105P,SPAN,STORE) ANAL 306
C PRINT THE SHEAR LOADINGS IF REQUIRED ANAL 307
001460 IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(IJJ), JJJ = 1, 10) ANAL 308
001477 GO TO 1220 ANAL 309
C CONSTRUCT THE LINE FOR THE RIGHT OF THE INTERIOR SUPPORTS ANAL 310
001500 1255 DO 1260 J = 1, NA ANAL 311
001502 SP = 0.0 ANAL 312
001503 IF (J .EQ. 1) SP = 1.0 ANAL 313
001507 DO 1261 K = 1, 3 ANAL 314
001511 WP = 0.0 ANAL 315
001512 DO 1262 KK = 1, 14 ANAL 316
001514 1262 WP = WP + RII(KK,J,K) ANAL 317
001524 IF (J .EQ. 1) WP = 2.00 ANAL 318
001530 1261 TIL(J,K) = SP - WP ANAL 319
001537 1260 CONTINUE ANAL 320
C LOAD THE LINE WITH THE LOAD SUBROUTINE ANAL 321
001541 CALL LOAD(TIL,IPT0, INLO,5,INLEN,NA,IPT2,LENGTH,COORD,I,4,HSUER,NS, ANAL 322
1105P,SPAN,STORE) ANAL 323
C PRINT THE SHEAR LOADINGS IF REQUIRED ANAL 324
001557 IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(IJJ), JJJ = 1, 10) ANAL 325
001576 GO TO 1220 ANAL 326
001577 1271 CONTINUE ANAL 327
C PRINT THE DESIGN SHEARS IF REQUIRED ANAL 328
001602 IF (IPT10 .EQ. 0) GO TO 102 ANAL 329
001603 WRITE (6,1231) ANAL 330
001606 1231 FORMAT (100,14HDESIGN SHEARS,.,.,/) ANAL 331

```

001606	KJ = 2 * PA - 2	ANAL 332
001610	DO 1232 I = 1, KJ	ANAL 333
001612	1232 WRITE (6,1120) T,DESSH(I,1), DESSH(I,2), DESSH(I,3), DESSH(I,4)	ANAL 334
001632	IF (UP14.GT. 9) WRITE (6,6010)	ANAL 335
001640	102 CONTINUE	ANAL 336
001640	RETURN	ANAL 337
001641	END	ANAL 338

SUBPROGRAM LENGTH
002665

UNUSED COMPILER SPACE
034400

	FUNCTION BASFAT (S,FU,Y,YP)	BASF 1
C	DETERMINES THE BASE METAL FATIGUE STRESS	BASF 2
000007	DIMENSION S(4)	BASF 3
C	DETERMINE THE REQUIRED STRESS RATIO	BASF 4
000007	SL = S(1) + S(2) + S(3)	BASF 5
000012	SR = S(1) + S(3) + S(4)	BASF 6
000015	R = SL / SR	BASF 7
000017	IF (ABS(SL).GT. ABS(SR)) R = SR / SL	BASF 8
000026	SK = 1.0 + .7R * (FU/SR - 1.0)	BASF 9
000032	IF (Y.LT. 0.0) GO TO 700	BASF 10
C	DETERMINE THE FATIGUE STRESS IN TENSION	BASF 11
000034	SFA = SK * 20.5 / (1.0 + .55 * R)	BASF 12
000040	GO TO 701	BASF 13
C	DETERMINE THE BASE METAL FATIGUE STRESS FOR COMPRESSION	BASF 14
000041	700 SFA = (.55*YP)/(1.0-((.55*YP/(SK*13.3))-1.0)*R)	BASF 15
000050	701 BASFAT = SFA	BASF 16
000052	RETURN	BASF 17
000052	END	BASF 18

SUBPROGRAM LENGTH
000113

UNUSED COMPILER SPACE
042700

	SUBROUTINE CONDES	COND 1
C	DESIGNS THE SHEAR CONNECTOR SPACING FOR A GIVEN PROBLEM	COND 2
000002	COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FTH(18),WERTH(18),	COND 3
	IDAF(18),IX(18),IY(18),NSECT(18),NOS(18),TSTEEL(20),CS(18),	COND 4
	ZILCON(18),CL(18),IMCON(18),CH(18),PLATH(18),PLWID(18),NF,	COND 5
	3SLATD,SLATH,MR,CODR(18),RT(15,18,3),CC(5,5),FM,SPAN(4),IDSP(5),	COND 6
	4TDFI(18),NA,IFNGTH,RO(18),SULLEN(18),NS,UNID(18),TALO(18),PROP(3),	COND 7
	SFC,STRESS(18,4,2),DESMON(18,4),JFSREA(S,4),DESSH(18,4),REAR(5,3),	COND 8
	6CGSI(18),CONSP(18,3,4)	COND 9


```

000002      C0440N/ONE/JPTR,JPTR      COND 10
000002      C0440N/TWO/LPCM          COND 11
000002      C0440N/THREE/AS,NCS,JPTR COND 12
000002      C0440N/FOUR/HATCH       COND 13
000002      REAL IX,IY,ISTEFF,ILCON,INCON,MN,LENGTH COND 14
000002      DIMENSION LPCM(10),SPACE(8),MMAX(4),C(16,3,4) COND 15
000002      C      ZERO OUT THE REQUIRED ARRAYS COND 16
000002      DO 2999 I = 1, 81          COND 17
000004      SPACE(I) = 0.0            COND 18
000005      2999 R0(I) = 0.0          COND 19
000010      DO 4200 I = 1, 4          COND 20
000011      DO 4200 J = 1, 16       COND 21
000012      DO 4200 K = 1, 3        COND 22
000013      4200 CONSP(I,J,K,I) = 0.0 COND 23
000026      C      DESIGN THE SPACING FOR 7/8 INCH SHEAR CONNECTOR STUDS COND 24
000030      IF (JPT3 .LT. 9) GO TO 6000 COND 25
000030      RETURN                     COND 26
000031      6000 DIA = .475              COND 27
000033      ZR = 10.6 * DIA**2          COND 28
000035      SLA44 = SLA40 * SLARTH/MR COND 29
000035      C      DETERMINE THE NUMBER OF CONNECTORS AT EACH SECTION COND 30
000037      FMIN = 50.0                COND 31
000041      DO 4000 I = 1, NF          COND 32
000042      IA = NOS(I)                 COND 33
000044      IF (FLWID(IA) .LT. FMIN) FMIN = FLWID(IA) COND 34
000050      3000 CONTINUE              COND 35
000053      NCS = 4                     COND 36
000054      IF (FMIN .LT. 12.0) NCS = 3  COND 37
000057      IF (FMIN .GE. 15.0) NCS = 5 COND 38
000063      SZ4 = NCS * ZR             COND 39
000063      C      DETERMINE THE ACTUAL REQUIRED SPACING COND 40
000066      DO 4001 I = 1, NA          COND 41
000067      DO 4010 J = 1, 10, 2        COND 42
000070      IF (LPCM(IJ) .EQ. 0) GO TO 3011 COND 43
000071      IF (I .GE. LPCM(IJ) .AND. I .LE. LPCM(IJ+1)) GO TO 3012 COND 44
000102      3010 CONTINUE            COND 45
000104      3011 IA = NOS(I)            COND 46
000106      IC = 2 * I - 1             COND 47
000110      ID = IC - 1                COND 48
000111      IF (I .EQ. NA) GO TO 3012  COND 49
000113      VR = DESSH(IC,2) - DESSH(IC,4) COND 50
000115      OD = DEPTH(IA) * PLATH(I,2) * HATCH * SLARTH COND 51
000122      SP = SLA44 * (OD - CH(I) - SLARTH/2.) / INCON(I) COND 52
000127      SPL = SZ4 / (VR * SP)     COND 53
000131      IF (I .EQ. 1) GO TO 3003  COND 54
000133      3012 VR = DESSH(ID,2) - DESSH(ID,4) COND 55
000136      IB = NOS(I-1)              COND 56
000140      OD = DEPTH(IB) * PLATH(I-1,2) * HATCH * SLARTH COND 57
000144      SP = SLA44 * (OD - CH(I-1) - SLARTH/2.) / INCON(I-1) COND 58
000151      SPL = SZ4 / (VR * SP)     COND 59
000154      IF (I .EQ. NA) GO TO 3004  COND 60
000156      SPACE(I) = SPL             COND 61
000157      IF (SPR .LT. SPL) SPACE(I) = SPR COND 62
000162      GO TO 3001                COND 63
000163      3013 SPACE(I) = SPR        COND 64
000165      GO TO 3001                COND 65
000166      3014 SPACE(I) = SPL        COND 66
000170      GO TO 3001                COND 67

```

```

000171 3012 SPACE(1) = 0.0 COND 68
000173 3001 CONTINUE COND 69
C DETERMINE REQUIRED SPACING TO THE NEAREST 3 INCH INCREMENT COND 70
000174 DO 4015 I = 1, NA COND 71
000177 N3 = SPACE(1) / 3. COND 72
000202 R0(1) = N3 * 3. COND 73
000205 IF (R0(1) .GT. 24.) R0(1) = 24. COND 74
000210 3015 CONTINUE COND 75
C FILL THE CONSP ARRAY WITH THE CORRECT VALUES COND 76
000213 MN = NS - 1 COND 77
C BEGIN THE LOOP FOR EACH SPAN COND 78
000215 DO 4051 I = 1, MN COND 79
C DETERMINE THE Crossover POINTS IN THE SPAN COND 80
000216 IF (MN .EQ. 1) GO TO 3050 COND 81
000220 IA = 1 COND 82
000221 IA = 10A COND 83
000222 GO TO 3053 COND 84
000223 3050 IF (I .EQ. 1) GO TO 3052 COND 85
000225 IA = 1 COND 86
000226 IA = LPCM(1) - 1 COND 87
000230 GO TO 3053 COND 88
000230 3052 IF (I .EQ. MN) GO TO 3054 COND 89
000232 DO 4055 JJ = 1, 10 COND 90
000234 IA = LPCM(JJ) + 1 COND 91
000236 IF (IA .GT. 10SP(MN)) GO TO 3056 COND 92
000242 3055 CONTINUE COND 93
000244 3056 IA = 1A COND 94
000246 GO TO 3053 COND 95
000246 3054 DO 4057 J = 2, 10, 2 COND 96
000250 IA = LPCM(J) + 1 COND 97
000252 IA = LPCM(J+1) - 1 COND 98
000254 IF (LPCM(J) .GE. 10SP(I) .AND. LPCM(J+1) .LE. 10SP(I+1)) GO TO 3053 COND 99
000265 3057 CONTINUE COND 100
C DETERMINE THE STARTING AND ENDING COORDINATE FOR THE CONNECTORS COND 101
000267 3053 NMAX(1) = IA COND 102
000271 DMAX = 0.0 COND 103
000272 DO 4059 J = 1A, 1B COND 104
000273 IF (DESMD(J,2) .LT. DMAX) GO TO 4058 COND 105
000276 DMAX = DESMD(J,2) COND 106
000277 NMAX(1) = J COND 107
000300 4050 CONTINUE COND 108
000303 SCOR = COOR(1A) COND 109
000305 FCOR = COOR(1A) COND 110
000306 J = 1 COND 111
000307 DO 4080 K = 1A, 1B COND 112
000310 IF (K .EQ. 1B) GO TO 3091 COND 113
000312 IF (ABS(R0(K) - R0(K+1)) .LT. .00001) GO TO 3090 COND 114
000316 IF (R0(K) .GT. R0(K+1)) GO TO 3085 COND 115
000322 XD = (R0(K+1) - SPACE(K)) * SURLEN(K) / (SPACE(K+1) - SPACE(K)) COND 116
000326 NRB = XD / R0(K) COND 117
000331 XD = R0(K) * NRB + R0(K) COND 118
000334 FCOR = FCOR + XD COND 119
000335 GO TO 3089 COND 120
000336 3085 XD = (SPACE(K) - R0(K)) * SURLEN(K) / (SPACE(K) - SPACE(K+1)) COND 121
000342 NRB = XD / R0(K) COND 122
000346 XD = R0(K) * NRB COND 123
000350 FCOR = FCOR + XD COND 124
000351 3089 CONSP(J+1,I) = R0(K) COND 125

```

```

000356      CONSP(J,2,1) = SCOR                      COND 126
000361      CONSP(J,3,1) = FCOR                      COND 127
000364      SCOR = FCOR                              COND 128
000365      FCOR = CONSP(K+1)                        COND 129
000366      J = J + 1                                  COND 130
000367      GO TO 3030                                COND 131
000370 3031 XD = CONSP(IH) = SCOR                  COND 132
000373      NRH = XD / R0(K)                          COND 133
000376      XD = XD - NRH * R0(K)                    COND 134
000401      FCOR = CONSP(IH) - XD                    COND 135
000403      CONSP(J,1,1) = R0(K)                     COND 136
000410      CONSP(J,2,1) = SCOR                      COND 137
000413      CONSP(J,3,1) = FCOR                      COND 138
000416      J = J + 1                                  COND 139
000417      GO TO 3030                                COND 140
000420 3040 FCOR = FCOR + SURLFN(K)                COND 141
000423 3040 CONTINUE                               COND 142
000426 3051 CONTINUE                               COND 143
C      ADJUST THE SPACINGS FOR THE LAST SPAN      COND 144
000430      DO 4100 J = 1, 16                        COND 145
000432      JJ = J                                    COND 146
000433      IF (ABS(CONSP(J,3,MN)) .LT. .0001) GO TO 3111 COND 147
000441 3100 CONTINUE                               COND 148
000442 3111 JJ = JJ - 1                             COND 149
000444      XD = FCOR(IH) - CONSP(JJ,3,MN)          COND 150
000450      DO 3102 J = 1, JJ                         COND 151
000452      CONSP(J,3,MN) = CONSP(J,3,MN) + XD      COND 152
000457 3102 CONSP(J,2,MN) = CONSP(J,2,MN) + XD    COND 153
C      CORRECT SPACING FOR INFLECTION POINTS      COND 154
C      DETERMINE NUMBER OF ROWS ADJACENT TO THE INFLECTION POINT COND 155
000464      SRS = (.93 * DIA**2) * SQRT(2500. * FC) COND 156
000472      PSLAB = AS * 40.                         COND 157
000474      NSLAB = PSLAB / (.65 * SRS) + 1           COND 158
000501      NSR = AS * 10. / ZR + 1                  COND 159
000506      IF (INSEV .GT. NSLAB) NSLAB = NSR        COND 160
000511      NR4 = NSLAB / NSR + 1                     COND 161
C      CORRECT THE CONSP ARRAY                     COND 162
000515      DO 4051 I = 1, MN                        COND 163
000517      DO 4052 J = 1, 16                        COND 164
000520      IF (LPCN(J) .EQ. 0) GO TO 4051          COND 165
000521      IA = LPCN(J)                              COND 166
000523      IF (IA .GT. IOSP(I) .AND. IA .LT. IOSP(I+1)) GO TO 4053 COND 167
000533      GO TO 4052                                COND 168
000533 4053 IA = IOSP(I)                            COND 169
000535      IF (CONSP(IA) .GT. CONSP(IH) + SPAN(I) / 2.) GO TO 4054 COND 170
000544      KK = 2                                     COND 171
000545      XD = DEFSDOM(IA+1,1) * SURLFN(IA) / (ABS(DEFSDOM(I+1,1)) + COND 172
-      ABS(DEFSDOM(IA,1)))                          COND 173
000552      GO TO 4155                                COND 174
000553 4054 KK = 3                                   COND 175
000554      XD = DEFSDOM(IA-1,1) * SURLFN(IA-1) / (ABS(DEFSDOM(IA-1,1)) + COND 176
-      ABS(DEFSDOM(IA,1)))                          COND 177
000562 4155 IF (KK .EQ. 3) GO TO 4055              COND 178
000564      NX = (CONSP(I,2,1) - CONSP(IA+1) + XD) / (CONSP(I,1,1) - COND 179
000575      X) = CONSP(I,1,1) * (NX + NP4)           COND 180
000601      CONSP(I,1,1) = CONSP(I,1,1) - XD      COND 181
000605      GO TO 4152                                COND 182
000605 4055 DO 4056 J6 = 1, 16                    COND 183

```

```

000607      K = JK - 1                                COND 184
000611      IF (ABS(CONSP(JK,1,1)) .LT. .00001) GO TO 4077 COND 185
000616 4056 CONTINUE                                COND 186
000620      X0 = (COORD(JA-1) + XD = CONSP(K,KK,1)) / CONSP(K,1,1) COND 187
000633      XD = CONSP(K,1,1) * (X0 + NDA)            COND 188
000640      CONSP(K,KK,1) = CONSP(K,3,1) + XD          COND 189
000645 4052 CONTINUE                                COND 190
000647 4051 CONTINUE                                COND 191
C      CORRECTION FOR ULTIMATE STRENGTH            COND 192
000652      DO 4080 I = 1, MM                       COND 193
000653      IA = MMAX(I)                              COND 194
000655      IB = NOS(IA)                               COND 195
000657      AR = SAREA(IA) + PLATH(IA,1)*PLWID(IA,1) + PLWID(IA,2)*PLATH(IA,2) COND 196
000664      PGOV = AR * PROP(3)                       COND 197
000666      PCOVC = 2.125 * FC * SLARWD * SLATH       COND 198
000671      IF (PCOVC .LT. PGOV) PGOV = PCOVC          COND 199
000674      NREQ = PGOV / (.85 * SUS) + 1            COND 200
000701      DO 4081 K = 1, 16                         COND 201
000703      IF (ABS(CONSP(K,1,1)) .LT. .0001) GO TO 4082 COND 202
000711      IF (COORD(IA).LE.CONSP(K,3,1).AND.COORD(IA).GE.CONSP(K,2,1)) JK = K COND 203
000727      KJ = K                                       COND 204
000731 4081 CONTINUE                                COND 205
000732      ROW = 1.0                                    COND 206
000736      J = 1                                       COND 207
000735 4084 IF (J .EQ. JK) GO TO 4083                COND 208
000737      ROW = ROW + (CONSP(J,3,1) - CONSP(J,2,1)) / CONSP(J,1,1) COND 209
000747      J = J + 1                                    COND 210
000750      GO TO 4084                                  COND 211
000751 4083 ROW = ROW + (COORD(IA) - CONSP(J,2,1)) / CONSP(J,1,1) COND 212
000762      NCON = ROW * NCS + 1                        COND 213
000766      IF (ABS(NCON - ROW * NCS) .GT. .00001) NCON = NCON + 1 COND 214
000775      IF (NCON .GE. NREQ) GO TO 4086            COND 215
001000      NADD = (NREQ - NCON) / NCS + 1            COND 216
001004      NAE = NADD                                  COND 217
001005      NTOT = 0                                    COND 218
001006      NT = 0                                       COND 219
001007      KKK = JK - 1                                COND 220
001010      IF (KKK .EQ. 0) GO TO 180                 COND 221
001011      NAE = NADD / KKK                            COND 222
001013      DO 4550 J = 1, KKK                         COND 223
001015      NTOT = NTOT + NAE                           COND 224
001017      IF (J .EQ. KKK .AND. NTOT .NE. NADD) NAE=NADD-NTOT/NAE COND 225
001027      DM = CONSP(J,3,1) - CONSP(J,2,1)          COND 226
001036      NSP = DM / CONSP(J,1,1) + NAE              COND 227
001042      SP = DM / NSP                              COND 228
001044      NP3 = SP / 3.                                COND 229
001047      SP = NP3 * 3.                                COND 230
001051      CONSP(J,1,1) = SP                          COND 231
001053      NT = NT + DM / SP + NSP + NAE              COND 232
001062      IF (NT .GE. NADD) GO TO 4086              COND 233
001064 4550 CONTINUE                                COND 234
001067      GO TO 4086                                  COND 235
001067 180 D = (COORD(IA) - CONSP(JK,2,1))          COND 236
001074      NSP = D / CONSP(KK,1,1) + NAE             COND 237
001102      SP = D / NSP                              COND 238
001104      NP3 = SP / 3.                                COND 239
001107      SP = NP3 * 3.                                COND 240
001111 185 KH = 15                                    COND 241

```

```

001112      TEMP = CONSP(1,2,1)                                COND 242
001116      188 DO I=6,KC = 1, 3                                COND 243
001120      186 CONSP(KI+1,KC,1) = CONSP(KI,KC,1)                COND 244
001134      IF (KH .EQ. 1) GO TO 187                              COND 245
001136      KH = KH - 1                                             COND 246
001137      GO TO 188                                              COND 247
001137      187 CONSP(1,1,1) = SP                                COND 248
001143      CONSP(1,2,1) = TEMP                                    COND 249
001146      CONSP(1,3,1) = CONSP(1,2,1) + SP * MSP              COND 250
001154      CONSP(2,2,1) = CONSP(1,3,1)                          COND 251
001157      4086 ROW = 1.0                                         COND 252
001161      J = KJ                                                  COND 253
001162      4089 IF (J .EQ. JK) GO TO 4090                          COND 254
001164      ROW = ROW + (CONSP(J,3,1) - CONSP(J,2,1)) / CONSP(J,1,1) COND 255
001174      J = J - 1                                              COND 256
001175      GO TO 4089                                             COND 257
001176      4090 ROW = ROW + (CONSP(J,3,1) - COOR(1A)) / CONSP(J,1,1) COND 258
001207      NCOS = ROW * NCS + 1                                    COND 259
001214      IF (ABS(NCOS) - ROW * NCS) .GT. .0001) NCOS = NCOS - 1 COND 260
001223      IF (NCOS .GE. NREF) GO TO 4500                       COND 261
001226      NADD = (NHEQ - NCOS) / NCS + 1                       COND 262
001232      NAF = NADD                                           COND 263
001233      IF (NJ .EQ. JK) GO TO 150                             COND 264
001235      KKK = KJ - JK                                          COND 265
001236      NAF = NADD / KKK                                       COND 266
001240      J = KJ                                                  COND 267
001241      NTOT = 0                                              COND 268
001242      NT = 0                                                  COND 269
001243      151 NTOT = NTOT + NAF                                  COND 270
001245      IF (J .EQ. JK+1 .AND. NTOT.NE. NADD) NAF=NADD-NTOT+NAF COND 271
001257      DM = CONSP(J,3,1) - CONSP(J,2,1)                    COND 272
001266      MSP = DM / CONSP(J,1,1) + NAF                        COND 273
001272      SP = DM / MSP                                         COND 274
001274      NP1 = SP / 3.                                         COND 275
001277      SP = NP1 * 3.                                         COND 276
001301      CONSP(J,1,1) = SP                                    COND 277
001303      NT = NT + DM/MSP - MSP + NAF                          COND 278
001312      IF (NT .GE. NADD) GO TO 4086                         COND 279
001314      J = J - 1                                             COND 280
001316      IF (J .EQ. JK) GO TO 4086                           COND 281
001317      GO TO 151                                             COND 282
001320      150 D = CONSP(JK,3,1) - COOR(1A)                    COND 283
001325      MSP = D / CONSP(JK,1,1) + NAF                       COND 284
001333      SP = D / MSP                                          COND 285
001335      NP1 = SP / 3.                                         COND 286
001340      SP = NP1 * 3.                                         COND 287
001342      CONSP(JK+1,1,1) = SP                                  COND 288
001346      CONSP(JK+1,3,1) = CONSP(JK,3,1)                    COND 289
001352      CONSP(JK+1,2,1) = CONSP(JK+1,3,1) - MSP * SP      COND 290
001356      CONSP(JK,3,1) = CONSP(JK+1,2,1)                    COND 291
001361      CONTINUE                                           COND 292
001361      4080 CONTINUE                                       COND 293
001364      DO 40 1 = 1, KA                                       COND 294
001365      KSCI = 1                                              COND 295
001366      DO 41 J = 1, 14                                         COND 296
001370      IF (ABS(CONSP(J,1,1)) .LT. .00001) GO TO 30         COND 297
001376      D = CONSP(J,3,1) - CONSP(J,2,1)                    COND 298
001403      NAF = D / CONSP(J,1,1)                              COND 299

```

```

001406 IF (ABS(MXD*CONSP(J,1,1) - D) .GT. .0001) MYD = MXD + 1 COND 300
001415 IF (MXD .GT. 3) GO TO 41 COND 301
001421 KSCI = J COND 302
001422 IF (J .EQ. 1) GO TO 42 COND 303
001424 IF (ABS(CONSP(J+1,1,1)) .LT. .0001) GO TO 43 COND 304
001430 CONSP(J,1,1) = CONSP(J-1,1,1) COND 305
001433 IF (CONSP(J+1,1,1) .LT. CONSP(J-1,1,1)) CONSP(J,1,1) = CONSP(J+1,1,1) COND 306
001440 GO TO 41 COND 307
001441 42 CONSP(1,3,1) = CONSP(1,2,1) + 3. * CONSP(1,1,1) COND 308
001450 CONSP(2,2,1) = CONSP(1,3,1) COND 309
001453 GO TO 41 COND 310
001453 43 CONSP(J,2,1) = CONSP(J,3,1) - 3. * CONSP(J,1,1) COND 311
001463 CONSP(J-1,3,1) = CONSP(J,2,1) COND 312
001466 41 CONTINUE COND 313
001470 39 IF (KSCI .EQ. 0) GO TO 40 COND 314
001471 DO 45 J = 1, 16 COND 315
001473 DO 45 K = 1, 3 COND 316
001474 C(J,K,1) = CONSP(J,K,1) COND 317
001504 45 CONSP(J,K,1) = 0.0 COND 318
001512 KR = 1 COND 319
001513 DO 46 J = 1, 16 COND 320
001515 IF (ABS(C(J,1,1)) .LT. .0001) GO TO 40 COND 321
001523 IF (J .EQ. 1) GO TO 47 COND 322
001524 IF (ABS(C(J-1,1,1) - C(J,1,1)) .GT. .0001) GO TO 47 COND 323
001535 CONSP(KR-1,3,1) = C(1,3,1) COND 324
001541 GO TO 46 COND 325
001541 47 CONSP(KR,1,1) = C(J,1,1) COND 326
001547 CONSP(KR,2,1) = C(1,2,1) COND 327
001552 CONSP(KR,3,1) = C(J,3,1) COND 328
001555 KR = KR + 1 COND 329
001556 46 CONTINUE COND 330
001560 40 CONTINUE COND 331
C FINAL ADJUSTMENTS OF THE CONNECTOR SPACING COND 332
001563 DO 4390 I = 1, MN COND 333
001564 DO 4390 J = 1, 16 COND 334
001565 IF (ABS(CONSP(J,1,1)) .LT. .0001) GO TO 4390 COND 335
001573 XD = CONSP(J,3,1) - CONSP(J,2,1) COND 336
001600 NXD = XD / CONSP(J,1,1) COND 337
001603 IF (ABS(MXD*CONSP(J,1,1) - XD) .GT. .0001) MYD = NXD + 1 COND 338
001612 XD = NXD * CONSP(J,1,1) COND 339
001616 IF (CONSP(J+1,1,1) .LT. CONSP(J,1,1)) GO TO 4390 COND 340
001622 CONSP(J,3,1) = CONSP(J,2,1) + XD COND 341
001625 CONSP(J+1,2,1) = CONSP(1,3,1) COND 342
001627 GO TO 4390 COND 343
001627 4092 IF (ABS(CONSP(J+1,1,1)) .LT. .0001) GO TO 4095 COND 344
001635 CONSP(J,3,1) = CONSP(1,2,1) + XD - CONSP(1,1,1) COND 345
001644 CONSP(J+1,2,1) = CONSP(J,3,1) COND 346
001647 GO TO 4390 COND 347
001647 4093 CONSP(J,3,1) = CONSP(1,2,1) + XD COND 348
001654 4390 CONTINUE COND 349
001663 IF (MN .EQ. 1) GO TO 19 COND 350
001665 DO 4095 I = 1, 16 COND 351
001666 JJ = J COND 352
001667 IF (ABS(CONSP(J,3,MN)) .LT. .0001) GO TO 4096 COND 353
001675 4095 CONTINUE COND 354
001676 4096 JJ = JJ + 1 COND 355
001700 XD = (E/J) * (1 - CONSP(1,3,MN)) COND 356
001704 DO 4097 J = 1, 16 COND 357

```


001706	CONSP(J,3,MN) = CONSP(J,3,MN) + XD	COND 358
001713	4097 CONSP(J,2,MN) = CONSP(J,2,MN) + XD	COND 359
001720	10 CONTINUE	COND 360
001720	RETURN	COND 361
001721	END	COND 362

SUBPROGRAM LENGTH

002531

UNUSED COMPILER SPACE

034100

	SUBROUTINE CPDS	CPDS 1
	DESIGN THE COVER PLATE ARRANGEMENTS FOR A ROLLED SECTION	CPDS 2
000002	COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FTH(18),WPTH(18),	CPDS 3
	1DAF(18),IX(18),IY(18),INSECT(18),NDS(80),ISTEEL(80),CS(80),	CPDS 4
	2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,	CPDS 5
	3SLAWD,SLABTH,MR,CORR(H1),R1(15,H1,2),CC(5,8),FM,SPAN(4),IOSP(5),	CPDS 6
	4TMDEI(H1),NA,LENGTH(40(H1),SUBLEN(80),NS,UNID(9),TRID(10),PROP(3),	CPDS 7
	5FC,STRESS(81,4,2),DESMON(81,4),DESPRA(5,4),DESSH(162,4),DEFA(5,3),	CPDS 8
	6CDS(9),CONSP(16,3,4),COVPL(12,6,2)	CPDS 9
000002	COMMON/ONE/JPT3, IPT10	CPDS 10
000002	COMMON/TWO/LPCM	CPDS 11
000002	COMMON/THREE	CPDS 12
000002	COMMON/FOUR/IST	CPDS 13
000002	COMMON/FIVE/HAUNCH	CPDS 14
000002	COMMON/SIX/NO,KC,SIS,SHS,SLS,ISS,ISS,DMX,DIST	CPDS 15
000002	COMMON/SEVEN/PTH,KT,I	CPDS 16
000002	REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH	CPDS 17
000002	DIMENSION LPCM(10),PTH(40,4),PSH(40,4),PHH(40,4)	CPDS 18
000002	IF (IPT10 .GE. 3) WRITE (6,1040)	CPDS 19
000011	1040 FORMAT (10I,20X,11HSTART CPDS//)	CPDS 20
	C ZERO OUT THE REQUIRED ARRAYS	CPDS 21
000011	DO 4400 I = 1, 12	CPDS 22
000013	DO 4400 J = 1, 6	CPDS 23
000014	DO 4400 K = 1, 2	CPDS 24
000015	9400 COVPL(I,J,K) = 0.0	CPDS 25
000031	MM = NS - 1	CPDS 26
000033	J = 2	CPDS 27
000034	6300 DO 4001 I = 1, NF	CPDS 28
000036	DO 4001 JJ = 1, 4	CPDS 29
000037	9001 PTH(I,JJ) = 0.0	CPDS 30
000047	DO 5061 I = 1, 40	CPDS 31
000051	DO 5061 K = 1, 4	CPDS 32
000052	5061 PHH(I,K) = 0.0	CPDS 33
000062	KH = 1	CPDS 34
000063	NC = KH	CPDS 35
	C DETERMINE THE PTH ARRAY	CPDS 36
000064	DO 4000 I = 1, NF	CPDS 37
000065	IF (I .GT. 1) GO TO 9500	CPDS 38
000070	9510 PTH(KH,1) = PLATH(I,1)	CPDS 39
000075	PTH(KH,2) = PLATH(I,2)	CPDS 40
000100	PTH(KH,3) = CORR(I)	CPDS 41


```

000301      KT = 0                                CPDS 100
000302      IF (IPTIG .EQ. 3) WRITE (6,1041) 1    CPDS 101
000303 104) FORMAT (5X,2)CALCULATIONS FOR I = .121 CPDS 102
000303      IF (PTH(1,1) .LT. .1) GO TO 9300      CPDS 103
000304      IF (KILL .EQ. 0) GO TO 190             CPDS 104
000305      IF (ABS(PTH(1,1)-PTH(1-1,1)) .GT. .0001) GO TO 190 CPDS 105
000305      GO TO 9400                             CPDS 106
000305 190 IF (ABS(PTH(1,1)-PTH(1-1,1)) .LT. .0001) GO TO 9300 CPDS 107
000305      IF (KST .EQ. 2) GO TO 9359            CPDS 108
C      DETERMINE THE PLATE END CONDITIONS      CPDS 109
000305      DO 9330 K = 1, NA                     CPDS 110
000305      IA = K                                CPDS 111
000305      IF (ABS(COOR(IA)-PTH(1,3)) .LT. .00001) GO TO 9331 CPDS 112
000305 9330 CONTINUE                            CPDS 113
000305 9331 KW = IA                             CPDS 114
000305      KC = 1                               CPDS 115
000305      IF (I .EQ. 1) GO TO 9333             CPDS 116
000305      IF (PTH(1-1,1) .GT. .1) KC = 2      CPDS 117
000305 9333 ISS = 1                             CPDS 118
000305      IF (PTH(1-1,1) .GT. PTH(1,1)) ISS = 2 CPDS 119
000305      IF (IPTIG .EQ. 4) WRITE (6,1012) K0, KC, ISS CPDS 120
000305 1012 FORMAT (5X,6HKO = ,12,2X,5HKC = ,12,2X,6HISS = ,12) CPDS 121
000305      IF (KNN .EQ. 1) GO TO 9329          CPDS 122
C      DETERMINE THE MAXIMUM ALLOWABLE CUTOFF CPDS 123
000305      DO 9310 K = 2, MN                     CPDS 124
000305      ID = 105P(K)                          CPDS 125
000305      IF (KPSI .EQ. 1) GO TO 7700          CPDS 126
000305      IF (COOR(ID) .GE. PTH(1,3) .AND. COOR(ID) .LE. PTH(1,4)) GO TO 9315 CPDS 127
000305      GO TO 9310                          CPDS 128
000305 7700 IF (COOR(ID) .GE. PTH(1-1,3) .AND. COOR(ID) .LE. PTH(1-1,4)) CPDS 129
000305 190 GO TO 9317                          CPDS 130
000305 9310 CONTINUE                            CPDS 131
000305 9329 IF (KILL .EQ. 1) GO TO 9317        CPDS 132
000305      DMAX = SUMLEN(IA)                     CPDS 133
000305      IF (ISS .EQ. 2) DMAX = SUMLEN(IA-1)   CPDS 134
000305      GO TO 9421                          CPDS 135
000305 9315 IF (KR .EQ. 2) GO TO 9317          CPDS 136
000305      NDD = DEPTH(151) + 19.              CPDS 137
000305      D = NDD * 1.0                       CPDS 138
000305      DMAX = COOR(ID) - D - PTH(1,3)      CPDS 139
000305      KR = 1                              CPDS 140
000305      GO TO 9421                          CPDS 141
000305 9317 NDD = DEPTH(151) + 19.            CPDS 142
000305      D = NDD * 1.0                       CPDS 143
000305      DMAX = PTH(1,4) - COOR(ID) - D      CPDS 144
000305      KPSI = 6                           CPDS 145
000305      IF (KILL .EQ. 1) DMAX = PTH(1,3) - COOR(ID) - D CPDS 146
000305 9321 KK = K0                             CPDS 147
000305      IF (IPTIG .EQ. 4) WRITE (6,1026) DMAX CPDS 148
000305 1026 FORMAT (6X,7HDMAX = ,F15,2)        CPDS 149
000305      IF (KILL .EQ. 1) GO TO 940        CPDS 150
000305      IF (ISTEFL(K0) .GT. ISTEFL(K0-1)) KK = K0 - 1 CPDS 151
000305      IF (J .EQ. 2) GO TO 6024           CPDS 152
C      DETERMINE THE SECTION PROPERTIES AT THE PLATE END CPDS 153
000305      TD = DEPTH(151) + PLATH(KK,1) + PLATH(KK,2) CPDS 154
000305      GO TO 6025                          CPDS 155
000305 6024 TD = 0.                            CPDS 156
000305 6025 S15 = ISTEFL(KK) / ABS(TD - CS(KK)) CPDS 157

```

```

000542      PLT = PLATH(KK,2)                                CPNS 154
000543      PLW = PLW(KK,2)                                  CPNS 155
000545      IF (JPTX.GT. 9) GO TO 490                       CPNS 160
000550      SIS = IICCN(KK) / ABS(ID - CI(KK))              CPNS 161
000554      SHS = IICCN(KK) / ABS(ID - CH(KK))              CPNS 162
000560      490 APIOD = PLT*PLW*(DEPTH(IST)/2.+PLT/2.)*1000 CPNS 163
000565      AISFC = (PLW * PLT*1000) / 12.                 CPNS 164
000570      SS = IX(IST) + 2.*AISFC + 2.*APIOD              CPNS 165
000575      SS = SS / (DEPTH(IST)/2. + PLT)                 CPNS 166
000600      IF (JPTX.GT. 9) SIS = SS                         CPNS 167
000604      IF (JPTX.GT. 9) SHS = SS                       CPNS 168
000607      IF (JPTX.GT. 9) SHS = SS                       CPNS 169
000612      IF (KILL.EQ. 1) GO TO 590                      CPNS 170
000614      IF (ABS(PLATH(KK,1)).LT..0001 .AND. 1.EQ. 1) SS = SIS CPNS 171
000630      GO TO 340                                         CPNS 172
000631      240 PLT = PTH(I-1,1)                            CPNS 173
000633      PLW = PTH(I-1,2)                                CPNS 174
000635      IF (PTH(I,1).LT. PTH(I-1,1)) PLT = PTH(I,1)     CPNS 175
000641      IF (PTH(I,1).LT. PTH(I-1,1)) PLW = PTH(I,2)     CPNS 176
000645      IF (JPTX.GT. 9) GO TO 490                       CPNS 177
000652      ID = DEPTH(IST) + PLT                          CPNS 178
000654      IF (JPTX.GT. 2) ID = 0.0                       CPNS 179
000657      HP = DEPTH(IST)/2. + PLT                       CPNS 180
000663      CALL ICLC(0.0,0.0,PLW,PLT,DEPTH(IST),SAFE(IST),IX(IST),HP,0., CPNS 181
        ISC,IST)                                           CPNS 182
000677      TSA = SAFE(IST) + PLT*PLW                     CPNS 183
000703      H = 2. * (DEPTH(IST) + PLT - SC)              CPNS 184
000706      CALL ICLC(SLABW/(3.*H),SLABTH,0.0,0.0,H,TSA,SIS,SC,HAUNCH,CSL,SL) CPNS 185
000723      CALL ICLC(SLABW/4.,SLABTH,0.0,0.0,H,TSA,SIS,SC,HAUNCH,CSH,SH) CPNS 186
000740      SIS = SIS/ABS(ID - SC)                          CPNS 187
000744      SHS = SH/ABS(ID - CSH)                        CPNS 188
000747      SLS = SL/ABS(ID - CSL)                        CPNS 189
000753      GO TO 490                                       CPNS 190
000753      540 CONTINUE                                     CPNS 191
000753      340 KILL = 0                                     CPNS 192
000754      IF (IPTIO.EQ. 4) WRITE (6,1027) SIS, SHS, SLS, SS CPNS 193
000772      1027 FORMAT (2X,6H SIS = ,F10.3,2X,6H SHS = ,F10.3,2X,6H SLS = ,F10.3,2X, CPNS 194
        1SHS = ,F10.3)                                     CPNS 195
C      CALL ON THE CUTOFF SUBROUTINE                      CPNS 196
000772      CALL TFCIN                                     CPNS 197
C      PRINT THE CUTOFF DISTANCE IF REQUIRED               CPNS 198
000773      IF (IPTIO.GE. 3) WRITE (6,110) DIST            CPNS 199
001004      110 FORMAT (/20X,10H CUTOFF DISTANCE = ,F15.5//) CPNS 200
C      ADJUST THE PTH ARRAY                               CPNS 201
001004      IF (PT.EQ. 1) GO TO 9360                      CPNS 202
001004      PTH(I-1,4) = PTH(I-1,4) + DIST                CPNS 203
001011      PTH(I,3) = PTH(I-1,4)                        CPNS 204
001012      7000 IF (IPTIO.EQ. 4) WRITE (6,1002) (PTH(I-1,K), K = 1, 4) CPNS 205
001030      IF (IPTIO.EQ. 4) WRITE (6,1002) (PTH(I,K), K = 1, 4) CPNS 206
001046      IF (KK.GT. 9) GO TO 9354                      CPNS 207
001051      GO TO 9354                                       CPNS 208
001051      9340 PTH(I,4) = PTH(I,4) + DIST                CPNS 209
001054      IF (I.EQ. 10) GO TO 9360                      CPNS 210
001056      PTH(I,1,3) = PTH(I,1)                          CPNS 211
001057      IF (IPTIO.EQ. 4) WRITE (6,1002) (PTH(I,K), K = 1, 4) CPNS 212
001075      IF (IPTIO.EQ. 4) WRITE (6,1002) (PTH(I,1,K), K = 1, 4) CPNS 213
001113      IF (ABS(PTH(I-1,4)-PTH(I,1,3)).GT. .0001) GO TO 9360 CPNS 214
001121      K = 0                                           CPNS 215

```

001121	KILL = 1	CPDS 216
001122	IF (PTH(I+2,1) .LT. PTH(I,1)) GO TO 94	CPDS 217
001125	PTH(I+1,1) = PTH(I,1)	CPDS 218
001126	PTH(I+1,2) = PTH(I,2)	CPDS 219
001130	GO TO 9400	CPDS 220
001130	94 PTH(I+1,1) = PTH(I+2,1)	CPDS 221
001132	PTH(I+1,2) = PTH(I+2,2)	CPDS 222
001134	PTH(I+1,3) = PTH(I+1,3) - DIST	CPDS 223
001136	PTH(I,4) = PTH(I+1,3)	CPDS 224
001137	KPSI = 1	CPDS 225
001140	GO TO 9300	CPDS 226
001141	9359 IF (I .EQ. NC) GO TO 9368	CPDS 227
001143	IF (PTH(I+1,1) .GT. .1) GO TO 9300	CPDS 228
001147	IF (KR .LT. .5) GO TO 9368	CPDS 229
001152	9358 KR = 2	CPDS 230
001153	9368 KT = 1	CPDS 231
C	CUIOFF THE RIGHT END IF REQUIRED	CPDS 232
001154	DO 9357 K = 1, NA	CPDS 233
001156	IH = K	CPDS 234
001157	IF (ABS(COOR(IH)-PTH(I,4)).LT..00001) GO TO 9356	CPDS 235
001165	9357 CONTINUE	CPDS 236
001167	9356 KQ = IH	CPDS 237
001171	KC = 1	CPDS 238
001172	ISS = 2	CPDS 239
001173	IF (IP110 .EQ. 4) WRITE (6,1012) KQ, KC, ISS	CPDS 240
001206	IF (KR .EQ. 2) GO TO 9371	CPDS 241
001210	GO TO 9329	CPDS 242
001211	9371 IF (PTH(I+1,1) .GT. .1) KC = 2	CPDS 243
001215	IF (PTH(I,1) .LT. PTH(I+1,1)) ISS = 1	CPDS 244
001221	IF (IP110 .EQ. 4) WRITE (6,1012) KQ, KC, ISS	CPDS 245
001235	GO TO 9315	CPDS 246
001236	9300 CONTINUE	CPDS 247
001241	NTOT = NC	CPDS 248
C	SET UP THE PHH ARRAY	CPDS 249
001242	DO 5060 I = 1, NTOT	CPDS 250
001243	DO 5060 K = 1, 4	CPDS 251
001244	5060 PHH(I,K) = PTH(I,K)	CPDS 252
001257	KIPR = 0	CPDS 253
001260	NDOP = 37. + 2. * DEPTH(IST)	CPDS 254
001264	DOP = NDOP * 1.	CPDS 255
C	SET UP THE OPTIMIZATION PROCEDURE IF END PLATES ARE TOO SHORT	CPDS 256
001267	I25 = 1	CPDS 257
001270	I2F = NTOT	CPDS 258
001271	IF (PHH(I,1) .LT. .1) I25 = 2	CPDS 259
001274	IF (PHH(NTOT,1) .LT. .1) I2F = NTOT - 1	CPDS 260
001301	DS = PHH(I25,4) - PHH(I25,3)	CPDS 261
001304	DF = PHH(I2F,4) - PHH(I2F,3)	CPDS 262
001306	IF (DS .GT. DOP .AND. DF .GT. DOP) GO TO 5082	CPDS 263
001317	FTOT = 999999999.	CPDS 264
001320	KIP2 = 0	CPDS 265
001321	IF (IS .GE. DOP) GO TO 5080	CPDS 266
001323	KIP2 = 1	CPDS 267
001324	IF (I25 .EQ. 1) GO TO 5080	CPDS 268
001326	IF (DOP - DS .GT. PHH(I,4) - PHH(I,3)) GO TO 5080	CPDS 269
001333	KIP2 = 2	CPDS 270
001334	5080 KIPR = 0	CPDS 271
001335	IF (IF .GE. DOP) GO TO 5081	CPDS 272
001340	KIPR = 1	CPDS 273

```

001341      IF (1ZF .EQ. N10T) GO TO 508)          CPDS 274
001343      IF (DOP = DF .LT. PTH(N10T,4) - PTH(N10T,3)) GO TO 508) CPDS 275
001350      KIPF = 2                                  CPDS 276
001351      508) KIPPER = 1                          CPDS 277
001352      KPS = KIPF                               CPDS 278
001354      KPF = KIPF                               CPDS 279
001355      5500 IF (KPS .EQ. 0) GO TO 54          CPDS 280
001356      IF (KPS .EQ. 1) GO TO 54                CPDS 281
001360      PTH(1ZS,3) = PTH(1ZS,4) - DOP         CPDS 282
001363      PTH(1ZS-1,4) = PTH(1ZS,3)            CPDS 283
001364      GO TO 55                                CPDS 284
001365      54 PTH(1ZS,1) = PTH(1ZS,1,1)          CPDS 285
001367      55 IF (KPF .EQ. 0) GO TO 5025        CPDS 286
001370      IF (KPF .EQ. 1) GO TO 56            CPDS 287
001372      PTH(1ZF,4) = PTH(1ZF,3) + DOP       CPDS 288
001375      PTH(1ZF+1,3) = PTH(1ZF,4)          CPDS 289
001376      GO TO 5025                           CPDS 290
001377      56 PTH(1ZF,1) = PTH(1ZF-1,1)         CPDS 291
001401      5025 N44 = N10T                     CPDS 292
001403      K1 = 1                                CPDS 293
001404      DO 40 I = 1, N10T                     CPDS 294
001405      IF (I .EQ. 1) GO TO 39               CPDS 295
001407      IF (ABS(PTH(I,1)-PTH(I-1,1)).GT..0001) GO TO 39 CPDS 296
001414      PTH(N1-1,4) = PTH(I,4)             CPDS 297
001416      N44 = N44 - 1                      CPDS 298
001420      GO TO 40                            CPDS 299
001420      39 PTH(N1,1) = PTH(I,1)            CPDS 300
001423      PTH(N1,2) = PTH(I,2)              CPDS 301
001424      PTH(N1,3) = PTH(I,3)              CPDS 302
001426      PTH(N1,4) = PTH(I,4)              CPDS 303
001427      K1 = K1 + 1                        CPDS 304
001431      40 CONTINUE                        CPDS 305
001434      NC = N44                           CPDS 306
001435      IF (1PT10 .LT. 3) GO TO 101         CPDS 307
001440      WRITE (6,1038)                     CPDS 308
001443      1038 FORMAT (1H1,10X,44HPH ARRAY AFTER CUTOFF AND LENGTH ADJUSTMENT//) CPDS 309
001443      DO 1009 I = 1, NC                  CPDS 310
001445      1009 WRITE (6,1002) (PTH(I,K), K = 1, 4) CPDS 311
001463      IF (KIPPER .EQ. 1) WRITE (6,5800) KPF,KIPF,KPS,KIPS CPDS 312
001500      5800 FORMAT (1H0,10X,87HTHE FLOPT SUBROUTINE IS CALLED A NUMBER OF TIME CPDS 313
      15 BECAUSE THE END PLATES ARE TOO SHORT. 15X,6HKPF = ,11,2X, CPDS 314
      27HKIPF = ,12,2X,6HKPS = ,12,2X,7HKIPS = ,12) CPDS 315
      CALL DD FLOPT TO DETERMINE THE SPLICE LOCATIONS CPDS 316
001500      C 101 CALL FLOPT (PTH,NC,COST,FT) CPDS 317
001503      IF (KIPPER .EQ. 0) GO TO 5100       CPDS 318
001504      IF (FT .GT. F10T) GO TO 5110         CPDS 319
001510      DO 5111 I = 1, 40                  CPDS 320
001511      DO 5111 K = 1, 4                   CPDS 321
001512      5111 PSH(I,K) = 0.0                 CPDS 322
001522      DO 5112 I = 1, NC                  CPDS 323
001523      DO 5112 K = 1, 4                   CPDS 324
001524      5112 PSH(I,K) = PTH(I,K)          CPDS 325
001537      N10 = NC                           CPDS 326
001540      F10 = FT                             CPDS 327
001541      IF (1PT10 .EQ. 4) WRITE (6,81)      CPDS 328
001547      81 FORMAT (1H1,10X,24HTHE CURRENT PSH ARRAY...//) CPDS 329
001547      IF (1PT10 .NE. 4) GO TO 82        CPDS 330
001551      DO 43 I = 1, N10                   CPDS 331

```

```

001553      23 WRITE (6,1002) (PSM(I,K), K = 1, 4) CPDS 332
001571      22 IF (IPTIC .GE. 3) WRITE (6,5401) FTOT CPDS 333
001602      5401 FORMAT (1H0,30X,7HFTOT = ,F15.5) CPDS 334
001602      5110 IF (KPF .EQ. 0) GO TO 5120 CPDS 335
001603      KPF = KPF + 1 CPDS 336
001605      IF (KPF .EQ. 0) GO TO 5120 CPDS 337
001606      GO TO 5100 CPDS 338
001606      5120 IF (KIPS .EQ. 0) GO TO 5400 CPDS 339
001607      KPS = KPS + 1 CPDS 340
001611      IF (KPS .EQ. 0) GO TO 5400 CPDS 341
001612      KPF = KIPF CPDS 342
001613      5300 DO 5310 I = 1, 40 CPDS 343
001615      DO 5310 K = 1, 4 CPDS 344
001616      5310 PTH(I,K) = 0.0 CPDS 345
001626      DO 5320 I = 1, NTOT CPDS 346
001627      DO 5320 K = 1, 4 CPDS 347
001630      5320 PTH(I,K) = PHH(I,K) CPDS 348
001643      GO TO 5500 CPDS 349
001644      5400 DO 5440 I = 1, 40 CPDS 350
001646      DO 5440 K = 1, 4 CPDS 351
001647      5440 PTH(I,K) = 0.0 CPDS 352
001657      NC = NTO CPDS 353
001660      DO 5441 I = 1, NC CPDS 354
001662      DO 5441 K = 1, 4 CPDS 355
001663      5441 PTH(I,K) = PSH(I,K) CPDS 356
001676      5100 KH = 0 CPDS 357
C      SET UP THE COVER PLATE ARRAY CPDS 358
001677      DO 8000 I = 1, NC CPDS 359
001701      IF (PTH(I,1) .LT. .1) GO TO 8000 CPDS 360
001704      IF (ABS(PTH(I,1)-PTH(1-1,1)) .LT. .00001) GO TO 8003 CPDS 361
001710      KH = KH + 1 CPDS 362
001712      COVPL(KH,1,J) = PTH(I,1) CPDS 363
001716      COVPL(KH,2,J) = PTH(I,4) CPDS 364
001721      COVPL(KH,3,J) = PTH(I,1) CPDS 365
001724      COVPL(KH,4,J) = PTH(I,2) CPDS 366
001727      G) TO 8000 CPDS 367
001730      8003 COVPL(KH,2,J) = PTH(I,4) CPDS 368
001735      8010 CONTINUE CPDS 369
C      PRINT THE COVPL ARRAY IF REQUIRED CPDS 370
001740      IF (IPTIC .LT. 3) GO TO 102 CPDS 371
001742      WRITE (6,1050) CPDS 372
001746      1050 FORMAT (1H0,10X,10HCOVPL ARRAY.....//) CPDS 373
001746      DO 1051 I = 1, 12 CPDS 374
001750      1051 WRITE (6,1053) (COVPL(I,K,J), K = 1, 4) CPDS 375
001767      1053 FORMAT (10X,4(F15.5,5X)) CPDS 376
C      SET UP THE PLATH AND PLMID ARRAYS CPDS 377
001767      102 DO 6110 I = 1, NF CPDS 378
001771      IF (I .EQ. 1) GO TO 10 CPDS 379
001773      DO 6250 II = 2, MN CPDS 380
001774      ID = IOSP(II) CPDS 381
001776      IF (I .EQ. ID-1 .OR. I .EQ. ID) GO TO 6110 CPDS 382
002005      6250 CONTINUE CPDS 383
002007      10 D1 = (COOR(I) + COOR(I+1)) / 2, CPDS 384
002013      DO 6109 JJ = 1, 12 CPDS 385
002014      IF (ABS(COVPL(JJ,1,J)) .LT. .00001 .AND. ABS(COVPL(JJ,2,J)) CPDS 386
002034      1, .LT. .00001) GO TO 6007 CPDS 387
002034      IF (DT .GE. COVPL(JJ,1,J) .AND. DT .LE. COVPL(JJ,2,J)) GO TO 6108 CPDS 388
002046      6119 CONTINUE CPDS 389

```

```

002053 6007 PLATH(I,J) = 0.00 CPDS 390
002054 PLWID(I,J) = 0.000 CPDS 391
002055 GO TO 6110 CPDS 392
002057 6108 CONTINUE CPDS 393
002057 IF (ABS(DT - COVPL(JJ,1,J)) .LT. .00001) GO TO 4001 CPDS 394
002066 IF (ABS(DT - COVPL(JJ,2,J)) .LT. .00001) GO TO 4000 CPDS 395
002075 PLATH(I,J) = COVPL(JJ,3,J) CPDS 396
002103 PLWID(I,J) = COVPL(JJ,4,J) CPDS 397
002110 GO TO 6110 CPDS 398
002111 4000 KJST = JJ CPDS 399
002113 IF (COVPL(JJ+1,3,J) .LT. COVPL(JJ,3,J)) KJST = JJ + 1 CPDS 400
002123 PLATH(I,J) = COVPL(KJST,3,J) CPDS 401
002131 PLWID(I,J) = COVPL(KJST,4,J) CPDS 402
002136 GO TO 6110 CPDS 403
002137 4001 KJST = JJ CPDS 404
002141 IF (COVPL(JJ-1,3,J) .LT. COVPL(JJ,3,J)) KJST = JJ - 1 CPDS 405
002151 PLATH(I,J) = COVPL(KJST,3,J) CPDS 406
002157 PLWID(I,J) = COVPL(KJST,4,J) CPDS 407
002164 6110 CONTINUE CPDS 408
002167 IF (J .EQ. 1) GO TO 6200 CPDS 409
002171 IF (JPT3 .GT. 9) GO TO 6350 CPDS 410
002174 J = 1 CPDS 411
C SET UP THE TOP PLATES OVER THE INTERIOR SUPPORTS CPDS 412
002174 DO 6140 I = 1, 10, 2 CPDS 413
002176 IF (LPCM(I) .EQ. 0) GO TO 6145 CPDS 414
002177 IA = LPCM(I) - 1 CPDS 415
002201 IB = LPCM(I+1) CPDS 416
002203 DO 6141 JJ = IA, IB CPDS 417
002205 PLATH(JJ,1) = PLATH(JJ,2) CPDS 418
002207 6141 PLWID(JJ,1) = PLWID(JJ,2) CPDS 419
002212 6140 CONTINUE CPDS 420
002214 6145 CALL SFMIC CPDS 421
C PRINT THE SECTION INFORMATION IF REQUIRED CPDS 422
002215 IF (IPT10 .LT. 4) GO TO 6300 CPDS 423
002220 CALL PRSEC (NF,NOS,NAME,PLWID,PLATH,ISTEEI,CS,1) CON,CL,1HCON,CH, CPDS 424
1COOK,SUHLIN,0,NCY) CPDS 425
GO TO 6300 CPDS 426
002236 6350 DO 6251 I = 1, 12 CPDS 427
002237 DO 6251 K = 1, 6 CPDS 428
002241 6251 COVPL(I,K,1) = COVPL(I,K,2) CPDS 429
002255 DO 6252 I = 1, NF CPDS 430
002256 PLATH(I,1) = PLATH(I,2) CPDS 431
002260 6252 PLWID(I,1) = PLWID(I,2) CPDS 432
002263 6200 RETURN CPDS 433
002264 END CPDS 434

```

SUBPROGRAM LENGTH
003304

UNUSED COMPILER SPACE
0032500

SUBROUTINE CPWDFS

CPWD 1


```

C      DESIGNS ALL OF THE COVER PLATE WELDS
000002 COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FETH(18),WETH(18), CPWD 2
000002 10AF(18),IX(18),IY(18),NEFCT(18),NDS(80),TSTEE(80),CS(80), CPWD 3
000002 2ILCON(80),CL(80),THCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF, CPWD 4
000002 3SLARW(8),SLATH(8),MR,CORR(8),RTI(5,8),3),CC(5,8),FM,SPAN(4),IUSP(5), CPWD 5
000002 4TMOI(80),NA,LENGTH(80)(8)),SHLEF(80),NS,UNDO(80),TMO(10),PHOP(3),CPWD 6
000002 5FC,STRESS(81,4,2),DESMON(81,4),DESFUA(5,4),CFSSH(16,4),RFAR(5,3),CPWD 7
000002 6COST(4),CONSP(16,3,4),COVPL(12,6,2) CPWD 8
000002 COMMON/ONE/JPT3,1PT10 CPWD 9
000002 COMMON/STX/FU CPWD 10
000002 COMMON/FTGH/IST CPWD 11
000002 REAL IX,IY,ISTEE,ILCON,THCON,MR,LENGTH,THINT CPWD 12
000002 DIMENSION IMST(4),TS(4) CPWD 13
C      DETERMINE THE ALLOWABLE WELD STATIC STRESS
000002 FV = 12.4 CPWD 14
000002 IF (ARS(PROP(3)) = 36.1 .GT. 1.45) FV = 14.5 CPWD 15
000002 BEGIN FOR THE TOP PLATES CPWD 16
C      K = 1
000012 K = 1 CPWD 17
000012 IM = IST CPWD 18
000012 IF (JPT3 .LT. 9) SLARA = SLARWD * SLATH CPWD 19
000012 930 DO /80 I = 1, 12 CPWD 20
000022 IF (ARS(COVPL(I,1,K)),LT,.0001.AND,ARS(COVPL(I,2,K)),LT,.0001) CPWD 21
000022 160 TO 700 CPWD 22
C      DETERMINE STARTING AND ENDING MODES
000042 IA = 0 CPWD 23
000042 IB = 0 CPWD 24
000042 DO /80 J = 1, NF CPWD 25
000042 IF (IA .NE. 0) GO TO 782 CPWD 26
000042 IF (COVPL(I,1,K).GE,CORR(J).AND,COVPL(I,1,K).IF,CORR(J+1)) IA=J+1 CPWD 27
000042 782 IF (COVPL(I,2,K).GE,CORR(J).AND,COVPL(I,2,K).IF,CORR(J+1)) IB=J CPWD 28
000102 IF (IA .NE. 0 .AND. IB .NE. 0) GO TO 783 CPWD 29
000110 781 CONTINUE CPWD 30
C      DETERMINE THE END CONDITIONS OF THE PLATE
000112 783 IF (I .EQ. 1) GO TO 801 CPWD 31
000114 IF (I .EQ. 12) GO TO 802 CPWD 32
000116 IF (ARS(COVPL(I,1,K)-COVPL(I+1,2,K)).GT,.0001) GO TO 801 CPWD 33
000122 IF (ARS(COVPL(I,2,K)-COVPL(I+1,1,K)).GT,.0001) GO TO 802 CPWD 34
000136 COVPL(I,5,K) = 0.00 CPWD 35
000140 GO TO 900 CPWD 36
000141 801 IF (ARS(COVPL(I,2,K)-COVPL(I+1,1,K)).GT,.0001) GO TO 830 CPWD 37
000153 837 0 = COVPL(I,1,K) + 1.5*COVPL(I,4,K) - CORR(I) CPWD 38
000164 KT = 1.0 CPWD 39
000166 IF (L .LT. 0.0) KT = 2.0 CPWD 40
000171 IT = IA CPWD 41
000173 IS = IA CPWD 42
000174 DP = ARS(0) CPWD 43
000175 GO TO 7000 CPWD 44
000176 802 IF (ARS(COVPL(I,1,K)-COVPL(I+1,2,K)).GT,.0001) GO TO 830 CPWD 45
000210 835 0 = (COVPL(I,2,K) - 1.5*COVPL(I,4,K) - CORR(I)) CPWD 46
000221 KT = 1.1 CPWD 47
000223 IF (L .LT. 0.0) KT = 2.0 CPWD 48
000226 IT = IB CPWD 49
000230 IS = IB - 1 CPWD 50
000232 DP = ARS(0) CPWD 51
000234 GO TO 7000 CPWD 52
000236 830 DO /3 I = 1, 4 CPWD 53
000236 831 IMST(J) = DESMON(IA+J) CPWD 54
000245 IL = STMAX(IMST) CPWD 55

```

```

000247      DO 432 J = 1, 4                                CPWD 60
000250      R32 TMSI(J) = DESMOD(IR, J)                    CPWD 61
000257      TR = STMXMAX(TMST)                               CPWD 62
000261      IF (ABS(TR) .GT. ABS(TL)) GO TO 435              CPWD 63
000266      GO TO 437                                         CPWD 64
C      DETERMINE THE WELD DESIGN STRESSES                CPWD 65
000266      7000 DO 702 JK = 1, 4                          CPWD 66
000270      DO 703 J = 1, NA                               CPWD 67
000271      703 R0(J) = DESMOD(I, JK)                      CPWD 68
000300      702 TMSI(JK) = IINT(IT, R0, COOR, K), DP, LENGTH, SUB, FN) CPWD 69
000313      IF (K .EQ. 1) GO TO 710                        CPWD 70
000315      TMSI(1) = TMST(1) * CS(1S) / ISTEEL(1S)      CPWD 71
000320      IF (JPT3 .GT. 9) GO TO 100                    CPWD 72
000324      TMSI(2) = TMST(2) * CH(1S) / IHCON(1S)      CPWD 73
000326      TMSI(3) = TMST(3) * CL(1S) / ILCON(1S)      CPWD 74
000331      TMSI(4) = TMST(4) * CH(1S) / IHCON(1S)      CPWD 75
000333      GO TO 720                                       CPWD 76
000334      710 TD = DEPTH(1S) * PLATH(1S,1) + PLATH(1S,2) CPWD 77
000340      TMSI(1) = TMST(1) * (TD - CS(1S)) / ISTEEL(1S) CPWD 78
000344      IF (JPT3 .GT. 9) GO TO 100                    CPWD 79
000347      TMSI(2) = TMST(2) * (TD - CH(1S)) / IHCON(1S) CPWD 80
000352      TMSI(3) = TMST(3) * (TD - CL(1S)) / ILCON(1S) CPWD 81
000354      TMSI(4) = TMST(4) * (TD - CH(1S)) / IHCON(1S) CPWD 82
000361      GO TO 720                                       CPWD 83
000362      100 TMSI(2) = TMST(2) * CS(1S) / ISTEEL(1S)   CPWD 84
000365      TMSI(3) = 0.0                                CPWD 85
000366      TMSI(4) = TMST(4) * CS(1S) / ISTEEL(1S)      CPWD 86
000370      720 TDS = STMXMAX(TMST)                      CPWD 87
000373      F = ABS(TDS * COVPL(1,4,K) * COVPL(1,3,K))   CPWD 88
C      DETERMINE THE ALLOWABLE STRESSES                  CPWD 89
000402      SL = TMST(1) + TMST(2) + TMST(3)             CPWD 90
000405      SR = TMST(1) + TMST(3) + TMST(4)             CPWD 91
000407      IF (ABS(SL) .GT. .0001 .AND. ABS(SR) .GT. .0001) GO TO 620 CPWD 92
000423      R = 0.0                                         CPWD 93
000424      GO TO 610                                       CPWD 94
000424      600 R = SL / SR                                CPWD 95
000426      IF (ABS(SL) .GT. ABS(SR)) R = SR / SL          CPWD 96
000432      610 SK = 1.0 + .36 * (FJ/SR - 1.0)          CPWD 97
000437      AFS = 10.8 * SK / (1.0 - .55 * R)            CPWD 98
000443      ASTR = FV                                       CPWD 99
000445      IF (AFS .LT. ASTR) ASTR = AFS                 CPWD 100
C      DETERMINE THE SIZE OF THE WELD                   CPWD 101
000450      TWS = F / (2.828 * COVPL(1,4,K) * ASTR)      CPWD 102
000456      N16 = TWS / .0625                             CPWD 103
000461      IF (ABS(TWS - N16*.0625) .GT. .0001) N16 = N16 + 1 CPWD 104
000470      TWS = .0625 * N16                             CPWD 105
000472      TMAX = FLTH(1S)                               CPWD 106
000474      TTM = FLTH(1S)                                 CPWD 107
000476      IF (COVPL(1,3,K) .GT. TMAX) TMAX = COVPL(1,3,K) CPWD 108
000506      IF (COVPL(1,3,K) .LT. TMAX) TTM = COVPL(1,3,K) CPWD 109
000516      TMIN = TOP(TMAX)                               CPWD 110
C      STORE THE REQUIRED WELD SIZE                      CPWD 111
000521      IF (TWS .GT. TMIN) GO TO 500                  CPWD 112
000524      IF (TTM .LT. TMIN) GO TO 503                  CPWD 113
000526      COVPL(1,5,K) = TMIN                            CPWD 114
000531      GO TO 400                                       CPWD 115
000532      503 COVPL(1,5,K) = TMIN                        CPWD 116
000534      IF (TWS .LT. TTM) COVPL(1,5,K) = TTM         CPWD 117

```



```

000544      GO TO 900                                CPWD 118
000545      500 COVPL(I,4,K) = TWS                      CPWD 119
C      CALCULATE THE THICKNESS OF THE SEAL WELD    CPWD 120
000551      900 CPA = COVPL(I,3,K) * COVPL(I,4,K)    CPWD 121
000557      DSS = 0.0                                CPWD 122
C      DETERMINE THE POINT OF MAXIMUM SHEAR        CPWD 123
      DO 750 J = 1A, 1R                            CPWD 124
000560      IL = 2 * J - 2                          CPWD 125
000562      IR = 2 * J - 1                          CPWD 126
000566      DO 751 JK = 1, 4                        CPWD 127
000567      751 TMSI(JK) = DESSH(IL,JK)             CPWD 128
000576      IL = STIRMAX(TMSI)                      CPWD 129
000600      DO 752 JK = 1, 4                        CPWD 130
000601      752 TMSI(JK) = DESSH(IR,JK)             CPWD 131
000610      IR = STIRMAX(TMSI)                      CPWD 132
000612      IF (ABS(IL) .LT. DSS) GO TO 753         CPWD 133
000615      DSS = ABS(IL)                          CPWD 134
000616      IC = J - 1                              CPWD 135
000620      IL = J                                  CPWD 136
000621      ID = IL                                CPWD 137
000622      753 IF (ABS(IR) .LT. DSS) GO TO 750     CPWD 138
000626      DSS = ABS(IR)                          CPWD 139
000627      IC = J                                  CPWD 140
000630      IR = J                                  CPWD 141
000631      ID = IR                                CPWD 142
000632      750 CONTINUE                          CPWD 143
000635      IF (DESSH(ID,1) .LT. 0.0) GO TO 3000    CPWD 144
000637      IF (DPT3 .GT. 4) GO TO 3000            CPWD 145
000642      IF (K .EQ. 2) GO TO 3012              CPWD 146
C      DETERMINE THE TOP DESIGN STRESSES IN THE POSITIVE REGION CPWD 147
      TS(1) = DESSH(ID,1)*CPA*(DEPTH(IM)+PLATH(IC,2)-CS(IC)+ CPWD 148
      1PLATH(IC,1)/2.) / ISTEEL(IC)                CPWD 149
000655      TS(2) = DESSH(ID,2)*SLARA*(DEPTH(IM)+PLATH(IC,2)-CH(IC)+SLARTH/2.+CPWD 150
      1HARDCH) / (MR * IHCON(IC))                  CPWD 151
000671      TS(3) = DESSH(ID,3)*SLARA*(DEPTH(IM)+PLATH(IC,2)-CL(IC)+SLARTH/2.+CPWD 152
      1HARDCH) / (3.0 * MR * ILCON(IC))             CPWD 153
000705      TS(4) = DESSH(ID,4)*SLARA*(DEPTH(IM)+PLATH(IC,2)-CH(IC)+SLARTH/2.+CPWD 154
      1HARDCH) / (MR * IHCON(IC))                   CPWD 155
000721      GO TO 3001                             CPWD 156
C      DETERMINE THE BOTTOM DESIGN STRESSES IN THE POSITIVE REGION CPWD 157
000722      3012 TS(1) = DESSH(ID,1)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CS(IC))/ CPWD 158
      1ISTEEL(IC)                                    CPWD 159
000733      TS(2) = DESSH(ID,2)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CH(IC))/ CPWD 160
      1IHCON(IC)                                     CPWD 161
000743      TS(3) = DESSH(ID,3)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CL(IC))/ CPWD 162
      1ILCON(IC)                                    CPWD 163
000754      TS(4) = DESSH(ID,4)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CH(IC))/ CPWD 164
      1IHCON(IC)                                    CPWD 165
000764      GO TO 3001                             CPWD 166
C      DETERMINE THE DESIGN STRESSES IN THE NEGATIVE REGION CPWD 167
000765      3000 IF (K .EQ. 2) GO TO 3010          CPWD 168
000767      CONST = CPA * (DEPTH(IM) + PLATH(IC,2) - CS(IC) + PLATH(IC,1)/2.) CPWD 169
      1/ ISTEEL(IC)                                CPWD 170
000777      GO TO 3011                             CPWD 171
001000      3010 CONST = CPA * (DEPTH(IM) + PLATH(IC,2)/2. - CS(IC))/ISTEEL(IC) CPWD 172
001007      3011 TS(1) = DESSH(ID,1) * CONST      CPWD 173
001012      TS(2) = DESSH(ID,2) * CONST            CPWD 174
001013      TS(3) = DESSH(ID,3) * CONST            CPWD 175

```

```

001015      TS(4) = DESSH(10,4) * CONST                      CPWD 176
C          DETERMINE THE TOTAL DESIGN STRESS                CPWD 177
001016      3001 TS0 = STMAX(TS)                             CPWD 178
C          CALCULATE THE FATIGUE ALLOWABLE STRESS           CPWD 179
001021      SL = TS(1) + TS(2) + TS(3)                      CPWD 180
001024      SR = TS(1) + TS(3) + TS(4)                      CPWD 181
001026      IF (ARS(SL).GT..0001 .AND. ARS(SR).GT..0001) GO TO 2700 CPWD 182
001040      AFS = SR * 10.6                                     CPWD 183
001042      GO TO 2701                                         CPWD 184
001042      2700 R = SL / SR                                  CPWD 185
001044      IF (ARS(SL) .GT. ARS(SR)) R = SR / SL            CPWD 186
001050      AFS = SR * 10.6 / (1.0 + .95 * R)                CPWD 187
001055      2701 ASST = FV                                     CPWD 188
001057      IF (AFS .LT. ASST) ASST = AFS                    CPWD 189
C          DETERMINE THE WFLD THICKNESS                     CPWD 190
001062      TWS = ARS(TS0) / (1.414 * ASST)                CPWD 191
001067      N16 = TWS / .0625 + 1                           CPWD 192
001073      IF (ARS(TWS - (N16 - 1) * .0625) .LT. .0001) N16 = N16 - 1 CPWD 193
001103      TWS = N16 * .0625                                CPWD 194
001105      COVPL(1,6,K) = TWS                                CPWD 195
001111      IF (TWS .LT. .1875) COVPL(1,6,K) = .1875      CPWD 196
001116      780 CONTINUE                                     CPWD 197
001120      790 IF (K .EQ. 2) GO TO 840                      CPWD 198
001122      IF (JPT3 .GT. R) GO TO 101                      CPWD 199
001126      K = K + 1                                         CPWD 200
001127      GO TO 930                                         CPWD 201
001127      101 DO 102 II = 1, 12                           CPWD 202
001131      COVPL(II,5,2) = COVPL(II,5,1)                  CPWD 203
001133      102 COVPL(II,6,2) = COVPL(II,6,1)              CPWD 204
001136      840 RETURN                                       CPWD 205
001137      END                                             CPWD 206

```

SUBPROGRAM LENGTH

001330

UNUSED COMPILER SPACE

036500

```

SUBROUTINE DEFLFC                      DEFL 1
C          CALCULATE THE REQUIRED DEFLECTIONS FOR THE FINAL DESIGN DEFL 2
000002      COMMON NAME(18), SARFA(18), DEPTH(18), FLWD(18), FLTH(18), WFRTH(18), DEFL 3
      1DAF(18), IX(18), IY(18), NSECT(18), NDS(R0), ISTEFL(90), CS(R0), DEFL 4
      2ILCON(R0), CL(R0), IHCON(R0), CH(R0), PLATH(R,2), PLWID(R0,2), NF, DEFL 5
      3SLAWD, SLABTH, MR, CDOR(R1), RII(5,R1,3), CC(S,5), FM, SPAN(4), IOSP(5), DEFL 6
      4TMORI(R0), NA, LENGTH, R0(R1), SUBLEN(R0), NS, MIN(9), TRI(10), PROP(3), DEFL 7
      5FC, STRESS(81,4,2), DESMOD(R1,4), DESRFA(S,4), DESSH(162,4), REAR(S,3), DEFL 8
      6CD5(9), CONSP(16,3,4), COVPL(12,6,2) DEFL 9
000002      COMMON/ONE/JPT3, IPT10 DEFL 10
000002      COMMON/IDEF(20), DESDEF(20,4), INDEF, PERCFN, NCYC DEFL 11
000002      COMMON/E/IPT2 DEFL 12
000002      REAL IX, IY, ISTEFL, ILCON, IHCON, MR, LENGTH DEFL 13
000002      DIMENSION ILC(1,3), STORE(10) DEFL 14
C          ZERO OUT THE REQUIRED ARRAYS DEFL 15

```

```

000002      DO 4500 I = 1, NA                                DEFL 16
000004      DO 4600 J = 1, 3                                  DEFL 17
000006      9800 TIL(I,J) = 0.0                               DEFL 18
000015      DO 4501 I = 1, NDEF                                DEFL 19
000017      DO 4601 J = 1, 4                                    DEFL 20
000020      4801 DEF(I,J) = 0.0                                 DEFL 21
000030      C PRINT THE HEADING IF REQUIRED                      DEFL 22
000030      IF (IPT10 .GE. 3) WRITE (6,9700)                  DEFL 23
000037      9700 FORMAT (1H),22HDEFLECTION LOADINGS...//)     DEFL 24
000037      C START THE LOOP FOR EACH DEFLECTION REQUIRED        DEFL 25
000037      DO 4500 I = 1, NDEF                                  DEFL 26
000041      IA = IDEF(I)                                          DEFL 27
000043      DO 4600 K = 1, 3                                      DEFL 28
000043      C DETERMINE THE BENDING MOMENTS DUE TO A UNIT LOAD DEFL 29
000044      IF (JPT1 .GT. 9 .AND. K .EQ. 2) GO TO 750         DEFL 30
000054      DO 4510 J = 1, NA                                   DEFL 31
000055      WP = 0.0                                              DEFL 32
000056      DO 4511 JJ = 1, NS                                    DEFL 33
000060      JD = IOSP(JJ)                                         DEFL 34
000062      IF (JJ .LE. 10) GO TO 9511                          DEFL 35
000064      WR = WP + RIL(JJ,IA,K) * (COORD(J) - COORD(ID))    DEFL 36
000075      9511 CONTINUE                                        DEFL 37
000100      SP = 0.0                                               DEFL 38
000101      IF (JJ .GT. 1A) SP = COORD(J) - COORD(IA)          DEFL 39
000105      RO(J) = WR - SP                                       DEFL 40
000110      9510 CONTINUE                                        DEFL 41
000110      C DETERMINE THE M/EI ARRAY                          DEFL 42
000112      DO 4515 J = 1, NF                                   DEFL 43
000114      UI = ISIFEL(J)                                       DEFL 44
000116      IF (K .EQ. 2) UI = ILCON(J)                         DEFL 45
000121      IF (K .EQ. 3) UI = IHCON(J)                         DEFL 46
000125      IMOEI(J) = (RO(J) + RO(J+1)) / (2. * EI * UI)      DEFL 47
000133      9515 CONTINUE                                        DEFL 48
000133      C DETERMINE THE TANGENT AT THE LEFT END            DEFL 49
000136      WP = 0.0                                              DEFL 50
000137      DO 4516 J = 1, NF                                       DEFL 51
000140      WP = WP + IMOEI(J) * SURLEN(J) * (LENGTH - COORD(J) - SURLEN(J)/2.) DEFL 52
000146      9516 CONTINUE                                        DEFL 53
000150      TALE = WP / LENGTH                                    DEFL 54
000150      C DETERMINE THE DEFLECTION AT EACH ANALYSIS POINT DEFL 55
000152      TIL(I,K) = 0.0                                       DEFL 56
000155      TIL(IA,K) = 0.0                                       DEFL 57
000160      DO 4520 J = 2, NF                                     DEFL 58
000162      WP = 0.0                                              DEFL 59
000163      IR = J - 1                                              DEFL 60
000165      DO 4521 JJ = 1, IH                                    DEFL 61
000165      C STORE THE DEFLECTIONS AS THE INFLUENCE COEFFICIENTS DEFL 63
000166      9521 WP = WP + IMOEI(JJ) * SURLEN(JJ) * (COORD(J) - COORD(IR) - SURLEN(JJ)/2.) DEFL 62
000177      9520 TIL(J,K) = TALE * COORD(J) - WP               DEFL 64
000207      9600 CONTINUE                                        DEFL 65
000211      750 CONTINUE                                        DEFL 66
000211      C LOAD THE INFLUENCE LINES TO DETERMINE THE DEFLECTIONS DEFL 67
000211      CALL LOAD(TIL,IPT0,JNLO,SURLEN,NA,IPT2,LENGTH,COORD,IA,4HDEFL,NS, DEFL 68
000211      IOSP,SPAN,STORE)                                       DEFL 69
000227      IF (IPT10 .GE. 3) WRITE (6,9701) IA, (STORE(KK), KK = 1, 10) DEFL 70
000246      9701 FORMAT (15,X,10(F11.8,X))                      DEFL 71
000246      C DETERMINE THE DEFLECTIONS                        DEFL 72
000246      DEF(I,1) = STORE(1)                                  DEFL 73

```

```

000250      DESDEF(1,3) = STORE(10)                                DEFL 74
000252      POS = 0.0                                              DEFL 75
000253      SEG = 0.0                                              DEFL 76
000254      DO 4550 K = 2, 7                                       DEFL 77
000255          IF (STORE(K) .GT. POS) POS = STORE(K)              DEFL 78
000261          IF (STORE(K) .LT. SEG) SEG = STORE(K)              DEFL 79
000265      9550 CONTINUE                                           DEFL 80
000267      DESDEF(1,2) = POS + STORE(8)                           DEFL 81
000272      DESDEF(1,4) = SEG + STORE(9)                           DEFL 82
000275      9560 CONTINUE                                           DEFL 83
C          PRINT THE DESIGN DEFLECTIONS IF REQUIRED               DEFL 84
000277      IF (IPT10 .LT. 3) GO TO 20                             DEFL 85
000301      WRITE (6,4710)                                           DEFL 86
000305      4710 FORMAT (1H0,21HDESIGN DEFLECTIONS,.,./)          DEFL 87
000305      DO 4711 I = 1, NDEF                                     DEFL 88
000307      4711 I = IDEF(I)                                         DEFL 89
000311      4711 WRITE (6,4702) IA, (DESDEF(I,K), K = 1, 4)        DEFL 90
000331      4702 FORMAT (15,4X,4(F12.9,4X))                       DEFL 91
000331      20 CONTINUE                                           DEFL 92
000331      RETURN                                                 DEFL 93
000332      END                                                    DEFL 94

```

SUBPROGRAM LENGTH

001013

UNUSED COMPILER SPACE

041100

```

SUBROUTINE EOST(R0,SURLN,LENGTH,COORD,NF,TMT,FM,NS,IOSP,SPAN,RI) EOST 1
C      SOLVES FOR THE INITIAL VALUES IN THE 211 MATRIX WHICH ARE THE EOST 2
C      DEFLECTIONS OF A SIMPLE BEAM AT THE SUPPORT POINTS          EOST 3
000016      REAL LENGTH                                           EOST 4
000016      DIMENSION RU(8),SURLN(80),COORD(80),TMT(80),IOSP(5),SPAN(4), EOST 5
000016      RI(4),WA(5),WB(5),TMOE1(80)                          EOST 6
000016      DO 40 I = 1, 4                                           EOST 7
000017      40 WA(I) = 0.0                                           EOST 8
000020      40 WB(I) = 0.0                                           EOST 9
C          CALCULATE TANGENT AT THE LEFT END                       EOST 10
000023      4P = 0.0                                              EOST 11
000024      DO 41 J = 1, NF                                           EOST 12
000025      41 TMOE1(J) = (RU(J)+RU(J+1))/(2.*TMT(J)*FM)            EOST 13
000036      41 4P = WP*TMOE1(J)*SURLN(J)*(LENGTH-COORD(J)-SURLN(J)/2.) EOST 14
000050      41 TALE = 4P/LENGTH                                       EOST 15
C          CALCULATE DEFLECTIONS AT THE INTERIOR SUPPORTS        EOST 16
000051      MN = NS - 1                                              EOST 17
000053      DO 42 J = 2, MN                                           EOST 18
000055      42 IA = IOSP(J-1)                                         EOST 19
000057      42 IB = IOSP(J) - 1                                       EOST 20
000062      42 IC = IOSP(J)                                           EOST 21
000065      DO 43 K = IA, IB                                         EOST 22
000067      43 WA(K) = WA(K) + TMOE1(K) * SURLN(K)                EOST 23
000074      43 WB(K) = WB(K) + TMOE1(K) * SURLN(K) * (COORD(IC)-COORD(K)-SURLN(K)/2.) EOST 24
000112      43 WA(K) = WA(K) + WA(K-1)                             EOST 25

```

000114	WH(I) = WH(J) + WH(J-1) + WA(J-1)*SPAN(I-1)	EQST	26
000122	RP PI(I-1) = CORR(PI) * TALE = WH(J)	EQST	27
000133	RETURN	EQST	28
000133	END	EQST	29

SUBPROGRAM LENGTH
000334

UNUSED COMPILER SPACE
042400

	SUBROUTINE FLOPT (PTH,NC,COST,FT)	FOPT	1
C	OPTIMIZE THE LOCATIONS OF THE COVER PLATE SPIICES	FOPT	2
000007	COMMON/ONE/JPT3,IP110	FOPT	3
000007	DIMENSION STTH(40),PTH(40,4),PCIS(40),COST(9),CT(40),IP(40,40)	FOPT	4
000007	IF (IP110 .GE. 3) WRITE(6,100)	FOPT	5
000020	100 FORMAT (1H1,10X,14HENTRY TO FLOPT//)	FOPT	6
000020	490 DO 1900 I = 1, NC	FOPT	7
000022	IF (PTH(I,2) .LT. 4.0) GO TO 1900	FOPT	8
000025	PLW = PTH(I,2)	FOPT	9
000027	GO TO 1902	FOPT	10
000027	1900 CONTINUE	FOPT	11
000032	1902 DO 970 I = 1, NC	FOPT	12
000034	970 STTH(I) = 0.0	FOPT	13
C	DETERMINE THE STTH ARRAY	FOPT	14
000037	STTH(1) = PTH(I,1)	FOPT	15
000040	K = 2	FOPT	16
000041	IR = 1	FOPT	17
000042	DO 971 I = 2, NC	FOPT	18
000044	DO 972 KK = 1, IR	FOPT	19
000045	IF (ABS(STTH(KK) - PTH(I,1)) .LT. .0001) GO TO 971	FOPT	20
000052	972 CONTINUE	FOPT	21
000055	STTH(K) = PTH(I,1)	FOPT	22
000057	K = K + 1	FOPT	23
000060	IR = K - 1	FOPT	24
000061	971 CONTINUE	FOPT	25
C	PLACE THE STTH ARRAY IN ASCENDING ORDER	FOPT	26
000064	CALL SORT (STTH,IR)	FOPT	27
000066	IF (IP110 .LT. 3) GO TO 400	FOPT	28
000073	WRITE (6,980)	FOPT	29
000077	980 FORMAT (1H0,10X,15HSTTH ARRAY.....//)	FOPT	30
000077	DO 981 I = 1, IR	FOPT	31
000103	981 WRITE (6,982) I, STTH(I)	FOPT	32
000117	982 FORMAT (10X,12,5X,F10,4)	FOPT	33
000117	400 DO 2900 I = 1, NC	FOPT	34
000121	DO 2900 JK = 1, IR	FOPT	35
000122	2900 IP(I,JK) = 0.0	FOPT	36
000133	DO 2901 I = 1, IR	FOPT	37
C	DETERMINE THE STARTING SECTION	FOPT	39
000134	2901 PCIS(I) = 0.0	FOPT	38
000137	I = 1	FOPT	40
000140	2904 IF (ABS(PTH(I,1)) .GT. .0001) GO TO 2903	FOPT	41
000144	I = I + 1	FOPT	42

000147	GO TO 2904	F0PT	43
000147	2903 IS = 1	F0PT	44
	C DETERMINE THE ENDING SECTION	F0PT	45
000151	I = NC	F0PT	46
000152	2906 IF (ABS(PTH(1,1)) .GT. .0001) GO TO 2905	F0PT	47
000160	I = I - 1	F0PT	48
000161	GO TO 2906	F0PT	49
000161	2905 IF = 1	F0PT	50
	C BEGIN THE OPTIMIZATION OF THE COVER PLATES	F0PT	51
000163	DO 4000 I = 1, IF	F0PT	52
000165	PLEN = PTH(1,4) - PTH(1,3)	F0PT	53
000170	IF (IPTIV .EQ. 4) WRITE (6,101) IS,IE,I,PLW,PLEN	F0PT	54
000213	101 FORMAT(1H0,5HIS = ,12,X,5HIE = ,12,X,4HI = ,12,X,6HPLW = ,F5.1, IX,8HPLEN = ,F10.2)	F0PT	55
	C INITIALIZE THE STARTING CONSTANTS AND COSTS	F0PT	57
000213	DO 4001 K = 1, IH	F0PT	58
000215	4001 CT(K) = 0.0	F0PT	59
000220	IF (I .EQ. 1) GO TO 4000	F0PT	60
	C VARY THE THICKNESS OF THE ADJACENT SECTION	F0PT	61
000222	DO 4010 K = 1, IH	F0PT	62
000224	TCT = 99999.	F0PT	63
000226	IF (STH(K) .GT. PTH(1+1,1)) GO TO 4020	F0PT	64
000232	IF (ABS(STH(K) - PTH(1+1,1)) .LT. .00001) GO TO 4020	F0PT	65
000236	CT(K) = 0.0	F0PT	66
000237	GO TO 4010	F0PT	67
	C VARY THE THICKNESS IN THE DESIGN SECTION	F0PT	68
000240	4020 DO 4030 KK = 1, IH	F0PT	69
000242	IF (STH(KK) .GT. PTH(1,1)) GO TO 4040	F0PT	70
000247	IF (ABS(STH(KK) - PTH(1,1)) .LT. .00001) GO TO 4040	F0PT	71
000253	GO TO 4030	F0PT	72
000254	4040 IF (I .NE. 1) GO TO 4050	F0PT	73
	C DETERMINE THE COST OF THE INITIAL SECTION	F0PT	74
000256	TMAX = F0PT (STH(KK))	F0PT	75
000262	CFW = COST(6)*PLW*COST(8)+.5*TMAX**2*PLW*COST(7)	F0PT	76
000273	CSW = 2.*COST(6)+2.* PLEN *(COST(8)+.017594*COST(7))	F0PT	77
000301	IF (ABS(STH(KK)) .GT. .000001) GO TO 4041	F0PT	78
000306	CBW = COST(6)*PLW*COST(8)+PLW*TMAX**2*COST(7)/2.	F0PT	79
000315	GO TO 4042	F0PT	80
	C DETERMINE THE COST OF THE BUTT WELD IF REQUIRED	F0PT	81
000315	4041 TH = STH(K)	F0PT	82
000317	IF (STH(KK) .LT. STH(K)) TH = STH(KK)	F0PT	83
000324	IF (ABS(STH(KK) - STH(K)) .LT. .00001) GO TO 4043	F0PT	84
000331	CHW = COST(6)*PLW*(COST(6)+(.125*TH+.2HH675*TH**2)*COST(7))	F0PT	85
000341	GO TO 4042	F0PT	86
000342	4043 CHW = 0.0	F0PT	87
	C DETERMINE THE MATERIAL COST OF THE PLATE	F0PT	88
000343	4042 CPT = PLEN *PLW*STH(KK)*COST(1)*490./1728.	F0PT	89
	C DETERMINE THE TOTAL COST OF THE SECTION	F0PT	90
000350	TC = CPT + CHW + CFW + CSW + PCTS(KK)	F0PT	91
000356	GO TO 4070	F0PT	92
	C DETERMINE THE COST OF THE SEAL WELDS	F0PT	93
000356	4050 IF (ABS(STH(KK)) .LT. .000001) GO TO 4055	F0PT	94
000362	IF (ABS(STH(KK)) .LT. .000001) GO TO 4056	F0PT	95
000366	CSW = 2.* PLEN *(COST(8)+.017594*COST(7))	F0PT	96
000373	CFW = 0.0	F0PT	97


```

000374      GO TO 4041                                FORT 98
000374      4055 IF (ABS(STH(K)) .LT. .00001) GO TO 4057 FORT 99
C      DETERMINE THE COST OF THE APPROXIMATE END WELDS FORT 100
      TMAX = TOP(STH(K)) FORT 101
      CEW = COST(6)*3.*PLW*COST(8)*PLW*.5*TMAX**2*COST(7) FORT 102
      TC = CEW + PCTS(KK) FORT 103
      GO TO 4070 FORT 104
000421      4056 TMAX = TOP(STH(KK)) FORT 105
      CEW = COST(6)*PLW*COST(6)*.5*TMAX**2*COST(7)*PLW FORT 106
      CEW = 0.0 FORT 107
      CSW = 2.*PLEN*(COST(8)+.017594*COST(7)) FORT 108
      GO TO 4042 FORT 109
000444      4057 TC = PCTS(KK) FORT 110
000446      4070 IF (IP110 .EQ. 4) WRITE (6,102) STH(K), STH(KK), TC FORT 111
000465      102 FORMAT (5A,2I)THICKNESS ON RIGHT = ,F10.5,2A,20THICKNESS ON LEFT FORT 112
      I = ,F10.5,2A,7HCOST = ,F10.2 FORT 113
      IF (IC .GT. ICI) GO TO 4030 FORT 114
C      STORE THE COST AND THE THICKNESS FORT 115
      ICI = IC FORT 116
      INP = KK FORT 117
000473      4030 CONTINUE FORT 118
C      STORE THE THICKNESS REQUIRED IN THE TP ARRAY FORT 119
      TP(I,K) = STH(INP) FORT 120
      CT(K) = ICI FORT 121
      IF (IP110 .EQ. 4) WRITE (6,110) CT(K), TP(I,K) FORT 122
000522      110 FORMAT (1H0,10A,7HCOST = ,F15.5,5A,5HTP = ,F15.5//) FORT 123
000522      4010 CONTINUE FORT 124
C      DETERMINE THE PCTS ARRAY FORT 125
      DO 4090 KKK = 1, 10 FORT 126
000526      PCTS(KKK) = CT(KKK) FORT 127
000532      IF (IP110 .LT. 4) GO TO 4000 FORT 128
000534      WRITE (6,410) FORT 129
000540      410 FORMAT (1H0,10HPCTS ARRAY//) FORT 130
000540      WRITE (6,103) (PCTS(KKK), KKK = 1, 10) FORT 131
000553      103 FORMAT (4(F15.5,5X)) FORT 132
000553      4000 CONTINUE FORT 133
C      VARY THE THICKNESS OF THE LAST SECTION FORT 134
000560      4500 DO 4100 K = 1, 10 FORT 135
000562      IF (STH(K) .GT. PTH(LE,1)) GO TO 4120 FORT 136
000567      IF (ABS(STH(K) - PTH(LE,1)) .LT. .000001) GO TO 4120 FORT 137
000573      PCTS(K) = 0.0 FORT 138
000575      GO TO 4100 FORT 139
C      DETERMINE THE COST OF THE LAST SECTION FORT 140
000575      4120 CPC = STH(K)* PLW * PLEN * COST(1)*490./1728. FORT 141
000602      CSW = 2.*PLEN*(COST(8)+.017594*COST(7)) FORT 142
000607      IF (LE .EQ. 15) CSW = CSW + 2.*COST(6) FORT 143
000614      TMAX = TOP(STH(K)) FORT 144
000620      CEW = COST(6)* PLW *COST(8)*.5*TMAX**2*COST(7)* PLW FORT 145
000631      PCTS(K) = PCTS(K) + CPC + CSW + CEW FORT 146
000636      IF (LE .EQ. 15) PCTS(K) = PCTS(K) + CEW FORT 147
000642      4100 CONTINUE FORT 148
000645      TCT = 99999. FORT 149
C      DETERMINE THE THICKNESS OF THE LAST SECTION FORT 150
000646      DO 4130 K = 1, 10 FORT 151
000650      IF (ABS(PCTS(K)) .LT. .000001) GO TO 4130 FORT 152
000654      IF (PCTS(K) .LT. TCT) GO TO 4130 FORT 153
000657      TCT = PCTS(K) FORT 154
000660      TCH = K FORT 155

```

```

000660 4130 CONTINUE                                FORT 156
000663 IF (IP110 .LT. 3) GO TO 420                FORT 157
000665 IT = IE - 1                                FORT 158
000667 WRITE (6,450)                             FORT 159
000673 450 FORMAT (1H0,10X,1RHFINAL TP ARRAY...//) FORT 160
000673 DO 5520 I = 15, IT                        FORT 161
000700 5520 WRITE (6,5521) I, ( IP(I,K), K = 1, 18) FORT 162
000723 5521 FORMAT (5X,12,5X,10(F7.4,3X))        FORT 163
000723 WRITE (6,455)                             FORT 164
000727 455 FORMAT (1H0,10X,3HHCALCULATIONS FOR THE LAST SEGMENT....//) FORT 165
000727 DO 5522 K = 1, 18                         FORT 166
000733 5522 WRITE (6,5523) K, SITH(K), PCTS(K)    FORT 167
000751 5523 FORMAT (5X,4H K = ,12,2X,7HSITH = ,F10.5,2X,7HCOST = ,F15.5) FORT 168
000751 420 CONTINUE                             FORT 169
000751 PTH(1F,1) = SITH(1CR)                     FORT 170
000754 FI = PCTS(1CR)                             FORT 171
000755 IF (IPT10 .GE. 3) WRITE (6,460) IE, SITH(1CR), FI FORT 172
001000 460 FORMAT (1H0,21HTHICKNESS IN SEGMENT ,I2,3H = ,F10.5,15H WITH A COSE FORT 173
      IT = ,F10.5)                                FORT 174
C DETERMINE THE THICKNESSES OF THE REMAINING SECTIONS FORT 175
001000 I = 1F                                     FORT 176
001002 IF (IS .EQ. 1F) GO TO 4160                FORT 177
001003 4170 DO 4140 K = 1, 18                     FORT 178
001005 1CR = K                                     FORT 179
001006 IF (ABS(SITH(K) - PTH(1,1)) .LT. .00001) GO TO 4150 FORT 180
001013 4140 CONTINUE                             FORT 181
001016 WRITE (6,5000)                             FORT 182
001021 5000 FORMAT (1H0,10X,20HERROR FOUND IN FLOPT) FORT 183
001021 STOP                                         FORT 184
001023 4150 PTH(1-1,1) = TP(1-1,1CR)             FORT 185
001032 PTH(1-1,2) = PLW                           FORT 186
001034 IF (ABS(PTH(1-1,1)) .LT. .00001) PTH(1-1,2) = 0.0 FORT 187
001042 IF (1 .EQ. 15 + 1) GO TO 4160             FORT 188
001045 I = I - 1                                   FORT 189
001046 GO TO 4170                                 FORT 190
001046 4160 CONTINUE                             FORT 191
001046 IF (IP110 .LT. 3) GO TO 470              FORT 192
001051 WRITE (6,1100)                             FORT 193
001054 1100 FORMAT (1H1,10X,1RHPTH ARRAY IN FLOPT//) FORT 194
001054 DO 1101 I = 1, NC                         FORT 195
001060 1101 WRITE (6,1102) PTH(I,1), PTH(I,2), PTH(I,3), PTH(I,4) FORT 196
001114 1102 FORMAT (2X,9HPHATE IS ,F10.5,3H X ,F6.2,10X,RHSTART = ,F10.2,10X, FORT 197
      1RHEND = ,F10.2)                             FORT 198
001114 WRITE (6,104)                             FORT 199
001117 104 FORMAT (1H0,10X,15HRETURN TO CPRES//) FORT 200
001117 470 CONTINUE                             FORT 201
001117 RETURN                                     FORT 202
001120 END                                         FORT 203

```

SUBPROGRAM LENGTH
007770

UNUSED COMPILER SPACE
037000

	SUBROUTINE ICALC (A,B,D,C,H,AS,IS,HP,HA,LEF,ITOT)	ICAL	1
	CALCULATES MOMENT OF INERTIA FOR A PARTICULAR SECTION	ICAL	2
000016	REAL ITOT, LEF, IPART, IS	ICAL	3
	CHECKS FOR COVER PLATES	ICAL	4
	CALCULATES I FOR NO COVER PLATES	ICAL	5
000016	IF (A .GT. .0001) .OR. D .GT. .0001) GO TO 10	ICAL	6
000027	ITOT = IS	ICAL	7
000030	LEF = HP	ICAL	8
000032	RETURN	ICAL	9
	SETS INITIAL ZERO VALUES	ICAL	10
000033	10 ALP = 0.0	ICAL	11
000034	AUP = 0.0	ICAL	12
000034	DL = 0.0	ICAL	13
000035	DU = 0.0	ICAL	14
	CALCULATES VALUES FOR THE UPPER PLATE OR SLAB	ICAL	15
000036	IF (A .LT. .0001) GO TO 11	ICAL	16
000041	AUP = A * B	ICAL	17
000042	DU = HA + (H + R)/2.	ICAL	18
	CALCULATE VALUES FOR THE LOWER PLATE	ICAL	19
000046	11 IF (D .LT. .0001) GO TO 12	ICAL	20
000051	ALP = D * C	ICAL	21
000052	DL = (H + C)/2.	ICAL	22
000055	12 ATOT = AS + AUP + ALP	ICAL	23
	CALCULATE DISTANCE FROM CENTROID OF STEEL SECTION TO THE	ICAL	24
	CENTROID OF THE OVERALL SECTION, ALSO CALCULATE I FOR SECTION.	ICAL	25
000060	DIST = (AUP * DU + ALP * DL)/ATOT	ICAL	26
000064	LEF = HP + DIST	ICAL	27
000066	IPART = IS + AUP*B*B/12. + ALP*C*C/12.	ICAL	28
000076	ITOT = IPART + AUP*DU**2 + ALP*DL**2 + ATOT*DIST**2	ICAL	29
000111	RETURN	ICAL	30
000112	END	ICAL	31

SUBPROGRAM LENGTH

000142

UNUSED COMPILER SPACE

042500

	REAL FUNCTION ILIN (PT,RO,COORD,ISW,DP,LENGTH,SUBLEN)	ILIN	1
	INTERPOLATES BETWEEN POINTS ON THE INFLUENCE LINE	ILIN	2
000012	DIMENSION RU(R1), COOR(R1), SUBLEN(R0)	ILIN	3
000012	REAL LENGTH	ILIN	4
000012	INTEGER PT	ILIN	5
000012	IF (ISW .EQ. 2) GO TO 520	ILIN	6
	CALCULATES VALUE OF INFLUENCE LINE TO THE RIGHT OF POINT PT	ILIN	7
000014	SP = COOR(PT) * DP	ILIN	8
	RETURNS A ZERO IF THE POINT IS OFF THE GIRDER	ILIN	9

000016	IF (SP .GT. LENGTH) GO TO 510	ILIN	10
000022	GO TO 512	ILIN	11
000022	510 ILIN = 0.0	ILIN	12
000023	RETURN	ILIN	13
C	DETERMINES THE SURLEMENT LOCATION	ILIN	14
000024	512 J = PT + 1	ILIN	15
000026	513 IF (COORD(J) .GE. SP) GO TO 515	ILIN	16
000032	J = J + 1	ILIN	17
000033	GO TO 513	ILIN	18
000034	515 K = J - 1	ILIN	19
C	INTERPOLATION FORMULA	ILIN	20
000036	ILIN = RO(K) + (RO(J)-RO(K)) * (SP-COOR(K))/SURLEN(K)	ILIN	21
000052	RETURN	ILIN	22
C	CALCULATES VALUE OF INFLUENCE LINE TO THE LEFT OF POINT PT	ILIN	23
000052	520 SP = COOR(PT) - DP	ILIN	24
C	RETURNS A ZERO IF THE POINT IS OFF THE GRID	ILIN	25
000054	IF (SP .LT. 0.0) GO TO 522	ILIN	26
000056	GO TO 525	ILIN	27
000057	522 ILIN = 0.0	ILIN	28
000060	RETURN	ILIN	29
C	DETERMINES THE SURLEMENT LOCATION	ILIN	30
000061	525 J = PT - 1	ILIN	31
000063	527 IF (COORD(J) .LE. SP) GO TO 526	ILIN	32
000067	J = J - 1	ILIN	33
000070	GO TO 527	ILIN	34
000071	526 K = J + 1	ILIN	35
000073	J = K	ILIN	36
000074	K = J - 1	ILIN	37
C	INTERPOLATION FORMULA	ILIN	38
000075	ILIN = RO(K) + (RO(J)-RO(K)) * (SP-COOR(K))/SURLEN(K)	ILIN	39
000111	RETURN	ILIN	40
000112	END	ILIN	41

SURPROGRAM LENGTH

000133

UNUSED COMPILER SPACE

042600

000013	SUBROUTINE ILPROP (RO,SURLEN,NA,KOP,ANEG,APOS,MAXOL,MINOL)	ILPR	1
	DIENSION RO(1), SURLEN(10), KOP(10)	ILPR	2
C	DETERMINES THE PROPERTIES OF A GIVEN INFLUENCE LINE	ILPR	3
000013	MAXOL = 1	ILPR	4
000013	MINOL = 1	ILPR	5
000015	ANEG = 1.0	ILPR	6
000016	APOS = 0.0	ILPR	7
000016	OMAX = -.0001	ILPR	8
000017	OMIN = .0001	ILPR	9
000021	DO 570 I = 1, 10	ILPR	10
000023	570 KOP(I) = 0	ILPR	11
C	DETERMINE THE CROSSOVER POINTS	ILPR	12
000026	KOP(1) = 1	ILPR	13
000027	J = 1	ILPR	14

000030	NE = NA = 1	ILPR	15
000032	DO 550 I = 2, NF	ILPR	16
000034	IF (ABS(R0(I)) .GT. 10.0**(-6)) GO TO 551	ILPR	17
000043	KOP(J) = I	ILPR	18
000045	J = J + 1	ILPR	19
000047	GO TO 550	ILPR	20
000047	551 IF (I .EQ. 1) GO TO 550	ILPR	21
000051	IF (R0(I)*R0(I-1) .GT. 0.0) GO TO 550	ILPR	22
000056	IF (ABS(R0(I-1)) .LT. 10.0**(-6)) GO TO 550	ILPR	23
000063	KOP(J) = I	ILPR	24
000065	J = J + 1	ILPR	25
000067	550 CONTINUE	ILPR	26
000072	KOP(J) = NA	ILPR	27
000074	NKOP = J	ILPR	28
	C DETERMINE MAXIMUM AND MINIMUM ORDINATE LOCATIONS	ILPR	29
000075	DO 552 I = 1, NA	ILPR	30
000076	IF (R0(I) .GT. OMAX) GO TO 555	ILPR	31
000102	IF (R0(I) .LT. OMIN) GO TO 554	ILPR	32
000105	GO TO 552	ILPR	33
000105	555 OMAX = R0(I)	ILPR	34
000107	MAXO(= I	ILPR	35
000111	GO TO 552	ILPR	36
000112	554 OMIN = R0(I)	ILPR	37
000114	MINO(= I	ILPR	38
000116	552 CONTINUE	ILPR	39
	C DETERMINE THE POSITIVE AND NEGATIVE AREAS UNDER THE LINE	ILPR	40
000121	DO 560 K = 1, NKOP	ILPR	41
000122	IF (K .EQ. NKOP) GO TO 561	ILPR	42
000124	IA = KOP(K)	ILPR	43
000125	IB = KOP(K+1) - 1	ILPR	44
000127	A = 0.0	ILPR	45
000131	DO 562 I = IA, IB	ILPR	46
000133	IF (I .NE. IA) GO TO 563	ILPR	47
000135	IF (I .EQ. 1) GO TO 563	ILPR	48
000137	IF (ABS(R0(I)) .LT. 10.0**(-6)) GO TO 563	ILPR	49
000145	A = SUBLEN(I-1)*R0(I)*ABS(R0(I))/(2.0*(ABS(R0(I))+ABS(R0(I-1))))	ILPR	50
000160	GO TO 564	ILPR	51
000161	563 IF (I .NE. IB) GO TO 564	ILPR	52
000163	IF (I+1 .EQ. NA) GO TO 564	ILPR	53
000165	IF (ABS(R0(I+1)) .LT. 10.0**(-6)) GO TO 564	ILPR	54
000173	A = A+SUBLEN(I)*R0(I+1)*ABS(R0(I+1))/(2.0*(ABS(R0(I))+ABS(R0(I+1))))	ILPR	55
000210	GO TO 562	ILPR	56
000211	564 A = A + (R0(I) + R0(I+1)) * SUBLEN(I)/2.0	ILPR	57
000220	562 CONTINUE	ILPR	58
000223	IF (A .GT. 0.0) APOS = APOS + A	ILPR	59
000225	560 IF (A .LT. 0.0) ANEG = ANEG + A	ILPR	60
000233	561 CONTINUE	ILPR	61
000233	RETURN	ILPR	62
000234	END	ILPR	63

SURPRO - DAM LENGTH

000277

UNUSED COMPILER SPACE

042200

```

REAL FUNCTION IMPACT (TYPE,NP,IOSP,NS,SPAN,CFAD,COORD)
C      DETERMINES THE IMPACT FACTOR
DIMENSION IOSP(5), SPAN(4), COOR(8)
INTEGER TYPE
MN = NS - 1
IF (MN .EQ. 1) GO TO 7000
IF (TYPE .NE. 4HDFAC) GO TO 600
C      IMPACT DISTANCE FOR THE REACTION INFLUENCE LINES
IF (NP .NE. 1) GO TO 601
DING = SPAN(1)
GO TO 650
601 IF (NP .NE. NS) GO TO 602
DING = SPAN(NS-1)
GO TO 650
602 DING = SPAN(NP) + SPAN(NP-1)
GO TO 650
600 IF (TYPE .NE. 4HDMOMT) GO TO 610
C      IMPACT DISTANCE FOR THE MOMENT INFLUENCE LINES
DO 611 I = 2, NS
IF (IOSP(I) .LT. NP) GO TO 611
IF (CFAD .LT. 0.0) GO TO 612
DING = SPAN(I-1)
GO TO 650
612 IF (NP .LT. (IOSP(I-1)+IOSP(I))/2) GO TO 625
DING = (SPAN(I-1) + SPAN(I))/2.0
GO TO 650
625 DING = (SPAN(I-1) + SPAN(I-2))/2.0
GO TO 650
611 CONTINUE
610 IF (TYPE .NE. 4HSHER) GO TO 620
C      IMPACT DISTANCE FOR THE SHEAR INFLUENCE LINES
DO 621 I = 2, NS
IF (IOSP(I) .LT. NP) GO TO 621
II = IOSP(I)
JJ = IOSP(I-1)
DING = COOR(II) - COOR(NP)
DONG = COOR(NP) - COOR(JJ)
IF (DONG .GT. DING) DING = DONG
GO TO 650
621 CONTINUE
C      IMPACT DISTANCE FOR THE DEFLECTION INFLUENCE LINES
620 DO 630 I = 2, NS
IF (NP .GT. IOSP(I)) GO TO 630
DING = SPAN(I-1)
GO TO 650
630 CONTINUE
GO TO 650
7000 DING = SPAN(1)
C      IMPACT FACTOR CALCULATION
650 HT = 50./((DING/12.) + 125.)
IF (HT .GT. .3) HT = .3
HT = HT + 1.0
IMPACT = HT
RETURN
END

```

```

IMPT 1
IMPT 2
IMPT 3
IMPT 4
IMPT 5
IMPT 6
IMPT 7
IMPT 8
IMPT 9
IMPT 10
IMPT 11
IMPT 12
IMPT 13
IMPT 14
IMPT 15
IMPT 16
IMPT 17
IMPT 18
IMPT 19
IMPT 20
IMPT 21
IMPT 22
IMPT 23
IMPT 24
IMPT 25
IMPT 26
IMPT 27
IMPT 28
IMPT 29
IMPT 30
IMPT 31
IMPT 32
IMPT 33
IMPT 34
IMPT 35
IMPT 36
IMPT 37
IMPT 38
IMPT 39
IMPT 40
IMPT 41
IMPT 42
IMPT 43
IMPT 44
IMPT 45
IMPT 46
IMPT 47
IMPT 48
IMPT 49
IMPT 50
IMPT 51
IMPT 52
IMPT 53
IMPT 54
IMPT 55

```

SUBPROGRAM LENGTH
000213

UNUSED COMPILER SPACE
042400

	SUBROUTINE KEEP (NE,NOS,PLATH,PLWID,COVPL,CMIN,WMIN,IPTA)	KEEP	1
	C STORES THE ACCEPTABLE GIRDER	KEEP	2
000013	COMMON/FOUR/WTOT,TCOST	KEEP	3
000013	COMMON/A/ISAVE(R0),SAVE(12,6,2),PLTSAV(R0,2),PLWSAV(R0,2)	KEEP	4
000013	DIMENSION NOS(R0),PLATH(R0,2),PLWID(R0,2),COVPL(12,6,2)	KEEP	5
	C CHECK IF THE GIRDER IS BETTER THAN THE PREVIOUS GIRDER	KEEP	6
000013	IF (IPTA.GT..5) GO TO 7003	KEEP	7
000017	IF (TCOST.LT.CMIN) GO TO 7004	KEEP	8
000021	RETURN	KEEP	9
000021	7003 IF (WTOT.LT.WMIN) GO TO 7004	KEEP	10
000024	RETURN	KEEP	11
	C STORE THE ACCEPTABLE GIRDER	KEEP	12
000025	7004 DO /000 I = 1, NF	KEEP	13
000027	DO /999 J = 1, 2	KEEP	14
000030	PLTSAV(I,J) = PLATH(I,J)	KEEP	15
000036	6999 PLWSAV(I,J) = PLWID(I,J)	KEEP	16
000043	7000 ISAVE(I) = NOS(I)	KEEP	17
000047	CMIN = TCOST	KEEP	18
000050	WMIN = WTOT	KEEP	19
000051	DO /001 I = 1, 12	KEEP	20
000053	DO /001 J = 1, 6	KEEP	21
000054	DO /001 K = 1, 2	KEEP	22
000055	7001 SAVE(I,J,K) = COVPL(I,J,K)	KEEP	23
000076	RETURN	KEEP	24
000077	END	KEEP	25

SUBPROGRAM LENGTH
000120

UNUSED COMPILER SPACE
042500

	SUBROUTINE LOAD (IIL,TRLO,UNLO,SURLFN,NA,IPT3,LENGTH,COORD,NP,TYPE,LOAD	LOAD	1
	INS,IOSP,SPAN,STORE)	LOAD	2
	C LOADS A GIVEN INFLUENCE LINE	LOAD	3
000021	DIMENSION IIL(R1,3),TRLO(10),UNLO(9),RO(R1),XOP(10),STORE(10),	LOAD	4
	ISURLFN(R0),IOSP(5),COORD(R1),SPAN(4)	LOAD	5
000021	INTEGER TYPE	LOAD	6
000021	REAL LENGTH, ILINT, IMPACT	LOAD	7
000021	COMMON/ONE/JPT3,IPT10	LOAD	8
000021	MN = NS = 1	LOAD	9
	C ZERO OUT THE STORE ARRAY	LOAD	10
000023	DO ALL I = 1, 10	LOAD	11

```

000024      R11 STORF(1) = 0.0                                LOAD 12
C          SET UP THE STEEL SECTION INFLUENCE LINE          LOAD 13
000030      DO 400 I = 1, NA                                  LOAD 14
000031      R01() = TIL(I,1)                                    LOAD 15
000040      CALL ILPROP(R0,SURLEN,NA,KDP,ANEG,APDS,MAXDI,MINDI)  LOAD 16
000047      IF (TYPE.NE. 4HSHER) GO TO 1850                   LOAD 17
000055      IF (MN.EQ. 1) GO TO 1850                           LOAD 18
000057      DO 1860 II = 2, MN                                  LOAD 19
000060      IF (MP.NE. 10SP(II)) GO TO 1860                    LOAD 20
000063      IF (TIL(INP,1).GT. 0.0) APDS = APDS + SURLEN(MP)/2.0  LOAD 21
000071      IF (TIL(INP,1).LT. 0.0) ANEG = ANEG + SURLEN(MP-1)/2.0  LOAD 22
000077      1860 CONTINUE                                       LOAD 23
000102      1850 ATOT = APDS + ANEG                             LOAD 24
000104      IF (UPT3.LT. 9) GO TO R01                           LOAD 25
C          READ LOAD REACTION FOR A NON - COMPOSITE SECTION  LOAD 26
000107      UDL = TRLO(7) + TRLO(8) + TRLO(10)                 LOAD 27
000113      IF (UDL.LT. UNLO(4)+UNLO(5)+UNLO(9)) UNL = UNLO(4)+UNLO(5)+UNLO(9)  LOAD 28
000125      STORF(1) = UDL + ATOT                               LOAD 29
000130      STORF(10) = 0.0                                       LOAD 30
000131      GO TO R01                                             LOAD 31
C          READ LOAD AND LOW MODULUS REACTION FOR COMPOSITE SECTION  LOAD 32
000131      R01 UDL = TRLO(8) + TRLO(10)                       LOAD 33
000134      IF (UDL.LT. UNLO(5)+UNLO(9)) UNL = UNLO(5) + UNLO(9)  LOAD 34
000142      STORF(1) = UNL + ATOT                               LOAD 35
000145      DO 402 I = 1, NA                                     LOAD 36
000146      R02() = TIL(I,2)                                    LOAD 37
000155      CALL ILPROP(R0,SURLEN,NA,KDP,ANEG,APDS,MAXDI,MINDI)  LOAD 38
000164      IF (TYPE.NE. 4HSHER) GO TO 1852                     LOAD 39
000172      IF (MN.EQ. 1) GO TO 1852                           LOAD 40
000174      DO 1861 II = 2, MN                                  LOAD 41
000175      IF (MP.NE. 10SP(II)) GO TO 1861                     LOAD 42
000200      IF (TIL(INP,2).GT. 0.0) APDS = APDS + SURLEN(MP)/2.0  LOAD 43
000206      IF (TIL(INP,2).LT. 0.0) ANEG = ANEG + SURLEN(MP-1)/2.0  LOAD 44
000214      1861 CONTINUE                                       LOAD 45
000217      1852 ATOT = APDS + ANEG                             LOAD 46
000221      ACA = UNLO(4)                                         LOAD 47
000223      IF (ACA.LT. TRLO(7)) ACA = TRLO(7)                   LOAD 48
000227      STORF(10) = ATOT + ACA                               LOAD 49
C          SET UP THE HIGH MODULUS INFLUENCE LINE IF REQUIRED  LOAD 50
000232      R03 IF (UPT3.GT. 9) GO TO R04                       LOAD 51
000236      DO 405 I = 1, NA                                     LOAD 52
000237      R05() = TIL(I,3)                                    LOAD 53
000244      CALL ILPROP(R0,SURLEN,NA,KDP,ANEG,APDS,MAXDI,MINDI)  LOAD 54
000255      IF (TYPE.NE. 4HSHER) GO TO R04                       LOAD 55
000263      IF (MN.EQ. 1) GO TO R04                             LOAD 56
000265      DO 1862 II = 2, MN                                  LOAD 57
000266      IF (MP.NE. 10SP(II)) GO TO 1862                     LOAD 58
000271      IF (TIL(INP,3).GT. 0.0) APDS = APDS + SURLEN(MP)/2.0  LOAD 59
000277      IF (TIL(INP,3).LT. 0.0) ANEG = ANEG + SURLEN(MP-1)/2.0  LOAD 60
000305      1862 CONTINUE                                       LOAD 61
C          BEGIN THE LIVE LOAD LOADINGS                      LOAD 62
000310      R04 IF (IPT2.EQ. 2) GO TO R07                       LOAD 63
000312      ACA = UNLO(3)                                         LOAD 64
000313      IF (TYPE.EQ. 4HMMNT) ACR = UNLO(2)                   LOAD 65
C          CALCULATE THE IMPACT FACTOR                       LOAD 66
000317      IF (UNLO(10).LT. .1) GO TO 1800                     LOAD 67
000322      HT = 1.0                                              LOAD 68
000324      GO TO 1800                                           LOAD 69

```



```

000324      LOAD HT = IMPACT(TYPE,NO,IOSP,NS,SPAN,STORE(1),COORD)      LOAD 70
C          CALCULATE THE POSITIVE AND NEGATIVE LANE LOADINGS      LOAD 71
000340      LOAD STORE(2) = HT * (UNLO(1) * APOS * ACR * RO(MAXOL))      LOAD 72
000353      SPEK = 0.      LOAD 73
000354      IF (TYPE .NE. 4HMMNT) GO TO R12      LOAD 74
000355      IF (MN .EQ. 1) GO TO R12      LOAD 75
000360      DO 406 J = 1, MN      LOAD 76
000361      IF (MINOL .LT. IOSP(J)) GO TO R06      LOAD 77
000364      IF (MINOL .GT. IOSP(J+1)) GO TO R06      LOAD 78
000367      IF (J .NE. 1) GO TO R07      LOAD 79
000370      L = IOSP(2)      LOAD 80
000372      M = IOSP(3)      LOAD 81
000373      DO 408 K = L, M      LOAD 82
000375      R08 IF (RO(K) .LT. SPEK) SPEK = RO(K)      LOAD 83
000405      GO TO R12      LOAD 84
000405      R07 IF (J .NE. MN) GO TO R09      LOAD 85
000407      L = IOSP(MN - 1)      LOAD 86
000412      M = IOSP(MN)      LOAD 87
000414      DO 413 K = L, M      LOAD 88
000416      R13 IF (RO(K) .LT. SPEK) SPEK = RO(K)      LOAD 89
000426      GO TO R12      LOAD 90
000426      R09 L = IOSP(J-1)      LOAD 91
000431      M = IOSP(J)      LOAD 92
000433      DO 414 K = L, M      LOAD 93
000435      R14 IF (RO(K) .LT. SPEK) SPEK = RO(K)      LOAD 94
000445      L = IOSP(J+1)      LOAD 95
000447      M = IOSP(J+2)      LOAD 96
000451      DO 415 K = L, M      LOAD 97
000453      R15 IF (RO(K) .LT. SPEK) SPEK = RO(K)      LOAD 98
000463      GO TO R12      LOAD 99
000463      R06 CONTINUE      LOAD 100
000466      R12 SPEK = RO(MINOL) + SPEK      LOAD 101
000471      STORE(3) = HT * (UNLO(1) * ANEG * ACR * SPEK)      LOAD 102
C          CALCULATE THE SIDEWALK LOADINGS      LOAD 103
000477      STORE(8) = APOS * UNLO(7)      LOAD 104
000502      STORE(9) = ANEG * UNLO(7)      LOAD 105
C          INTERSTATE LOADINGS ( POS AND NEG BASED ON A 4 FOOT SPACING)      LOAD 106
000504      RR = ILINT(MAXOL,RO,COORD,1.4R,.LENGTH,SURFEM)      LOAD 107
000520      PR = RR + RO(MAXOL)      LOAD 108
000523      RR = ILINT(MAXOL,RO,COORD,2.4R,.LENGTH,SURFEM)      LOAD 109
000536      SR = RR + RO(MAXOL)      LOAD 110
000541      IF (PR .LT. SR) PR = SR      LOAD 111
000544      RR = ILINT(MAXOL,RO,COORD,1.24,.LENGTH,SURFEM)      LOAD 112
000560      TR = ILINT(MAXOL,RO,COORD,2.24,.LENGTH,SURFEM)      LOAD 113
000574      SR = RR + TR      LOAD 114
000576      IF (PR .LT. SR) PR = SR      LOAD 115
000601      STORE(4) = HT * (UNLO(6) * PR)      LOAD 116
000605      RR = ILINT(MINOL,RO,COORD,1.4R,.LENGTH,SURFEM)      LOAD 117
000620      PR = RR + RO(MINOL)      LOAD 118
000623      RR = ILINT(MINOL,RO,COORD,2.4R,.LENGTH,SURFEM)      LOAD 119
000636      SR = RR + RO(MINOL)      LOAD 120
000641      IF (PR .GT. SR) PR = SR      LOAD 121
000644      RR = ILINT(MINOL,RO,COORD,1.24,.LENGTH,SURFEM)      LOAD 122
000660      TR = ILINT(MINOL,RO,COORD,2.24,.LENGTH,SURFEM)      LOAD 123
000674      SR = RR + TR      LOAD 124
000676      IF (PR .GT. SR) PR = SR      LOAD 125
000701      STORE(5) = HT * (UNLO(6) * PR)      LOAD 126
000705      R30 IF (IPT2 .EQ. 1) GO TO R70      LOAD 127

```

000707	C	START THE TRUCK LOADING	LOAD 128
000710		POS = 0.0	LOAD 129
000711		SEF = 0.0	LOAD 130
000712		PFRA = .4	LOAD 131
000713		OCA = TRLO(5) + TRLO(4)	LOAD 132
000715		OCB = TRLO(6) + TRLO(4)	LOAD 133
000717	C	MOVE THE TRUCK FROM LEFT TO RIGHT	LOAD 134
		J = 1	LOAD 135
	C	REAR AXLE ON THE POINT	LOAD 136
	C	CALCULATE FOR THE MINIMUM SPACING	LOAD 137
000720	R34	RAL = TRLO(3) * RO(J)	LOAD 138
000723		RR = ILINT(J,RO,COORD,1,TRLO(5),LENGTH,SURLEN)	LOAD 139
000741		SR = ILINT(J,RO,COORD,1,OCA,LENGTH,SURLEN)	LOAD 140
000755		TTL = RAL + TRLO(2) * RR + TRLO(1) * SR	LOAD 141
000763		IF (TTL .GT. POS) POS = TTL	LOAD 142
000764		IF (TTL .LT. SEF) SEF = TTL	LOAD 143
	C	CALCULATE FOR THE MAXIMUM SPACING	LOAD 144
000771		RR = ILINT(J,RO,COORD,1,TRLO(6),LENGTH,SURLEN)	LOAD 145
001007		SR = ILINT(J,RO,COORD,1,OCB,LENGTH,SURLEN)	LOAD 146
001023		TTL = RAL + TRLO(2) * RR + TRLO(1) * SR	LOAD 147
001031		IF (TTL .GT. POS) POS = TTL	LOAD 148
001034		IF (TTL .LT. SEF) SEF = TTL	LOAD 149
	C	FRONT AXLE ON THE POINT	LOAD 150
001037		FAL = RO(J) * TRLO(1)	LOAD 151
001042		RR = ILINT(J,RO,COORD,2,TRLO(4),LENGTH,SURLEN)	LOAD 152
001057		PAL = FAL + RR * TRLO(2)	LOAD 153
001062		IF (PAL .GT. POS) POS = PAL	LOAD 154
001065		IF (PAL .LT. SEF) SEF = PAL	LOAD 155
001070		IF (COORD(J) - OCA .LT. 0.0) GO TO R35	LOAD 156
001074		IF (PAL .LT. PFRA * POS) GO TO R31	LOAD 157
001100		GO TO R32	LOAD 158
001100	R31	IF (PAL .GT. PFRA * POS) GO TO R35	LOAD 159
	C	START WIGGLING THE BACK AXLE	LOAD 160
001105	R32	RR = ILINT(J,RO,COORD,2,OCA,LENGTH,SURLEN)	LOAD 161
001121		TTL = PAL + RR * TRLO(3)	LOAD 162
001124		IF (TTL .GT. POS) POS = TTL	LOAD 163
001127		IF (TTL .LT. SEF) SEF = TTL	LOAD 164
001132		SR = ILINT(J,RO,COORD,2,OCB,LENGTH,SURLEN)	LOAD 165
001144		TTL = PAL + SR * TRLO(3)	LOAD 166
001151		IF (TTL .GT. POS) POS = TTL	LOAD 167
001154		IF (TTL .LT. SEF) SEF = TTL	LOAD 168
001157		K = J + 1	LOAD 169
001161	R33	IF (K .EQ. 1) GO TO R35	LOAD 170
001163		K = K - 1	LOAD 171
001164		IF (COORD(K) .GT. COORD(J) - OCA) GO TO R33	LOAD 172
001173		IF (COORD(K) .LT. COORD(J) - OCB) GO TO R35	LOAD 173
001201		TTL = PAL + RO(K) * TRLO(3)	LOAD 174
001205		IF (TTL .GT. POS) POS = TTL	LOAD 175
001210		IF (TTL .LT. SEF) SEF = TTL	LOAD 176
001213		GO TO R33	LOAD 177
	C	PLACE THE SECOND AXLE ON THE POINT	LOAD 178
001214	R35	CAL = TRLO(2) * RO(J)	LOAD 179
001217		RR = ILINT(J,RO,COORD,1,TRLO(4),LENGTH,SURLEN)	LOAD 180
001235		PAL = CAL + RR * TRLO(1)	LOAD 181
001240		IF (PAL .GT. POS) POS = PAL	LOAD 182
001243		IF (PAL .LT. SEF) SEF = PAL	LOAD 183
001246		IF (COORD(J) - TRLO(5) .LT. 0.0) GO TO R40	LOAD 184
001253		IF (PAL .LT. PFRA * POS) GO TO R34	LOAD 185


```

001255      GO TO B37                                LOAD 186
001257      R36 IF (PAL .GT. PERA * SFG) GO TO R40    LOAD 187
C          START WIGGLING THE BACK AXLE           LOAD 188
001264      R37 RR = ILINT(J,R0,COORD,2,TLO(5),LENGTH,SURFEX)  LOAD 189
001302      TTL = PAL + RR * TLO(3)                LOAD 190
001305      IF (TTL .GT. POS) POS = TTL             LOAD 191
001310      IF (TTL .LT. SFG) SFG = TTL            LOAD 192
001313      SR = ILINT(J,R0,COORD,2,TLO(6),LENGTH,SURFEX)    LOAD 193
001331      TTL = PAL + SR * TLO(3)                LOAD 194
001334      IF (TTL .GT. POS) POS = TTL            LOAD 195
001337      IF (TTL .LT. SFG) SFG = TTL           LOAD 196
001342      K = J + 1                              LOAD 197
001344      R38 IF (K .EQ. 1) GO TO R40             LOAD 198
001346      K = K - 1                               LOAD 199
001347      IF (COORD(K) .GT. COOR(J) - OCA) GO TO R39  LOAD 200
001356      IF (COORD(K) .LT. COOR(J) - OCB) GO TO R40  LOAD 201
001364      TTL = PAL + KO(K) * TLO(3)             LOAD 202
001370      IF (TTL .GT. POS) POS = TTL            LOAD 203
001373      IF (TTL .LT. SFG) SFG = TTL           LOAD 204
001376      GO TO R36                             LOAD 205
001377      R40 J = J + 1                          LOAD 206
001401      IF (J .LE. NA) GO TO R34              LOAD 207
C          MOVE THE TRUCK FROM THE RIGHT TO THE LEFT  LOAD 208
001403      J = NA                                LOAD 209
C          REAR AXLE ON THE POINT                  LOAD 210
C          CALCULATE FOR THE MINIMUM SPACING        LOAD 211
001403      R41 RAL = TLO(3) * PH(J)              LOAD 212
001406      RR = ILINT(J,R0,COORD,2,TLO(5),LENGTH,SURFEX)  LOAD 213
001424      SR = ILINT(J,R0,COORD,2,OCA,LENGTH,SURFEX)    LOAD 214
001440      TTL = RAL + TLO(2) * RR + TLO(1) * SR    LOAD 215
001446      IF (TTL .GT. POS) POS = TTL            LOAD 216
001451      IF (TTL .LT. SFG) SFG = TTL           LOAD 217
C          CALCULATE FOR THE MAXIMUM SPACING        LOAD 218
001454      RR = ILINT(J,R0,COORD,2,TLO(6),LENGTH,SURFEX)  LOAD 219
001472      SR = ILINT(J,R0,COORD,2,OCB,LENGTH,SURFEX)    LOAD 220
001506      TTL = RAL + TLO(2) * RR + TLO(1) * SR    LOAD 221
001514      IF (TTL .GT. POS) POS = TTL            LOAD 222
001517      IF (TTL .LT. SFG) SFG = TTL           LOAD 223
C          FRONT AXLE ON THE POINT                 LOAD 224
001522      FAL = R0(J) * TLO(1)                LOAD 225
001525      RR = ILINT(J,R0,COORD,1,TLO(4),LENGTH,SURFEX)  LOAD 226
001542      PAL = FAL + RR * TLO(2)              LOAD 227
001545      IF (PAL .GT. POS) POS = PAL           LOAD 228
001550      IF (PAL .LT. SFG) SFG = PAL          LOAD 229
001553      IF (COORD(J) + OCA .GT. LENGTH) GO TO R45  LOAD 230
001562      IF (PAL .LT. PERA * POS) GO TO R42      LOAD 231
001565      GO TO R43                             LOAD 232
001566      R42 IF (PAL .GT. PERA * SFG) GO TO R45    LOAD 233
C          START WIGGLING THE BACK AXLE           LOAD 234
001573      R43 RR = ILINT(J,R0,COORD,1,OCA,LENGTH,SURFEX)  LOAD 235
001607      TTL = PAL + RR * TLO(2)              LOAD 236
001612      IF (TTL .GT. POS) POS = TTL           LOAD 237
001615      IF (TTL .LT. SFG) SFG = TTL          LOAD 238
001620      SR = ILINT(J,R0,COORD,1,OCB,LENGTH,SURFEX)    LOAD 239
001634      TTL = PAL + SR * TLO(2)              LOAD 240
001637      IF (TTL .GT. POS) POS = TTL           LOAD 241
001642      IF (TTL .LT. SFG) SFG = TTL           LOAD 242
001645      DO 444 K = J, NA                      LOAD 243

```

```

001647      IF (COORD(K) .LT. COORD(J) + DCA) GO TO R44      LOAD 244
001655      IF (COORD(K) .GT. COORD(J) + DCH) GO TO R45      LOAD 245
001665      TTL = PAL + R0(K) * TRLO(3)      LOAD 246
001670      IF (TTL .GT. POS) POS = TTL      LOAD 247
001673      IF (TTL .LT. SEG) SEG = TTL      LOAD 248
001675      R44 CONTINUE      LOAD 249
C      PLACE THE SECOND AXLE ON THE POINT      LOAD 250
001701      R45 CAL = TRLO(2) * R0(J)      LOAD 251
001704      RR = ILINT(J,R0,COORD,2,TRLO(4),LENGTH,SURLEN)      LOAD 252
001722      PAL = CAL + RR * TRLO(1)      LOAD 253
001729      IF (PAL .GT. POS) POS = PAL      LOAD 254
001730      IF (PAL .LT. SEG) SEG = PAL      LOAD 255
001733      IF (COORD(J) + TRLO(5) .GT. LENGTH) GO TO R50      LOAD 256
001742      IF (PAL .LT. PERA * POS) GO TO R46      LOAD 257
001745      GO TO R48      LOAD 258
001746      R46 IF (PAL .GT. PERA * SEG) GO TO R50      LOAD 259
C      START WIGGLING THE BACK AXLE      LOAD 260
001753      R48 RR = ILINT(J,R0,COORD,1,TRLO(5),LENGTH,SURLEN)      LOAD 261
001771      TTL = PAL + RR * TRLO(3)      LOAD 262
001774      IF (TTL .GT. POS) POS = TTL      LOAD 263
001777      IF (TTL .LT. SEG) SEG = TTL      LOAD 264
002002      SR = ILINT(J,R0,COORD,1,TRLO(4),LENGTH,SURLEN)      LOAD 265
002020      TTL = PAL + SR * TRLO(3)      LOAD 266
002023      IF (TTL .GT. POS) POS = TTL      LOAD 267
002024      IF (TTL .LT. SEG) SEG = TTL      LOAD 268
002031      DO 147 I = J, NA      LOAD 269
002033      IF (COORD(K) .LT. COORD(J) + DCA) GO TO R47      LOAD 270
002041      IF (COORD(K) .GT. COORD(J) + DCH) GO TO R50      LOAD 271
002051      TTL = PAL + R0(K) * TRLO(3)      LOAD 272
002054      IF (TTL .GT. POS) POS = TTL      LOAD 273
002057      IF (TTL .LT. SEG) SEG = TTL      LOAD 274
002062      R47 CONTINUE      LOAD 275
002065      R50 J = J - 1      LOAD 276
002067      IF (J .GE. 1) GO TO R41      LOAD 277
C      CALCULATE THE IMPACT FACTOR      LOAD 278
002070      IF (TRLO(4) .LT. .1) GO TO 1802      LOAD 279
002073      HT = 1.0      LOAD 280
C      STORE THE MAXIMUM POSITIVE AND NEGATIVE TRUCK LOADS      LOAD 281
002075      1812 STOR(6) = HT * POS      LOAD 282
002100      STOR(7) = HT * SEG      LOAD 283
002102      R70 CONTINUE      LOAD 284
002102      RETURN      LOAD 285
002103      END      LOAD 286

```

SUBPROGRAM LENGTH
002367

UNUSED COMPILER SPACE
034500

```

SUBROUTINE OBJECT      OBJECT 1
C      DETERMINES THE WEIGHT AND/OR COST OF A DESIGN      OBJECT 14
000002      COMMON NAME(14),SALEPA(14),DEPTH(14),FWIDTH(14),FUTHT(14),WORTH(14),OBJT 2

```

```

10AF(1R),1X(1R),1Y(1R),1NSCT(1R),1NOS(2R),1STEE(1R),1CS(4R),      OHJT 3
21LC(1R),1CL(1R),1CON(4R),1CH(1R),1PLATH(1R),1PWID(1R),1NF,      OHJT 4
3SL(1R),1LATH(1R),1C(1R),1RT(1R),1CO(1R),1F(1R),1SPAN(1R),1TOSP(1R), OHJT 5
4T(1R),1NA(1R),1L(1R),1S(1R),1NS(1R),1T(1R),1PL(1R),1PROP(1R), OHJT 6
5FC(1R),1FESS(1R),1FESSM(1R),1FESSA(1R),1FESS(1R),1FESS(1R),1FESS(1R), OHJT 7
6COST(1R),1CONSP(1R),1COVPL(1R),1COVPL(1R),1COVPL(1R),1COVPL(1R), OHJT 8
COMMON/ONE/1PT3,1PT10      OHJT 9
COMMON/THREE/AS,ACS,1PTA      OHJT 10
COMMON/FOUR/WTOT,1COST      OHJT 11
COMMON/EIGHT/1ST      OHJT 12
REAL LENGTH      OHJT 13
C      DETERMINE WEIGHT AND/OR COST OF ROLLED SECTIONS      OHJT 14
      MN = NS = 1      OHJT 15
      WRS = LENGTH * SARFA(1ST) * 490./1728.      OHJT 16
      IF (1PTA.EQ.1) GO TO 6000      OHJT 17
      CRS = WRS * COST(1)      OHJT 18
      6000 CONTINUE      OHJT 19
C      DETERMINE WEIGHT AND/OR COST OF COVER PLATES      OHJT 20
      6005 CPW = 0.0      OHJT 21
      CPC = 0.0      OHJT 22
      DO 6006 K = 1, 2      OHJT 23
      DO 6007 I = 1, 12      OHJT 24
      IF (ABS(COVPL(I,1,K)).LT..0001.AND.ABS(COVPL(I,2,K)).LT..0001)      OHJT 25
      GO TO 6006      OHJT 26
      CPL = COVPL(I,2,K) - COVPL(I,1,K)      OHJT 27
      CPA = COVPL(I,3,K) * COVPL(I,4,K)      OHJT 28
      PARW = CPA * CPL * 490./1728.      OHJT 29
      CPW = CPW + PARW      OHJT 30
      IF (1PTA.EQ.1) GO TO 6007      OHJT 31
      CPC = CPC + PARW * COST(2)      OHJT 32
      6007 CONTINUE      OHJT 33
      6006 CONTINUE      OHJT 34
C      DETERMINE WEIGHT AND/OR COST OF BEARING STIFFENERS      OHJT 35
      WRS = 0.0      OHJT 36
      CRS = 0.0      OHJT 37
      CHSW = 0.0      OHJT 38
      DO 6010 I = 1, NS      OHJT 39
      IF (ABS(HEAR(I,1)).LT..0001) GO TO 6010      OHJT 40
      IA = IOSP(I)      OHJT 41
      IF (I.EQ.NS) IA = IOSP(I) - 1      OHJT 42
      IA = NOS(IA)      OHJT 43
      BH = DEPTH(IA) * 2. * FLTH(IA)      OHJT 44
      BV = 2. * BH * REAR(I,1) * REAR(I,2)      OHJT 45
      PARW = BV * 490./1728.      OHJT 46
      WRS = WRS + PARW      OHJT 47
      IF (1PTA.EQ.1) GO TO 6010      OHJT 48
      CRS = CRS + COST(3) * PARW      OHJT 49
      WV = 2. * BH * REAR(I,3) * REAR(I,3)      OHJT 50
      CHSW = CHSW + 4. * COST(4) * WV * COST(7) + 4. * BH * COST(4)      OHJT 51
      6010 CONTINUE      OHJT 52
C      DETERMINE WEIGHT AND/OR COST OF THE SHEAR CONNECTORS      OHJT 53
      WCON = 0.0      OHJT 54
      CCOW = 0.0      OHJT 55
      IF (1PT3.GT.9) GO TO 6015      OHJT 56
      NR02 = NN      OHJT 57
      DO 6012 K = 1, NN      OHJT 58
      DO 6013 I = 1, 16      OHJT 59
      IF (ABS(CONSP(I,1,K)).LT..0001) GO TO 6013      OHJT 60

```

```

000152      NPA = (CONSP(1,2,K) - CONSP(1,2,K)) / CONSP(1,1,K)      ORJT 61
000161      IF (ABS(NPA*CONSP(1,1,K)-CONSP(1,2,K)+CONSP(1,2,K)) .GT. .0001) ORJT 62
      1 NPA = NPA + 1      ORJT 63
000171      NROW = NROW + NPA      ORJT 64
000173      6013 CONTINUE      ORJT 65
000175      6012 CONTINUE      ORJT 66
000200      NCN = NROW * NCS      ORJT 67
000202      WCON = NCON * .77541443      ORJT 68
000205      IF (IP1A .EQ. 1) GO TO 6015      ORJT 69
000207      CCN = WCON * COST(4) + NCON * COST(5)      ORJT 70
000213      6015 WTI = WRS + CPW + WRS + WCON      ORJT 71
000217      IF (IP1A .NE. 1) GO TO 6050      ORJT 72
000221      RETURN      ORJT 73
C      DETERMINE THE COST OF THE WELDS      ORJT 74
000222      6050 CWELD = CHSW      ORJT 75
C      DETERMINE THE COST OF THE COVER PLATE WELDS      ORJT 76
000224      CPWC = 0.0      ORJT 77
000225      DO 6020 K = 1, 2      ORJT 78
000226      IA = 0      ORJT 79
000227      DO 6021 I = 1, 12      ORJT 80
000231      IF (ABS(COVPL(I,1,K)) .LT. .0001 .AND. ABS(COVPL(I,2,K)) .LT.      ORJT 81
      1.0001) GO TO 6022      ORJT 82
      IA = I      ORJT 83
000251      6021 CONTINUE      ORJT 84
000252      6022 IF (IA .EQ. 0) GO TO 6020      ORJT 85
000254      DO 6023 I = 1, IA      ORJT 86
000255      CW = 0.0      ORJT 87
000260      IF (I .EQ. 1) GO TO 6025      ORJT 88
000262      IF (ABS(COVPL(I,1,K)-COVPL(I-1,2,K)) .LT. .0001) GO TO 6024      ORJT 89
000272      6025 WLEN = COVPL(I,4,K) * 4.      ORJT 90
000277      WV = WLEN * COVPL(I,5,K) * COVPL(I,5,K)/2.      ORJT 91
000303      CW = CW + WLEN*COST(8) + WV*COST(7) + COST(4)      ORJT 92
000312      GO TO 6020      ORJT 93
000312      6024 TH = COVPL(I,3,K)      ORJT 94
000316      IF (COVPL(I-1,3,K) .LT. COVPL(I,3,K)) TH = COVPL(I-1,3,K)      ORJT 95
000326      WID = COVPL(I,4,K)      ORJT 96
000332      IF (COVPL(I-1,4,K) .LT. COVPL(I,4,K)) WID = COVPL(I-1,4,K)      ORJT 97
000341      WV = (.125*TH + .284675*TH**2) * WID      ORJT 98
000346      BWC = WV*COST(7) + WID*COST(8) + COST(6)      ORJT 99
000352      CW = CW + BWC      ORJT 100
000354      6026 IF (I .EQ. IA) GO TO 6027      ORJT 101
000356      IF (ABS(COVPL(I+1,1,K) - COVPL(I,2,K)) .LT. .0001) GO TO 6029      ORJT 102
000366      6027 WLEN = COVPL(I,4,K) * 4.      ORJT 103
000373      WV = WLEN * COVPL(I,5,K) * COVPL(I,5,K)/2.      ORJT 104
000377      CW = CW + WLEN*COST(8) + WV*COST(7) + COST(4)      ORJT 105
000406      6029 SWL = COVPL(I,2,K) - COVPL(I,1,K)      ORJT 106
000415      IF (I .EQ. 1 .OR. I .EQ. IA) GO TO 6030      ORJT 107
000424      IF (ABS(COVPL(I+1,1,K) - COVPL(I,2,K)) .GT. .0001)      ORJT 108
      ISWL = SWL - 1.5 * COVPL(I,4,K)      ORJT 109
      IF (ABS(COVPL(I,1,K) - COVPL(I-1,2,K)) .GT. .0001)      ORJT 110
      ISWL = SWL - 1.5 * COVPL(I,4,K)      ORJT 111
      GO TO 6031      ORJT 112
000451      6030 SWL = SWL - 1.5 * COVPL(I,4,K)      ORJT 113
000452      IF (I .EQ. 1 .AND. ABS(COVPL(2,1,K)-COVPL(I,2,K)) .GT. .0001)      ORJT 114
      ISWL = SWL - 1.5 * COVPL(I,4,K)      ORJT 115
000501      IF (I .EQ. IA .AND. ABS(COVPL(I,1,K)-COVPL(I-1,2,K)) .GT. .0001)      ORJT 116
      ISWL = SWL - 1.5 * COVPL(I,4,K)      ORJT 117
000524      6031 WV = SWL * COVPL(I,5,K) * COVPL(I,5,K)/2.      ORJT 118

```

000532	CSLW = WV * COST(7) + SXL * COST(8) + COST(4)	04J1 119
000536	CW = CW + 2. * CSLW	04J1 120
000541	CPWC = CPWC + CW	04J1 121
000542	6023 CONTINUE	04J1 122
000545	6020 CONTINUE	04J1 123
000547	CWFID = CWFID + CPWC	04J1 124
000551	TOST = CRS + CPC + CRS + CCOR + CWFID	04J1 125
000556	RETURN	04J1 126
000557	END	04J1 127

SUBPROGRAM LENGTH

000677

UNUSED COMPILER SPACE

040100

	SUBROUTINE PRCON(NS,CONSP,NC5)	PRCO 1
	PRINT THE CONNECTOR INFORMATION	PRCO 2
000006	COMMON/ONE/JPI3,IP110	PRCO 3
000006	DIMENSION CONSP(16,3,4)	PRCO 4
000006	MC = NS - 1	PRCO 5
	ONLY PRINT TABLE IN DESIGN IS COMPOSITE	PRCO 6
000010	IF (JPI3 .GT. 4) GO TO 5000	PRCO 7
	PRINT THE HEADING	PRCO 8
000013	WRITE (6,4191)	PRCO 9
000016	4190 FORMAT (1H0,10X,31HSHEAR CONNECTOR INFORMATION,...//20X, 137PLACE THE SHEAR CONNECTORS AS FOLLOWS)	PRCO 10
	PRINT THE CONNECTOR INFORMATION FOR EACH SPAN	PRCO 12
000016	DO 4191 I = 1, MC	PRCO 13
000022	WRITE (6,4194) I	PRCO 14
000027	4194 FORMAT (20X,16HFOR SPAN NUMBER ,I),3H...)	PRCO 15
000027	DO 4192 J = 1, 16	PRCO 16
000033	IF (ABS(CONSP(J,1,1)) .LT. .0001) GO TO 4191	PRCO 17
000041	X0 = CONSP(J,2,1) - CONSP(J,2,1)	PRCO 18
000047	NX0 = X0 / CONSP(J,1,1)	PRCO 19
000052	IF (ABS(NX0*CONSP(J,1,1) - X0) .GT. .0001) NX0 = NX0 + 1	PRCO 20
000061	4192 WRITE (6,4193) CONSP(J,2,1), NX0, CONSP(J,1,1), X0	PRCO 21
000111	4193 FORMAT (25X,12HSTARTING AT ,F9.2,H, USE ,I2.11H SPACES AT ,F4.1, 1164 INCHES/SPACE = ,F10.2,H INCHES.)	PRCO 22
000111	4191 CONTINUE	PRCO 24
	PRINT THE NUMBER OF CONNECTIONS PER SECTION	PRCO 25
000116	WRITE (6,4195) NC5	PRCO 26
000121	4195 FORMAT (10X,20X,4HUSE ,I1,24H CONNECTIONS IN EACH ROW.)	PRCO 27
000121	GO TO 5001	PRCO 28
	PRINT THE NON-COMPOSITE DESIGN STATEMENT	PRCO 29
000124	5010 WRITE (6,5002)	PRCO 30
000130	5002 FORMAT (1H0///10X,75H THERE ARE NO SHEAR CONNECTORS REQUIRED BECAUSE	PRCO 31
	IF THE DESIGN IS NON-COMPOSITE.//)	PRCO 32
000130	5001 RETURN	PRCO 33
000131	END	PRCO 34

SUBPROGRAM LENGTH

000235

UNUSED COMPILER SPACE

042300

	SUBROUTINE PRCDV (COVPL)	PRCV 1
	C PRINT THE COVER PLATE INFORMATION	PRCV 2
000003	COMMON/ONE/JPT3, IPT10	PRCV 3
000003	DIMENSION COVPL (12,5,2)	PRCV 4
	C PRINT THE HEADINGS	PRCV 5
000003	WRITE (6,2000)	PRCV 6
000007	2000 FORMAT (1H0,25HCOVER PLATE INFORMATION...//)	PRCV 7
	C BEGIN THE LOOP FOR TOP AND BOTTOM PLATES	PRCV 8
	DO 2001 K = 1, 2	PRCV 9
000012	IF (K .EQ. 1) WRITE (6,2002)	PRCV 10
000020	IF (K .EQ. 2) WRITE (6,2003)	PRCV 11
000027	2002 FORMAT (5X,10HON THE TOP OF THE GIRDER USE...)	PRCV 12
000027	2003 FORMAT (5X,10HON THE BOTTOM OF THE GIRDER USE...)	PRCV 13
000027	IF (ABS(COVPL (1,1,K)).LT.,.0001.AND.ABS(COVPL (1,2,K)).LT.,.0001)	PRCV 14
	GO TO 2005	PRCV 15
	C BEGIN THE LOOP FOR EACH PLATE	PRCV 16
000045	DO 2004 I = 1, 12	PRCV 17
000046	IF (ABS(COVPL (I,1,K)).LT.,.0001.AND.ABS(COVPL (I,2,K)).LT.,.0001)	PRCV 18
	GO TO 2001	PRCV 19
	C PRINT THE INFORMATION FOR EACH PLATE	PRCV 20
000067	WRITE (6,2006) COVPL (1,4,K), COVPL (1,3,K), COVPL (1,1,K),	PRCV 21
	COVPL (1,2,K), COVPL (1,5,K), COVPL (1,6,K)	PRCV 22
000122	2006 FORMAT (10X,4HUSE ,F6,2,3H X ,F6,3,17H INCH PLATE FROM ,	PRCV 23
	1F6,1,4H TO ,F6,1,6H WITH ,F6,3,17H END WELDS AND A ,	PRCV 24
	2F6,4,11H SEAL WELD.)	PRCV 25
000122	2004 CONTINUE	PRCV 26
000125	GO TO 2001	PRCV 27
	C PRINT THE NO PLATE STATEMENT	PRCV 28
000126	2005 WRITE (6,2010)	PRCV 29
000132	2010 FORMAT (1H0,10X,25HNO COVER PLATES REQUIRED...//)	PRCV 30
000132	2001 CONTINUE	PRCV 31
	C PRINT THE NON-COMPOSITE STATEMENT	PRCV 32
000135	IF (JPT3 .GT. 9) WRITE (6,2200)	PRCV 33
000144	2200 FORMAT (1H0,10X,25HTHE DESIGN IS NON-COMPOSITE...)	PRCV 34
000144	RETURN	PRCV 35
000145	END	PRCV 36

SUBPROGRAM LENGTH

000247

UNUSED COMPILER SPACE

042300

SUBROUTINE PRDEF (ND,D,NS,S,IO,IPT2,USLI)

PRDEF 1


```

C      PRINT THE DEFLECTION INFORMATION
COMMON/DEF/ IPT3, IPT10
DIMENSION D(20,4), S(4), ID(20)
MN = NS - 1
C = 800.
IF (IPT2 .EQ. 2) GO TO 250
IF (ABS(USLL) .GT. .0001) C = 1000.
C      PRINT THE DEFLECTION TABLE
250 WRITE (6,210)
210 FORMAT (1H0,23HGIROER DEFLECTIONS.....//10X,7HTHE GIRDERS DEFLECTI
IONS ARE SHOWN IN THE FOLLOWING TABLE...//10X,9HIVE LOAD/20X,
22HPI,10X,9HOFAD 1000,10X,12HSUPERIMPOSED,10X,9HDOWNWARD,10X,
36HUPWARD//)
DO 215 I = 1, ND
IA = ID(I)
215 WRITE (6,211) IA, D(I,1), D(I,3), D(I,2), D(I,4)
211 FORMAT (20X,12,10X,F7.4,14X,F7.4,13X,F7.4,11X,F7.4)
WRITE (6,212)
212 FORMAT (20X,36HTHE ABOVE DEFLECTIONS ARE IN INCHES.)
C      PRINT THE MAXIMUM ALLOWABLE LIVE LOAD DEFLECTIONS
WRITE (6,231)
231 FORMAT (1H0)
DO 230 I = 1, MN
DM = S(I) / C
230 WRITE (6,235) I, DM
235 FORMAT (11X,5)HTHE MAXIMUM ALLOWABLE LIVE LOAD DEFLECTION IN SPAN
1,11,3H = ,F7.4,9H INCHES.)
IF (IPT3 .GT. 9) WRITE (6,240)
240 FORMAT (1H0,10X,74HTHE SUPERIMPOSED DEFLECTIONS ARE ZERO BECAUSE THE
DESIGN IS NON-COMPOSITE.)
RETURN
END

```

SUBPROGRAM LENGTH
000263

UNUSED COMPILER SPACE
042200

```

SUBROUTINE PRSECTINE,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CL,
ILCON,CH,COORD,SURLFN,KEY,NCY)
C      PRINTS THE SECTION PROPERTIES FOR A DESIGN
COMMON/ONE/ IPT3, IPT10
DIMENSION NOS(40), NAME(18), PLWID(80,2), PLATH(80,2), ISTEEL(80),
ICS(80), ILCON(80), CL(80), INCON(80), CH(80), COOR(81), SURLFN(80)
REAL ILCON, INCON, ISTEEL
C      DETERMINE THE HEADING FOR THE TABLE
IF (KEY .EQ. 1) GO TO 140
IF (KEY .EQ. 2) GO TO 60
WRITE (6,170) 10Y
170 FORMAT (1H1//40X,36HTHE OPTIMUM DESIGN FOR CYCLE NUMBER ,12//)
GO TO 180
60 WRITE (6,61)

```

```

000042      61 FORMAT (11H1//50X,15HTHE FINAL DESIGN//)      PRSF 15
000042      G7 TO 140      PRSF 16
000046      150 WRITE (6,161)      PRSF 17
000052      161 FORMAT (1H1)      PRSF 18
000052      140 IF (JPT3 .GT. 9) GO TO 1000      PRSF 19
C      PRINTS THE TABLE HEADING FOR A COMPOSITE DESIGN      PRSF 20
000061      WRITE (6,151)      PRSF 21
000064      151 FORMAT (110,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HROT,4X,3HROT,      PRSF 22
      15X,4HLEFT,4X,5HRIGHT/2X,1H1,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,      PRSF 23
      25HPLWID,2X,5HPLATH,2X,6HSTFEEL,5X,2HCS,5X,5H1 CON,5X,2H1,5X,      PRSF 24
      35H1 CON,5X,2HCH,5X,4HCOR,5X,4HCOR,3X,6HSHOEFN//)      PRSF 25
000064      G7 TO 1100      PRSF 26
C      PRINTS THE TABLE HEADING FOR A NON-COMPOSITE DESIGN      PRSF 27
000070      1070 WRITE (6,1151)      PRSF 28
000074      1151 FORMAT (110,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HROT,4X,3HROT,      PRSF 29
      122X,4HLEFT,4X,5HRIGHT/2X,1H1,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,      PRSF 30
      25HPLWID,2X,5HPLATH,2X,6HSTFEEL,5X,2HCS,5X,      PRSF 31
      3      4HCOR,5X,4HCOR,3X,6HSHOEFN//)      PRSF 32
C      PRINTS THE TABLE FOR A COMPOSITE DESIGN      PRSF 33
000074      1001 G7 155 I = 1, NF      PRSF 34
000101      K = 105(J)      PRSF 35
000103      IF (JPT3 .GT. 9) GO TO 1155      PRSF 36
000107      WRITE (6,152) J,NAME(K),PLWID(J,1),PLATH(1,1),PLWID(J,2),      PRSF 37
      1PLATH(J,2),1STEEL(J),CS(J),11 CON(J),CL(J),11 CON(J),CH(J),COOR(J),      PRSF 38
      2COOR(J,1),SHRLEN(J)      PRSF 39
000247      152 FORMAT (X,12,X,47,X,F6,2,X,F6,3,X,F6,2,X,F6,2,X,F6,1,X,F7,3,X,      PRSF 40
      1FH,1,X,F7,3,X,F6,1,X,F7,3,X,F6,1,X,F6,1,X,F7,2)      PRSF 41
000247      G7 TO 155      PRSF 42
C      PRINTS THE TABLE FOR A NON-COMPOSITE DESIGN      PRSF 43
000253      1155 WRITE (6,1152) J,NAME(K),PLWID(J,1),PLATH(1,1),PLWID(J,2),      PRSF 44
      1PLATH(J,2),1STEEL(J),CS(J),COOR(J),COOR(J,1),SHRLEN(J)      PRSF 45
000360      1152 FORMAT (X,12,X,47,X,F6,2,X,F6,3,X,F6,2,X,F6,2,X,F6,1,X,F7,3,X,      PRSF 46
      1      FH,1,X,F7,3,X,F6,1,X,F7,2)      PRSF 47
000360      155 CONTINUE      PRSF 48
000366      IF (JPT3 .GT. 9) WRITE (6,1003)      PRSF 49
000400      1003 FORMAT (11H0,10X,2HTHE DESIGN IS NON-COMPOSITE.)      PRSF 50
000400      RETURN      PRSF 51
000401      EN7      PRSF 52

```

SUBPROGRAM LENGTH

000550

UNUSED COMPILER SPACE

041100

```

C      SUBROUTINE PRSTF (NS,REAR)      PRST 1
C      PRINTS THE STEEPLER DESIGN INFORMATION      PRST 2
000005      DIMENSION HEAD(5,1)      PRST 3
C      PRINT THE HEADINGS      PRST 4
000005      WRITE (6,201)      PRST 5
000010      201 FORMAT (110,30HBEARING STEEPLER DESIGNS.....)      PRST 6
000010      KPI = 0      PRST 7
C      PRINT THE INFORMATION FOR EACH SUPPORT      PRST 8

```



```

000011      DO 200 I = 1, NS                      PRST 9
000014      IF (ABS(HEAR(I,1)) .LE. .00001) GO TO 200 PRST 10
000020      KPT = 1 PRST 11
000021      WRITE (A,300) I, HEAR(I,1), HEAR(I,2), HEAR(I,3) PRST 12
000043      200 CONTINUE PRST 13
000047      300 FORMAT (10X,12HSUPPORT NO. ,I1,5X,PHUSE 2 = ,F7.3,2H X,F4.3, PRST 14
          1411 PLATES. CONNECT ONE TO EACH SIDE WITH A ,F7.4,19H INCH FILLET PRST 15
          2WELLS.) PRST 16
          C      ONLY PRINT IF NO STIFFENERS ARE REQUIRED PRST 17
000047      IF (KPT .EQ. 0) WRITE (A,202) PRST 18
000055      202 FORMAT (20X,36HREARING STIFFENERS ARE NOT REQUIRED.) PRST 19
000055      RETURN PRST 20
000056      END PRST 21

```

SUBPROGRAM LENGTH

000125

UNUSED COMPILER SPACE

042700

```

          SUBROUTINE PRSTW (STRESS,NA)                      PRSR 1
          C      PRINTS THE ENTIRE STRESS TABLE FOR A GIRDER PRSR 2
000005      COMMON/ONE/JPT3,IPT10 PRSR 3
000005      DIMENSION STRESS(81,4,2) PRSR 4
          C      PRINT THE HEADING FOR THE TABLE PRSR 5
000005      WRITE (A,890) PRSR 6
000010      890 FORMAT (11H1,37HSTRESSES ALONG THE DESIGN GIRDER.....//5X,2HPT, PRSR 7
          15X,10HBOT STRESS,4X,10HTOP STRESS,4X,11HCOMB STRESS,4X, PRSR 8
          212HSHEAR STRESS,11X,22HSAFE STRESS INDICATOR//) PRSR 9
          C      PRINT THE STRESSES FOR EACH POINT PRSR 10
000010      DO 2500 I = 1, NA PRSR 11
000013      2500 WRITE (A,891) I, STRESS(I,1,1), STRESS(I,2,1), STRESS(I,3,1), PRSR 12
          1STRESS(I,4,1), STRESS(I,1,2), STRESS(I,2,2), STRESS(I,3,2), PRSR 13
          2STRESS(I,4,2) PRSR 14
000064      891 FORMAT (4X,13,3X,4(F10.3,5X),8X,4(F5.1,X)) PRSR 15
          C      PRINT THE NON-COMPOSITE DESIGN STATEMENT PRSR 16
000064      IF (JPT3 .GT. 9) WRITE (A,870) PRSR 17
000074      870 FORMAT (10H,10X,2HTHE DESIGN IS NON-COMPOSITE.) PRSR 18
000074      RETURN PRSR 19
000075      END PRSR 20

```

SUBPROGRAM LENGTH

000142

UNUSED COMPILER SPACE

042600

SUBROUTINE HFAC

HFAC 1

```

C          SOLVES FOR THE REACTION INFLUENCE LINES
000002      COMMON NAME(18),SARFA(14),DEPTH(14),FLWID(14),FLTH(14),WENTH(14),
      IDAF(18),IX(14),IY(14),NSECT(14),NOS(40),ISTEEL(40),CS(40),
      2ILCON(40),CL(40),THCON(40),CH(40),PLATH(40),PIWID(40),PI,
      3SLA4WD,SLA4IH,MR,COORD(41),R1(15,41,3),CC(5,41),FM,SPAN(4),IOSP(5),
      4TMU(14),NA,LENGTH,RO(R1),SUBLEN(40),NS
      000002      COMMON/ONE/JP13,TP110
      000002      DIMENSION R1(3), R12(3), R13(3),R1(5,R1)
      000002      REAL IX, IY, ISTEEL, ILCON, THCON, MR, LENGTH
C          ZERO OUT VARIOUS MATRICES
000002      DO 41 I = 1, 5
      000004      DO 40 J = 1, R1
      000005      DO 40 K = 1, 3
      000006      10 RIL(I,J,K) = 0.0
      000017      DO 41 I1 = 1, 5
      000020      31 CC(I,I1) = 0.0
      000027      DO 43 I = 1, 40
      000030      33 140(1(I)) = 0.0
      000033      DO 44 I = 1, R1
      000034      34 RO(I) = 0.0
C          BEGIN TO CALCULATE FOR EACH ANALYSIS POINT
000037      DO 49 I = 1, NA
C          CALCULATE REACTIONS AND MOMENTS FOR A SINGLE BEAM
000040      RR = COOR(I)/LENGTH
      000042      RL = 1.0 - RR
      000044      DO 42 J = 1, 1
      000046      32 RO(J) = RL * COOR(J)
      000052      MM = I * 1
      000054      DO 45 J = MM, NA
C          CALL ON EUSE1 TO SET UP THE RIL ARRAY WHICH IS THE RIGHT
C          HAND SIDE OF THE EQUATION.
000055      35 RO(J) = RL * COOR(J) - (COOR(J) - COOR(1))
      000064      CALL EUSE1(RO,SUBLEN,LENGTH,COORD,NE,ISTEEL,FM,NS,IOSP,SPAN,R11)
      000076      CALL EUSE1(RO,SUBLEN,LENGTH,COORD,NE,ILCON,FM,NS,IOSP,SPAN,R12)
      000111      CALL EUSE1(RO,SUBLEN,LENGTH,COORD,NE,THCON,FM,NS,IOSP,SPAN,R13)
      000124      1335 K = 3
      000125      DO 44 J = 1, 3
      000127      RIL(1,I,J) = 1.0
      000134      RIL(2,I,J) = COOR(I)
      000140      IF (K.GT. NS) GO TO 44
      000144      RIL(K,I,1) = R11(J)
      000147      RIL(K,I,2) = R12(J)
      000151      RIL(K,I,3) = R13(J)
      000154      K = K + 1
      000155      44 CONTINUE
      000157      39 CONTINUE
C          ASSEMBLE COEFFICIENT MATRIX AND SOLVE THE EQUATIONS
000162      MN = NS - 1
      000164      DO 49 K = 1, 3
      000165      DO 40 I = 1,NS
      000166      CC(I,I) = 1.0
      000171      IF (I.NE.1) GO TO 41
      000173      CC(2,1) = 0.0
      000174      GO TO 40
      000174      41 CC(2,I) = CC(2,I-1) * SPAN(I-1)
      000202      40 CONTINUE
      000205      DO 45 I = 2, MN
      000206      DO 45 J = 1, NS

```

```

000207      JJ = IOSP(J)                                RFAC 60
000211      45 CC(1+1,J) = KJL(1+1,J,K)              RFAC 61
000224      DO 47 J = 1, NS                             RFAC 62
000226      DO 47 J = 1, NA                             RFAC 63
000227      37 RI(1,J) = RII(1,J,K)                   RFAC 64
000244      CALL SOLVE (NS,NA,CC,RI,NA)               RFAC 65
000247      DO 48 J = 1, NS                             RFAC 66
000251      DO 48 J = 1, NA                             RFAC 67
000252      38 RIL(1,J,K) = RI(1,J)                   RFAC 68
000267      49 CONTINUE                                RFAC 69
000271      1400 RETURN                                RFAC 70
000272      END                                          RFAC 71

```

SUBPROGRAM LENGTH

001162

UNUSED COMPILER SPACE

041500

```

      SUBROUTINE READIN                                RFAD 1
      READS IN THE DATA FOR THE PROBLEM              RFAD 2
000002      COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WENTH(18), RFAD 3
      IDAR(18),IX(18),IY(18),HSECT(18),NOS(18),ISTEEL(18),CS(18), RFAD 4
      2ILCON(18),CL(18),ILCON(18),CH(18),PLAT(18),PI(18),ID(18),NF, RFAD 5
      3SLABWD,SLABTH,MX,COORD(18),RJI(18),CC(18),F,SPAN(4),IDSP(5), RFAD 6
      4TDR(18),NA,LENGTH,R0(18),SUBLEN(18),NS,UM(18),TRD(18),PROP(3), RFAD 7
      5FCS,STRESS(18,4,2),DESMOM(18,4),DESBFA(5,4),DESSM(162,4),OFAR(5,3), RFAD 8
      6COSI(9)                                          RFAD 9
000002      COMMON/ONE/JP13,JP14                      RFAD 10
000002      COMMON/TWO/LPCM                             RFAD 11
000002      COMMON/THREE/AS,MCS,JP16                  RFAD 12
000002      COMMON/FOUR/HANDCH                          RFAD 13
000002      COMMON/FIVE/IDFF(20),DESDFF(20,4),IDFF,PERCENT,XYC RFAD 14
000002      COMMON/SIX/JP12                             RFAD 15
000002      COMMON/SEVEN/JP17                          RFAD 16
000002      REAL IX,IY,ISTEEL,ILCON,ILCON,MX,LENGTH RFAD 17
000002      DIMENSION TITLE(36),FPS(4),LPCM(18) RFAD 18
000002      INTEGER FPS,TITLE                             RFAD 19
      C ZERO THE REQUIRED INPUT ARRAYS                RFAD 20
000002      DO 50 I = 1, 18                             RFAD 21
000004      NOS(I) = 0                                    RFAD 22
000005      ISTEEL(I) = 0.0                             RFAD 23
000006      CS(I) = 0.0                                  RFAD 24
000007      ILCON(I) = 0.0                             RFAD 25
000010      CL(I) = 0.0                                  RFAD 26
000011      IACON(I) = 0.0                             RFAD 27
000012      CH(I) = 0.0                                  RFAD 28
000013      SUBLEN(I) = 0.0                             RFAD 29
000014      DO 50 K = 1, 2                                RFAD 30
000016      PLAT(1,K) = 0.0                             RFAD 31
000021      50 PLAT(1,K) = 0.0                             RFAD 32
000030      SLABWD = 0.0                                    RFAD 33
000031      SLABTH = 0.0                                    RFAD 34

```

```

000032      MR = 0.0                                HEAD 35
000033      DO 501 I = 1, 4                          HEAD 36
000034      SPAN(I) = 0.0                             HEAD 37
000035      501 F25(I) = 0                             HEAD 38
000040      DO 502 I = 1, 36                         HEAD 39
000041      502 TITL(I) = 0                           HEAD 40
000044      DO 503 I = 1, 5                         HEAD 41
000045      503 JOSP(I) = 0                           HEAD 42
000050      DO 504 I = 1, 10                        HEAD 43
000051      LPO4(I) = 0                             HEAD 44
000052      504 T(L)(I) = 0.0                       HEAD 45
000055      DO 505 I = 1, 4                         HEAD 46
000056      505 UNL(I) = 0.0                       HEAD 47
000061      DO 506 I = 1, 3                         HEAD 48
000062      506 PAMP(I) = 0.0                       HEAD 49
000065      DO 508 I = 1, 20                      HEAD 50
000066      508 JOEX(I) = 0                         HEAD 51
000071      DO 509 I = 1, 51                      HEAD 52
000072      509 CDOX(I) = 0.0                     HEAD 53
C      READ TITLE AND INPUT PARAMETER DATA CARDS HEAD 54
000075      READ (5,100) (TITL(I), I = 1, 36)     HEAD 55
000106      100 FORMAT (18A4/18A4)                 HEAD 56
000106      READ (5,102) IPT1,IPT2,IPT3,IPT4,IPT5,IPT6,IPT7,IPT8,IPT9,IPT10 HEAD 57
000136      102 FORMAT (10I5)                       HEAD 58
C      PRINT TITLE OF THE PROBLEM                HEAD 59
000136      WRITE (6,101) (TITL(I), I = 1, 36)    HEAD 60
000150      101 FORMAT (10H)///24X,18A4///24X,18A4/// HEAD 61
C      PRINT THE HEADINGS IF IPT10 IS GREATER THAN 0 HEAD 62
000150      IF (IPT10 .LT. 3) GO TO 1011          HEAD 63
000153      WRITE (6,103)                           HEAD 64
000156      103 FORMAT (10H,24X,66HTHIS PROBLEM WILL BE SOLVED USING THE METHOD OF HEAD 65
      TITL OF THE HIGHWAY/25X,70HBRIDGE SPECIFICATIONS PUBLISHED IN 1969 HEAD 66
      2 BY THE AMERICAN ASSOCIATION OF/25X,60HSTATE HIGHWAY OFFICIALS. THEAD 67
      THE OPTIMUM PROGRAM USED IN THIS DESIGN WAS/24X,66HDEVELOPED BY ROHREAD 68
      SEPT E. ROSEK DURING HIS STUDY FOR A MASTERS DEGREE/25X,60HAT PURDUE HEAD 69
      SE UNIVERSITY. THE RESEARCH WAS FINANCED BY THE JOINT HIGHWAY/25X, HEAD 70
      67HRESEARCH BOARD. THE RESEARCH WAS DIRECTED BY THE FOLLOWING MEMBEAD 71
      25HOF THE/25X,25HTHE RESEARCH COMMITTEE,.,25X,25X, PROFESSOR J. THEAD 72
      A. GANN/25X,25H2. PROFESSOR A. D. W. LEITCH/25X, HEAD 73
      92H3. PROFESSOR C. W. LOVELL) HEAD 74
000156      WRITE (6,104)                          HEAD 75
000162      104 FORMAT (10H,24X,72HTHE PROGRAM, AS DEVELOPED, WILL DESIGN THE OPTIMUM HEAD 76
      10H4 HIGHWAY BRIDGE GIRDER/25X,67HCOMPOSED OF A ROLLED SECTION AND THEAD 77
      25H COMPOSITELY WITH THE DECK SLAB./25X,72HTHE PROGRAM CAN SOLVE A HEAD 78
      36H SPAN CONTINUOUS BRIDGE GIRDER AS WELL AS THE/25X,71HSIMPLE SPREAD HEAD 79
      40H GIRDER PROBLEM. THE SOLUTION IS FOUND BY OPTIMIZING EITHER/ HEAD 80
      52H,64HTHE WEIGHT OR THE TOTAL COST INCLUDING THE COSTS OF FABRICATEAD 81
      67H10,72H) HEAD 82
C      PRINT THE INPUT PARAMETERS IF IPT10 IS GREATER THAN 1 HEAD 83
000162      1010 IF (IPT10 .LT. 3) GO TO 1011      HEAD 84
000165      WRITE (6,105) IPT1,IPT2,IPT3,IPT4,IPT5,IPT6,IPT7,IPT8,IPT9,IPT10 HEAD 85
000214      105 FORMAT (11H,24X,25HTHE INPUT PARAMETERS ARE,.,24X,25H) HEAD 86
      15X,24X,5H1. ELEMENT INFORMATION/24X,25H) IPT2 = ,12, 3,24X,5H) TYPE OF LOAD HEAD 87
      25H) 1000/24X,25H) IPT3 = ,12,50,24X,5H) INITIAL GIRDER CROSS-SECTION / HEAD 88
      13X,24X,5H) IPT4 = ,12,50,13H) UNITS OF STAY/24X,5H) IPT5 = ,12, 3,40H) ACTION HEAD 89
      40H) 1. CLAMPED, 2. FREE, 3. ON DESIGNATION/24X,5H) IPT6 = ,12,50,25H) TYPE HEAD 90
      50H) OF JAIL/24X,5H) IPT7 = ,12,50,13H) STAY CYCLES/24X, HEAD 91
      67H) 1. = ,12,50,25H) MATERIAL USED IN THE BRIDGE/24X,5H) IPT8 = , HEAD 92

```

```

712.4.1.100 DEFLECTIONS DESIRED/14X, DIRECTION = .12, 0X, 14X DIRECTION PARAMETER HEAD 93
RTF(1) HEAD 94
000214 WRITE (4,106) HEAD 95
000220 106 FORMAT (1H0,24X,73 (SEE INPUT INSTRUCTIONS FOR THE DEFINITIONS OF HEAD 96
THE INPUT PARAMETER VALUES.//) HEAD 97
C READ THE LENGTHS OF SPANS HEAD 98
000220 101 READ (5,110) (SPAN(I), I = 1, 4) HEAD 99
000232 110 FORMAT (4F10,4) HEAD 100
C DETERMINE THE NUMBER OF SUPPORTS AND THE TOTAL LENGTH HEAD 101
000232 NS = 1 HEAD 102
000233 LENGTH = 0.0 HEAD 103
000234 DO 111 I = 1, 4 HEAD 104
000236 LENGTH = LENGTH + SPAN(I) HEAD 105
000240 IF (SPAN(1) .LT. .001) GO TO 112 HEAD 106
000243 111 NS = NS + 1 HEAD 107
000246 112 MN = NS - 1 HEAD 108
C DETERMINE THE NUMBER OF ELEMENTS PER SPAN HEAD 109
C DETERMINE THE COORDINATE ARRAY HEAD 110
C DETERMINE THE INDEX OF THE SUPPORT POINTS HEAD 111
C DETERMINE THE SURLENGTH ARRAY HEAD 112
000250 IF (IPT1 .NE. 0) GO TO 113 HEAD 113
000251 DO 114 I = 1, MN HEAD 114
000253 114 EPS(I) = 10 HEAD 115
000257 GO TO 115 HEAD 116
000257 113 IF (IPT1 .NE. 2) GO TO 120 HEAD 117
000261 READ (5,116) (EPS(I), I = 1, MN) HEAD 118
000274 116 FORMAT (4F10) HEAD 119
000274 115 IOSP(1) = 1 HEAD 120
000275 NE = 0 HEAD 121
000276 NA = 1 HEAD 122
000277 COOR(1) = 0.0 HEAD 123
000300 DO 117 I = 1, MN HEAD 124
000302 IOSP(I+1) = IOSP(1) + EPS(I) HEAD 125
000304 NE = NE + EPS(I) HEAD 126
000306 117 NA = NA + EPS(I) HEAD 127
000311 DO 118 I = 1, MN HEAD 128
000313 IA = IOSP(I) HEAD 129
000315 IB = IOSP(I+1) - 1 HEAD 130
000317 SUR = SPAN(I)/EPS(I) HEAD 131
000321 DO 119 J = IA, IB HEAD 132
000323 SURLEN(J) = SUR HEAD 133
000325 119 COOR(J+1) = COOR(J) + SURLEN(J) HEAD 134
000331 118 CONTINUE HEAD 135
000334 GO TO 125 HEAD 136
000334 120 READ (5,121) NA HEAD 137
000342 121 FORMAT (110) HEAD 138
000342 READ (5,122) (COOR(I), I = 1, NA) HEAD 139
000355 122 FORMAT (7F10,2) HEAD 140
000355 IOSP(1) = 1 HEAD 141
000356 NE = NA - 1 HEAD 142
000360 DO 123 I = 1, NE HEAD 143
000362 123 SURLEN(I) = COOR(I+1) - COOR(I) HEAD 144
000367 J = 1 HEAD 145
000370 DO 124 I = 1, MN HEAD 146
000371 TEMP = 0.0 HEAD 147
000372 K = 1 HEAD 148
000373 IF K = TEMP + SURLEN(I) HEAD 149
000376 127 IF (TEMP .GT. SPAN(I)) GO TO 125 HEAD 150

```

```

000575      WRITE (6,1031) NAME(I), I = 1, 14          HEAD 209
000607 1031 FORMAT (1H0,2X,6(A7,2X)/21X,6(A7,2X)/21X,6(A7,2X)) HEAD 210
000607      GO TO 140                                     HEAD 211
000610      139 WRITE (6,1030)                             HEAD 212
000614      WRITE (6,1002)                                 HEAD 213
000620 1002 FORMAT (1H0,6X,4HNAME,4X,5HSAKEA,4X,5HDEPTH,4X,4HFLTH,4X,4HFLTH, HEAD 214
      16X,5HDEPTH,7X,3HDAF,10X,2HIX,13X,2HTY,6X,5HSECT//) HEAD 215
000620      DO 135 I = 1, 14                             HEAD 216
000622      135 WRITE (6,1004) NAME(I),SARFA(I),DEPTH(I),FLTH(I),FLTH(I), HEAD 217
      1HFLTH(I),IAF(I),IX(I),IY(I),ASECT(I) HEAD 218
000653 1004 FORMAT (1X,A7,2(X,F6,2),3(SX,F6,2),5X,F5,2,~(SY,F10,2),5X,I2) HEAD 219
      C      READ THE COMPOSITE SECTION DATA IF REQUESTED HEAD 220
000653      140 IF (IPT1 .LT. 9) GO TO 141                HEAD 221
000654      JPI3 = IPT3                                     HEAD 222
000657      IPI3 = JPI3 - 10                               HEAD 223
000660      WRITE (6,1141)                                 HEAD 224
000664 1141 FORMAT (1H0/1H,20X,49HTHE GIRDER DESIGN WILL BE A NON-COMPOSITE DESIGN) HEAD 225
      1ESI3(I) HEAD 226
000664      GO TO 142                                     HEAD 227
000665      141 REA(5,143) SLABWD,SLANTH, MM, EPC, AS, HANCH HEAD 228
000705 143 FORMAT (F10,2,F10,4)                          HEAD 229
000705      JPI3 = IPT3                                     HEAD 230
      C      PRINT THE COMPOSITE SECTION DATA IF REQUESTED HEAD 231
000707      WRITE (6,1040) SLABWD, SLANTH, MM, EPC, AS, HANCH HEAD 232
000726 1040 FORMAT (1H0,10X,49HTHE DESIGN WILL BE BASED ON A COMPOSITE SECTION HEAD 233
      1 HAVING THE FOLLOWING PROPERTIES,.../20X,26 EFFECTIVE WIDTH OF SLAB HEAD 234
      2H = ,F7,2,MM INCHES,/20X,17HSLAB THICKNESS = ,F6,2,MM INCHES,/20X, HEAD 235
      31HCONCRETE RATIO = ,F5,1,1H,/20X,4HUSE ,F6,2,14H PSI CONCRETE,/ HEAD 236
      420X,29HAREA OF SLAB REINFORCEMENT = ,F6,2,MM SQ. IN./20X, HEAD 237
      52HTHICKNESS OF HANCH = ,F6,4,MM INCHES.) HEAD 238
      C      DETERMINE THE INITIAL SECTION PROPERTIES HEAD 239
000726      142 IF (IPI3 .NE. 0) GO TO 149              HEAD 240
000727      DO 145 I = 1, 14                             HEAD 241
000731      NOS(I) = 1H                                     HEAD 242
000733      DO 145 K = 1, 2                               HEAD 243
000734      PLWI(I,K) = 0.0                               HEAD 244
000737      145 PLAT(I,K) = 0.0                             HEAD 245
000747      GO TO 140                                     HEAD 246
000747      149 IF (IPI3 .NE. 2) GO TO 144              HEAD 247
000751      DO 146 I = 1, 14                             HEAD 248
000753      146 REA(5,147) NOS(I),PLWID(I,1),PLATH(I,1),PI ID(I,2),PLATH(I,2) HEAD 249
000773      147 FORMAT (110,4F10,4)                       HEAD 250
000773      GO TO 150                                     HEAD 251
000773      144 REA(5,147) NOS(I),PLWID(I,1),PLATH(I,1),PI ID(I,2),PLATH(I,2) HEAD 252
001011      DO 148 I = 2, 14                             HEAD 253
001013      NOS(I) = NOS(I)                               HEAD 254
001015      DO 148 K = 1, 2                               HEAD 255
001016      PLWI(I,K) = PLWID(I,K)                       HEAD 256
001023      148 PLAT(I,K) = PLATH(I,K)                   HEAD 257
001032      150 CALL SECTC                                HEAD 258
      C      PRINT THE INITIAL PROPERTIES IF IT IS FASTER THAN A HEAD 259
001033      IF (IPT1 .LT. 2) GO TO 160                   HEAD 260
001036      WRITE (6,1040)                                HEAD 261
001041      47.0 FORMAT (1H)/275X,54HPROPERTIES OF THE SECTION USED FOR THE INITIAL HEAD 262
      1 ANALYSIS(//) HEAD 263
001041      IF (IPT1 .GT. 9) GO TO 1160                  HEAD 264
001045      WRITE (6,151)                                    HEAD 265
001050      151 FORMAT (1H0,5X,4HSECT,4X,3HIDP,4X,3HTOP,4X,3H0.4X,3HBDT,

```



```

1554,4HLEFT,4X,4HRIGHT/2X,1H],3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,
25HPLXID,2X,5HPLATH,2X,6HISTEEL,5X,2HCS,6X,5HPLCOM,5X,2HCL,5X,
3541401.5X,2HCL,6X,4HCOOR,5X,4HCOOR,2X,6HSP,1F/2X)
GO TO 1171
001050 1160 WRITE (5,1151)
001051 1151 FORMAT (11F0,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HROT,4X,3HROT,
001055 122X,4HLEFT,4X,4HRIGHT/2X,1H],3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,
25HPLXID,2X,5HPLATH,2X,6HISTEEL,5X,2HCS,6X,
3 4HCOOR,5X,4HCOOR,2X,6HSP,1F/2X)
001055 1170 DO 155 J = 1, 15
001057 K = TOS(J)
001061 IF (JPTX .GT. 4) GO TO 1155
001064 WRITE (5,152) J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(1,2),
PLATH(1,2),ISTEEL(J),CS(J),11COM(J),CL(J),THCOM(J),CH(J),COOR(J),
2COMX(J,1),SHRLEN(J)
001125 152 FORMAT (1X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,FR,1,X,F7.3,X,
FR,1,X,F7.3,X,FR,1,X,F7.3,X,FR,1,X,FR,1,X,F7.2)
001125 GO TO 155
001125 1155 WRITE (5,1152) J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(1,2),
PLATH(1,2),ISTEEL(J),CS(J),COOR(J),COOR(1,1),SHRLEN(J)
001160 1152 FORMAT (1X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,FR,1,X,F7.3,X,
FR,1,X,FR,1,X,F7.2)
001160 155 CONTINUE
001163 WRITE (5,1156)
001166 1156 FORMAT (1H0,4CHTHE UNITS OF THE ABOVE TABLE ARE INCHES.)
HEAD IN THE LOAD INFORMATION
001166 160 IF (LPTS .NE. 0) GO TO 170
001167 READ (5,161) LPTS, PLL, SLAWT, SOL, GDI, SLL, OI, OIL
001213 161 FORMAT (A6,F6.3,4F10.4)
001213 IF (LPTS .EQ. 4H H10) O = PLL * .5
001217 IF (LPTS .EQ. 4H H15 .OR. LPTS .EQ. 4H H15) O = PLL * .75
001230 IF (LPTS .EQ. 4H H20 .OR. LPTS .EQ. 4H H20) O = PLL
001240 IF (LPTS .EQ. 2) GO TO 175
001242 UNL(1) = O * .6533
001244 UNL(2) = O * 1P.
001246 UNL(3) = O * 2P.
001250 UNL(4) = SOL
001251 UNL(5) = GDI
001252 UNL(6) = 11.0 - OIL) * PLL * 24.
001256 UNL(7) = SLL
001260 UNL(8) = OI
001261 UNL(9) = SLAWT
001263 175 IF (LPTS .EQ. 1) GO TO 176
001265 TPL(1) = O * H.
001267 TPL(2) = O * 32.
001271 TPL(4) = 16P.
001272 TPL(7) = 20L
001274 TPL(8) = GDI
001275 IF (LPTS .EQ. 4H H15 .OR. LPTS .EQ. 4H H20) GO TO 177
001305 TPL(3) = 0.0
001306 TPL(5) = 90000.
001307 TPL(6) = 90001.
001311 GO TO 178
001311 177 TPL(3) = O * 32.
001313 TPL(5) = 16P.
001315 TPL(6) = 16P.
001316 178 TPL(9) = OI
001320 TPL(10) = SLAWT

```

```

001321      GO TO 176                                READ 325
001322      170 IF (IPT2.EQ. 1) GO TO 179            READ 326
001324      READ (5,180) (TRLO(I), I = 1, 10)        READ 327
001336      180 FORMAT (10F7.3)                     READ 328
001336      179 IF (IPT2.EQ. 2) GO TO 176            READ 329
001340      READ (5,181) (UNLO(I), I = 1, 9)         READ 330
001352      181 FORMAT (9F8.4)                       READ 331
C          CONVERT THE LOAD INFORMATION TO UNITS OF INCHES
001352      176 IF (IPT4.EQ. 0) GO TO 190            READ 332
001353      IF (IPT5.EQ. 0) GO TO 185                READ 333
001354      TRLO(4) = TRLO(4) * 12.                  READ 334
001356      TRLO(5) = TRLO(5) * 12.                  READ 335
001357      TRLO(6) = TRLO(6) * 12.                  READ 336
001360      UNLO(1) = UNLO(1)/12.                    READ 337
001361      185 TRLO(7) = TRLO(7)/12.                READ 338
001363      TRLO(8) = TRLO(8)/12.                    READ 339
001364      TRLO(10) = TRLO(10)/12.                 READ 340
001366      UNLO(4) = UNLO(4)/12.                   READ 341
001367      UNLO(5) = UNLO(5)/12.                   READ 342
001371      UNLO(7) = UNLO(7)/12.                   READ 343
001372      UNLO(9) = UNLO(9)/12.                   READ 344
C          PRINT THE LOAD INFORMATION              READ 345
001374      190 CONTINUE                             READ 346
001374      WRITE (6,195)                             READ 347
001400      195 FORMAT (1H1,3X,24HLOADING INFORMATION.....//) READ 348
001400      IF (IPT5.EQ. 0) WRITE (6,200) LDIS        READ 349
001407      200 FORMAT (1H0,3X,25HTHE DESIGN IS BASED ON A 14.8H LOADING//) READ 350
001411      IF (IPT2.EQ. 1) GO TO 191                READ 351
001415      WRITE (6,201)                             READ 352
001425      201 FORMAT (40X,25HTRUCK LOAD INFORMATION...//) READ 353
001425      WRITE (6,202) TRLO(1),TRLO(2)            READ 354
001425      202 FORMAT (50X,1PHFIRST AXLE LOAD = ,24X,FR.3,FW KIPS/50X,1PHSECONO AXLE READ 355
001425      AXLE LOAD = ,24X,FR.3,5H KIPS)            READ 356
001425      TP = TRLO(4)/12.                          READ 357
001427      WRITE (6,203) TP                          READ 358
001435      203 FORMAT (50X,4CHSPACING BETWEEN FIRST AND SECOND AXLE = ,4X,FR.3, READ 359
001435      15H FEET)                                READ 360
001435      IF (IPT5.EQ.1.OR.LDIS.EQ.4HHS15.OR.LDIS.EQ.6HHS20) GO TO 192 READ 361
001450      GO TO 143                                READ 362
001451      192 TP1 = TRLO(5)/12.                      READ 363
001453      TP2 = TRLO(6)/12.                        READ 364
001455      WRITE (6,204) TP(0(3), TP1, TP2          READ 365
001467      204 FORMAT (50X,1PHTHIRD AXLE LOAD = ,24X,FR.3,FW KIPS/50X,44HMINIMUM READ 366
001467      1SPACING OF SECOND AND THIRD AXLES = ,FR.3,FW FEET/50X,44HMAXIMUM READ 367
001467      2PACING OF SECOND AND THIRD AXLES = ,FR.3,5H FEET) READ 368
001471      193 TP1 = TRLO(7) * 12.                  READ 369
001473      TP2 = TRLO(8) * 12.                      READ 370
001474      TP3 = TRLO(10) * 12.                     READ 371
001506      WRITE (6,205) TP1, TP2, TP3             READ 372
001506      205 FORMAT (50X,25HSUPERIMPOSED DEAD LOAD = ,10X,FR.3,8H KIPS/FT/50X, READ 373
001514      12HMINIMUM DEAD WEIGHT = ,23X,FR.3,8H KIPS/FT/50X, READ 374
001514      22HDEAD WEIGHT OF SLAB = ,22X,FR.3,8H KIPS/FT) READ 375
001514      IF (TRLO(9) .GT. .1) WRITE (6,206)        READ 376
001514      206 FORMAT (50X,29HNO IMPACT FACTOR WILL BE USED//) READ 377
001522      191 IF (IPT2.EQ. 2) GO TO 249            READ 378
001522      WRITE (6,210)                             READ 379
001522      210 FORMAT (1H0,3X,27HLANE LOADING INFORMATION...//) READ 380
001522      TP = UNLO(1) * 12.                      READ 381
001522      READ 382

```



```

001524      WRITE (6,211) TP, UNLO(2), UNLO(3)      HEAD 383
001536 211 FORMAT (5X,20H)IVE LANE LOADING = ,24X,FR,3,4H KIPS/FT/50X,      HEAD 384
      13HCONCENTRATED LOAD FOR MOMENT = ,17X,FR,3,4H KIPS/50X,      HEAD 385
      23HCONCENTRATED LOAD FOR SHEAR = ,16X,FR,3,4H KIPS)      HEAD 386
001536      TP1 = UNLO(4) * 12.      HEAD 387
001540      TP2 = UNLO(5) * 12.      HEAD 388
001542      TP3 = UNLO(9) * 12.      HEAD 389
001543      WRITE (6,205) TP1, TP2, TP3      HEAD 390
001555      TP4 = UNLO(7) * 12.      HEAD 391
001557      WRITE (6,212) TP1, UNLO(6)      HEAD 392
001567 212 FORMAT (5X,21H)SIDEWALK LIVE LOAD = ,23X,FR,3,4H KIPS/FT/50X,      HEAD 393
      13HCONCENTRATED INTERSTATE LOADING = ,10X,FR,3,4H KIPS)      HEAD 394
001567      IF (UNLO(8) .GT. .1) WRITE (6,206)      HEAD 395
001575 244 IF (TRLO(4) .LT. .1 .AND. UNLO(8) .LT. .1) WRITE (6,215)      HEAD 396
001610 215 FORMAT (1H0,39X,34H)IMPACT IS INCLUDED IN THE ANALYSIS)      HEAD 397
      C      HEAD THE MATERIAL PROPERTIES IF REQUIRED      HEAD 398
001610 250 IF (IP18 .EQ. 1) GO TO 260      HEAD 399
001612      PROP(1) = 20.0      HEAD 400
001614      PROP(2) = 12.0      HEAD 401
001615      PROP(3) = 36.0      HEAD 402
001617      EM = 29000.      HEAD 403
001620      FC = .4 * FPC/1000.      HEAD 404
001623      GO TO 270      HEAD 405
001623 260 RE4) (5,261) PROP(3)      HEAD 406
001631 261 FORMAT (F10,2)      HEAD 407
001631      PHOP(1) = .35 * PROP(3)      HEAD 408
001633      PHOP(2) = .33 * PHOP(3)      HEAD 409
001635      EM = 29000.      HEAD 410
001636      FC = .4 * FPC/1000.      HEAD 411
      C      PRINT THE MATERIAL PROPERTIES      HEAD 412
001641 270 CONTINUE      HEAD 413
001641      WRITE (6,275) PROP(1), PROP(2), PROP(3), EM, FC      HEAD 414
001657 275 FORMAT (1H0,39X,27H)MATERIAL PROPERTIES...//52X,27HALLOWABLE BENDING      HEAD 415
      1G STRESS = ,17X,FR,3,4H KSI/50X,25HALLOWABLE SHEAR STRESS = ,19X      HEAD 416
      2FR,3,4H KSI/50X,15H)YIELD STRESS = ,29X,FR,3,4H KSI/50X,      HEAD 417
      32H)MODULUS OF ELASTICITY = ,18X,FR,3,4H KSI/50X,      HEAD 418
      42H)ALLOWABLE CONCRETE STRESS = ,16X,FR,3,4H KSI)      HEAD 419
001657 300 CONTINUE      HEAD 420
      C      HEAD THE COST INFORMATION IF REQUIRED      HEAD 421
001657      IF (IP16 .GT. 0) GO TO 1300      HEAD 422
001662      READ(5,1301) (COST(I), I = 1, 9)      HEAD 423
001673 1301 FORMAT (9F6,2)      HEAD 424
      C      PRINT THE COST INFORMATION      HEAD 425
001673      WRITE (6,1302) COST(1),COST(2),COST(3),COST(4),COST(5),COST(6),      HEAD 426
      1COST(7),COST(8)      HEAD 427
001717 1302 FORMAT (1H0,39X,24H)UNIT COST INFORMATION...//50X,      HEAD 428
      12H)COST OF WF SECTION = 23X,FR,3,4H DOLS/LRS/50X,17H)PLATE COST = ,HEAD 429
      23X,FR,3,4H DOLS/LRS/50X,29H)COST OF BEARING STIFFENERS = ,HEAD 430
      3NEXS = ,15X,FR,3,4H DOLS/LRS/50X,27H)COST OF SHEAR CONNECTORS = ,      HEAD 431
      417X,FR,3,4H DOLS/LRS/50X,33H)COST OF CONNECTOR INSTALLATION = ,11X,HEAD 432
      5FR,3,4H DOLS/CONNECTOR/50X,18H)FIXED WELD COST = ,24X,FR,3,      HEAD 433
      6104 DOLS/WELD/50X,24H)COST OF WELD MATERIAL = ,22X,FR,3,      HEAD 434
      7194 DOLS/CU IN OF WELD/50X,22H)COST OF WELD LENGTH = ,22X,FR,3,      HEAD 435
      8184 DOLS/IN OF LENGTH)      HEAD 436
001717 1300 CONTINUE      HEAD 437
      C      DETERMINE THE NUMBER OF CYCLE INFORMATION AND PRINT      HEAD 438
001717      IF (IP17 .EQ. 0) GO TO 2300      HEAD 439
001720      IF (IP17 .EQ. 2) GO TO 2301      HEAD 440

```

```

001722      READ (5,2302) NCYC, PERCEN                      HEAD 441
001731      2302 FORMAT (I10,F10.2)                        HEAD 442
001731      WRITE (6,2303) PERCEN,NCYC                     HEAD 443
001741      2303 FORMAT (I10,20X,52H"THE DESIGN CYCLE IS REPEATED UNTIL THE FINAL MOREAD 444
1MENIS/21X,22H"IF CHANGED LESS THAN .F6,2,22H" PERCENT. THE MAXIMUMREAD 445
2/21X,20H"NUMBER OF CYCLES IS .I2,1H,////")              HEAD 446
001741      GO TO 2350                                       HEAD 447
001742      2301 READ (5,2304) NCYC                          HEAD 448
001750      2304 FORMAT (I10)                               HEAD 449
001750      PERCEN = 0.0                                    HEAD 450
001751      WRITE (6,2305) NCYC                             HEAD 451
001757      2305 FORMAT (I10,20X,40H"THE DESIGN CYCLE IS REPEATED A TOTAL OF .I2, 452
17H"TIMES,////")                                         HEAD 453
001757      GO TO 2350                                       HEAD 454
001760      2300 NCYC = 1                                    HEAD 455
001761      PERCEN = 0.0                                    HEAD 456
001762      WRITE (6,2306)                                   HEAD 457
001766      2306 FORMAT (I10,20X,50H"THE DESIGN CYCLE IS REPEATED ONLY ONCE. THE HAHEAD 458
1SIC/21X,52H"SECTION DETERMINED IN THE DESIGN CYCLE IS REDESIGNED/ READ 459
221X,48H"USING THE NEW DESIGN LOADINGS AND THE PROGRAM IS/ READ 460
321X,16H"THEN TERMINATED.////")                         READ 461
C      DETERMINE THE REQUIRED DEFLECTION LOCATIONS        READ 462
001766      2350 IF (IPT9.NE. 0) GO TO 2400                READ 463
001767      DO 2401 I = 1, MN                                READ 464
001771      ID = IOSP(I)                                    READ 465
001773      DT = COOR(ID) + SPAN(I) / 2.                  READ 466
001776      AS = 999999.                                     READ 467
002000      DO 2402 J = 1, NA                               READ 468
002001      IF (ABS(COOR(J) - DT) .GT. AS) GO TO 2402     READ 469
002007      IA = J                                           READ 470
002007      AS = ABS(COOR(J) - DT)                         READ 471
002011      2402 CONTINUE                                  READ 472
002014      IDEF(I) = IA                                    READ 473
002016      2401 CONTINUE                                  READ 474
002020      NDEF = MN                                       READ 475
002021      GO TO 2450                                       READ 476
002021      2400 IF (IPT9.NE. 1) GO TO 2460               READ 477
002023      IR = 1                                           READ 478
002024      DO 2410 I = 1, MN                               READ 479
002026      D = .2                                           READ 480
002030      ID = IOSP(I)                                    READ 481
002031      2425 DT = COOR(ID) + D * SPAN(I)              READ 482
002035      AS = 999999.                                     READ 483
002037      DO 2411 J = 1, NA                               READ 484
002040      IF (ABS(COOR(J) - DT) .GT. AS) GO TO 2411     READ 485
002046      IA = J                                           READ 486
002046      AS = ABS(COOR(J) - DT)                         READ 487
002050      2411 CONTINUE                                  READ 488
002053      IDEF(IA) = IA                                    READ 489
002055      IR = IR + 1                                       READ 490
002056      IF (D .LT. .4) GO TO 2421                     READ 491
002061      IF (D .LT. .6) GO TO 2422                     READ 492
002063      GO TO 2410                                       READ 493
002064      2421 D = .5                                       READ 494
002066      GO TO 2425                                       READ 495
002066      2422 D = .8                                       READ 496
002070      GO TO 2425                                       READ 497
002070      2410 CONTINUE                                  READ 498

```

```

002073      NDEF = 18 - 1                      READ 499
002075      GO TO 2450                          READ 500
002075      2460 READ (5,2465) NDEF, (IDFF(I), I = 1, NDEF) READ 501
002112      2465 FORMAT (11U, 20I3)          READ 502
002112      C      PRINT THE REQUIRED DEFLECTION LOCATIONS READ 503
002112      2450 WRITE (6,2455) (IDFF(I), I = 1, NDEF) READ 504
002125      2455 FORMAT (1H0,20X,50HDEFLECTIONS ARE FOUND AT THE FOLLOWING POINTS., READ 505
002125      1,...//24X,20I3)                READ 506
002125      RETURN                          READ 507
002126      EN                                READ 508

```

SUBPROGRAM LENGTH
003672

UNUSED COMPILER SPACE
031000

```

SUBROUTINE SEMIC                                SFMC 1
C      SEMIC SETS THE CONDITIONS FOR USING ICALC TO SOLVE FOR I VALUES SFMC 2
000002      COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FTH(18),WEHTH(18), SFMC 3
      IDFF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80), SFMC 4
      2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF, SFMC 5
      3SLABWD,SLATH,NR SFMC 6
000002      COMMON/ONE/JPT3,IPT10 SFMC 7
000002      COMMON/TWO/LPCM SFMC 8
000002      COMMON/THREE/HALINCH SFMC 9
000002      DIMENSION LPCM(10) SFMC 10
000002      REAL IX, IY, ISTEEL, ILCON, IHCON, MR SFMC 11
000002      INTEGER NAME SFMC 12
C      SET INITIAL ZERO VALUES SFMC 13
000002      ISA = 0. SFMC 14
000003      H = 0. SFMC 15
000004      WDIF = 0. SFMC 16
000005      THDIF = 0. SFMC 17
000006      IF (JPT3 .LT. 9) ISWID1 = SLABWD / (3. * MR) SFMC 18
000013      IF (JPT3 .LT. 9) ISWID2 = SLABWD / MR SFMC 19
C      START ITERATION LOOP FOR EACH SHELEMENT SFMC 20
000017      DO 25 I = 1, NF SFMC 21
C      CHECK PREVIOUS SECTION FOR SIMILARITY TO THE CURRENT ONE SFMC 23
000021      IF (I .EQ. 1) GO TO 22 SFMC 22
000023      K = 1 - 1 SFMC 24
C      CHECK FOR SIMILAR STEEL SECTIONS SFMC 25
000024      IF (NOS(1) .NE. NOS(K)) GO TO 22 SFMC 26
C      CHECK FOR SIMILAR TOP AND BOTTOM COVER PLATES SFMC 27
000026      DO 26 KK = 1, 2 SFMC 28
000027      THIF = ABS(PLATH(I,KK) - PLATH(K,KK)) SFMC 29
000036      IF (THIF .GT. .00001) GO TO 22 SFMC 30
000042      WDIF = ABS(PLWID(I,KK) - PLWID(K,KK)) SFMC 31
000047      IF (WDIF .GT. .00001) GO TO 22 SFMC 32
000052      26 CONTINUE SFMC 33
C      SET I VALUES TO THOSE OF THE SIMILAR SECTION SFMC 34
000054      ISTEEL(I) = ISTEEL(K) SFMC 35
000056      CS(I) = CS(K) SFMC 36

```

```

000060      ILCON(I) = ILCON(K)                      SFMC 37
000061      CL(I) = CL(K)                            SFMC 38
000063      INCON(I) = INCON(K)                      SFMC 39
000064      CH(I) = CH(K)                            SFMC 40
000066      GO TO 25                                  SFMC 41
000066 22 CONTINUE                                SFMC 42
C      CALCULATES I VALUES WHEN THERE IS NO SIMILAR SECTION SFMC 43
000066 20 J = NOS(I)                              SFMC 44
000070      HP = (DEPTH(J)/2. + PLATH(I,2)         SFMC 45
000073      CALL ICALC(PLWID(I,1),PLATH(I,1),PLWID(I,2),PLATH(I,2),DEPTH(J), SFMC 46
      ISAREA(J),IA(I),HP,H,CS(I),ISTEEL(I))      SFMC 47
000114      D = 0.                                SFMC 48
000115      IF (UPTI .GE. 10) GO TO 225             SFMC 49
000120      C = 0.                                SFMC 50
000121      TSA = SAREA(J)*PLWID(I,1)*PLATH(I,1)+PLWID(I,2)*PLATH(I,2) SFMC 51
000127      H = 2. * (DEPTH(I) + PLATH(I,2) - CS(I)) SFMC 52
000134      CALL ICALC(ISWID,SLATH,D,C,H,TSA,ISTEEL(I),CS(I),HAUNCH,CL(I), SFMC 53
      ILCON(I))                                  SFMC 54
000156      CALL ICALC(ISWID,SLATH,D,C,H,TSA,ISTEEL(I),CS(I),HAUNCH,CH(I), SFMC 55
      INCON(I))                                  SFMC 56
000201      GO TO 25                                  SFMC 57
000202 225 ILCON(I) = ISTEEL(I)                   SFMC 58
000204      INCON(I) = ISTEEL(I)                   SFMC 59
000206      CH(I) = CS(I)                          SFMC 60
000207      CL(I) = CS(I)                          SFMC 61
000211 25 CONTINUE                                SFMC 62
C      CORRECT FOR THE COMPOSITE ACTION SFMC 63
000214      IF (UPTI .GE. 10) GO TO 30             SFMC 64
000216      DO 41 I = 1, 10, 2                     SFMC 65
000220      IA = LPCM(I)                          SFMC 66
000222      IH = LPCM(I+1) - 1                   SFMC 67
000224      IF (IA .EQ. 0) GO TO 30               SFMC 68
000225      DO 41 J = IA, IH                       SFMC 69
000226      ILCON(J) = ISTEEL(J)                 SFMC 70
000230      INCON(J) = ISTEEL(J)                 SFMC 71
000231      CH(J) = CS(J)                         SFMC 72
000233      CL(J) = CS(J)                       SFMC 73
000234 41 CONTINUE                                SFMC 74
000240 30 CONTINUE                                SFMC 75
000240      RETURN                                SFMC 76
000241      END                                    SFMC 77

```

SUBPROGRAM LENGTH

000320

UNUSED COMPILER SPACE

042000

```

      SUBROUTINE SOLVE (N,A,P,NA)                SOLV 1
C      SOLVES THE MATRIX EQUATION FOR THE REACTION INFLUENCE LINES SOLV 2
      DIMENSION A(5,5), P(5,5), NA(5,2)       SOLV 3
      201 FORMAT (// 17H SINGULAR MATRIX //)    SOLV 4
      202 FORMAT (2113)                          SOLV 5

```

C	GAUSS ELIMINATION TO UNIT MATRIX WITH PIVOTING. A IS THE	SOLV	6
C	COEFFICIENT MATRIX SIZE NS BY NS. P IS THE RIGHT HAND SIDE	SOLV	7
C	MATRIX WITH NA COLUMNS AND NS ROWS. EVERYTHING IS SCRAMBLED	SOLV	8
C	IN THE A MATRIX AND THE RESULTS ARE ON TOP OF P. IF A IS	SOLV	9
C	SINGULAR THE PROGRAM STOPS.	SOLV	10
C	KEEP OUT THE REQUIRED VARIABLES	SOLV	11
000010	DO 203 I = 1, N	SOLV	12
000011	DO 203 J = 1, 2	SOLV	13
000012	203 MAVE (I,J) = 0	SOLV	14
000021	IS = 0	SOLV	15
000022	IS = 0	SOLV	16
000023	NMAX = 0	SOLV	17
000024	MMAX = 0	SOLV	18
000025	HITLF = 0.0	SOLV	19
000026	INDEX = 0	SOLV	20
C	BEGIN THE SOLUTION LOOP	SOLV	21
000027	DO 206 I = 1, N	SOLV	22
000030	IS = A(I,1)	SOLV	23
000033	NMAX = I	SOLV	24
000034	MMAX = I	SOLV	25
C	DETERMINE THE LARGEST ELEMENT	SOLV	26
000036	DO 210 J = 1, N	SOLV	27
000036	DO 210 M = 1, N	SOLV	28
000040	IF (ABS(A(J,M)) .LT. IS) GO TO 210	SOLV	29
000045	IS = ABS(A(J,M))	SOLV	30
000050	NMAX = J	SOLV	31
000051	MMAX = M	SOLV	32
000052	210 CONTINUE	SOLV	33
000057	IF (NMAX .EQ. 1) GO TO 220	SOLV	34
C	DO THE SIMPLE ROW AND COLUMN OPERATIONS	SOLV	35
000061	DO 215 K = 1, N	SOLV	36
000063	IS = A(NMAX,K)	SOLV	37
000066	A(NMAX,K) = A(I,K)	SOLV	38
000073	215 A(I,K) = IS	SOLV	39
000100	DO 219 K = 1, L	SOLV	40
000101	IS = P(NMAX,K)	SOLV	41
000104	P(NMAX,K) = P(I,K)	SOLV	42
000111	219 P(I,K) = IS	SOLV	43
000116	220 IF (NMAX .EQ. 1) GO TO 230	SOLV	44
000120	DO 225 K = 1, N	SOLV	45
000122	IS = A(K,I)	SOLV	46
000126	A(K,1) = A(K,MMAX)	SOLV	47
000132	225 A(K,MMAX) = IS	SOLV	48
000137	INDEX = INDEX + 1	SOLV	49
000140	MAVE (INDEX,1) = MMAX	SOLV	50
000142	MAVE (INDEX,2) = 1	SOLV	51
C	DETERMINE THE SINGULARITY CONDITION	SOLV	52
000144	230 IF (ABS(A(I,1)) .GT. HITLF) GO TO 240	SOLV	53
C	EXIT IF A SINGULAR MATRIX IS FOUND	SOLV	54
000153	WRITE (6,201)	SOLV	55
000156	WRITE (6,202) 1, INDEX	SOLV	56
000165	CALL EXIT	SOLV	57
C	CONTINUE THE ROW AND COLUMN OPERATIONS	SOLV	58
000167	240 J = N + 1	SOLV	59
000174	IF (1 .EQ. N) GO TO 243	SOLV	60
000175	241 IF (J .EQ. 1+1) GO TO 243	SOLV	61
000200	J = J + 1	SOLV	62
000201	IS = A(I,J)/A(I,1)	SOLV	63

000206	A(I,J) = TS	SOLV	64
000211	GO TO 241	SOLV	65
000211	243 DO 244 J = 1, L	SOLV	66
000213	TS = P(I,J)/A(I,I)	SOLV	67
000221	244 P(I,J) = TS	SOLV	68
000226	245 J = J + 1	SOLV	69
000230	IF (J .EQ. N) GO TO 257	SOLV	70
000231	K = 0	SOLV	71
000232	246 K = K + 1	SOLV	72
000234	TI = A(K,I)	SOLV	73
000240	J = J + 1	SOLV	74
000241	IF (J .EQ. N) GO TO 252	SOLV	75
000242	247 IF (J .EQ. I+1) GO TO 252	SOLV	76
000245	J = J - 1	SOLV	77
000246	TI = A(K,I)	SOLV	78
000251	TU = A(I,J)	SOLV	79
000254	TV = A(K,J)	SOLV	80
000257	TS = TV - TI * TU	SOLV	81
000262	A(K,J) = TS	SOLV	82
000265	GO TO 250	SOLV	83
000266	252 DO 255 J = 1, L	SOLV	84
000270	TS = P(K,J) - A(K,I) * P(I,J)	SOLV	85
000301	255 P(K,J) = TS	SOLV	86
000306	IF (K .EQ. I0) GO TO 257	SOLV	87
000310	GO TO 245	SOLV	88
000310	257 I0 = I + 1	SOLV	89
000312	IF (I0 .EQ. N+1) GO TO 266	SOLV	90
000314	DO 265 K = I0, N	SOLV	91
000315	TI = A(K,I)	SOLV	92
000321	TS = I + 1	SOLV	93
000323	IF (TS .EQ. N+1) GO TO 262	SOLV	94
000325	J = K + 1	SOLV	95
000326	IF (J .EQ. N) GO TO 262	SOLV	96
000327	260 IF (J .EQ. I+1) GO TO 262	SOLV	97
000332	J = J - 1	SOLV	98
000333	TI = A(K,I)	SOLV	99
000336	TU = A(I,J)	SOLV	100
000341	TV = A(K,J)	SOLV	101
000344	TS = TV - TI * TU	SOLV	102
000347	A(K,J) = TS	SOLV	103
000352	GO TO 260	SOLV	104
000353	262 DO 265 J = 1, L	SOLV	105
000355	TS = P(K,J) - A(K,I) * P(I,J)	SOLV	106
000366	265 P(K,J) = TS	SOLV	107
000375	266 CONTINUE	SOLV	108
000400	IF (INDEX .EQ. 0) GO TO 280	SOLV	109
000401	C ADJUST COLUMNS BACK TO THEIR ORIGINAL ORDER	SOLV	110
000401	I = INDEX + 1	SOLV	111
000402	270 IF (I .EQ. 1) GO TO 280	SOLV	112
000404	I = I - 1	SOLV	113
000405	J = N+1(I,2)	SOLV	114
000407	M = N+1(I,1)	SOLV	115
000410	DO 275 K = 1, L	SOLV	116
000412	TS = A(I,K)	SOLV	117
000415	P(J,K) = TS	SOLV	118
000422	275 P(M,K) = TS	SOLV	119
000427	GO TO 270	SOLV	120
000427	280 RETURN	SOLV	121

000430 E P

SLV 122

SUPERPROGRAM LENGTH

000507

UNUSED COMPILER SPACE

041300

	SUBROUTINE SORT (PAR,N)	SORT	1
000005	DIMENSION PAR(40), IO(40), P(40)	SORT	2
000005	DO 1 I=1, N	SORT	3
000005	1 IO(I) = I	SORT	4
000011	NSIDE = N - 1	SORT	5
000013	DO 2 K = 1, NSIDE	SORT	6
000014	J = N - K	SORT	7
000015	IS4 = 1	SORT	8
000016	DO 3 I = 1, J	SORT	9
000020	IOI = IO(I)	SORT	10
000021	IOJ = IO(I+1)	SORT	11
000023	IF (PAR(IOI) .LE. PAR(IOJ)) GO TO 3	SORT	12
000030	II = IOI	SORT	13
000030	IOI = IOJ	SORT	14
000032	IO(I+1) = II	SORT	15
000034	IS4 = 2	SORT	16
000035	3 CONTINUE	SORT	17
000040	IF (IS4 .EQ. 1) GO TO 10	SORT	18
000042	2 CONTINUE	SORT	19
000044	10 DO 11 I = 1, N	SORT	20
000046	PA = IO(I)	SORT	21
000047	11 P(I) = PAR(PA)	SORT	22
000054	DO 12 I = 1, N	SORT	23
000055	12 PAR(I) = P(I)	SORT	24
000061	RETURN	SORT	25
000061	END	SORT	26

SUPERPROGRAM LENGTH

000226

UNUSED COMPILER SPACE

042700

	SUBROUTINE STIFF (ISD,NS,DESPA,POS,FLTH,FLWID,FBTH,DEPTH,PROP,	STIF	1
	LEN,REAR)	STIF	2
	C DIMENSION THE READING STIFFENERS WHERE DEFINED	STIF	3
000016	(IMENSION ISD(4),DESPA(5,4),RL(4),REAR(5,3),NOS(40),FLTH(18),	STIF	4
	FLWID(18),WEATH(18),DEPTH(18),PROP(3))	STIF	5
000016	DO 420 I = 1, NS	STIF	6
000017	DO 420 J = 1, 3	STIF	7


```

000020      920 BEAR(I,J) = 0.0                                STIF 8
C          SET UP THE BEAR FOR EACH SUPPORT                STIF 9
000030      DO 400 I = 1, NS                                  STIF 10
C          CHECK TO SEE IF A STIFFENER IS REQUIRED          STIF 11
000031      DO 401 K = 1, 4                                    STIF 12
000032      901 RE(K) = DESRFA(I,K)                          STIF 13
000042      TMX = SIRMUX(RE)                                  STIF 14
000045      IR = IOSP(I)                                       STIF 15
000053      IA = NOS(IR)                                       STIF 16
000055      IF (I.EQ. NS) IA = NOS(IR - 1)                   STIF 17
000060      SS = TMX/(WEITH(IA) * (DEPTH(IA) - 2.0 * FLTH(I))) STIF 18
000071      IF (ABS(SS) .LT. .75*PROP(2)) GO TO 902          STIF 19
C          DESIGN OF THE BEARING STIFFENER                STIF 20
C          DETERMINE THE WIDTH OF THE PLATE                 STIF 21
000076      WIDMAX = (FLWID(IA) + WEITH(IA))/2.0           STIF 22
000103      WID = 4.0                                          STIF 23
000104      905 IF ((WID+.25) .GT. WIDMAX) GO TO 904         STIF 24
000111      WID = WID + .25                                    STIF 25
000112      GO TO 905                                         STIF 26
C          DETERMINE THE MINIMUM THICKNESS                 STIF 27
000112      904 TMN = SORT(PROP(3)/33.1) * WID / 12.       STIF 28
000121      THICK = .25                                       STIF 29
000122      907 IF (THICK .GE. TMN) GO TO 906               STIF 30
000130      THICK = THICK + .0625                             STIF 31
000132      GO TO 907                                         STIF 32
C          DETERMINE AND CHECK THE BEARING STRESS         STIF 33
000132      906 BS = TMX/((WID-1.25*WEITH(IA)/2.)*THICK*.0) STIF 34
000143      IF (BS .LE. .8*PROP(3)) GO TO 908               STIF 35
000147      THICK = THICK + .0625                             STIF 36
000151      GO TO 906                                         STIF 37
C          DETERMINE AND CHECK THE COLUMN STRESS          STIF 38
000151      908 COLA = 18.0 * WEITH(IA)**2 + WID * THICK * 2.0 STIF 39
000160      COLI = 1.5*WEITH(IA)**4 + (THICK*WID**3)/6. +   STIF 40
      12.0*WID*THICK*(WID/2. + WEITH(IA)/2. )**2           STIF 41
000175      COLR = SORT(COLI/COLA)                           STIF 42
000201      SL = (DEPTH(IA) - 2. * FLTH(IA)) * .75/COLR   STIF 43
000213      FA = PROP(1)*(1.-SL*SL*PROP(3)/(39.47835*FM))/1.25 STIF 44
000223      CSIR = TMX/COLA                                    STIF 45
000225      IF (CSIR .LE. FA) GO TO 909                     STIF 46
000227      THICK = THICK + .0625                             STIF 47
000231      GO TO 906                                         STIF 48
C          SET UP THE BEAR ARRAY                            STIF 49
000232      909 BEAR(I,1) = WID                              STIF 50
000235      BEAR(I,2) = THICK                                STIF 51
C          DESIGN THE CONNECTING WELDS                     STIF 52
000237      WL = 4. * (DEPTH(IA) - 2. * FLTH(IA))          STIF 53
C          DETERMINE THE ALLOWABLE WELD METAL STRESS     STIF 54
000246      EV = 12.4                                         STIF 55
000247      IF (ABS(PROP(3) - 36.0) .GT. 1.45) EV = 14.5    STIF 56
000256      SSX = 1.0 + .36 * (EV/ER - 1.0)                  STIF 57
000263      RP = DESRFA(I,1) + DESRFA(I,2) + DESRFA(I,3)    STIF 58
000270      RN = DESRFA(I,1) + DESRFA(I,3) + DESRFA(I,4)     STIF 59
000275      RATIO = RN / RP                                     STIF 60
000277      IF (ABS(RN) .GT. RP) RATIO = RP / RN             STIF 61
000302      FATS = SSX * 10.8 / (1.0 + .55 * RATIO)          STIF 62
000307      IF (FATS .LT. EV) EV = FATS                     STIF 63
C          CALCULATE THE WELD THICKNESS                   STIF 64
000312      TR = TMX/(.707 * EV * WL)                       STIF 65

```



```

000316      N1A = 1/2.0625
000321      BEAR(I,3) = N1A * .0625 * .0625
000325      IF (BEAR(I,3) .LT. .25) BEAR(I,3) = .25
000331      GO TO 900
C          SET UP THE BEAR ARRAY FOR NO STIFFENER
000332      902 BEAR(I,1) = 0.0
000335      BEAR(I,2) = 0.0
000336      BEAR(I,3) = 0.0
000337      900 CONTINUE
000342      RETURN
000342      END

```

STIF 66
STIF 67
STIF 68
STIF 69
STIF 70
STIF 71
STIF 72
STIF 73
STIF 74
STIF 75
STIF 76

SUBPROGRAM LENGTH
000477

UNUSED COMPILER SPACE
041500

```

C          SUBROUTINE STROCK
C          CALCULATES THE STRESSES AT EACH ANALYSIS POINT
000002      COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEHTH(18),
1DAF(18),IX(18),IY(18),NSFCT(18),NOS(80),TSTEEL(80),CS(80),
2ILCON(80),CL(80),THCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,
3SLABWD,SLABTH,MR,CORR(81),RIL(5,81,2),CC(5,81,FM),SPAN(4),TOSP(5),
4THICK(80),NA,LENGTH,PO(R1),SUBLEN(80),NS,UNLC(9),TRLO(10),PHOP(3),STRC
5FC,STRESS(81,4,2),DESMOM(R1,4),DESFREA(5,4),DESSH(162,4),BEAR(5,3),STRC
6COST(9),CONSP(16,3,4),COVPL(12,6,2)
000002      COMMON/ONE/JPT3,IPT10
000002      COMMON/TWO/LPCM
000002      COMMON/STX/FU
000002      COMMON/LEN/HAUNCH
000002      REAL IX,IY,TSTEEL,ILCON,THCON,MR,LENGTH
000002      DIMENSION LPCM(10),TMST(4)
000002      IF (JPT3 .LT. 9) TSW10 = SLABWD / (3. * MR)
000010      IF (JPT3 .LT. 9) TSW102 = SLABWD / MR
000014      DO 1800 I = 1, NA
000016      DO 1800 J = 1, 4
000017      DO 1800 K = 1, 2
000020      1800 STRESS(I,J,K) = 0.0
C          BEGIN THE LOOP TO CALCULATE THE STRESSES
000034      DO 800 I = 1, NA
000036      PLT1 = 0.0
000037      PLW1 = 0.0
000040      PLT4 = 0.0
000041      PLW4 = 0.0
C          DETERMINE THE PLATE THICKNESS AT EACH NODE
000042      DO 803 K = 1, 2
000043      DO 801 J = 1, 12
000044      IF (ABS(COVPL(J,1,K)) .LT. .0001 .AND. ABS(COVPL(J,2,K)) .LT. .0001)
1GO TO 803
000064      IF (CORR(1).GE.COVPL(J,1,K).AND.CORR(1).LE.COVPL(J,2,K)) GO TO 802
000077      GO TO 801
000077      802 IF (K .EQ. 2) GO TO 810

```

STRC 1
STRC 2
STRC 3
STRC 4
STRC 5
STRC 6
STRC 7
STRC 8
STRC 9
STRC 10
STRC 11
STRC 12
STRC 13
STRC 14
STRC 15
STRC 16
STRC 17
STRC 18
STRC 19
STRC 20
STRC 21
STRC 22
STRC 23
STRC 24
STRC 25
STRC 26
STRC 27
STRC 28
STRC 29
STRC 30
STRC 31
STRC 32
STRC 33
STRC 34
STRC 35

```

000101      PLT1 = COVPL(J,3,K)                                STCR 36
000105      PLW1 = COVPL(J,4,K)                                STCR 37
000110      GO TO R17                                           STCR 38
000110      H10 PLT4 = COVPL(J,3,K)                                STCR 39
000114      PLW4 = COVPL(J,4,K)                                STCR 40
000117      GO TO R03                                           STCR 41
000120      R01 CONTINUE                                         STCR 42
000122      R03 CONTINUE                                         STCR 43
C          CALCULATE THE SECTION PROPERTIES                     STCR 44
000124      IA = I                                               STCR 45
000126      IF (I .EQ. NA) GO TO R21                             STCR 46
000130      IF (ABS(PLT1-PLATH(IA,1)) .LT. .0001 .AND. ABS(PLTR-PLATH(IA,2)) STCR 47
1,LT.,.0001) GO TO R20                                         STCR 48
000146      IF (ABS(PLT1-PLATH(IA-1,1)) .LT. .0001 .AND. ABS(PLTR-PLATH(IA-1,2)) STCR 49
1,LT.,.0001) GO TO R21                                         STCR 50
000164      JJ = NOS(IA)                                          STCR 51
000166      HP = DEPTH(JJ)/2. * PLTR                               STCR 52
000171      CALL ICALC (PLT1,PLW1,PLTR,PLWB,DEPTH(JJ),SAREA(JJ),IX(JJ),HP,N., STCR 53
ISC,SIS)                                                       STCR 54
000205      IF (OPT1 .GT. 9) GO TO R30                           STCR 55
000211      IF (DESMON(1,1) .LT. 0.0) GO TO R30                 STCR 56
000213      H = 2. * (DEPTH(JJ) * PLTR - SC)                     STCR 57
000217      TSA = SAREA(JJ) * PLT1*PLW1 + PLTR*PLWB           STCR 58
000225      CALL ICALC (ISW1,SLATH,0.,0.,H,TSA,SIS,SC,HAUNCH,SI,SIS) STCR 59
000237      CALL ICALC (TSK12,SLATH,0.,0.,H,TSA,SIS,SC,HAUNCH,SH,S4S) STCR 60
000252      GO TO R50                                             STCR 61
000253      R21 IA = I - 1                                         STCR 62
000255      R20 SIS = ISTEEL(IA)                                   STCR 63
000257      SC = CS(IA)                                           STCR 64
000261      IF (OPT1 .GT. 9) GO TO R30                           STCR 65
000264      IF (DESMON(1,1) .LT. 0.0) GO TO R30                 STCR 66
000266      SLS = ICON(IA)                                         STCR 67
000270      SL = CL(IA)                                           STCR 68
000271      SHS = INCON(IA)                                       STCR 69
000273      SI = CH(IA)                                           STCR 70
000274      GO TO R50                                             STCR 71
000275      R30 SLS = SIS                                         STCR 72
000277      SL = SC                                               STCR 73
000300      SHS = SIS                                             STCR 74
000301      SH = SC                                               STCR 75
000302      R50 IS = NOS(IA)                                       STCR 76
000304      T010 = DEPTH(1S) * PLT1 + PLTH                      STCR 77
C          CALCULATE THE BOTTOM FIBER STRESS                     STCR 78
000307      TMS1(1) = DESMON(1,1) * SC / SIS                     STCR 79
000312      TMS1(2) = DESMON(1,2) * SH / SHS                     STCR 80
000315      TMS1(3) = DESMON(1,3) * SL / SLS                     STCR 81
000317      TMS1(4) = DESMON(1,4) * SH / SHS                     STCR 82
000322      T0S = STMAX(TMS1)                                       STCR 83
000324      STRESS(1,1,1) = T0S                                    STCR 84
C          DETERMINE THE ALLOWABLE BOTTOM FIBER STRESSES        STCR 85
000326      ASIR = ALLOW(DESMON(1,1),FLW10(1S),IPCM,TSP,I,S,PROP,COND, STCR 86
IDESMON,NBLEN,FM,1)                                             STCR 87
000343      IF (I .EQ. 1) GO TO 301                               STCR 88
000345      IF (I .EQ. NA) GO TO 303                               STCR 89
000347      AFS = H0SFAL(TMS1,F1,STRESS(1,1,1),PROP(2))          STCR 90
000353      IF (AFS .LT. ASIR) ASIR = AFS                          STCR 91
000356      301 STRESS(1,1,2) = 0.0                               STCR 92
000360      IF (ABS(STRESS(1,1,1)) .GT. ABS(ASIR)) STRESS(1,1,2) = 1.0 STCR 93

```


SUBPROGRAM LENGTH
000735

UNUSED COMPILER SPACE
040100

	FUNCTION STIRMAX (TSSS)	STIRMAX	1
	DETERMINES THE DESIGN STRESS AT A POINT GIVEN	STIRMAX	2
	THE COMPONENT DESIGN STRESSES FOR THAT POINT	STIRMAX	3
000003	DIMENSION TSSS(4)	STIRMAX	4
000003	TEMP = TSSS(1) + TSSS(3)	STIRMAX	5
000004	PAR1 = TEMP + TSSS(2)	STIRMAX	6
000006	PAR2 = TEMP + TSSS(4)	STIRMAX	7
000010	ANS = PAR1	STIRMAX	8
000012	IF (ABS(PAR2) .GT. ABS(ANS)) ANS = PAR2	STIRMAX	9
000017	STIRMAX = ANS	STIRMAX	10
000020	RETURN	STIRMAX	11
000021	END	STIRMAX	12

SUBPROGRAM LENGTH
000043

UNUSED COMPILER SPACE
043100

	SUBROUTINE TERMIN	TERMIN	1
	DETERMINE THE CUTOFF DISTANCE FOR A GIVEN PLATE	TERMIN	2
000002	COMMON NAME(18),SAREA(18),DEPTH(18),FLWD(18),FLTH(18),WEPTH(18),	TERMIN	3
	IDAF(18),IX(18),IY(18),NSECT(18),NOS(40),ISTEEL(40),CS(40),	TERMIN	4
	2ILCON(40),CL(40),IHCON(40),CH(40),PLATH(40,2),PLKID(40,2),NF,	TERMIN	5
	3SLABHD,SLABTH,MP,CORR(41),RTL(5,41,2),CC(5,51),F1,SPAN(4),IDSP(5),	TERMIN	6
	4TMDP(18),NA,(LENGTH,RO(41),SMBLEN(40),NS,IMMO(41),TQID(10),PROP(3),	TERMIN	7
	5FC,STRESS(81,4,2),DESMOM(41,4),DESRFA(5,4),DESSW(162,4),HFA(5,31),	TERMIN	8
	6COSI(9),CONSP(16,3,4),COMPL(12,6,2)	TERMIN	9
000002	COMMON/ONE/UP13,IP110	TERMIN	10
000002	COMMON/TWO/LPCM	TERMIN	11
000002	COMMON/SIX/FH	TERMIN	12
000002	COMMON/EIGHT/IST	TERMIN	13
000002	COMMON/TEN/HALINCH	TERMIN	14
000002	COMMON/ELEVEN/KD,KC,SIS,SHS,SLS,TSS,IJ,SS,DMAX,MTST	TERMIN	15
000002	COMMON/TEEN/PTH,PT,ITT	TERMIN	16
000002	REAL IX,IY,ISTEEL,ILCON,IHCON,MP,LENGTH,RTLT	TERMIN	17
000002	DIMENSION LPCM(10),TMST(4),PTH(40,4)	TERMIN	18
	INITIALIZE THE REQUIRED VARIABLES	TERMIN	19
000002	MN = NS = 1	TERMIN	20
000004	ITN = 0.0	TERMIN	21
000005	KLL = 0	TERMIN	22
000006	KTI = 0	TERMIN	23
000007	DP = 0.0	TERMIN	24

```

000010      AV = 9999999.
000011      IF (TERMINE THE MOMENT AND STRESS AT THE POINT
000012      DO 5601 I = 2, NM
000013      IF (IOSP(I) .EQ. 0) GO TO 5602
000014      IR = IOSP(I)
000015      IF (ABS(COOR(IR) - COOR(IA)) .GT. AV) GO TO 5601
000016      IA = IOSP(I)
000017      IF (KQ .LT. IA) IA = IA - 1
000018      AV = ABS(COOR(IR) - COOR(IA))
000019      5601 CONTINUE
000020      5602 CONTINUE
000021      IF (NM .EQ. 1) IA = KQ
000022      T1 = 0.0
000023      DO 5603 K = 1, 4
000024      5603 IMST(K) = DESMOM(KQ, K)
000025      GO TO 5604
000026      5604 DO 5605 K = 1, 4
000027      DO 5606 KK = 1, NA
000028      5606 RO(KK) = DESMOM(KK, K)
000029      5607 IMST(K) = ILINT(KQ, RO, COOR, ISW, DP, LENGTH, SURFEN)
000030      5608 CONTINUE
000031      DMS = IMST(1)
000032      IF (DMS .LT. 0.0) GO TO 5700
000033      IMST(1) = IMST(1) / SIS
000034      IMST(2) = IMST(2) / SHS
000035      IMST(3) = IMST(3) / SLS
000036      IMST(4) = IMST(4) / SHS
000037      GO TO 5701
000038      5700 IMST(1) = IMST(1) / SS
000039      IMST(2) = IMST(2) / SS
000040      IMST(3) = IMST(3) / SS
000041      IMST(4) = IMST(4) / SS
000042      5701 TDS = STMAX(IMST)
000043      C      DETERMINE THE ALLOWABLE STRESSES
000044      ASTRT = PROP(1)
000045      ASFRB = ALLOW(DMS, FLUID(TST), LPCM, IOSP, IA, NS, PROP, COOR, DESMOM,
000046      ISURLEN, FM, 2)
000047      SMIN = IMST(1) + IMST(2) + IMST(3)
000048      SMAX = IMST(1) + IMST(3) + IMST(4)
000049      IF (ABS(SMIN) .GT. .001 .AND. ABS(SMAX) .GT. .0001) GO TO 5100
000050      R = 0.0
000051      GO TO 5110
000052      5100 RATIO = SMIN / SMAX
000053      IF (ABS(SMAX) .LT. ABS(SMIN)) RATIO = SMAX / SMIN
000054      5110 SK = 1.0 + .25 * (FU/FR - 1.0)
000055      IF (KQ .EQ. 2) GO TO 5060
000056      ASFI = 12. / (1.0 - RATIO)
000057      ASFB = ASFI
000058      GO TO 5061
000059      5060 ASFI = .55 * PROP(3) / (1. - (.55 * PROP(3) / (SK * 10.6) - 1.) * RATIO)
000060      ASFB = SK * 17.2 / (1. - .62 * RATIO)
000061      IF (DMS .LT. 0.0) GO TO 5061
000062      TFMF = ASFI
000063      ASFI = ASFB
000064      ASFB = TFMF
000065      5061 IF (IJ .EQ. 2) GO TO 6000
000066      AS = ASFI
000067      IF (ASFI .LT. ASTRT) AS = ASFI

```

```

TERM 25
TERM 26
TERM 27
TERM 28
TERM 29
TERM 30
TERM 31
TERM 32
TERM 33
TERM 34
TERM 35
TERM 36
TERM 37
TERM 38
TERM 39
TERM 40
TERM 41
TERM 42
TERM 43
TERM 44
TERM 45
TERM 46
TERM 47
TERM 48
TERM 49
TERM 50
TERM 51
TERM 52
TERM 53
TERM 54
TERM 55
TERM 56
TERM 57
TERM 58
TERM 59
TERM 60
TERM 61
TERM 62
TERM 63
TERM 64
TERM 65
TERM 66
TERM 67
TERM 68
TERM 69
TERM 70
TERM 71
TERM 72
TERM 73
TERM 74
TERM 75
TERM 76
TERM 77
TERM 78
TERM 79
TERM 80
TERM 81
TERM 82

```

000237	GO TO 6001	IFRM 83
000240	6000 AS = ASER	IFRM 84
000242	IF (ASER .LT. ASTER) AS = ASER	IFRM 85
	C START CUTTING OFF THE PLATE BY USING AN INCREASING	IFRM 86
	C STEP SIZE OPTIMIZATION PROCEDURE	IFRM 87
000245	6001 IF (ABS(UP) .LT. .00001) GO TO 5056	IFRM 88
000252	IF (ABS(DS) .LT. AS) GO TO 5057	IFRM 89
000255	GO TO 5056	IFRM 90
000256	5057 ISW = ISS	IFRM 91
000257	GO TO 5059	IFRM 92
000257	5058 ISW = 1	IFRM 93
000260	KLL = 1	IFRM 94
000261	IF (ISS .EQ. 1) ISW = 2	IFRM 95
000264	GO TO 5059	IFRM 96
000265	5056 IF (ABS(DS) .GT. AS .AND. KLL .EQ. 0) GO TO 5059	IFRM 97
000276	IF (ABS(DS) .GT. AS .AND. KLL .EQ. 1) GO TO 5059	IFRM 98
000307	GO TO 5057	IFRM 99
000307	5059 HL = 0.	IFRM 100
000310	YS = 1.	IFRM 101
000312	IF (ISW .EQ. 2) YK = -1.0	IFRM 102
000315	VDS = 4/999.	IFRM 103
000317	V = .5	IFRM 104
000320	6000 IF (ABS(OP-DMAX) .LT. .00001) GO TO 6000	IFRM 105
000325	5100 V = V * 2.	IFRM 106
000327	IT = OP	IFRM 107
000330	OP = HL * V	IFRM 108
000332	IF (OP .GT. DMAX) OP = DMAX	IFRM 109
000335	IF (ABS(OP - VDS) .LT. .00001) GO TO 5057	IFRM 110
000341	GO TO 5059	IFRM 111
000342	5052 IF (V .LT. VDS) VDS = OP	IFRM 112
000345	IF (ABS(OP - VDS) .LT. 1.5) GO TO 5055	IFRM 113
000352	HL = IT	IFRM 114
000353	V = .5	IFRM 115
000354	GO TO 5100	IFRM 116
000355	6000 HL = OP	IFRM 117
000357	5055 DIST = YK * H	IFRM 118
000361	IF (ABS(ABS(DIST)-DMAX) .GT. 1) GO TO 6500	IFRM 119
000370	IF (IT .EQ. 1) GO TO 6500	IFRM 120
000372	IF (ABS(DS) .LT. AS) GO TO 6500	IFRM 121
000375	EL = OP	IFRM 122
	C ADJUST DMAX IF REQUIRED	IFRM 123
000377	ITN = ITN + 1.0	IFRM 124
000401	DMAX = DMAX * DMAX/ITN	IFRM 125
000403	IF (IT .EQ. 1) GO TO 7000	IFRM 126
000405	IF (DMAX .GT. (COORD(KO)-PTH(IT+1.3)) GO TO 7001	IFRM 127
000413	GO TO 7002	IFRM 128
000413	7001 DMAX = COORD(KO) - PTH(IT+1.3)	IFRM 129
000416	KIT = 1	IFRM 130
000417	GO TO 7002	IFRM 131
000420	7000 IF (DMAX .GT. PTH(IT+1.4)-COORD(KO)) GO TO 7003	IFRM 132
000426	GO TO 7002	IFRM 133
000426	7003 DMAX = PTH(IT+1.4) - COORD(KO)	IFRM 134
000431	KIT = 1	IFRM 135
000432	7002 IF (PTH .EQ. 4) WRITE (6,1) DMAX	IFRM 136
000442	1 EXHAUST (13X,7HDMAX = ,F12.5)	IFRM 137
000442	V = .5	IFRM 138
000444	GO TO 5100	IFRM 139
000444	6500 CONTINUE	IFRM 140

```

000444      REFIN
000445      END

```

```

TERM 141
TERM 142

```

```

SUMPROGRAM LENGTH
000567

```

```

UNUSED COMPILER SPACE
040400

```

```

SUBROUTINE THRES
C      DETERMINE THE REQUIRED THICKNESSES IN EACH SHELEMENT
000002      COMMON NAME(18),SAREA(18),DEPTH(18),FLWD(18),FLTH(18),WCHTH(18),
      IDAF(18),IA(18),TY(18),INSECT(18),NDS(80),ISTEEL(80),CS(80),
      2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,
      3SLAWD,SLABTH,MR,CORR(R1),RT(15,81,3),CORF(15,81,3),SPAN(4),TOSP(5),
      4THI(18),NA,LENGTH,RD(8),SHELEM(80),NS,UNFA(9),TPO(10),PROP(3),THDE
      5FCS(81),RENS(81,4,2),DESMOM(R1,4),DESDFA(S,4),DESSM(162,4),REFR(5,3),THDE
      6COSI(9),CONSP(16,3,4),COMPL(12,6,2)
      7      COMMON/UNIT/UNIT,IP10
      8      COMMON/TAO/LPCW
      9      COMMON/SLX/SLX
      10      COMMON/THI/THI
      11      COMMON/PLN/PLN
      12      COMMON/IFN/HATCH
      13      REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH
      14      DIMENSION IMST(4),LPCW(10)
      15      C      ZERO OUT THE DESIGN ARRAYS
      16      DO 4600 I = 1, NF
      17      DO 4600 J = 1, 2
      18      PLATH(I,J) = 0.0
      19      PLWID(I,J) = 0.0
      20      IF (JPT3 .LT. 9) ISWID1 = SLAWD / (3. * MR)
      21      IF (JPT3 .LT. 9) ISWID2 = SLAWD / MR
      22      PLW = 1.0
      23      IF (FLWD(IST) .GT. 13.0) PLW = 14.0
      24      IF (FLWD(IST) .GT. 16.0) PLW = 15.0
      25      C      BEGIN THE LOOP FOR EACH ELEMENT
      26      DO 4900 I = 1, NF
      27      C      DETERMINE CRITICAL MOMENT LOCATION IN THE ELEMENT
      28      ISW1 = 0
      29      IF (DESMOM(I,1) .LT. 0.0 .OR. DESMOM(I+1,1) .LT. 0.0) GO TO 4500
      30      KEY = 0
      31      T1 = DESMOM(I,1) * DESMOM(I,2) * DESMOM(I,3)
      32      T2 = DESMOM(I+1,1) * DESMOM(I+1,2) * DESMOM(I+1,3)
      33      GO TO 4501
      34      4500 KEY = 1
      35      IF (DESDFA(I,1) .GT. 0.0 .OR. DESDFA(I+1,1) .GT. 0.0) GO TO 4502
      36      T1 = DESDFA(I,1) * DESDFA(I,3) * DESDFA(I,4)
      37      T2 = DESDFA(I+1,1) * DESDFA(I+1,3) * DESDFA(I+1,4)
      38      4501 IA = I
      39      IF (ABS(T1) .LT. ABS(T2)) IA = I + 1
      40      GO TO 4503
      41      4502 IA = I
      42      4503 IA = I
      43      4504 IA = I
      44

```



```

000120      IF (DESMOM(IA,1) .LT. 0.0) TA = 1 + 1      THDE 45
      C      START THE PLATE DETERMINATION      THDE 46
000123      R513 CONTINUE      THDE 47
000123      R511 KP = 0      THDE 48
000124      IF (DESMOM(IA,1) .LT. 0.0) DSMN = DESMOM(IA,1) + DESMOM(IA,3) +      THDE 49
      10ESMOM(IA,7)      THDE 50
      C      START WITH NO PLATES      THDE 51
000131      PLTT = 0.0      THDE 52
000132      PLTR = 0.0      THDE 53
      C      DETERMINE THE SECTION PROPERTIES      THDE 54
000133      R545 HP = DEPTH(IST)/2. + PLTR      THDE 55
000137      CALL ICALC (PLW,PLTT,PLA,PLTR,DEPTH(IST),SAREA(IST),TX(IST),HP,      THDE 56
      10,0,SECC,SECI)      THDE 57
000153      IF (DESMOM(IA,1) .LT. 0.0) GO TO R540      THDE 58
000155      IF (JPT3 .GT. 9) GO TO 700      THDE 59
000161      TSA = SAREA(IST) + PLW * (PLTT + PLTR)      THDE 60
000165      H = 2. * (DEPTH(IST) + PLTR - SECC)      THDE 61
000171      CALL ICALC (TSWID,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECL,SECIL)      THDE 62
000204      CALL ICALC (TSWID,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECH,SECIL)      THDE 63
000217      TD = DEPTH(IST) + PLTR + PLTT      THDE 64
000223      TDI = TD - PLTT + SLABTH + HAUNCH      THDE 65
000226      GO TO 701      THDE 66
000227      700 SECH = SECC      THDE 67
000231      SECL = SECC      THDE 68
000232      SECIL = SECI      THDE 69
000233      SECIL = SECI      THDE 70
000234      701 CONTINUE      THDE 71
      C      DETERMINE THE BOTTOM STRESSES IN POSITIVE SECTION      THDE 72
000234      TMS1(1) = DESMOM(IA,1) * SECC / SECI      THDE 73
000237      TMS1(2) = DESMOM(IA,2) * SECH / SECIH      THDE 74
000242      TMS1(3) = DESMOM(IA,3) * SECI / SECIL      THDE 75
000244      TMS1(4) = DESMOM(IA,4) * SECH / SECIL      THDE 76
      C      DETERMINE THE ALLOWABLE STRESS      THDE 77
000247      AFS = BASFAT(TMS1,FIL,1000.,PROP(3))      THDE 78
000253      ASTRH = PROP(1)      THDE 79
000254      IF (ABS(AFS) .LT. ASTRH) ASTRH = ABS(AFS)      THDE 80
000261      TDS = STREMAX(TMS1)      THDE 81
000264      IF (ABS(TDS) .LT. ABS(ASTRH)) GO TO R541      THDE 82
000270      GO TO R542      THDE 83
      C      DETERMINE THE BOTTOM STRESS IN NEGATIVE REGION      THDE 84
000271      R540 EFS = DSMN * SECC / SECI      THDE 85
000274      DJ4661 KI = 1.4      THDE 86
000275      R661 TMS1(KI) = DESMOM(IA,KI)      THDE 87
      C      DETERMINE THE ALLOWABLE STRESS      THDE 88
000304      AFS4 = BASFAT(TMS1,FIL,1000.,PROP(3))      THDE 89
000310      ASTRB = ALLOW(DESMOM(IA,1),FILWID(IST),LPCM,10SP,TA,NS,PROP,COORD,      THDE 90
      10DESMOM,SUHLN,EM,1)      THDE 91
000325      IF (ABS(AFS4) .LT. ASTRA) ASTRB = ABS(AFS4)      THDE 92
000331      AFS1 = BASFAT(TMS1,FIL,1000.,PROP(3))      THDE 93
000335      ASTRT = PROP(1)      THDE 94
000336      IF (ABS(AFS1) .LT. ASTRT) ASTRT = ABS(AFS1)      THDE 95
000343      IF (ABS(EFS) .GT. ABS(ASTRB)) GO TO R542      THDE 96
000350      IF (ABS(EFS) .GT. ABS(ASTRT)) GO TO R542      THDE 97
000355      GO TO R541      THDE 98
      C      USE INTERVAL HALVING TO VARY THE PLATE SIZE      THDE 99
000355      R542 IF (KP .EQ. 1) GO TO R700      THDE 100
000357      IF (ABS(PLTR) .LT. .0001) GO TO R540      THDE 101
000363      IF (ABS(PLTT) .GT. 1.5 * FLTH(IST)) GO TO R540      THDE 102

```



```

000370      BL = PLTR                                END 103
000371      KS = 1                                    END 104
000372      GO TO R565                                END 105
000372  R560  BL = 2.1875                            END 106
000374      BL = .1875                                END 107
000375      IF (IST .LT. 3) BL = 4.1875              END 108
000401      GO TO R565                                END 109
000402  R561  IF (KP .EQ. 1) GO TO R701              END 110
000404      IF (ABS(PLTH) .LT. .0001) GO TO R581      END 111
000410      BL = PLTR                                    END 112
000411      KS = 0                                       END 113
000412  R565  TH = (BL + BL) / 2.                    END 114
000415      IF (ABS(TH - TH) .LT. .0021) GO TO R582    END 115
000421      PLTH = TH                                    END 116
000422      PLTH = TH                                    END 117
000423      IF (JPTL .GT. 9) GO TO R545                END 118
000426      IF (DEFCON(IA,1) .GE. 0.0) PLTH = 0.0      END 119
000431      GO TO R545                                END 120
000432  R582  IF (KS .EQ. 0) GO TO R587              END 121
000433      PLTH = PLTH + .0025                            END 122
000435      KP = 1                                       END 123
000436  R587  IF (PLTH .GT. 1.5 * PLTH(IST)) GO TO R588  END 124
C      STORE THE THICKNESS REQUIRED IN THE PROPER ARRAY  END 125
000443      PLATH(1,2) = PLTR                            END 126
000444      PLATH(1,2) = PLW                                END 127
000446      IF (JPTL .GT. 9) GO TO R0                  END 128
000451      IF (DEFCON(IA,1) .GE. 0.0) GO TO R585        END 129
000453      PLATH(1,1) = PLTR                                END 130
000455      PLATH(1,1) = PLW                                END 131
000457      GO TO R701                                    END 132
000457  R585  (1) 4509 J = 1, 2                        END 133
000461      PLATH(1,J) = 0.0                                END 134
000464  R589  PLATH(1,J) = 0.0                            END 135
000471      IF (DEFCON(IA,1) .LT. 0.0) GO TO R900        END 136
000473      GO TO R585                                    END 137
000474  R585  PLATH(1,1) = 0.0                            END 138
000476      PLATH(1,1) = 0.0                                END 139
000477      GO TO R589                                    END 140
000477  R580  CONTINUE                                    END 141
000477      ISPLIT = 1                                       END 142
000500      IF (JPTL .EQ. 4) WRITE (6,7000) NAME(IST)      END 143
000510  7000  FORMAT (///10X,4HTHE ,47,25H SECTION CAN NOT BE USED.)  END 144
000510      RETURN                                           END 145
000511  R701  PLATH(1,1) = PLTH                            END 146
000513      PLATH(1,2) = PLTR                                END 147
000515      PLWH(1,1) = PLW                                    END 148
000516      PLWH(1,2) = PLW                                    END 149
000517  R575  IF (KS .EQ. 1 .AND. KP .NE. 1) GO TO R597  END 150
C      COMPUTE THE SECTION PROPERTIES IF REQUIRED        END 151
000526      RP = DEPT(IST) / 2. + PLATH(1,2)            END 152
000532      CALL ICALC (PLTH(1,1), PLATH(1,1), PLWH(1,2), PLTH(1,2),  END 153
000546      , IDEPT(IST), SAREA(IST), IX(IST), RP, 0.0, SECC, SECI)  END 154
000546      KS = 0                                       END 155
000547      IF (JPTL .GT. 9) GO TO 702                    END 156
000553      TSA = SAREA(IST) + PLWH(1,1) * PLATH(1,1) + PLWH(1,2) * PLTH(1,2)  END 157
000561      H = 2. * (IDEPT(IST) + PLATH(1,2)) - SECC      END 158
000566      CALL ICALC (TSA, IDEPT, SLABTH, 0.0, 0.0, TSA, SECC, SECC, LAUNCH, SECH, SECI)  END 159
000600      CALL ICALC (TSA, IDEPT, SLABTH, 0.0, 0.0, H, TSA, SECC, SECC, LAUNCH, SECH, SECI)  END 160

```

000613	TO = DESMOD(I1) * PLATH(1,1) * PLATH(1,2)	100E 161
000617	TJ = TJ + PLATH(1,1) * SLATH * HATHCH	100E 162
000623	GO TO 8547	100E 163
000624	762 SECH = SECC	100E 164
000626	SECH = SECC	100E 165
000627	SECH = SECH	100E 166
000630	SECH = SECH	100E 167
000631	GO TO 8547	100E 168
000631	8710 IF (LMS(PLTH) .LT. .0001) PLTH = .1475	100E 169
000636	PLTH = PLTH + .0625	100E 170
000640	IF (PLTH .LE. 1.5 * FLTH(IST)) GO TO 8545	100E 171
000644	ISWIT = 1	100E 172
000645	RETURN	100E 173
000646	8527 IF (LPTC .GT. 9) GO TO 8900	100E 174
000652	C DETERMINE THE TOP STRESSES	100E 175
000652	INST(1) = DESMOD(IA,1) * (TD - SECC) / SECH	100E 176
000654	INST(2) = DESMOD(IA,2) * (TD - SECH) / SECH	100E 177
000662	INST(3) = DESMOD(IA,3) * (TD - SECH) / SECH	100E 178
000666	INST(4) = DESMOD(IA,4) * (TD - SECH) / SECH	100E 179
000672	C DETERMINE THE ALLOWABLE STRESS	100E 180
000672	AFS = BASEFAT(INST,F1,-1000.,PROP(3))	100E 181
000676	ASTRT = PROP(1)	100E 182
000677	IF (AFS(AFS) .LT. ASTRT) ASTRT = AFS(AFS)	100E 183
000704	IDS = SIGMAX(INST)	100E 184
000707	IF (IDS(IDS) .LE. ASTRT) GO TO 6060	100E 185
000712	C INCREASE THE TOP PLATE THICKNESS IF REQUIRED	100E 186
000712	IF (LMS(PLTH) .LT. .0001) PLTH = .1475	100E 187
000716	K2 = 1	100E 188
000717	PLTH = PLTH + .0625	100E 189
000721	IF (PLTH .LE. 1.5 * FLTH(IST)) GO TO 8545	100E 190
000725	ISWIT = 1	100E 191
000726	IF (LPTC .EQ. 4) WRITE (6,7000) NAME(IST)	100E 192
000735	RETURN	100E 193
000736	6060 IF (LPTC .GT. 9) GO TO 8900	100E 194
000742	C CALCULATE THE CONCRETE STRESS	100E 195
000742	INST(1) = 0.0	100E 196
000742	INST(2) = DESMOD(IA,2) * (TD - SECH) / (SECH * MR)	100E 197
000750	INST(3) = DESMOD(IA,3) * (TD - SECH) / (SECH * 3. * MR)	100E 198
000757	INST(4) = DESMOD(IA,4) * (TD - SECH) / (SECH * MR)	100E 199
000764	IDS = SIGMAX(INST)	100E 200
000767	IF (IDS(IDS) .LE. FC) GO TO 8900	100E 201
000772	C INCREASE THE TOP PLATE IF REQUIRED	100E 202
000772	IF (LMS(PLTH) .LT. .0001) PLTH = .1475	100E 203
000776	K2 = 1	100E 204
000777	PLTH = PLTH + .0625	100E 205
001001	IF (PLTH .LE. 1.5 * FLTH(IST)) GO TO 8545	100E 206
001005	ISWIT = 1	100E 207
001006	IF (LPTC .EQ. 4) WRITE (6,7000) NAME(IST)	100E 208
001015	RETURN	100E 209
001016	8900 CONTINUE	100E 210
001021	C PRINT THE PLATE SIZES IF REQUIRED BY INT	100E 211
001023	IF (LPTC .NE. 4) GO TO 7200	100E 212
001026	WRITE (5,7100)	100E 213
001026	7100 FORMAT(10I,2X,4HSECT,4X,3HTOP,4X,3HTOP,5X,3HBT,4X,3HBT/22X, 11I,3X,4HBASE,3X,5HPLTH,2X,5HPLTH,2X,5HPLTH,2X,5HPLTH/22X, 10/7101) = 1.0	100E 214
001026	GO TO 7101	100E 215
001026	IF (LMS(PLTH) .LT. .0001) PLTH = .1475	100E 216
001030	PLTH = PLTH + .0625	100E 217
001032	IF (PLTH .LE. 1.5 * FLTH(IST)) GO TO 8545	100E 218

	IF (X1(1,2)	TIME 219
001054	7100 F1X44T (216,T1,1,07,0,0,2,0,16,3,0,0,2,0,0,0,3)	TIME 220
001054	7200 CONVTIME	TIME 221
001054	RETURN	TIME 222
001055	END	TIME 223

SUBPROGRAM LENGTH

001265

UNUSED COMPILER SPACE

037000

	FUNCTION TOPT (T)	TOPT 1
000003	IF (1.LE. .5) TMIN = .1875	TOPT 2
000007	IF (1.GT. .5 .AND. 1.LE. .75) TMIN = .25	TOPT 3
000021	IF (1.GT. .75 .AND. 1.LE. 1.5) TMIN = .3125	TOPT 4
000033	IF (1.GT. 1.5 .AND. 1.LE. 2.25) TMIN = .375	TOPT 5
000045	IF (1.GT. 2.25 .AND. 1.LE. 6.0) TMIN = .5	TOPT 6
000057	TOPT = TMIN	TOPT 7
000061	RETURN	TOPT 8
000061	END	TOPT 9

SUBPROGRAM LENGTH

000111

UNUSED COMPILER SPACE

042700

	SUBROUTINE UNIFST	UNIF 1
	DESIGNS THE UNIFORM SECTION GIRDER	UNIF 2
000002	COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WDEPTH(18),	UNIF 3
	IDAF(18),IX(18),IY(18),NSFCT(18),NOS(80),ISTEFF(80),CS(80),	UNIF 4
	ZILCON(80),CL(80),THCON(80),CH(80),PLATH(2,2),PLWID(80,2),NF,	UNIF 5
	JSLANW,SLATH,MU,CDD(R1),RTI(5,81,3),CCF(5,81),F0,SPAN(4),TOSP(5),	UNIF 6
	4TMOI(80),NA,LENGTH(80),SURLEN(80),NS,UNICO(2),TOLC(10),PROP(3),	UNIF 7
	SFC,STRESS(R1,4,2),DESMOM(R1,4),DESDFA(5,4),DESS(162,4),FAR(5,3),	UNIF 8
	6CONSI(9),CONSDI(6,3,4),CONVBL(12,6,2)	UNIF 9
000002	COMMON/UNIF/JPT3,TRT10	UNIF 10
000002	COMMON/THW/LPCM	UNIF 11
000002	COMMON/THREE/AS,NCS,TRT4	UNIF 12
000002	COMMON/FOUR/WTCT,ICOST	UNIF 13
000002	COMMON/SIX/FU	UNIF 14
000002	COMMON/SEVEN/DMTH,CMTN	UNIF 15
000002	COMMON/EIGHT/IST	UNIF 16
000002	COMMON/TEEN/HADINCH	UNIF 17
000002	COMMON/EIGHTEEN/TCR	UNIF 18
000002	REAL IX,IY,ISTEFF,IILCON,THCON,MU,LENGTH	UNIF 19
000002	DIMENSION IITL(26),EPS(4),IIL(21,3),STOR(12),LPCM(10)	UNIF 20

```

000003  -   DETERMINE THE CRITICAL DESIGN SECTION FOR A CONTINUOUS GIRDER UNIF 21
000004  M4 = MS = 1 UNIF 22
000004  IF (OPT1 .LT. 9) ISX101 = SLAMW / 13. * 200 UNIF 23
000012  IF (OPT1 .LT. 9) ISX102 = SLAMW / 10 UNIF 24
000016  IF (M4 .EQ. 1) GO TO 4000 UNIF 25
000020  C   DETERMINE THE CRITICAL DESIGN SECTION FOR A CONTINUOUS GIRDER UNIF 26
000020  D54 = 0.0 UNIF 27
000021  ICR = 1 UNIF 28
000022  DO 1500 I = 1, 10, 2 UNIF 29
000023  IA = LPM(I) UNIF 30
000025  IB = LPM(I+1) UNIF 31
000026  IF (IA .EQ. 0 .AND. IB .EQ. 0) GO TO 1500 UNIF 32
000034  DO 1501 K = 1A, IB UNIF 33
000035  DO 1502 J = 1, 4 UNIF 34
000036  1502 TMS1(J) = DESMOM(K,J) UNIF 35
000045  TMS = SMAX(TMS1) UNIF 36
000047  IF (ABS(TMS) .LE. DSM) GO TO 1501 UNIF 37
000052  ICR = K UNIF 38
000053  DSM = ABS(TMS) UNIF 39
000054  1501 CONTINUE UNIF 40
000057  1500 CONTINUE UNIF 41
000061  SMFEC = DESMOM(ICR,1) + DESMOM(ICR,2) + DESMOM(ICR,4) UNIF 42
000065  GO TO 1562 UNIF 43
000065  C   DETERMINE THE CRITICAL DESIGN SECTION FOR A SIMPLE SPAN UNIF 44
000067  4000 AV = 9999999. UNIF 45
000067  DT = LENGTH / 2. UNIF 46
000071  DO 4010 J = 1, NA UNIF 47
000072  IF (ABS(COOR(J) - DT) .GT. AV) GO TO 4010 UNIF 48
000100  ICR = J UNIF 49
000100  AV = ABS(COOR(J) - DT) UNIF 50
000102  4010 CONTINUE UNIF 51
000105  WRITE (6,4030) ICR UNIF 52
000112  4030 FORMAT (1H0,10X,62HON THE SIMPLE BEAM, THE CRITICAL MOMENT IS AT POINT UNIF 53
000112  ID(NT,12) UNIF 54
000112  SMFEC = DESMOM(ICR,1) UNIF 55
000114  IF (OPT3 .GT. 9) SMFEC = DESMOM(ICR,1) + DESMOM(ICR,2) UNIF 56
000121  GO TO 4020 UNIF 57
000122  1562 WRITE (6,1510) ICR UNIF 58
000130  1510 FORMAT (1H0,10X,34HCRITICAL NEGATIVE MOMENT POINT IS ,12) UNIF 59
000130  C   DESIGNS THE CONTINUOUS GIRDER UNIF 60
000130  C   DETERMINE THE ALLOWABLE STRESSES UNIF 61
000130  DO 4200 J = 1, 4 UNIF 62
000132  4200 TMS1(J) = DESMOM(ICR,J) UNIF 63
000141  ASFC = HASEAT(TMS1,FU,-1000.,PROP(3)) UNIF 64
000145  ASFI = HASEAT(TMS1,FU,1000.,PROP(3)) UNIF 65
000151  ASF = ASFI UNIF 66
000152  IF (ABS(ASFC) .LT. ABS(ASFI)) ASF = ABS(ASFC) UNIF 67
000157  PLT = 0.0 UNIF 68
000160  PLW = 0.0 UNIF 69
000160  C   ITERATES THE SECTIONS TO DETERMINE THE MINIMUM WEIGHT SECTION UNIF 70
000161  I = 1 UNIF 71
000162  IST = 0 UNIF 72
000163  1561 IF (I .EQ. 19) GO TO 2603 UNIF 73
000165  EFS = DSM * DEPTH(I) / (2. * IX(I)) UNIF 74
000171  ASTC = ALLOW(ESMOM(ICR,1),FLWID(I),LPCM,IOSD,ICR,NS,PROP,COOR, UNIF 75
000171  IDPS40M,SUBLEN,F,1) UNIF 76
000206  IF (ABS(ASF) .LT. ABS(ASTR)) ASTR = ABS(ASF) UNIF 77
000213  IF (ABS(EFS) .GT. ASTR) GO TO 1560 UNIF 78

```

```

000220      ISF = 1                                UNIF 79
000221      I = I + 1                                UNIF 80
000222      GO TO 1501                                UNIF 81
000223 4000 CONTINUE                                UNIF 82
C          DESIGN THE SIMPLE SPAN PROBLEM          UNIF 83
C          ITERATES THE SECTIONS TO DETERMINE THE MINIMUM WEIGHT DESIGN UNIF 84
000223      I = 1                                UNIF 85
000224      ISF = 0                                UNIF 86
000225 4000 IF (I.EQ. 19) GO TO 2603                UNIF 87
000227      SIS = ISF(I)                            UNIF 88
000231      SC = DEPTH(I) / 2.                      UNIF 89
000233      IF (DPTH.GT. 4) GO TO 700                UNIF 90
000236      CALL ICALC(ISWID1,SLATH,0.,0.,DEPTH(I),SABF(I),SIS,SC,LAUNCH, UNIF 91
000236      ICIC,SLC)                                UNIF 92
000250      CALL ICALC(ISWID2,SLATH,0.,0.,DEPTH(I),SABF(I),SIS,SC,LAUNCH, UNIF 93
000250      ICIC,SLC)                                UNIF 94
000253      GO TO 701                                UNIF 95
000264 700 CIC = SC                                UNIF 96
000266      SIC = SIS                                UNIF 97
000267      CLC = SC                                UNIF 98
000270      SLC = SIS                                UNIF 99
C          DETERMINES THE ALLOWABLE STRESSES        UNIF 100
000271 701 TMS(1) = DESMOD(ICR,1) * SC / SIS        UNIF 101
000274      TMS(2) = DESMOD(ICR,2) * CIC / SLC        UNIF 102
000277      TMS(3) = DESMOD(ICR,3) * CLC / SLC        UNIF 103
000301      TMS(4) = DESMOD(ICR,4) * CIC / SLC        UNIF 104
000304      TMS = STMAX(TMS)                          UNIF 105
000306      ASF = HASTAT(TMS,FJ,1000.,PROP(3))        UNIF 106
000312      ASF = PROP(1)                              UNIF 107
000313      IF (ASF.LT. PROP(1)) ASTR = ASF            UNIF 108
000317      IF (TMS(IDS).GT. ASTR) GO TO 1500         UNIF 109
000324      ISF = 1                                UNIF 110
000326      I = I + 1                                UNIF 111
000326      GO TO 4060                                UNIF 112
000327 1500 IF (IST.EQ. 0) GO TO 2605                UNIF 113
000330 2603 IF (DPTH.GE. 1) WRITE (6,1588) NAME(I)SFI UNIF 114
000341 1588 FORMAT (1H0,20X,PHASE THE .A7.23H FOR A UNIFORM SECTION.////) UNIF 115
C          SETS UP THE DESIGN ARRAYS                UNIF 116
000341      DO 1510 J = 1, NF                        UNIF 117
000343      PLAIN(J,1) = PL1                            UNIF 118
000345      PLAIN(J,2) = PLT                            UNIF 119
000346      PLWID(J,1) = PLW                            UNIF 120
000347      PLWID(J,2) = PLW                            UNIF 121
000350 1510 NOST(J) = IST                              UNIF 122
000354      CALL SEFIC                                UNIF 123
000355      GO TO 2602                                UNIF 124
000356 2605 WRITE (6,2606)                            UNIF 125
000360 2606 FORMAT (1H0,42HMIN SECTION WITHOUT PLATES IS SATISFACTORY.) UNIF 126
000362      RETURN                                    UNIF 127
000363 2602 DO 3000 I = 1, 12                        UNIF 128
000365      DO 3100 J = 1, 6                            UNIF 129
000366      DO 3200 K = 1, 2                            UNIF 130
000367 3000 CUMPL(I,J,K) = 0.0                        UNIF 131
000403      CALL CUMPLS                               UNIF 132
C          DESIGN THE BEARING STIFFENERS            UNIF 133
000404      CALL STIFF(COSP,S,DESFRA,DS,FLTH,FLWID,FLFTH,FRTH,PRDS,PM, UNIF 134
000404      I,J,K)                                       UNIF 135
C          DESIGN THE SHEAR CONNECTORS              UNIF 136

```

000417	CALL COEFES	UNIF 137
	C DEFINE THE VALUES OF THE OBJECTIVE FUNCTION	UNIF 138
000420	CALL OBJECT	UNIF 139
	C AFTER THE DESIGN AS A STARTING POINT	UNIF 140
000421	CALL KEEPINF (NS, PLATH, PLWID, COVPL, CHIN, CH, IPTA)	UNIF 141
	C RETRIEVE THE DESIGN INFORMATION	UNIF 142
000431	IF (ITFIC .LE. 1) GO TO 2611	UNIF 143
000434	CALL PRSEFC (NF, NOS, TAPF, PLWID, PLATH, ISTEEL, CS, IICON, CI, IIRCON, CH,	UNIF 144
	ICONX, SUNIT, G, NOY)	UNIF 145
000452	CALL PROCV (COVPL)	UNIF 146
000454	WRITE (6,100)	UNIF 147
000460	100 FORMAT (131)	UNIF 148
000460	CALL PRSTIF (NS, HEAR)	UNIF 149
000462	CALL PROCV (NS, COVSP, NOS)	UNIF 150
000465	WRITE (6,1510) WTOT, ICOST	UNIF 151
000475	1510 FORMAT (1H0,10X,15HTOTAL WEIGHT = ,F15.5X,13HTOTAL COST = ,	UNIF 152
	IF15.2)	UNIF 153
000475	IF (IPTA .GT. .5) GO TO 2610	UNIF 154
000501	WRITE (6,15H2) CHIN	UNIF 155
000506	15H2 FORMAT (1H0,30X,7HC4IN = ,F12.2)	UNIF 156
000506	GO TO 2611	UNIF 157
000507	2610 WRITE (6,35H2) WHIN	UNIF 158
000515	35H2 FORMAT (1H0,30X,7H4MIN = ,F12.2)	UNIF 159
000515	2611 CONTINUE	UNIF 160
000515	RETURN	UNIF 161
000516	END	UNIF 162

SUBPROGRAM LENGTH

001343

UNUSED COMPILER SPACE

040100