OPTIMIZATION OF THE SIZE AND THE LOCATION OF STEEL MEMBERS OF A GRIDWORK BY MATHEMATICAL PROGRAMMING

By

MENG SAM CHAN

Bachelor of Science
Heald Engineering College
San Francisco, California
1964

Master of Science
University of New Mexico
Albuquerque, New Mexico
1966

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1971

OPTIMIZATION OF THE SIZE AND THE
LOCATION OF STEEL MEMBERS OF
A GRIDWORK BY MATHEMATICAL
PROGRAMMING

Thesis Approved:

M. Tichacek
Thesis Adviser

R. L. Janes

N. Lawkins

P. L. Claypool

D. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I am dedicating this dissertation to the memory of my father, who strived hard all his life for the betterment of his family's welfare and of his children's education but missed seeing me fulfill his hopes and my goal by a few months. Besides my father, my mother, brothers, and sisters all have given encouragement throughout my studies, for which I grateful.

Special gratitude and appreciation should go to my thesis advisor, Dr. Tochacek, who spent an uncountable amount of time and effort in supervising and perfecting this research. Thanks are directed to Dr. Janes, Chairman of the Advisory Committee, to Dr. Dawkins, Chairman of the Structures Division, and to Dr. Claypool, who have served on my Committee and given invaluable advice in this research and in reading the thesis. No less thanks should be given to Dr. Mac-Alpine, former Chairman of the Structures Division, who served on my Committee until his retirement, and to Dr. Eldred, who also served on my Committee until his detachment from the university.

Since this research stemmed from the knowledge of my Civil Engineering study, I am grateful to all my professors at this university and others. Among them are especially Dr. James Yao and Professor Marion Cottrel, who laid the foundation for me and gave me encouragement to further my studies. Speaking of the academic community as a whole, besides the social community, I thank all my friends, who have shown me friendship and willingness to help, from the Civil Engineering and other departments. I must thank Miss Carol Manson for typing the final version of the thesis.

I would like to thank the following persons, among others, with whom I have had correspondence during my research: Professor J. Abadie, Professor Edward J. Beltrami, Mr. W. C. Mylander, and Mr. R. T. Mertz, who supplied information and suggestions about the RGM Code.

Last, but not the least, I am revealing the efforts, encouragement, and love from behind the scene: The person who works harder when I work hard, suffers more when I suffer, cheers me up when I am depressed and is ready to give encouragement whenever I need it, of whom I am proud and forever grateful, from the bottom of my heart--my dear wife, Elsa.

TABLE OF CONTENTS

Chapter | Page

I. INTRODUCTION 1

1.1 Statement of the Problem 1
1.2 Historical Sketch 1
1.3 Object of Investigations 3

II. METHOD OF DESIGN 4

2.1 Introduction 4
2.2 Selection of the Structural Analysis Method 5
2.3 Selection of the Mathematical Programming Technique 6
2.4 The Ricochet Gradient Method (RGM) 9
2.5 The Modified Sequence of Linear Programming Solutions Technique (MSLP) 13
2.6 Computer Code MSLP 17
2.7 The Design Procedure 22
2.8 Convexity 22

III. BEAMS SPANNING IN ONE DIRECTION PROBLEM 1 25

3.1 Section Properties 25
3.2 Design Variables 28
3.3 Design Criteria 28
3.4 Mathematical Model 33
3.5 Analytical Approach 35

IV. A GRID WITH BEAMS SPANNING IN TWO ORTHOGONAL DIRECTIONS--SIMPLE-CONNECTION (PROBLEM 2), RIGID-CONNECTION (PROBLEM 3) 41

4.1 Introduction 41
4.2 Geometry of the Grid 43
4.3 Governing Equation 48
4.4 Deflections and Moments 49
4.5 Design Criteria 54
4.6 Mathematical Model 60

Chapter Page

V. NUMERICAL EXAMPLES 62

5.1 Problem 1. 62
5.2 Problem 2. 67
5.3 Problem 3. 72
5.4 Discussion of Numerical Results. 72

VI. BOUNDARY CONDITIONS, LOADING CONDITIONS, AND BOUNDARY DIMENSIONS. 75

6.1 Introduction. 75
6.2 Timoshenko-Ritz Method. 75
6.3 A_{mn} for Clamped Supports. 78
6.4 Loading Conditions 80
6.5 Boundary Dimensions 83

VII. DISCUSSION OF RESULTS AND CONCLUSION 87

A SELECTED BIBLIOGRAPHY 92

APPENDIX A - COMPUTER PROGRAM LISTING-- THE MSLP CODE, (MODIFIED SEQUENCE OF LINEAR PROGRAMMING SOLUTIONS). 95

APPENDIX B - PROBLEM LISTINGS
(1) PROBLEM 1
(2) PROBLEM 3 126

APPENDIX C - DETERMINATION OF $C = C_{11}$ FROM DEFLECTION CONSIDERATION FOR CASE (2)--FORMULA (3.46) WITH THE EQUALITY SIGN 147

APPENDIX D - DETERMINATION OF C_{12} AND THE MOMENT OF INERTIA, I, FOR CASE (3). 150

LIST OF TABLES

Table Page

I. Data for Example (5. 1) 63

II. Solutions of Examples (5. 1) to (5. 5). 65

III. Three Constraint Values of Examples (5. 1) to (5. 5) 66

IV. Solution of Example (5. 6), (PROBLEM 1) 68

V. Solution of Example (5. 7), (PROBLEM 2) 70

VI. Final Design of Example (5. 7), (PROBLEM 2) 71

LIST OF FIGURES

Figure		Page
1.	Path of Ricochet Gradient Method	11
2.	Contours of One Constraint Function	12
3.	Tangent Lines Enclosing the Nonlinear Constraint Set	14
4.	Example Illustrating the Cutting Plane Method	15
5.	Block Diagram, "Modified Sequence of Linear Programming Solutions Technique"	19
6.	Illustration of Convex (G_1) and Nonconvex (G_2) Functions	24
7.	Beams Spanning in One Direction	29
8.	(a), (b), and (c): Types 1, 2, and 3	42
9.	Case 1	44
10.	Case 2	45
11.	Case 3	46
12.	Case 4	47
13.	A Grid Simply-Supported on All Four Sides	50
14.	Locations of Three Typical x-Beams	52
15.	Bending Stresses at a Joint	56
16.	A Grid Clamped on All Four Sides	76
17.	Local Uniform Load Shrinking to a Concentrated Load	81
18.	Simplest Grid with One Beam in Each Direction	84
19.	Castillated Beam	89

NOMENCLATURE

a	Spacing of beams oriented in the y-direction, ("y-beams")
A	Dimension in the x-direction of a given rectangular area spanned by beams
$\{A\}$	Column matrix of the unknown coefficients, (A_{11} A_{12} A_{21} A_{22})
A_f	Flange cross-sectional area
A_{mn}	Parameters associated with the assumed solution of deflection
A_t	Total cross-sectional area
A_w	Web cross-sectional area
b	Spacing of beams oriented in the x-direction, ("x-beams")
b_f	Flange width
b_o	Depth (width) of the narrow rectangular element
b_1	Constant
B	Dimension in the y-direction of a given rectangular area spanned by beams
BL_j	Lower bounds of x_j
B_1	Ratio of flange thickness to web thickness
B_2	Ratio of flange area to web area
C	Parameter changing the constraints from inequality to equality
$[C]$	A square matrix with elements defined by Formula (6.15)
C_1	Moment coefficient for simple supports and uniformly distributed load

C_2	Deflection coefficient for uniformly distributed load
C_3	Coefficient for deflection specified in building codes
C_4	Coefficient for critical stress
C_x, C_y	Unit torsional rigidities of the x- and y-beams
D	Decision function
$\{D\}$	Column matrix with four identical elements of (pAB)
D_x, D_y	Unit flexural rigidities of x-and y-beams
D_1, D_{xy}	Parameters associated with unit torsional rigidities
e_1, e_2	The distances from the neutral axis to the top and bottom fibers
E	Young's modulus of elasticity
f(X)	Function of X
f(x)	Loading function, varying in the x-direction
F(X)	Objective function
F_x	Flexural rigidity of each x-beam
F_y	Flexural rigidity of each y-beam
g(y)	Loading function, varying in the y-direction
$g_k(X)$	The k^{th} constraint
G	Shear modulus
$\overline{G}(X_1)$	Taylor series expansion of the constraint G(X), evaluated at the first solution X_1
$G_k(X)$	The k^{th} constraint
h	Web depth
H	Parameter associated with unit torsional rigidities
i	Total number of constraints
I	Moment of inertia with respect to the horizontal principal axis

I_x	Moment of inertia of the x-beam with respect to the horizontal principal axis
I_y	Moment of inertia of the y-beam with respect to the horizontal principal axis
j	Total number of design variables
J	Torsional moment of inertia
J_x, J_y	Torsional moments of inertia for the x- and y-beams, respectively
K	Web coefficient
K_w	Web coefficient
K_1	Coefficient depending on the stress distribution
LIN	Number of linear constraints
m	Summation index for the x-direction
M	Maximum bending moment
MN	Number of variables
M_{tot}	The total moment, covering a strip of width b, being centered at M_i
M_x	Bending moment per unit width in x-beam
M_y	Bending moment per unit width in y-beam
M_{xy}	Torsional moment per unit width in x-beam
M_{yx}	Torsional moment per unit width in y-beam
n	Summation index for the y-direction
NNL	Number of nonlinear constraints
p	Uniformly distributed load (UDL)
q(x, y)	Loading function per unit area
R	The reaction on the x-beam
R_1	Ratio of the two sides of the grid
s	Ratio of B to b
S	Section modulus

S_x	Section modulus of x-beam
S_y	Section modulus of y-beam
S_1, S_2	Section moduli for the top fibers 1 and bottom fibers 2
t	Thickness of the narrow rectangular element
t_f	Flange thickness
u	Dimension in the x-direction of the local uniform load
U	Strain energy of bending
UB_j	Upper bounds of x_j
v	Dimension in the y-direction of the local uniform load
V	Total volume of all beams used
V_w	Virtual work
w	Deflection
w_{all}	Allowable deflection
$x_1, x_2, \ldots, x_j$	Coordinates
X	A vector, $(x_1, \ldots, x_j)$
X_A	Point A
X_B	Point B

α	Section asymmetry
β	Constant between zero and one
δ	Web thickness
δ_x	Web thickness of x-beam
δ_y	Web thickness of y-beam
λ	Web slenderness, i.e., web depth-thickness ratio
λ_x	Web depth-thickness ratio of the x-beam, (web slenderness)
λ_y	Web depth-thickness ratio of the y-beam, (web slenderness)
ν	Poisson's ratio
π	3.1416
σ	Maximum bending stress
σ_{all}	Allowable bending stress
σ_{cr}	Critical buckling stress
σ_{jt}	Combined stress at a joint
σ_x	Bending stress in x-beam
σ_y	Bending stress in y-beam
σ_{xy}	Maximum shearing stress
Σ	Summation sign
∇	Gradient
∂	Notation for partial derivatives
$[\ ]$	Square matrix
$\{\ \}$	Column matrix
$(\ )^T$	Transpose of a matrix
$(\ )_{,x}$ through $(\ )_{,xxxx}$	First through fourth partial derivatives with respect to x

$(\ )_{,y}$ through $(\ )_{,yyyy}$	First through fourth partial derivatives with respect to y
$(\ )_{,xy}$ and $(\ )_{,xxyy}$	Second and Fourth mixed partial derivatives with respect to x and y

CHAPTER I

INTRODUCTION

1.1 Statement of the Problem

It is desired to optimize the location and the size of the steel members of a grid. For a particular combination of spacings and sizes of the steel members, the economical design of the grid is to be considered. It is evident that some design variables will originate from the key word "location" and some from the word "size." Although I-shaped steel plate girders are used throughout this investigation, other shapes can be handled using procedures developed herein. The approach to the optimal design of a grid is to use a suitable Structural Analysis method and to cast it into a form of Nonlinear Programming problem, usually referred to as NLP, so that a compatible NLP technique can be applied to solve the grid problem.

1.2 Historical Sketch

Optimization has long been industry's main concern. Contracts of construction or of manufactured products are made usually based on the lowest bid. The Armed Services started using optimization in its strategic decisions. G. B. Dantzig and M. Wood developed and applied the problem of Linear Programming, LP, for the U. S. Department of Air Force (1). Publications on optimization can be found in Operation Research, Management Science, Economics, Industrial Engineering, Aeronautical and Aerospace Engineering, Chemical Engineering, Civil Engineering, Mechanical Engineering, Applied Mathematics, and among various other science fields. A good account of optimization development in Civil Engineering can be found in papers by Sheu (2) and Wasiutynski (3). With increasing demands for solving complicated optimization problems, developments in Nonlinear Programming techniques have been flourishing.

Probably one of the earliest investigations in the Civil Engineering field using Mathematical Programming was the three-bar truss problem solved by Schmit (4). Some work was done on individual structural members by Goble (5), Razani (6), Mauch (7), Holt (8), and Seaburg (9). The building frame was first investigated through the use of plastic design by Pearson (10), and Livesley (11), whereby the simplified assumptions yielded linear constraints so that Linear Programming could be applied. As the NLP was becoming more popular, the building frame could be analyzed using elastic design with nonlinear functions; for example, in Ref. (12). Since it has taken some time to get plastic design theory up to a competitive level with elastic design theory, it is expected that this approach to optimization should earn its recognition in the Civil Engineering field in the similarly slow pace. At the present time, it still stays at the academic level, caused by rather slow demands for it. The reaction is mutual; the industry is not getting much stimulation from universities.

A historical survey on Mathematical Programming is included in Section 2.3.

1.3 Object of Investigations

Under the term "conventional design" of a structure, sizing of structural members is usually understood, the geometry of the structure being given or arbitrarily selected. On the other hand, in the present "optimum design" approach, the spacings of beams in the two directions of the grid are treated as design variables rather than given or preselected constants. Similarly, some other parameters (e.g., section properties) may be treated as design variables to be optimized. The economy achieved by the optimum design is the advantage over the conventional design method, the latter being based rather on experience and intuition. The designer can choose his own analysis method and his available NLP technique to do the job. This point will be discussed further in Chapter II, together with the design procedure.

The theme of this investigation starts off with a "degenerated" grid, with beams in one direction only (Chapter III). Chapter IV considers first a grid with beams in two orthogonal directions, joined by connections without torsional resistance, and also a grid with torsion-carrying connections. Numerical examples for these three systems with simple supports are given in Chapter V. Clamped ends (fixed supports) are treated in Chapter VI. Also in Chapter VI, boundary dimensions and general loading conditions are discussed. Finally, Chapter VII discusses results and summarizes conclusions.

CHAPTER II

METHOD OF DESIGN

2.1 Introduction

The conventional design of a grid proceeds as follows: A preliminary design of members with the choice of their spacings is based on experience and engineering judgment. Proposed sections are checked for effects determined by methods of Structural Analysis and Strength of Materials to satisfy safety and functional requirements of official building codes. If the resulting safety factors are too high, the design might be modified for economy and the sections reduced in size. Sometimes several different beam spacings are considered and economy of different arrangements is studied. As computer usage is becoming popular, the structural engineer is relieved of tedious calculator and slide rule work. Modifications in the design are done by changing parts of the computer program, or even just a few data cards. However, with the use of standard computer programs such as the MIT ICES (13) packages, the conventional procedure of design has remained basically unchanged.

Only in recent few years research scientists have started to apply optimization methods of Mathematical Programming to structural design: A physical model is set up and cast into a mathematical model. The physical model is formulated through Structural Analysis and Strength of Materials, and then transformed into a mathematical model treatable by Mathematical Programming. The objective function, the goal, is introduced, for instance, to minimize the cost of the structural system, or its volume. (The total cost of the structure varies from company to company and from city to city. Therefore, for the sake of generality, the volume of the material used is selected for the objective function in this study.) The functional and safety requirements define the so-called constraints. Techniques of Mathematical Programming enable one to arrive from a starting point within the feasible region (the preliminary design) at a final point (the optimum design).

2.2 Selection of the Structural Analysis Method

At the very beginning of this investigation, a Structural Analysis method must be chosen such that it will be suitable to be incorporated with the Mathematical Programming model. Various methods are available to treat the grid as

(a) a discrete system, or

(b) a continuous system.

(a) In the first group, the displacement method can be used to calculate deflections and moments. If evaluations of functions involved were done just a few times, the selection of Structural Analysis methods would not make too much difference. However, the Mathematical Programming technique requires many function evaluations; hence, the choice of a suitable Structural Analysis method will play an important role, especially in the consumption of computer time. The displacement method also depends on the number of nodes; its matrix operation, especially inversion, requires more execution time besides storage time as the number of nodes increases.

(b) In the second group, another possibility is to treat the grid as an orthotropic plate. Computation of deflections and moments is done by algebraic manipulation with double or single series. As far as computer time is concerned, this approach is preferred because the previous method (with too many redundant quantities) requires more time. Therefore, in this study the grid will be designed using the Theory of Plates.

2.3 Selection of the Mathematical Programming Technique

In the mathematical language, the problem can be formulated as follows:

Let $X = (x_1, x_2, \ldots, x_j)$,

j = total number of design variables, and

i = total number of constraints.

Minimize[1] the objective function $F(X)$ subject to constraints

$$G_k(X) \geq 0, \quad k = 1, \ldots, i \tag{2.1}$$

for the variables $X \geq 0$. Both the objective function $F(X)$ and the constraints $G_i(X)$ are nonlinear in this study; hence, the Linear Programming technique cannot be used directly and Nonlinear Programming must be applied.

Nonlinear Mathematical Programming techniques fall into five main categories:

(1) Methods to solve a problem with a nonlinear objective function but linear constraints,

---

[1] Minimization of a function $F(X)$ is the same as maximization of the negative function $-F(X)$.

(2) Sequential Unconstrained Minimization Technique (SUMT),

(3) Sequence of Linear Programming Solutions (SLP),

(4) Direct Search,

(5) Totally Nonlinear Programming.

2.3.1 First category

The objective function is approximated by a quadratic function and the algorithm is based on linear constraints. The method of solution was developed mainly by Zoutendijk (14) in 1960. An up-to-date reference on the method is given by Abadie (15) and Wolfe (16). They are among the pioneers who conducted researches along this line. In their adaptation, the eventual nonlinear constraints are linearized by Taylor series expansion. Successive iterations remove errors resulting from the truncation of terms of higher degrees. The accuracy of the method depends on the degree of nonlinearities of the objective function and the constraints. Computer Codes for this technique for a general case have not been released yet; a code is being developed by RAND Corporation, Santa Monica, California (17). IBM Corporation has a code, called "Separable Programming" (18).

2.3.2 Second category

Some researchers started the idea of Sequential Unconstrained Minimization Technique (SUMT) about 1959 but did not go into depths until Fiacco and McCormick (19), (20), and (21) developed the details and a computer code of the method under an Army research contract between the years 1963 (19) and 1965 (20). The constraints are incorporated with the objective function in the so-called penalty function and the algorithm treats the problem as an unconstrained one. A sample problem was solved in (20) with a cubic objective function and linear constraints. If a problem is highly nonlinear in both the objective function and the constraints, complications can be expected.

2.3.3 Third category

Kelly (22) suggested in 1960 a sequence of linear programming expressions, SLP, to approximate a nonlinear constraint set by a linear set, while the objective function is linear. He named his technique "the Cutting Plane Method" (22). A brief description follows in Section 2.5.

2.3.4 Fourth category

The Direct Search method does not require any quadratic or linear approximation. The main problem is the choice of the mesh density and of the methods of successively reducing the feasible region during the solution. Massachusetts Institute of Technology developed a package within the ICES, called OPTECH (13), in 1968. It works well for an unconstrained problem; however, no hints are given to handle a constrained problem.

The IBM Corporation has a code, termed the Ricochet Gradient Method (RGM) due to Mertz (23), released in 1967. He used the Steepest Descent Method with some modifications (23). A brief description of the RGM is in Section 2.4.

2.3.5 Fifth category

In recent years, researchers have been trying to improve the Totally Nonlinear Programming; a stimulation has been evident. Three techniques are mentioned here in passing: Geometric Programming by Duffin, Peterson, Zener (24) in 1967; Generalized Reduced Gradient Method by Abadie (15) in 1966; and Dynamic Programming by Bellman (25) and (26) in 1957.

Since Geometric Programming depends on the manipulation with the exponential powers of the variables appearing in the functions involved, an immediate drawback is when the method is applied to a problem with fractional functions.

The Generalized Reduced Gradient Method promises to be superior in both speed and accuracy.

Naturally, after a structural problem has been cast into a Mathematical Programming model, one can choose any suitable technique. From the viewpoint of availability and compatibility with the Structural Analysis method chosen, the Ricochet Gradient Method (RGM) and the Sequence of Linear Programming Solutions (SLP), from the fourth and third categories respectively, seem to be most suitable for this study.

The two methods chosen are explained in more detail in Sections 2.4 and 2.5.

2.4 The Ricochet Gradient Method (RGM)

The Ricochet Gradient Method, RGM Code, by Mertz (23), is essentially a Direct Search method using the Gradient Method, also known as the Steepest Descent[1] Method. It solves a Nonlinear Programming problem (Formula 2.1 in Section 2.3). Consider a two-dimensional case (Fig 1). The routine starts from a feasible point X_o and follows the path of steepest descent or the steepest gradient of the objective function. If a local minimum is inside the feasible region, then this path will lead to the local minimum. Otherwise, the path will lead to a nearby constraint at point B.

Consider Fig 2 which shows one constraint by itself. It is desired to locate a point B on the constraint function, $G_k = 0$. At point A in Fig 2 the constraint function value is $G_k > 0$; at point C its value is $G_k < 0$. Interpolation locates approximately the point B desired, at which $G_k = 0$.

The path is reflected (ricochets), following the constant value of the objective function, and heads towards another constraint across the feasible region. A point D is obtained in the same manner as point B. The line BD and the two tangents drawn at B and D form a triangle. With two base angles and the base of the triangle known, the point X_1 is determined and used as a starting point for the next cycle of the process. The process is repeated until an "optimum solution" is obtained.

A detailed description of this method and a listing of the Computer Code can be found in (23); therefore, no attempt is made to describe the method any further.

---

[1] Ascent for maximization, descent for minimization.

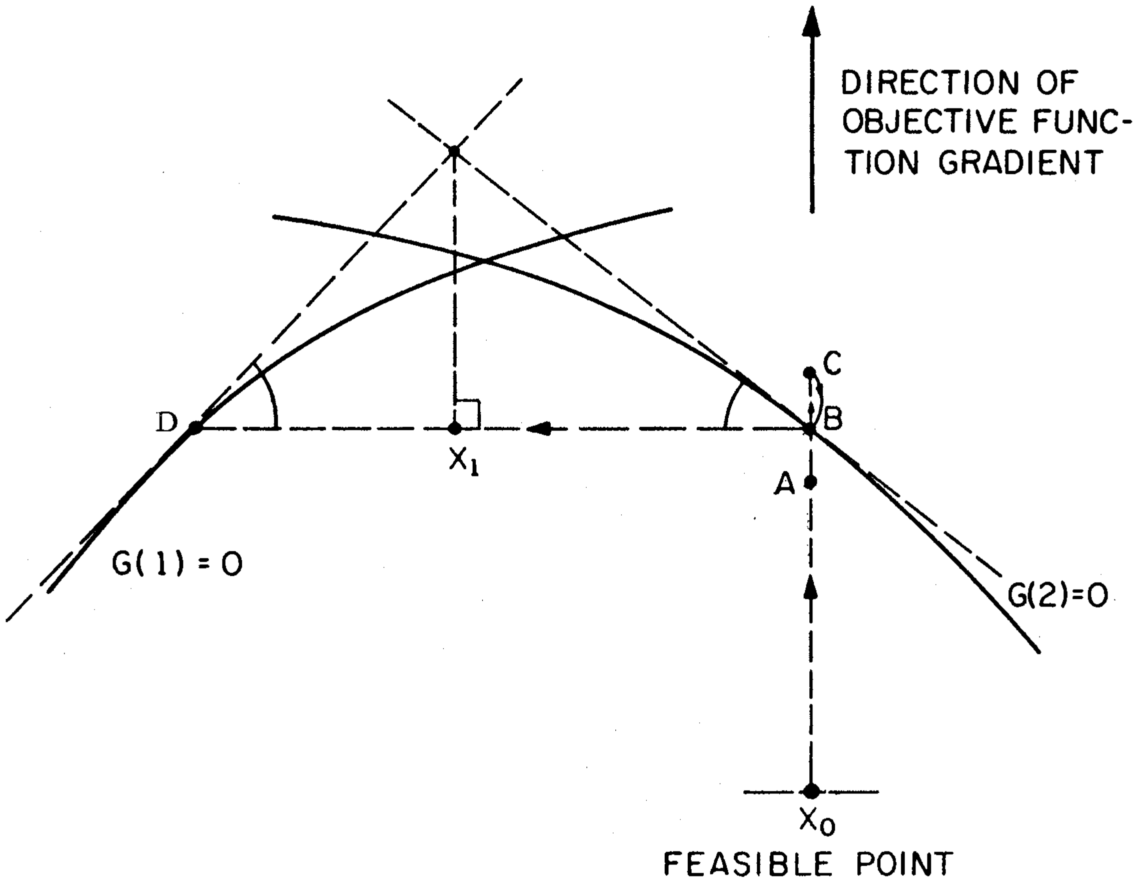


Figure 1. Path of Ricochet Gradient Method

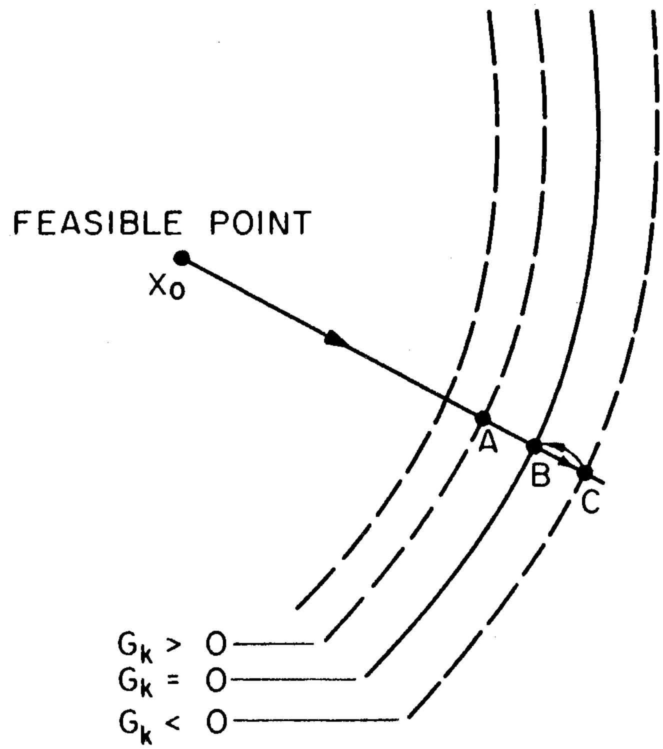


Figure 2. Contours of One Constraint Function

2.5 The Modified Sequence of Linear Programming Solutions Technique (MSLP)

Since the Gradient Method and Linear Programming have been well-documented in the literature, no attempt is made in this section to make a superfluous description of them. The Gradient Method used in Ricochet Gradient Method (RGM) was documented by Mertz (23). The Sequence of Linear Programming Solutions (SLP) was explained in (22); the Linear Programming subroutine used in MSLP was documented by Kuo (27).

Two things are changed in the SLP so that the modified version will be called "the Modified Sequence of Linear Programming Solutions Technique" (MSLP). The first modification is that the objective function is also linearized. The second modification is that the nonlinear constraint set is approximated by a set of lines tangent to the nonlinear constraint set, for example in a two-dimensional case (Fig 3).

A numerical example (22) is given to illustrate the "Cutting Plane Method" (Fig 4):

Let $X = (x_1, x_2, \ldots, x_j)$,

$(x_1, x_2, \ldots, x_j)$ = coordinates.

Minimize $F(X) = x_1 - x_2$ subject to the constraint

$$G(X) = 3x_1^2 - 2x_1x_2 + x_2^2 - 1 \leq 0$$

First, a large square, representing the upper and lower bounds of the variables, is placed to enclose the feasible region. Initially we solve the linear problem:

Minimize $F(X) = x_1 - x_2$ subject to the constraints

$$-2 \leq x_1 \leq 2 \text{ and } -2 \leq x_2 \leq 2.$$

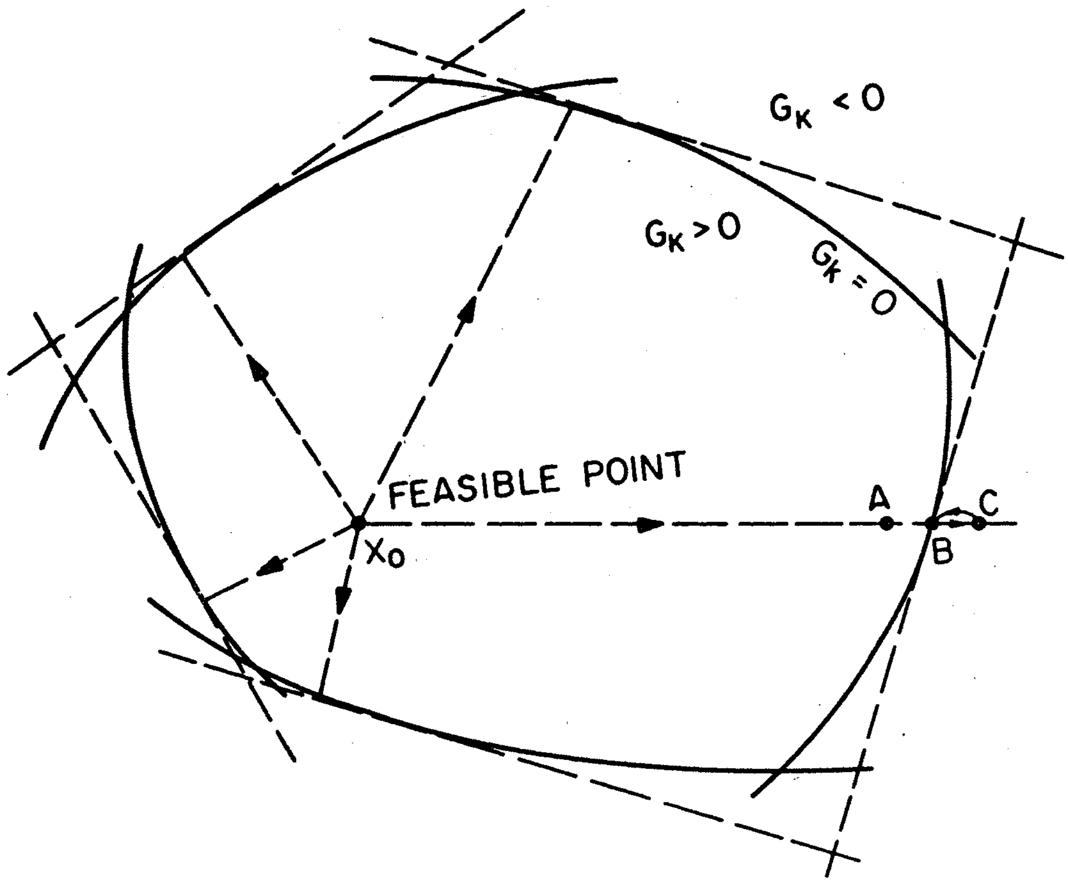


Figure 3. Tangent Lines Enclosing the Nonlinear Constraint Set

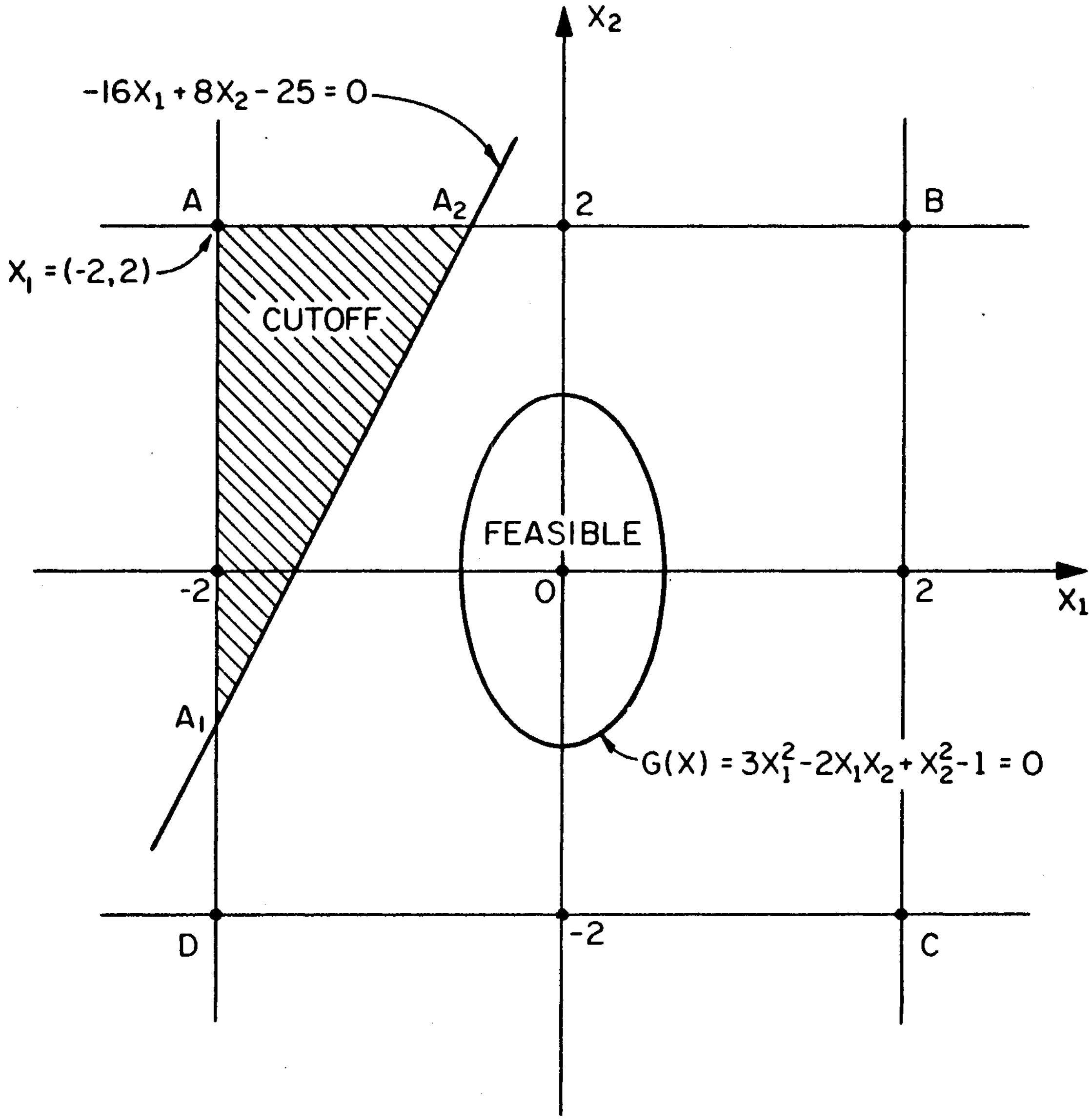


Figure 4. Example Illustrating the Cutting Plane Method

Its solution is $X_1 = (-2,\ 2)$ and $F(X_1) = -4$.

Evaluate the constraint value at the first solution X_1, resulting in

$$\overline{G}(X_1) = -16x_1 + 8x_2 - 25 \leq 0$$

where $\overline{G}(X_1)$ = Taylor series expansion of the constraint G(X), evaluated at the first solution X_1.

In general,

$$\overline{G}(X) \approx G(X_1) + \nabla G(X_1)^T (X - X_1)$$

where ∇ designates gradient, and T means transpose.

It is observed that the initial domain has been reduced from the square ABCD to the polygon A_1A_2BCD or a portion of the initial domain (triangle AA_1A_2) has been cut off (Fig 4). This process of a sequence of linear programming solutions will finally lead to close "contact" with the nonlinear constraint.

Returning to the second modification, two approaches to form a set of tangent lines are considered:

(1) an analytical method, and

(2) a numerical method.

Observing only one constraint (Fig 2) in two-dimensional space (Fig 3), in the analytical approach a normal to the boundary curve is passed through a starting point (feasible point) and at the intersection of this normal with the boundary curve a tangent is considered. (In a multi-dimensional space, curves and tangent lines are replaced by hypersurfaces and tangent hyperplanes.) A parametric representation of the normal in space, the linear representation of the space coordinates, and the equation of the nonlinear constraint constitute a set of simultaneous expressions, a mixture of nonlinear and linear equations. Fortunately, substitutions lead to only one nonlinear equation with only one unknown, the parameter from the parametric expression of the normal in space. This equation can be solved by methods such as Wegstein's (28), Newton's, Bailey's and Aitken's iterations (29), (30), etc. Once this parameter is computed, the coordinates will be known by back substitution. The solution represents the intersection point B desired (Fig 2). At point B a tangent is drawn.

In the numerical approach there is a subroutine available (31) using the Fletcher and Powell numerical method (32). Consider again any one of the constraints in Fig 3 separately as shown in Fig 2. The routine solves an unconstrained problem. It is desired to locate a point B on the constraint function, $G_k = 0$. The routine starts from a feasible point and follows a path of decreasing values of the constraint function. At point A in Fig 2 the constraint function value is $G_k > 0$; at point C its value is $G_k < 0$. Interpolation locates approximately the point B desired, at which $G_k = 0$.

After point B has been obtained, a tangent line can be expressed at this point, e.g., by expanding the constraint function in Taylor series. This process is to be repeated for all other constraints. A closed set of tangent lines is so formed (Fig 3). With this polygon of linear constraints as an approximation of the nonlinear constraint set, Linear Programming can now be applied.

2.6 Computer Code MSLP

The Computer Code MSLP (Modified Sequence of Linear Programming Solutions) is used to solve the optimum design of a gridwork, which mathematically is a Nonlinear Programming problem of the form as Formula (2.1). The objective function is nonlinear, and the constraint set is a mixture of nonlinear and linear functions. The MSLP Code is listed in Appendix A.

The Code is based on Kelly's method (22) with modifications; a sequence of linear programming problems is used as an approximation. In the beginning, a set of hyperplanes tangent to the set of nonlinear constraints is formed. The limitation on the Code is that when a variable has a small range between its upper bound and lower bound it must be made constant, being a close approximation anyway, and that the Code works for convex programming only.

<u>Outline of the MSLP (Fig 5)</u>

MAIN PROGRAM

(1) MAIN

SUBROUTINE	SUBROUTINE	SUBROUTINE
(2) PTCON	(7) BNEG1	(10) FCNGR
(3) FPINTP	(8) CALA2	(11) LFCN
(4) FPM	(9) LP2	
(5) FUNCT		
(6) INTP		

In the Computer Code MSLP, the MAIN program controls the iterations with the aim to generate a sequence of linear programming solutions until convergence. In addition, three groups of subroutines are employed. The first group of subroutines is used to form a closed set of hyperplanes tangent to the nonlinear constraint set. The second group sets up the Mathematical Programming problem so that the Linear Programming technique can be applied in subroutine LP2. The

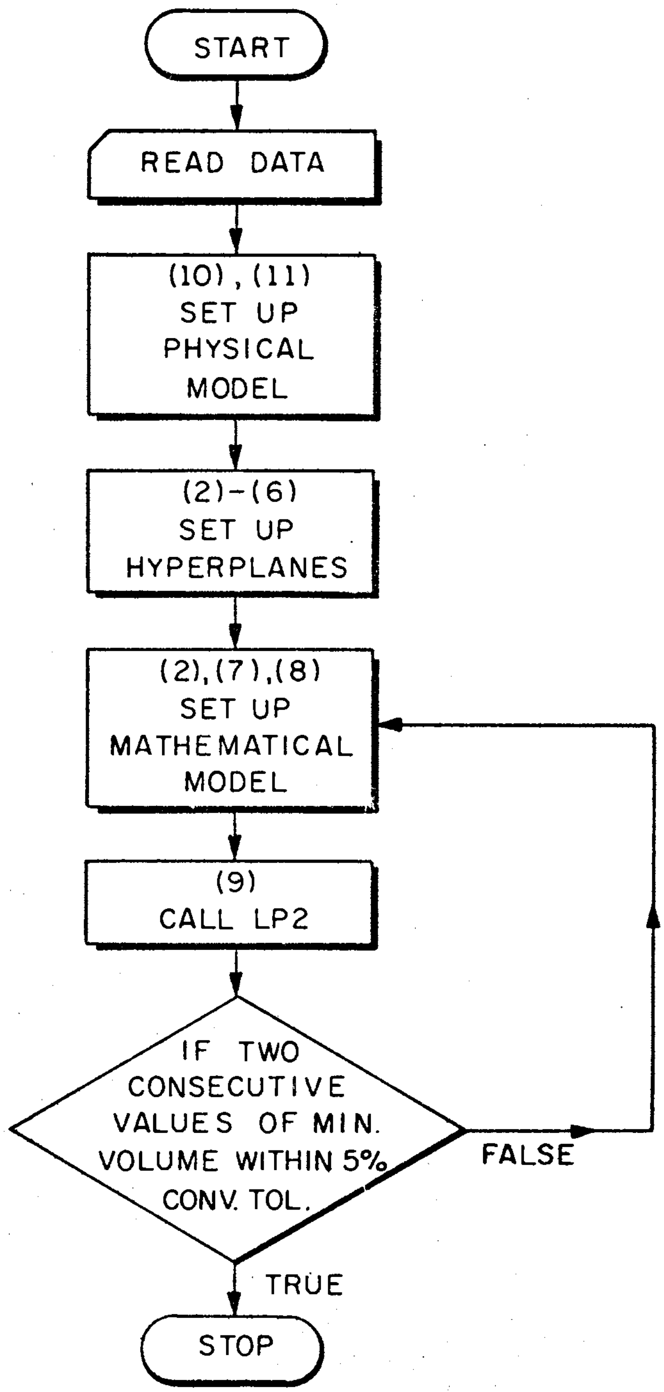


Figure 5. Block Diagram, "Modified Sequence of Linear Programming Solutions Technique"

third group is supplied by the user to calculate values of the functions involved and their first partial derivatives.

The distinct parts of the program are now explained in more detail:

MAIN Program — It reads in data: the initial point, number of variables and number of linear and nonlinear constraints. A closed set of hyperplanes tangent to the set of nonlinear constraints is formed. It goes through a sequence of Linear Programming problems. Upon convergence, the values of minimum objective function F of two consecutive iterations differing by less than five percent, the process is stopped with final solution, final point, and minimum value of objective function F printed out.

<u>First group of subroutines</u>

PTCON — It controls the formulation of Linear Programming, starting with a set of hyperplanes as a constraint set by calling FPINTP.

FPINTP — It furnishes a set of hyperplanes tangent to the nonlinear constraint set.

FPM — This is a modified version of the subroutine FMFP (31) which is an unconstrained function minimization routine due to Fletcher and Power (32).

FUNCT — This is required by FPM to calculate function values and their first partial derivatives.

INTP — As soon as the constraint value steps over zero (Fig 2) to the negative values, this point C and the previous point A (for which the constraint was positive) are interpolated; thus point B is obtained for which the constraint value is approximately equal to zero.

Second group of subroutines

BNEG1 — Owing to the nature of the problem, the constant term b_1 could be negative. The constraints,

$$G_k(X) \geq 0, \quad k = 1, \ldots, i, \tag{2.2}$$

can be rewritten as

$$g_k(X) \geq b_1, \quad k = 1, \ldots, i. \tag{2.3}$$

This subroutine will keep them in this standard form.

CALA2 — It transforms the problem in BNEG1 to the form required by LP2.

LP2 — It is the Linear Programming problem solved by Kuo (27), with modifications.

Third group of subroutines (user-supplied)

FCNGR — Given an initial point, the number of variables and the number of nonlinear constraints, it calculates the values of the objective function, of the nonlinear constraints, and of their first partial derivatives. It is the formulation of the structural design problem.

LFCN — Given the number of variables, the number of linear constraints and the upper and lower bounds, it forms a set of linear constraints.

See Appendix B. The listing of PROBLEM 2 is omitted because it is like PROBLEM 3 with the parameter H and M_{xy} being set equal to zero.

2.7 The Design Procedure

The design procedure can be summarized in the following steps:

Step 1--- Formulate the structural problem by the Theory of Plates, or by other Structural Analysis methods.

Step 2--- Cast it into the form of a Mathematical Programming problem.

Step 3--- Use the gradient technique or other techniques to solve the Mathematical Programming problem.

Step 4--- After the local minimum has been found, one can either be satisfied with the local minimum or try to search for more local minima by using several starting points. The comparison of the local minima can lead to a smaller value of the objective function and in many practical problems even to the global minimum.

The feasibility of this design procedure is evident. After the minimum number of design variables and that of constraints have been determined, one can still apply engineering judgment and reasonable assumptions to reduce that number. This choice has to be made by the designer to see whether the extra computer time could be compensated by the expenses due to extra preparation time and also sacrifice of fully automated procedure. Up to now, Step 4 is probably the best one available for finding a minimum.

2.8 Convexity

Convex programming (33) is defined as one where both the objective function and constraint functions are convex. A function $f(X) = f(x_1, x_2, \ldots, x_j)$, where $X = (x_1, x_2, \ldots, x_j)$ is a vector and $(x_1, x_2, \ldots, x_j)$ are coordinates is a convex function if, for any two points X_A and X_B and for $0 \leq \beta \leq 1$,

$$\beta f(X_A) + (1 - \beta) f(X_B) \geq f\left[\beta X_A + (1 - \beta) X_B\right]. \quad (2.4)$$

Alternately, f(X) is convex if

$$f(X_A) - f(X_B) \geq (X_A - X_B)^T \nabla f(X_B) \quad (2.5)$$

where T means transpose, and ∇ designates gradient.

Figure 6 shows the constraint G_1 being a convex function and the other constraint G_2 being a nonconvex function.

The Sequential Unconstrained Minimization Technique (SUMT) was originally developed for convex programming but it was reported that it also worked for some nonconvex programming problems.

The Ricochet Gradient Method (RGM) always works for convex programming problems but for non-convex programming it might run into difficulties. In the latter case, the Modified Sequence of Linear Programming Solutions (MSLP) might not even work.

The numerical examples in Chapter V will be tested for convexity using the second definition, Formula (2.5).

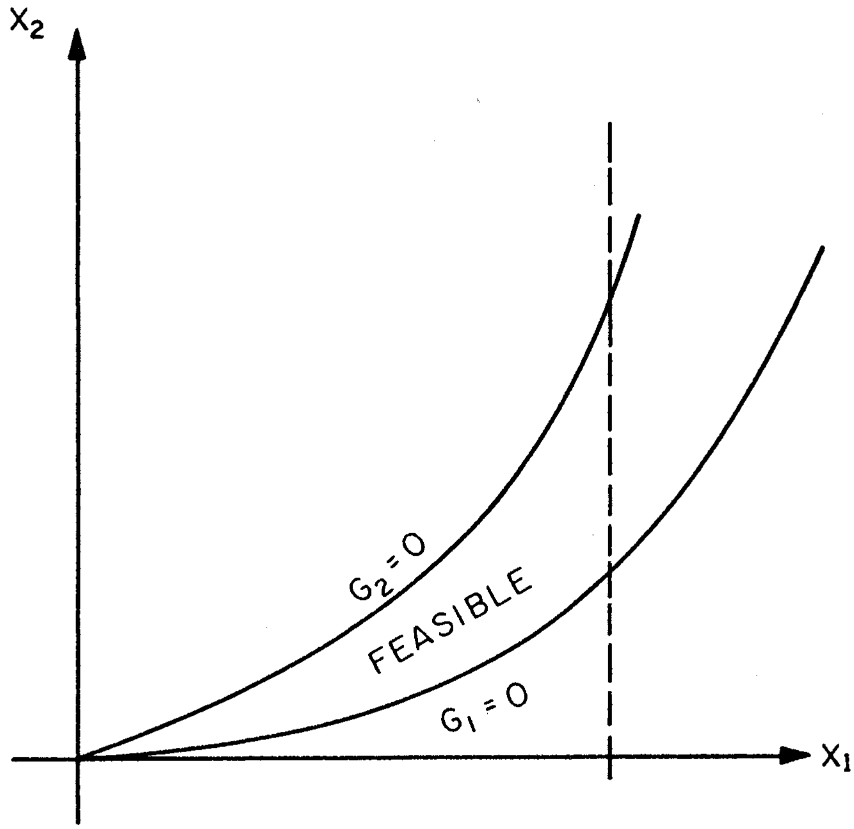


Figure 6. Illustration of Convex (G_1) and Nonconvex (G_2) Functions

CHAPTER III

BEAMS SPANNING IN ONE DIRECTION

PROBLEM 1

3.1 Section Properties

Section properties described in this section are used as design variables. Brown and Ang (12) plotted the section modulus S vs. the moment of inertia I, using the WF rolled sections in the American Institute of Steel Construction Manual. By fitting a curve over the plot, the discrete relationship was changed to a continuous function, S = S(I).

However, a more general approach can be used, appropriate for both welded I-sections and rolled beams. Tochacek (34) expressed all section properties for an unsymmetric I-section in terms of three parameters: Beam asymmetry α, web slenderness λ, and web coefficient K.

$$S_1 = \frac{6\alpha - K(\alpha + 1)^2}{6(\alpha + 1)} (\lambda K A_t^3)^{0.5} \tag{3.1}$$

$$S_2 = \frac{S_1}{\alpha} \tag{3.2}$$

$$A_w = KA_t \tag{3.3}$$

$$I = \frac{6\alpha - K(\alpha + 1)^2}{6(\alpha + 1)^2} \lambda K A_t^2 \tag{3.4}$$

where

$$S_1 = \frac{I}{e_1} \quad , \quad S_2 = \frac{I}{e_2} \; , \tag{3.5}$$

S_1, S_2 = section moduli for the top fibers 1 and bottom fibers 2,

I = moment of inertia with respect to the horizontal principal axis,

e_1, e_2 = the distances from the neutral axis to the top and bottom fibers,

$\alpha = \frac{S_1}{S_2} = \frac{e_2}{e_1}$ = section asymmetry,

$$\lambda = \frac{h}{\delta} \; , \tag{3.6}$$

λ = web slenderness, i.e., web depth-thickness ratio,

h = web depth,

δ = web thickness,

A_t = total cross-sectional area,

$$A_w = h\delta, \tag{3.7}$$

A_w = web cross-sectional area,

$K = \frac{A_w}{A_t}$ = web coefficient.

Assuming equal allowable stress σ_{all} for the bottom as well as for the top fibers, the I-section becomes symmetrical and $\alpha = 1$ or $S_1 = S_2 = S$. Formula (3.1) reduces to

$$S = (0.5 - 0.333K)(\lambda K A_t^3)^{0.5}. \tag{3.8}$$

The strength condition for flexure reads:

$$\sigma = \frac{M}{S} \leq \sigma_{all} \tag{3.9}$$

where

σ = maximum bending stress,

σ_{all} = allowable bending stress,

M = maximum bending moment,

S = section modulus.

For the fully stressed, most economical design,

$$M = \sigma_{all} S = \sigma_{all} (0.5 - 0.333K)(\lambda K A_t^3)^{0.5} . \qquad (3.10)$$

A constant cross-sectional area A_t is assumed throughout the beam span and for the beams used. For the given amount of material (A_t = constant), the maximum resisting moment M is achieved if

$K = 0.5$,

λ = maximum,

the latter value to be discussed later.

Substituting $\alpha = 1$, $K = 0.5$ into Formulas (3.4) and (3.8),

$$I = \frac{0.5 \lambda A_t^2}{6} , \qquad (3.11)$$

or

$$A_t = 3.46 \frac{I^{0.5}}{\lambda^{0.5}} , \qquad (3.12)$$

$$S = 0.235 \lambda^{0.5} A_t^{1.5} , \qquad (3.13)$$

or from Formula (3.12)

$$S = 1.51 \frac{I^{0.75}}{\lambda^{0.25}} . \qquad (3.14)$$

As obvious from Formulas (3.12) and (3.14) two independent section properties, i.e., I and λ, will be decisive for the sizing of beams, besides other parameters described later, e.g., b, $\frac{A}{B}$, etc.

3.2 Design Variables

It is desired to span beams in one direction over a rectangular area of specified dimensions A x B (Fig 7) in such a manner that the total volume of beams will be minimum. Such a design is known as an optimum design. The design procedure starts out with specifications to satisfy safety and functional requirements, goes through a physical model which is transformed into a mathematical model, and ends up with a solution.

Given: A = dimension in the x-direction of a given rectangular area spanned by beams,

B = dimension in the y-direction of a given rectangular area spanned by beams,

p = UDL = uniformly distributed load,

Simple supports,

Symmetrical I-shaped welded plate girders (or possibly WF-section or another rolled section).

Required: to optimize the location and the size of beams using the following variables:

b = spacing of beams oriented in the x-direction ("x-beams"),

I_x = moment of inertia of the x-beam with respect to its horizontal principal axis,

λ_x = web depth-thickness ratio of the x-beam (web slenderness).

3.3 Design Criteria

Although other secondary criteria such as shear and lateral buckling could be included as constraints, only three primary criteria for a safe design would be considered:

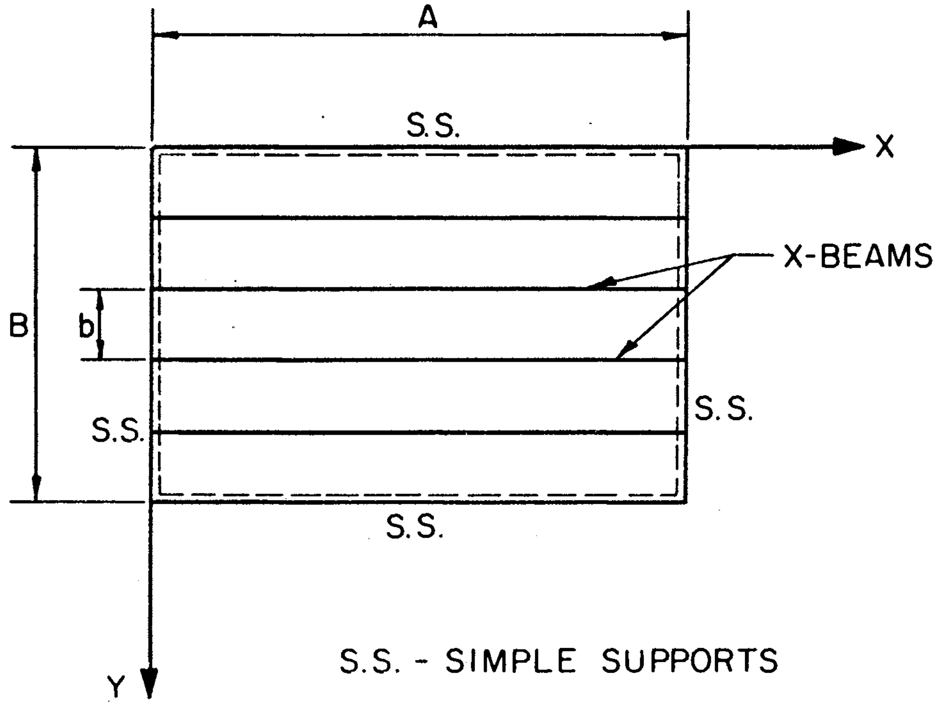


Figure 7. Beams Spanning in One Direction

(1) flexural strength,

(2) deflection,

(3) web buckling.

Shear can be checked by supplemental calculations, and lateral buckling is frequently insignificant in gridworks with compressive flanges supported laterally by slabs and the like.

3.3.1 Flexural Strength

The first criterion as a constraint implies that

$$\sigma_{all} - \sigma \geq 0 \tag{3.15}$$

rewritten from Formula (3.9). Maximum moment

$$M = C_1 p b A^2 \tag{3.16}$$

where for simple supports and for uniformly distributed load UDL = p, $C_1 = \frac{1}{8}$.

Symbols b and A have been explained in Section 3.2. Loadings other than a uniform load will be discussed in Chapter VI. Substitution of Formulas (3.14) and (3.16) in Formula (3.9) yields the first constraint,

$$G_1 = 1.51 \frac{I_x^{0.75}}{\lambda_x^{0.25}} - \frac{C_1 p b A^2}{\sigma_{all}} \geq 0. \tag{3.17}$$

3.3.2 Deflection

The maximum deflection of a beam should not exceed the allowable deflection; i.e.,

$$w_{all} - w \geq 0. \tag{3.18}$$

The maximum deflection,

$$w = C_2 \frac{pbA^4}{EI_x} \qquad (3.19)$$

where $C_2 = \frac{5}{384}$ for uniformly distributed load UDL = p and simple supports,

E = Young's modulus of elasticity.

The allowable deflection,

$$w_{all} = C_3 A \qquad (3.20)$$

with C_3 = coefficient for deflection specified in building codes. In the forthcoming numerical calculations, mainly the value of 1/360 will be considered as a characteristic one.

Substitution of Formulas (3.19) and (3.20) in Formula (3.18) leads to the second constraint,

$$G_2 = C_3 A - \frac{C_2 pbA^4}{EI_x} \geq 0. \qquad (3.21)$$

3.3.3 Web Buckling

The third criterion considered is web buckling. Buckling problems have been investigated by numerous authors. Some results are accumulated, e.g., in the book by Beedle, et al (35), or in the USS Steel Design Manual (36). The girder web can be treated as a plate with the in-plane loading as in (37). The main result of the solution is the critical stress σ_{cr}, under which an ideal plate would buckle. Because of the post-critical reserve in the load-carrying capacity of a compressed plate, the safety factor 1.0 for σ_{cr} is sufficient in the elastic range. Then the buckling condition reads

$$\sigma_{cr} - \sigma \geq 0 \qquad (3.22)$$

where the critical buckling stress

$$\sigma_{cr} = K_1 \frac{\pi^2 E}{12(1 - \nu^2)\lambda^2} \tag{3.23}$$

with

π = 3.1416,

ν = Poisson's ratio,

K_1 = coefficient depending on the stress distribution and support condition of unstiffened web: the value of 23.9 (36) is considered in the forthcoming numerical examples, corresponding to the unstiffened, simply supported web with the typical linear bending stress distribution.

Let

$$C_4 = K_1 \frac{\pi^2 E}{12(1 - \nu^2)} \tag{3.24}$$

then the critical stress in Formula (3.23) is

$$\sigma_{cr} = C_4 \frac{1}{\lambda^2} \tag{3.25}$$

and Formula (3.22) reads

$$\frac{C_4}{\lambda^2} - \frac{M}{S} \geq 0 . \tag{3.26}$$

From Formulas (3.14) and (3.16) and rearranging, the third constraint results in

$$G_3 = \frac{C_4}{\lambda^2} 1.51 \frac{I_x^{0.75}}{\lambda_x^{0.25}} - C_1 p b A^2 \geq 0 . \tag{3.27}$$

3.3.4 Objective Function

The objective of this chapter is to minimize the total volume V of all beams used over the area A x B. Hence, it is the volume V that has the character of the objective function F as explained in Section 2.3.

If two beams are located on the two side boundaries (e.g., supported by corner columns), then the total number of beams is equal to $(\frac{B}{b} + 1)$. If the beams are placed only inside the area but not on the sides, then the total number of beams is equal to $(\frac{B}{b} - 1)$. As the side beams are usually not employed, frequently being replaced by walls, only the latter case will be considered. With the girder area A_t from Formula (3.12), the total volume V of all beams used is

$$V = (\frac{B}{b} - 1)(3.46 \frac{I_x^{0.5}}{\lambda_x^{0.5}}) A . \qquad (3.28)$$

3.4 Mathematical Model

From Formulas (3.28), (3.17), (3.21), and (3.27), the problem can be formulated in a mathematical model as follows:

PROBLEM 1

Minimize the objective function

$$F(X) = V = (\frac{B}{b} - 1)(3.46 \frac{I_x^{0.5}}{\lambda_x^{0.5}}) A \qquad (3.28)$$

subject to the constraints

$$G_1 = 1.51 \frac{I_x^{0.75}}{\lambda_x^{0.25}} - \frac{C_1 p b A^2}{\sigma_{all}} \geq 0 \qquad (3.17)$$

$$G_2 = C_3 A - \frac{C_2 p b A^4}{E I_x} \geq 0 \qquad (3.21)$$

$$G_3 = \frac{C_4}{\lambda^2} 1.51 \frac{I_x^{0.75}}{\lambda_x^{0.25}} - C_1 p b A^2 \geq 0 \qquad (3.27)$$

with the variables $x_i = b, I_x, \lambda_x$ for $i = 1, 2, 3$ satisfying the non-negativity constraints,

$$\frac{B}{2} \geq b \geq 0, \tag{3.29}$$

$$I_x \geq 0, \tag{3.30}$$

$$\lambda_x \geq 0. \tag{3.31}$$

It is observed that this is a Nonlinear Programming problem with both the objective function and the constraints nonlinear. First the Modified Sequence of Linear Programming Solutions (MSLP) computer program was applied to the problem described in Ex (5.6) without the upper bound of Formula (3.29), but failed, thus indicating an unbounded solution. It was evident that an open set caused the trouble. Also, the lower bounds equalling zero in Formulas (3.29) to (3.31) are not acceptable for practical reasons; e.g., $b = 0$ (zero spacing between two beams) represents a continuous structural system (a slab), for which some applied expressions do not hold true anymore.

The upper and lower bounds on the variables are further considered in Section 3.5. Being subject to removal if necessary, constraints of Formulas (3.29) to (3.31) can generally be reformulated as

$$BL(1) \leq b \leq UB(1), \tag{3.32}$$

$$BL(2) \leq I_x \leq UB(2), \tag{3.33}$$

$$BL(3) \leq \lambda_x \leq UB(3). \tag{3.34}$$

These lower and upper bounds form a large envelope as the starting basis of the Cutting Plane Method (Fig 4). Although a more rigorous investigation of the need of all these bounds in Section 3.5 reveals that only an upper bound on b (or on another variable as in Fig 6) is absolutely necessary, together with one lower bound on any of the three variables (b, I_x, λ_x), the other bounds turn out to be helpful rather than obstructive, enabling one to observe some practical requirements.

3.5 Analytical Approach

3.5.1 Parameters in the Objective Function

The relative simplicity of PROBLEM 1 enables an analytic solution, permitting one to derive some quantitative results useful for checking numerical solutions of more involved problems. The section properties, derived from Formulas (3.1) to (3.4), originated from three parameters: α, λ, and K. For symmetry, $\alpha = 1$, Formula (3.8) holds true. The strength condition reads

$$\sigma = \frac{M}{S} = C\sigma_{all} \qquad (3.35)$$

where

C is the parameter changing the constraints from inequality to equality, and

$C \leq 1$.

Here the inequality sign indicates that the design is not fully stressed (e.g., when deflection or buckling is decisive rather than flexural strength). Substituting Formula (3.8) into Formula (3.35), the resisting moment

$$M = C\sigma_{all}(0.5 - 0.333K)(\lambda K A_t^3)^{0.5}. \qquad (3.36)$$

Letting $s = \frac{B}{b}$, and rewriting Formula (3.16), the maximum bending moment

$$M = C_1 p A^2 \frac{B}{s}. \qquad (3.37)$$

Equating Formula (3.36) and Formula (3.37) and rearranging,

$$A_t^{\,3} = \left(\frac{C_1 p A^2 B}{\sigma_{all}}\right)^2 \frac{1}{C^2 \lambda s^2} \frac{1}{K(0.5 - 0.333K)^2} . \qquad (3.38)$$

Rewriting Formula (3. 28),

$$V = (s - 1) A_t A , \qquad (3.39)$$

or

$$\left(\frac{V}{A}\right)^3 = (s - 1)^3 A_t^{\,3} . \qquad (3.40)$$

Multiplying both sides by a constant,

$$\left(\frac{\sigma_{all}}{C_1 p A^2 B}\right)^2 \left(\frac{V}{A}\right)^3 = (s - 1)^3 A_t^{\,3} \left(\frac{\sigma_{all}}{C_1 p A^2 B}\right)^2 . \qquad (3.41)$$

Since the term on the left-hand side of Formula (3. 41) is the cube of **the objective** function V^3 premultiplied by a constant, it can be called a decision function D, and

$$D = (s - 1)^3 A_t^{\,3} \left(\frac{\sigma_{all}}{C_1 p A^2 B}\right)^2 . \qquad (3.42)$$

Multiplying both sides of Formula (3. 38) by $(s - 1)^3 \left(\frac{\sigma_{all}}{C_1 p A^2 B}\right)^2$,

$$(s - 1)^3 A_t^{\,3} \left(\frac{\sigma_{all}}{C_1 p A^2 B}\right)^2 = \frac{(s - 1)^3}{C^2 \lambda s^2} \frac{1}{K(0.5 - 0.333K)^2} . \qquad (3.43)$$

Comparing Formula (3. 42) with (3. 43), the decision function finally reads

$$D = \frac{(s - 1)^3}{C^2 \lambda s^2} \frac{1}{K(0.5 - 0.333K)^2} . \qquad (3.44)$$

The decision function D and, hence, also the volume of material decreases with the increasing values of C, λ, $K(0.5 - 0.333K)^2$ and $\frac{s^2}{(s - 1)^3}$. The hints for proportioning the section then are as follows:

(1) Use fully stressed design ($C = 1$) wherever possible. In other cases, use the highest value of $C < 1$, as permitted by deflection of Formula (3. 51).

(2) Use webs with maximum possible slenderness λ, as permitted by web buckling.

(3) The expression $K(0.5 - 0.333K)^2$ assumes its maximum for $K = 0.5$.

(4) The value of $s = \frac{B}{b}$ for $s = 2, 3, 4, \ldots$ should be as low as possible or b as great as possible. The limit is $s = 2$ or $b = \frac{B}{2}$. This is based on the arrangement without the side beams (Section 3.3.4). Similarly, this also holds true for the other arrangement. The value of $b < \frac{B}{2}$ is determined usually by practical considerations, e. g., by the length of available floor slabs.

3.5.2 Reduction in Constraints

The decision function D (Formula 3. 44) has depended on four variables (C, λ, $b = \frac{B}{s}$, K). As the values of b and K have been determined ($K = 0.5$, b as great as practical considerations permit) only two unknowns, C and λ, should be found to satisfy the constraints

$$G_1 (I, \lambda) = G_1 (C, \lambda) \geq 0 \qquad \text{flexural strength} \qquad (3.45)$$

$$G_2 (I) = G_2 (C) \geq 0 \qquad \text{deflection} \qquad (3.46)$$

$$G_3 (I, \lambda) = G_3 (C, \lambda) \geq 0 \qquad \text{web buckling.} \qquad (3.47)$$

Variables I and C are correlated through Formulas (3. 14) and (3. 35).

As three expressions (3. 45) to (3. 47) for only two unknowns, C and λ, are available, just two of them can be equalities.

	Case (1)	Case (2)	Case (3)	
G_1	=	>	=	0
G_2	>	=	=	0
G_3	=	=	>	0

It depends on the given data, which pair would be chosen. Instead of using a method of Mathematical Programming to solve the problem, one can employ also another procedure (analytical approach): Eliminate one variable, e.g., λ, and use the constraints to find the remaining variable C. Then, determine λ by back substitution.

Case (1)

If the equality sign is considered in Formula (3.47), (Formula 3.22),

$$\sigma_{cr} = C\sigma_{all} \tag{3.48}$$

then the constraint G_3 can be eliminated. From Formulas (3.26) and (3.35), the expression for the web slenderness follows:

$$\lambda = \left(\frac{C_4}{C\sigma_{all}}\right)^{0.5} . \tag{3.49}$$

The problem has been reduced to the determination of C. For a fully stressed design, with the equality sign in Formula (3.45),

$$C = 1.0. \tag{3.50}$$

Case (2)

$$C_{11} \leq C \leq 1.0 \tag{3.51}$$

as much as the constraint G_2 (Formula 3.46) for deflection allows. The limit C_{11} follows from G_2 (with equality sign in Formula 3.46). See Appendix C, Formula (C 23), where

$$C_{11} = \left(\frac{C_1^{\frac{13}{18}} C_4^{\frac{1}{9}}}{A^{\frac{5}{9}} \sigma_{all}} \right) \left(\frac{0.354\, E\, C_3}{C_2} \right)^{\frac{2}{3}} \left(\frac{pb}{0.00305} \right)^{\frac{1}{18}} . \quad \text{(C 23)}$$

Case (3)

This case, with equality signs for G_1 and G_2 and an inequality sign for G_3, yields a constant C_{12} and hence I, where in Appendix D

$$C_{12} = \frac{2\, C_2\, \sigma_{all}}{C_1\, C_3\, E} , \quad \text{(D 4)}$$

$$I = \frac{C_1\, p\, b\, A^3\, C_{12}}{2\sigma_{all}} . \quad \text{(D 6)}$$

For a chosen feasible value of λ, there is a corresponding value of σ_{cr} (Formula 3.23) such that $G_3 > 0$; and there is a corresponding value of volume since $A_t = f(I, \lambda)$ from Formula (3.12).

The result obtained (three cases) demonstrates that it is not always the fully stressed design which is the optimum, the fully stressed design being not applicable for other conditions.

3.5.3 Steps of Computation

The computation can now be summarized in the following steps:

Step 1 The spacing b should be as great as the design allows but less than or equal to $\frac{B}{2}$. The web coefficient $K = 0.5$.

Step 2 For the given data, calculate C where $C = \text{Min}(1.0, C_{11})$ from Formula (3.51). The three cases in Section 3.5.2 are:

Case (1)

If $C = 1.0$, flexural strength is critical. Go to Step 3 to calculate λ and I.

Case (2)

If $C = C_{11}$, deflection is critical. Go to Step 3.

Case (3)

If web buckling is not critical, go to Step 4.

Step 3 $C = 1.0$ or $C = C_{11}$: Calculate λ from Formula (3.49) and I from Formula (3.17) for Case (1) or from Formula (3.21) for Case (2).

Step 4 $C = C_{12}$: Calculate I from

$$I = \frac{C_1 p b A^3 C_{12}}{2\sigma_{all}} . \qquad (D\ 6)$$

For a chosen feasible value of λ, calculate σ_{cr} (Formula 3.23), A_t (Formula 3.12), and volume (Formula 3.28).

This analytical approach is tested numerically by considering five examples, Exs (5.1) to (5.5), in Chapter V. One of these examples (Ex 5.3) is checked by the Code MSLP (Modified Sequence of Linear Programming Solutions Technique) as **Ex (5.6)**.

CHAPTER IV

A GRID WITH BEAMS SPANNING IN TWO ORTHOGONAL DIRECTIONS--SIMPLE-CONNECTION (PROBLEM 2), RIGID-CONNECTION (PROBLEM 3)

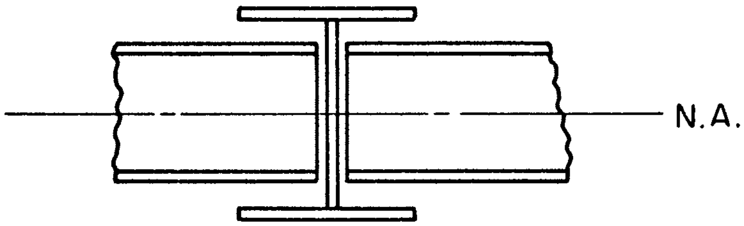
4.1 Introduction

The analysis of a grid requires more than the Theory of Beams as in Chapter III. Since the Theory of Plates (41) is adopted, the governing equation and its solution will lead to the formulation of a mathematical model.

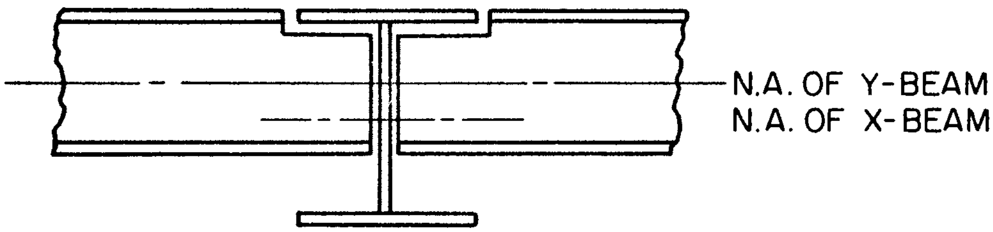
General expressions are derived to include torsional rigidities at the joints (Rigid-Connections). A simpler case is investigated first, namely the Simple-Connection. By Simple-Connection it is meant that girders are connected in such a way that the joint does not offer any torsional resistance. In such a case, hereafter referred to as PROBLEM 2, torsion in the expressions is excluded. On the other hand, the Rigid-Connection is supposed to provide torsional resistance and is hereafter referred to as PROBLEM 3.

With respect to the mutual position of the neutral axes of the intersecting beams, there are generally three types of arrangements at the joint, as shown in Figs 8(a) to (c) which are self-explanatory.

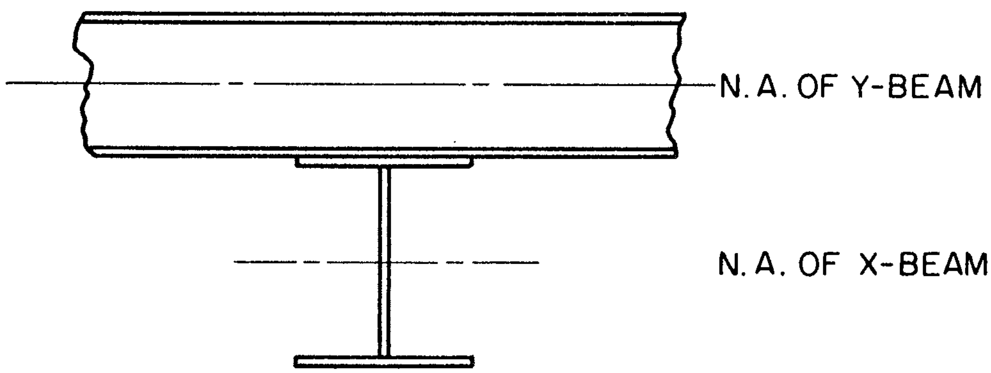
The Theory of Plates assumes the first type (Fig 8a) although it takes on the slight eccentricity in the second type (Fig 8b) without much error. As a matter of fact, most orthotropic-plate bridge decks are



(a)



(b)



(c)

Figure 8. (a), (b), and (c): Types 1, 2, and 3

built with the second type. The most drastic eccentricity is the third type (Fig 8c). However, it is an approximation good enough to warrant the application of the theory derived for the first type (Fig 8a) to all three types. It is important to distinguish the various arrangements when strength conditions (bending stresses) are applied. The second type is used throughout this study.

In general, the subscripts "x, y" denote the x- and y-direction, respectively. The parameters, with or without the subscript "x," used in Chapter III can be used here in Chapter IV for the x-direction and also for the y-direction with the subscripts changed from "x" to "y."

4.2 Geometry of the Grid

Let σ_{jt} = combined stress at a joint,

σ_x = bending stress in x-beam, and

σ_y = bending stress in y-beam.

Since the grid is assumed to be a continuous system, it will be conservative enough (more than safe) to consider σ_{jt}, σ_x, and σ_y at the center for this case as the strength criteria. These criteria will appear as constraints G(7), G(1), and G(2), respectively. In order to clarify this statement of "conservative enough," consider for a moment the grid as a discrete system. The four cases in Figs (9) to (12), which are self-explanatory, among in-between situations will occur in the process of Nonlinear Programming (NLP) being applied.

From a simple speculation, it is observed that the maximum of σ_{jt}, σ_x, and σ_y evaluated at the center (for this symmetric case, PROBLEM 3) is greater than any flexural stress evaluated at any critical points in all the four cases.

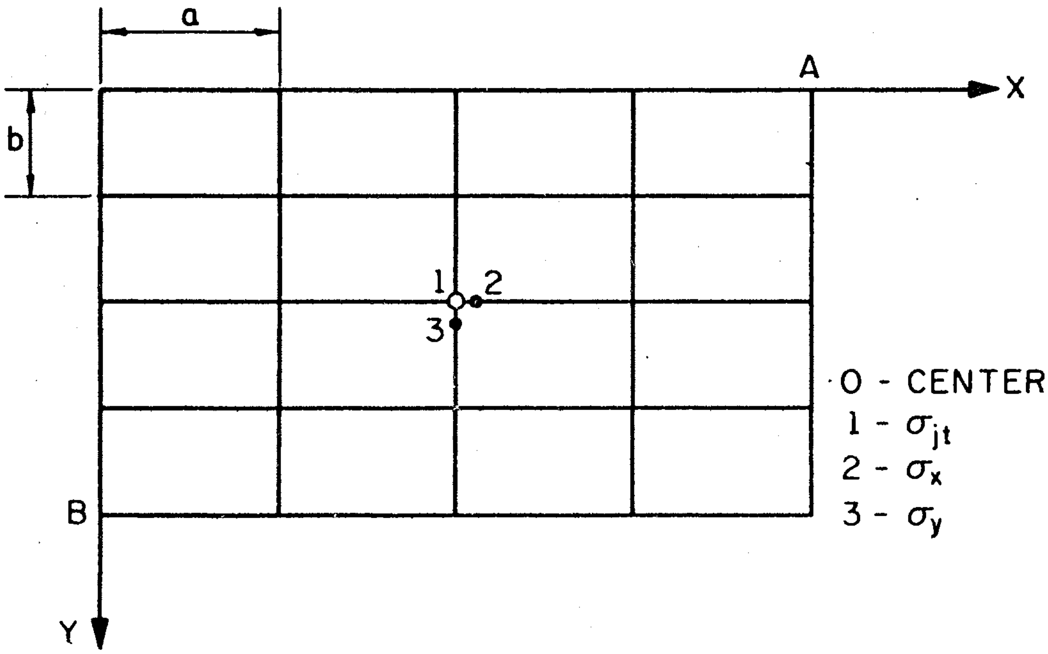


Figure 9. Case 1

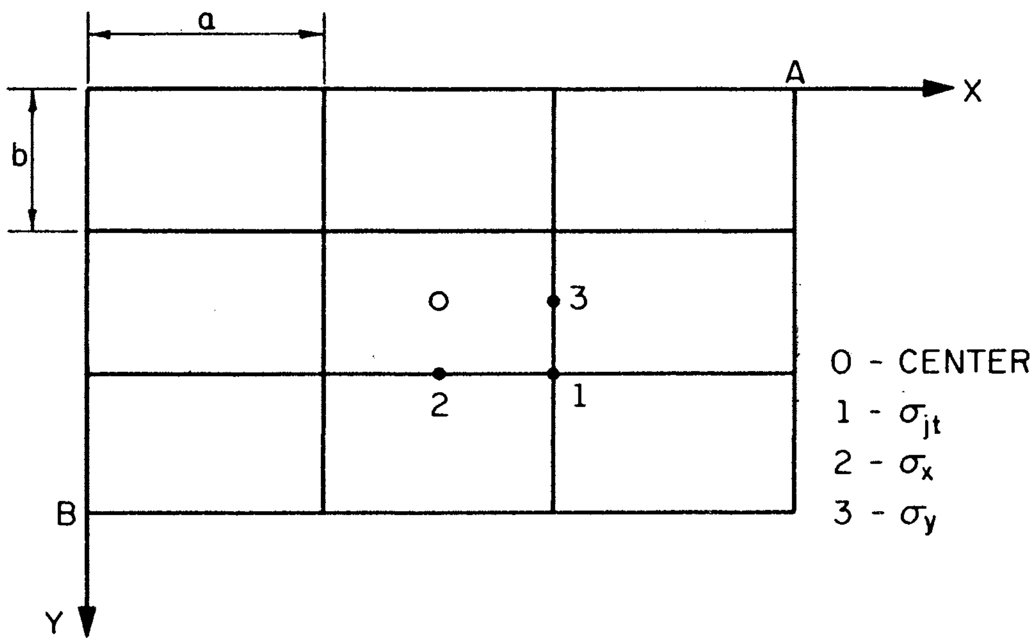


Figure 10. Case 2

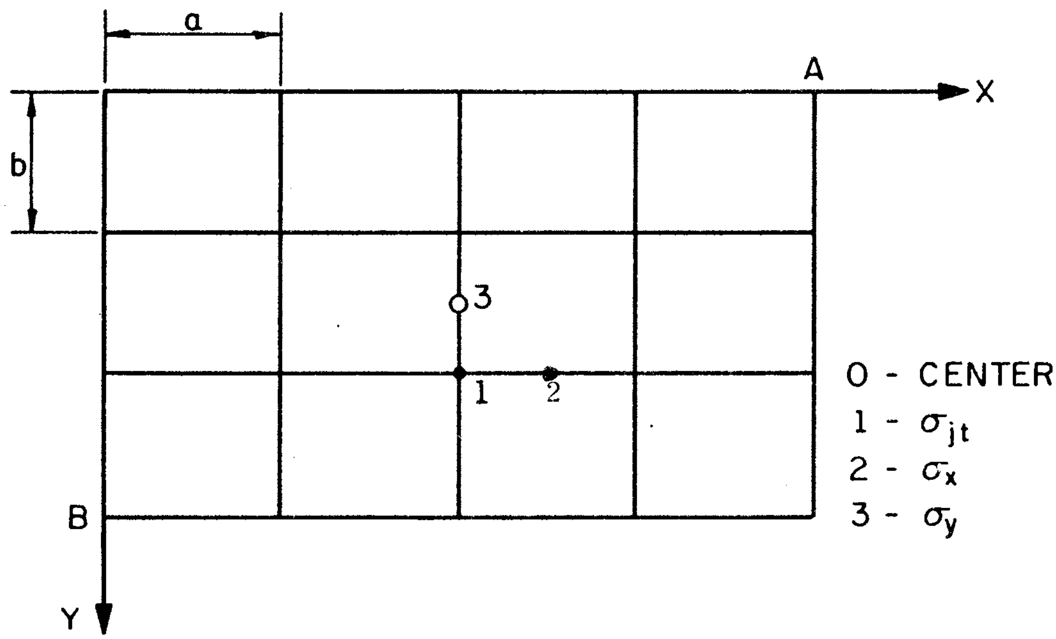


Figure 11. Case 3

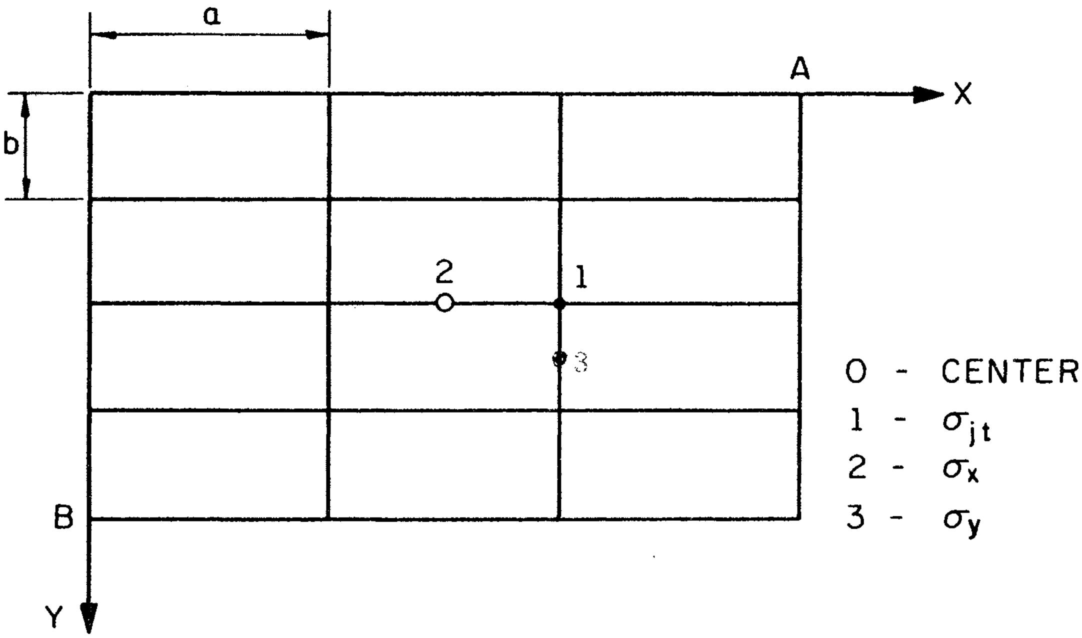


Figure 12. Case 4

As this conservative approach is safe as well as in accordance with the assumption of a continuous system, it will be adopted, although a rigorous analysis can be used if one chooses. A similar argument goes for the deflection and web buckling criteria. However, in the web buckling criterion, since web buckling unlikely occurs at the joint because of mutual reinforcement, the σ_{jt} will be omitted.

4.3 Governing Equation

From the analysis of an orthotropic plate (38), the bending moments per unit width are

$$M_x = -(D_x w,_{xx} + D_1 w,_{yy}) \tag{4.1}$$

$$M_y = -(D_y w,_{yy} + D_1 w,_{xx}) \tag{4.2}$$

and the torsional moment per unit width

$$M_{xy} = 2D_{xy} w,_{xy} = -M_{yx}. \tag{4.3}$$

The governing equation results in the following form:

$$D_x w,_{xxxx} + 2Hw,_{xxyy} + D_y w,_{yyyy} = q \tag{4.4}$$

where the subscript designates the direction and the comma its derivative. Let

$(\ ),_x$ through $(\ ),_{xxxx}$ = 1st through 4th partial derivatives with respect to x,

$(\ ),_y$ through $(\ ),_{yyyy}$ = 1st through 4th partial derivatives with respect to y,

$(\ ),_{xy}$ and $(\ ),_{xxyy}$ = 2nd and 4th mixed partial derivatives with respect to x and y,

M_x = bending moment per unit width in x-beam,

M_y = bending moment per unit width in y-beam,

M_{xy} = torsional moment per unit width in x-beam,

M_{yx} = torsional moment per unit width in y-beam,

$q(x, y)$	=	loading function per unit area,
w	=	deflection,
D_x, D_y	=	unit flexural rigidities,
H, D_1, D_{xy}	=	parameters associated with unit torsional rigidities.

4.4 Deflections and Moments

The rectangular grid (Fig 13), with sides A by B, is under consideration. If

$F_x = EI_x$ = flexural rigidity of each x-beam, and

$F_y = EI_y$ = flexural rigidity of each y-beam,

then their unit flexural rigidities are

$$D_x = \frac{F_x}{b} \qquad (4.5)$$

$$D_y = \frac{F_y}{a} \qquad (4.6)$$

where

a = spacing of beams oriented in the y-direction ("y-beams"),

b = spacing of beams oriented in the x-direction ("x-beams").

The parameter H is defined as

$$2H = \frac{C_x}{b} + \frac{C_y}{a}\,. \qquad (4.7)$$

C_x and C_y will be explained in Section 4.5.2.

The Navier solution (38) for the plate, uniformly loaded and simply supported on all sides (Fig 13), is the deflection

$$w(x, y) = \sum_{m=1,3,..}^{\infty} \sum_{n=1,3,..}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{A}\right) \sin\left(\frac{n\pi y}{B}\right) \qquad (4.8)$$

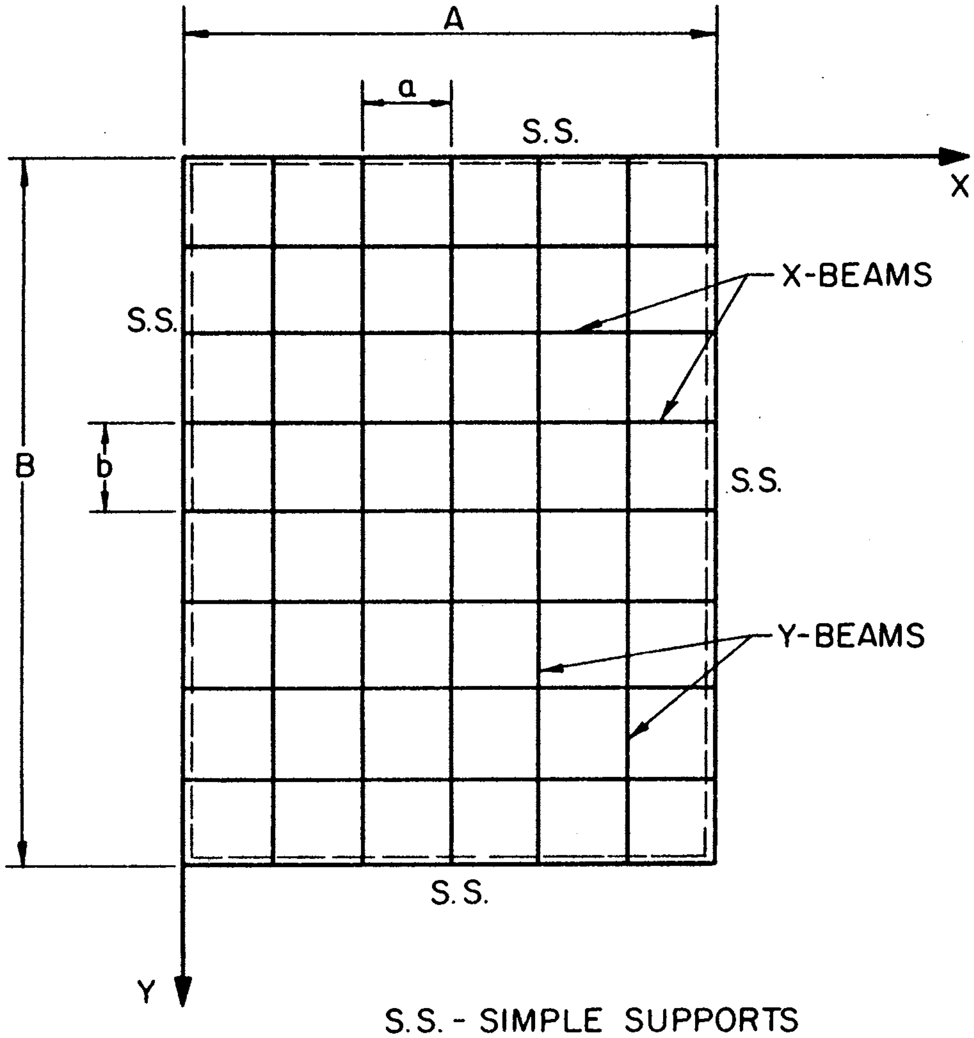


Figure 13. A Grid Simply-Supported on All Four Sides

where A_{mn} = parameters associated with the assumed solution of deflection, and

$$A_{mn} = \frac{16p}{\pi^6} \frac{1}{mn\left[\left(\frac{m}{A}\right)^4 D_x + \left(\frac{mn}{AB}\right)^2 2H + \left(\frac{n}{B}\right)^4 D_y\right]} \qquad (4.9)$$

or

$$w = \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} \frac{\sin\left(\frac{m\pi x}{A}\right) \sin\left(\frac{n\pi y}{B}\right)}{mn\left[\left(\frac{m}{A}\right)^4 D_x + \left(\frac{mn}{AB}\right)^2 2H + \left(\frac{n}{B}\right)^4 D_y\right]} \; . \qquad (4.10)$$

Derivatives needed for calculations of moments are:

$$w,_{xx} = \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} \frac{-\left(\frac{m\pi}{A}\right)^2 \sin\left(\frac{m\pi x}{A}\right) \sin\left(\frac{n\pi y}{B}\right)}{mn\left[\left(\frac{m}{A}\right)^4 D_x + \left(\frac{mn}{AB}\right)^2 2H + \left(\frac{n}{B}\right)^4 D_y\right]} \qquad (4.11)$$

$$w,_{yy} = \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} \frac{-\left(\frac{n\pi}{B}\right)^2 \sin\left(\frac{m\pi x}{A}\right) \sin\left(\frac{n\pi y}{B}\right)}{mn\left[\left(\frac{m}{A}\right)^4 D_x + \left(\frac{mn}{AB}\right)^2 2H + \left(\frac{n}{B}\right)^4 D_y\right]} \qquad (4.12)$$

$$w,_{xy} = \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} \frac{+\left(\frac{m\pi}{A}\right)\left(\frac{n\pi}{B}\right) \cos\left(\frac{m\pi x}{A}\right) \cos\left(\frac{n\pi y}{B}\right)}{mn\left[\left(\frac{m}{A}\right)^4 D_x + \left(\frac{mn}{AB}\right)^2 2H + \left(\frac{n}{B}\right)^4 D_y\right]} \; . \qquad (4.13)$$

As in Chapter III, the same three criteria are considered here but twofold, once for x-beams and once for y-beams. With regard to moments, Timoshenko (38) suggested for a fine-meshed grid to assume a parabolic moment variation among three adjacent x-beams (Fig 14)

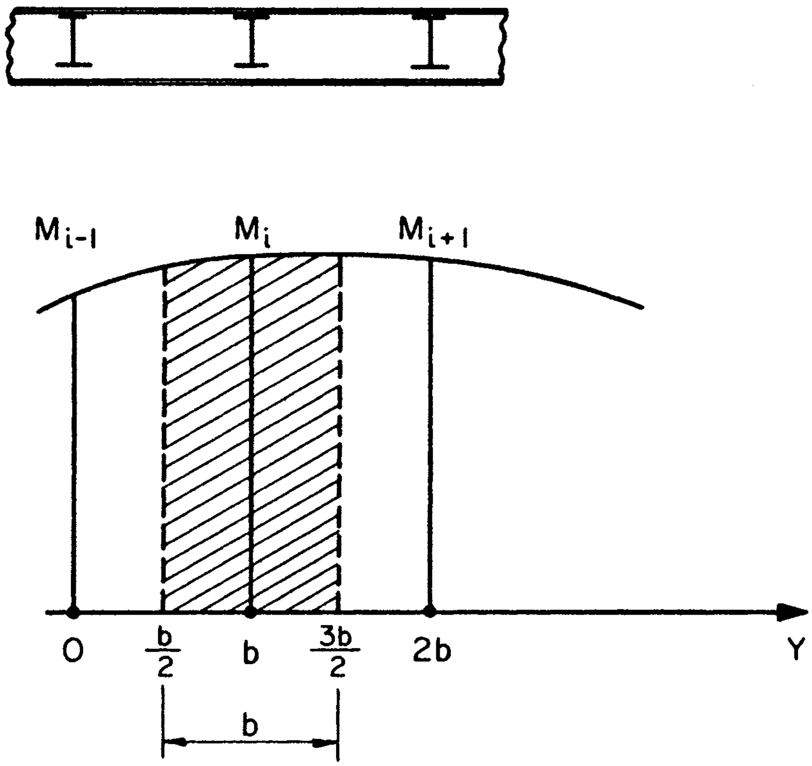


Figure 14. Locations of Three Typical x-Beams

with three ordinates M_{i-1}, M_i, and M_{i+1} at y = 0, b, 2b, respectively. A similar assumption can be made for the x-direction. The total moment, covering a strip of width b, being centered at M_i, is then

$$M_{tot} = \frac{b}{24}(M_{i-1} + 22M_i + M_{i+1}) \tag{4.14}$$

$$M_i = -D_x (w,_{xx})_i = \frac{-F_x}{b}(w,_{xx})_i \tag{4.15}$$

$$M_x = \frac{-F_x}{24}\left[(w,_{xx})_{i-1} + 22(w,_{xx})_i + (w,_{xx})_{i+1}\right]. \tag{4.16}$$

Similarly,

$$M_y = \frac{-F_y}{24}\left[(w,_{yy})_{i-1} + 22(w,_{yy})_i + (w,_{yy})_{i+1}\right] \tag{4.17}$$

$$M_{xy} = \frac{C_x}{24}\left[(w,_{xy})_{i-1} + 22(w,_{xy})_i + (w,_{xy})_{i+1}\right] \tag{4.18}$$

$$M_{yx} = \frac{-C_y}{24}\left[(w,_{xy})_{i-1} + 22(w,_{xy})_i + (w,_{xy})_{i+1}\right]. \tag{4.19}$$

At the center $(\frac{A}{2}, \frac{B}{2})$, Formula (4, 16) becomes

$$\max M_x = \frac{+F_x}{24} \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} \frac{(\frac{m\pi}{A})^2}{mn\left[(\frac{m}{A})^4 D_x + (\frac{mn}{AB})^2 2H + (\frac{n}{B})^4 D_y \right]}$$

$$\sin(\frac{m\pi}{2})\left[\sin\frac{n\pi}{B}(\frac{B}{2} - b) + 22\sin(\frac{n\pi}{2}) + \sin\frac{n\pi}{B}(\frac{B}{2} + b)\right]. \tag{4.20}$$

Similarly,

$$\max M_y = \frac{+F_y}{24} \frac{16p}{\pi^6} \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty}$$

$$\frac{(\frac{n\pi}{B})^2}{mn\left[(\frac{m}{A})^4 D_x + (\frac{mn}{AB})^2 2H + (\frac{n}{B})^4 D_y \right]}$$

$$\sin(\frac{n\pi}{2}) \left[\sin \frac{m\pi}{A}(\frac{A}{2} - a) + 22 \sin(\frac{m\pi}{2}) \right.$$

$$\left. + \sin \frac{m\pi}{A}(\frac{A}{2} + a) \right]. \qquad (4.21)$$

Torsional moment M_{xy} is usually negligible or equal to zero at the sections where bending moments assume their maxima.

4.5 Design Criteria

4.5.1 First Criterion (Bending)

There exists a spacial state of stress in the gridwork; stresses in the vertical direction being negligible, the two-dimensional state of stress is observed. This biaxial action calls for the selection of a failure theory. The theory of maximum distorsion strain energy has been chosen, leading to the strength condition at a joint. Let

σ_{jt} = combined stress at a joint, and

$$\sigma_{jt} = (\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2)^{0.5} \qquad (4.22)$$

where

$$\sigma_{\substack{1\\2}} = \frac{\sigma_x + \sigma_y}{2} \pm \left[(\frac{\sigma_x - \sigma_y}{2})^2 + \sigma_{xy}^2 \right]^{0.5} \qquad (4.23)$$

are principal stresses at a joint. The maximum bending stresses used in the constraints G(1) and G(2) as Formula (3.15) are

$$\sigma_x = \frac{M_x}{S_x}, \text{ and} \tag{4.24}$$

$$\sigma_y = \frac{M_y}{S_y}, \tag{4.25}$$

where S_x = section modulus of the x-beam, and

S_y = section modulus of the y-beam.

The maximum shearing stress is

$$\sigma_{xy} = \frac{M_{xy}}{J}\delta. \tag{4.26}$$

The torsional moment of inertia J will be discussed in Section 4.5.2. The torsional moments M_{xy}, which are absent in PROBLEM 2 due to zero torsional rigidities, have now come into play.

Recalling the joint of Type 2 (Fig 8b), it is observed that the maximum combined stress occurs at the top fiber as shown in Fig 15. The constraint at the joint becomes

$$G(7) = \sigma_{all} - (\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2)^{0.5} \geq 0. \tag{4.27}$$

4.5.2 Second Criterion (Deflection)

Deflection is expressed by Formula (4.10), where the unit torsional rigidities in Formula (4.7) are

$$C_x = GJ_x, \text{ and} \tag{4.28}$$

$$C_y = GJ_y \tag{4.29}$$

with G = shear modulus, and

J_x, J_y = torsional moments of inertia for the x- and y-beams, respectively.

Since an I-section is assumed which is made up of narrow rectangular elements, the torsional moment of inertia is

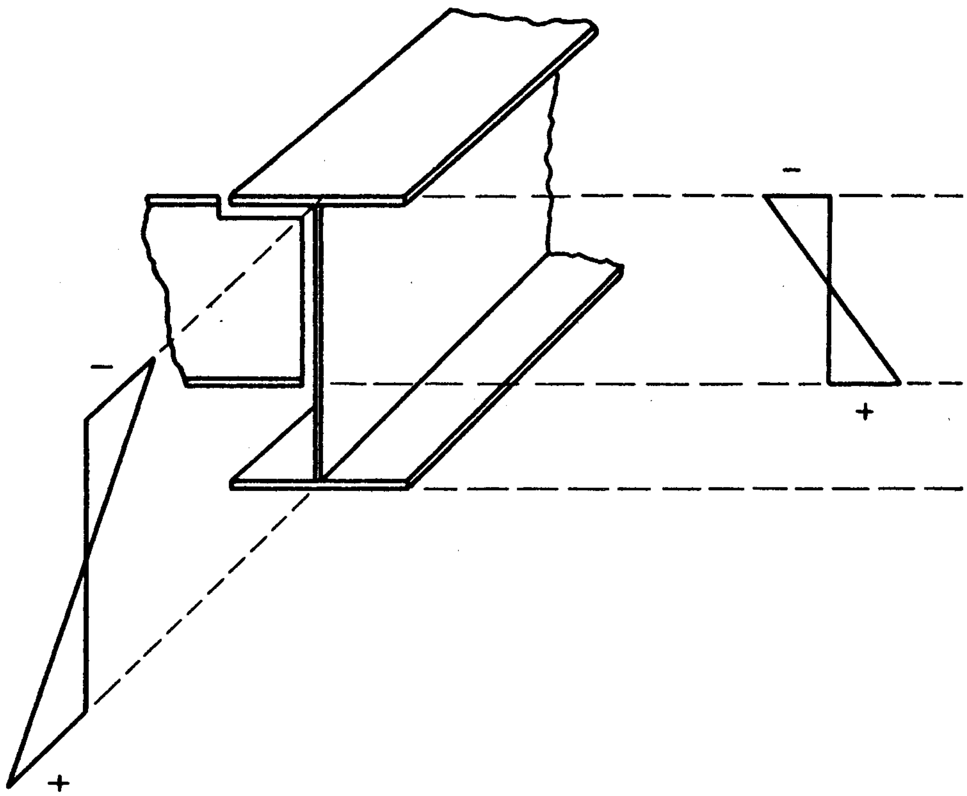


Figure 15. Bending Stresses at a Joint

$$J = \frac{1}{3} \sum_{i=1}^{3} b_o t^3 \tag{4.30}$$

or

$$J = \frac{1}{3}(2A_f t_f^2 + A_w \delta^2) \tag{4.31}$$

where b_o = depth (width) of the narrow rectangular element,

t = thickness of the narrow rectangular element,

A_f = flange cross-sectional area,

A_w = web cross-sectional area,

t_f = flange thickness,

δ = web thickness.

Letting

$$B_1 = \frac{t_f}{\delta} \tag{4.32}$$

$$B_2 = \frac{A_f}{A_w}, \tag{4.33}$$

Formula (4.31) becomes

$$J = \frac{1}{3}(2B_2 B_1^2 + 1)(A_w \delta^2). \tag{4.34}$$

Substituting Formulas (3.6) and (3.7) into Formula (4.34), for the respective x- and y-directions, Formula (4.34) reduces to

$$J_x = \frac{1}{3}(2B_2 B_1^2 + 1)(\lambda_x \delta_x^4) \tag{4.35}$$

and

$$J_y = \frac{1}{3}(2B_2 B_1^2 + 1)(\lambda_y \delta_y^4), \tag{4.36}$$

where δ_x = web thickness of the x-beam, and

δ_y = web thickness of the y-beam.

Efforts were made to suppress the addition of the two variables B_1 and B_2. The parameter B_2 is dealt with as follows:

$$A_t = 2A_f + A_w . \tag{4.37}$$

Recalling Formula (3. 3) and $K_w = K = 0.5$ (the optimal value for bending when bending prevails), where

$$A_w = K_w A_t , \tag{4.38}$$

Formula (4. 37) becomes

$$A_f = 0.5 A_w \tag{4.39}$$

or by Formula (4. 33)

$$B_2 = \frac{A_f}{A_w} = 0.5 \tag{4.40}$$

when torsion is negligible. When torsion is taken into consideration, B_2 may vary within a narrow range, similarly as K_w does.

Treating B_2 as K_w, it is desired to estimate the range of B_1. Rewriting Formula (4. 34),

$$J = \frac{\lambda \delta^4}{3} \left(1 + 2 \frac{A_f}{A_w} \frac{t_f^2}{\delta^2}\right) \tag{4.41}$$

and noting

$$K_w = \frac{A_w}{A_t} \tag{4.42}$$

the two ratios can be expressed as

$$\frac{2A_f}{A_w} = \frac{1 - K_w}{K_w} , \tag{4.43}$$

$$\frac{t_f}{\delta} = \frac{1 - K_w}{2K_w} \frac{h}{b_f} \tag{4.44}$$

where b_f = flange width. Substituting into Formula (4. 41),

$$J = \frac{\lambda \delta^4}{3} \left[1 + \frac{(1 - K_w)^3}{4K_w^3} \frac{h^2}{b_f^2}\right] . \tag{4.45}$$

A practical range of $\frac{h}{b_f}$ is between 2 and 5 for shallow and deep girders, respectively, although the ratio of 3 or 4 is generally preferred. Assuming constant cross-sectional area, a shallow girder will have smaller K_w, say $K_w = 0.4$, whereas a deep girder will have $K_w = 0.6$. Substituting these values of K_w, the factor in the brackets of J in Formula (4.45) varies between 2.85 and 4.38; and B_1 from Formulas (4.44) and (4.32) varies between 1.5 and 1.67. These small ranges of B_1 and J, and the fact that J is usually small compared to I (moment of inertia in bending) warrant the use of B_1 as a constant rather than as a variable. From Formulas (4.43) and (4.33), B_2 is found to be between 0.75 and 0.33 for K_w between 0.4 and 0.6, respectively.

Taking the averages of the narrow ranges of B_1 and B_2, consider

$B_1 = 1.59$, instead of $B_1 = (1.5\text{--}1.67)$, and

$B_2 = 0.54$, instead of $B_2 = (0.75\text{--}0.33)$.

By the speculations above, the necessity of introducing two new variables has been excluded. Because torsional resistance is of secondary importance, this simplification of the problem cannot affect the results. However, if desired, programs can be run for the limiting values of B_1 and B_2 to show the insignificant difference in results. With B_1 and B_2 being set constant, another advantage is that Formulas (4.37), (3.7), and (3.12) lead to the elimination of the two variables δ_x and δ_y because δ_x can now be expressed in terms of I_x and λ_x, and similarly δ_y in terms of I_y and λ_y. PROBLEM 3 has become a problem of at least six variables.

4.5.3 Third Criterion (Web Buckling)

As the beam-web is stiffened by the crossing beam, web buckling will likely occur in the span rather than at the joint. Since the difference between PROBLEM 3 and PROBLEM 2 is only in the joint condition, the web buckling criteria are the same for both problems. In Formula (3.26), λ, M, and S are changed to λ_x, M_x, and S_x for the x-beam and to λ_y, M_y, and S_y for the y-beam.

4.5.4 Objective function

The objective function is similar to Formula (3.28) except that now both for PROBLEMS 2 and 3 it has two terms, one for x-beams and the other for y-beams.

$$V = 3.46(\frac{B}{b} - 1)\frac{AI_x^{0.5}}{\lambda_x^{0.5}} + 3.46(\frac{A}{a} - 1)\frac{BI_y^{0.5}}{\lambda_y^{0.5}} . \qquad (4.46)$$

4.6 Mathematical Model

The necessity of an upper bound on the spacing, as discussed in Chapter III, again leads to two constraints for the two upper bounds on x-spacings, a, and y-spacings, b. The other upper bounds and lower bounds could be included. For example, the upper bounds on a and b can be replaced by the upper bounds on I_x and I_y. It should be noted that the bounds are placed here in general as discussed in Section 3.4. They are subject to removal if they are not needed physically or for the sake of programming.

PROBLEM 2 bears the form similar to PROBLEM 3, with the exclusion of the torsional rigidity expressed by the parameter H from the formulas where H appears. Instead of repeating the pertinent formulas, with V from Formula (4.46) and σ_x, σ_y, w_x, w_y, σ_{cr}, and σ_{jt} given by Formulas (4.24), (4.25), (4.10), (4.10), (3.23), and (4.22), respectively, PROBLEM 3 is formulated in a simplified form:

PROBLEM 3

Minimize $V(x_j)$, $j = 1, \ldots, 6$,

subject to the constraints

$$G(1) = \sigma_{all} - \sigma_x(x_j) \geq 0 \quad (4.47)$$

$$G(2) = \sigma_{all} - \sigma_y(x_j) \geq 0 \quad (4.48)$$

$$G(3) = w_{all} - w_x(x_j) \geq 0 \quad (4.49)$$

$$G(4) = w_{all} - w_y(x_j) \geq 0 \quad (4.50)$$

$$G(5) = \sigma_{cr}(x_3) - \sigma_x(x_j) \geq 0 \quad (4.51)$$

$$G(6) = \sigma_{cr}(x_6) - \sigma_y(x_j) \geq 0 \quad (4.52)$$

$$G(7) = \sigma_{all} - \sigma_{jt}(x_j) \geq 0 \quad (4.53)$$

$$BL_j \leq x_j \leq UB_j \quad (4.54)$$

where

$j = 1, \ldots, 6$ in Formulas (4.47) to (4.54),

$x_j = (a, I_x, \lambda_x, b, I_y, \lambda_y)$,

BL_j = lower bounds of x_j, and

UB_j = upper bounds of x_j.

CHAPTER V

NUMERICAL EXAMPLES

5.1 Problem 1

Six numerical examples were chosen to illustrate the procedure of design as discussed in Chapter III. In Ex (5.1) the theoretical upper bound of spacing b was used. Since it was a fully stressed design, the next example (Ex 5.2) made use of a higher allowable stress σ_{all} and lower allowable deflection coefficient C_3 to force the problem to be one that was not fully stressed design (or Case 2 in Section 3.5.2). The immediate question was that of a practical upper bound on b, (affected, e.g., by the available width of floor slabs), which was answered by Exs (5.3) and (5.4) similar to Exs (5.1) and (5.2), respectively, with the exception of the upper bound of b. Similarly, certain data in Ex (5.5) were chosen to make up a situation like Case (3) in Section 3.5.2. These five examples were done by the procedure described in Section 3.5, and their data and solutions are shown in Tables I and II, respectively. Table III shows three constraint values for Exs (5.1) to (5.5) which illustrate the three cases in Section 3.5.2. Finally, Ex (5.6) was the same as Ex (5.3) but was done by the Computer Code MSLP (Modified Sequence of Linear Programming Solutions) instead; its result is shown in Table IV. PROBLEM 1, with the data given in Ex (5.6), was found to be a nonconvex programming problem as defined in Section 2.8, using the definition of Formula (2.5).

TABLE I

DATA FOR EXAMPLE (5.1)

A	720	in
B	720	in
b	$\frac{B}{2} = 360$	in
σ_{all}	22	ksi
E	30,000	ksi
ν	0.3	
p	0.001	ksi
K_1	23.9	
C_1	$\frac{1}{8}$	
C_2	$\frac{5}{384}$	
C_3	$\frac{1}{360}$	

Example (5.1)

Simple supports,

p = UDL = uniformly distributed load,

Symmetrical I-section welded plate girders, and

Data given in Table I.

Example (5.2)

The data are the same as in Ex (5.1) except that

$b = \frac{B}{2} = 360$ in.,

$\sigma_{all} = 42$ ksi, and

$C_3 = \frac{1}{1000}$.

Example (5.3)

The data are the same as in Ex (5.1) except that

$b = \frac{B}{6} = 120$ in.,

$\sigma_{all} = 22$ ksi, and

$C_3 = \frac{1}{360}$.

Example (5.4)

The data are the same as in Ex (5.1) except that

$b = \frac{B}{6} = 120$ in.,

$\sigma_{all} = 42$ ksi, and

$C_3 = \frac{1}{1000}$.

Example (5.5)

The data are the same as in Ex (5.1) except that

$b = \frac{B}{6} = 120$ in.,

$\sigma_{all} = 42$ ksi, and

$C_3 = \frac{1}{360}$.

TABLE II

SOLUTIONS OF EXAMPLES (5.1) TO (5.5)

	(5.1)	(5.2)	(5.3)	(5.4)	(5.5)
CASE	(1)	(2)	(1)	(2)	(3)
C_{11}	1.39	0.369	1.09	0.289	---
C	1.0	0.369	1.0	0.289	---
h (in)	---	---	---	---	75.6
C_{14}	---	---	---	---	0.105
σ (ksi)	22.0[(A)]	15.6	22.0[(A)]	12.2	42.0[(A)]
b (in)	360.0	360.0	120.0	120.0	120.0
λ	170.0	205.0	170.0	232.0	100.0
I (in^4)	34,600.0	58,300.0	8,004.0	19,400.0	6,998.0
Min V (in^3)	35,345.0	42,018.0	84,976.0	114,120.0	104,200.0

[(A)]fully stressed

TABLE III

THREE CONSTRAINT VALUES OF EXAMPLES (5.1) TO (5.5)

	(5.1)	(5.2)	(5.3)	(5.4)	(5.5)
CASE	(1)	(2)	(1)	(2)	(3)
σ_{all}	22.0	42.0	22.0	42.0	42.0
σ	22.0[A]	15.6	22.0[A]	12.2	42.0[A]
w_{all}	2.0	0.72	2.0	0.72	2.0
w	1.99	0.719	2.0	0.72	2.0
σ_{cr}	22.0	15.5	22.0	12.1	65.1
σ	22.0	15.6	22.0	12.2	42.0
$G_1 \geq 0$	0 (=)	26.4 (>)	0 (=)	29.8 (>)	0 (=)
$G_2 \geq 0$	0.01 (>)	0 (=)	0 (=)	0 (=)	0 (=)
$G_3 \geq 0$	0 (=)	0 (=)	0 (=)	0 (=)	23.1 (>)

[A] fully stressed

Example (5. 6)

The data are the same as in Ex (5. 3) and the solution is shown in Table IV. In addition to the data given in Table I, more data pertinent to this example are:

MN = No. of variables = 3,

NNL = No. of nonlinear constraints = 3,

LIN = No. of linear constraints = 6,

X(1) = b (in),

X(2) = I_x (in^4),

X(3) = λ_x,

BL(1) = 24 (in),

BL(2) = 11. 3 (in^4),

BL(3) = 100,

UB(1) = 120 (in),

UB(2) = 349, 000 (in^4), and

UB(3) = 350.

5. 2 Problem 2

In Chapter IV general expressions were derived for PROBLEM 3, with the mathematical model given in Section (4. 6). It was realized that PROBLEM 2 was actually PROBLEM 3 with the exclusion of the parameter H. The result of PROBLEM 2 as Ex (5. 7) is shown in Table V. PROBLEM 2 was found to be a nonconvex programming problem, using the definition of Formula (2. 5).

From the last column of Table V, the values of a and b are 106. 5 inches, indicating the number of spacings, e. g., $\frac{A}{a}$, a non-integer. The two adjacent integers are 6 and 7, which top the second and fourth

TABLE IV

SOLUTION OF EXAMPLE (5.6), (PROBLEM 1)

	INITIAL POINT	MSLP FINAL POINT
X(1) = b (in)	103	120
X(2) = I_x (in^4)	9,000	8,022
X(3) = λ_x	100	172
Min V (in^3)	141,600	85,070
		<u>SOLUTION</u>

IBM 360/65

Execution time 3.98 seconds

Total time 20.82 seconds

columns of Table VI. The last column of Table V (solution) is inserted in Table VI for comparison. The last line of Table VI shows whether the three prospective solutions satisfy all the constraints or not. As the solution point in the last column of Table VI satisfies all the constraints, it is the final design.

Example (5.7)

Simple supports,

p = UDL = uniformly distributed load, and

Symmetrical I-section welded plate girders.

In addition to the data given in Table I, more data pertinent to this example are:

MN = No. of variables = 6,

NNL = No. of nonlinear constraints = 7,

LIN = No. of linear constraints = 12,

X(1) = a (in),

X(2) = I_x (in^4),

X(3) = λ_x,

X(4) = b (in),

X(5) = I_y (in^4),

X(6) = λ_y,

BL(1), (4) = 24 (in),

BL(2), (5) = 11.3 (in^4),

BL(3), (6) = 100,

UB(1), (4) = 120 (in),

UB(2), (5) = 349,000 (in^4), and

UB(3), (6) = 350.

TABLE V

SOLUTION OF EXAMPLE (5.7), (PROBLEM 2)

	Initial point	Intermediate point	RGM Final point
X(1) = a (in)	103	120	106.5
X(2) = $I_x(in^4)$	9,000	8,999	9,000
X(3) = λ_x	100	108	172
X(4) = b (in)	103	120	106.5
X(5) = $I_y(in^4)$	9,000	8,999	9,000
X(6) = λ_y	100	108	172
Min V (in^3)	283,100	227,700	207,600
			SOLUTION

IBM 360/65

Total time 14 min. 34 sec.

TABLE VI

FINAL DESIGN OF EXAMPLE (5.7), (PROBLEM 2)

		RGM Final point	
$s = \frac{A}{a} = \frac{B}{b}$	6	non-integer	7
X(1) = a (in)	120	106.5	103
X(2) = $I_x(in^4)$	9,000	9,000	9,000
X(3) = λ_x	172	172	172
X(4) = b (in)	120	106.5	103
X(5) = $I_y(in^4)$	9,000	9,000	9,000
X(6) = λ_y	172	172	172
Min V (in^3)	180,200	207,600	215,900
All constraints satisfied	NO	YES	YES
			FINAL DESIGN

5.3 Problem 3

The mathematical model of PROBLEM 3 given in Section 4.6 is tested here numerically. PROBLEM 3 yielded the same solution as PROBLEM 2 (Table V); therefore, the final design of PROBLEM 3 is identical to that of PROBLEM 2 (Table VI).

PROBLEM 3 was found to be a nonconvex programming problem, using the definition of Formula (2.5). The RGM Code (Ricochet Gradient Method) took 14 min. 21 sec. in computer time.

Example (5.8)

Simple supports,

p = UDL = uniformly distributed load, and

Symmetrical I-section welded plate girders.

In addition to the data given in Example (5.7), more data pertinent to this example are:

B_1 = 1.59,

B_2 = 0.54, and

G = 12,000 ksi.

5.4 Discussion of Numerical Results

PROBLEM 1 was solved by the analytical approach (Section 3.5) and also by using the Computer Code MSLP (Modified Sequence of Linear Programming Solutions). Exs (5.1) to (5.5) numerically verified the three cases in Section 3.5.2. Upper bounds are linear constraints. When a design variable reaches its upper bound in the solution of the problem, it steps on its linear constraint, outside which the solution is infeasible. It can be said that the variable has reached its maximum possible value allowed by its design upper bound or design linear constraint. The same can be said about a nonlinear constraint; the variable has reached its maximum possible value allowed by its design criterion or design nonlinear constraint. With these implications the spacing b was set at its maximum value allowed by its linear constraint (upper bound) according to Section 3. 5. 1 and the web slenderness λ reached its maximum value allowed by its design criterion or nonlinear constraint. In Ex (5. 5), λ was preselected and had nothing to do with bounds.

Ex (5. 6) was used to check on the validity of the Computer Code MSLP and also to compare with one of the previous examples, namely Ex (5. 3). The results of these two examples came to a good agreement, as seen from Tables II and IV. Both b and λ in Ex (5. 6) reached their maximum values as described above. The execution time for Ex (5. 3) was 3. 98 seconds (total time 20. 82 seconds) on an IBM 360/65 model. Certainly the preparation for a Mathematical Programming problem is shorter than that for the analytical approach. A problem of the size approximate to the size of PROBLEM 1 needed about 2 minutes 17 seconds using the RGM Code (Ricochet Gradient Method). This is no surprise since the gradient method is known for its slow convergence.

PROBLEM 2 revealed some interesting results (Table V). The web slenderness λ reached its maximum value as allowed by the web buckling constraint, in spite of the notion speculated in Section 3. 5. 1 that the spacing of beams was expected to attain its upper bound. As a matter of fact, at some intermediate point (Table V) the spacing did reach its upper bound with the web slenderness being still low. However, at this intermediate point, the volume was still going on decreasing.

From the viewpoint of Nonlinear Programming, the final point (Table V) is the local minimum. On the other hand, the designer may take the intermediate point as an alternate design if he prefers to use a larger spacing at the expense of about 10 percent bigger volume for this case. It was noted that the solution did not give $s = \frac{B}{b}$ as an integer. Comparing the two points for the two adjacent integers, the point at which all the constraints are satisfied will be the final design.

PROBLEM 3 yielded a solution identical to that of PROBLEM 2. In Formulas (4.10), (4.20), and (4.21), PROBLEM 3 and PROBLEM 2 differ in the parameter H, being zero for the latter. A typical section shows that H is very small compared to D_x or D_y. As H is comparatively negligible, PROBLEM 3 for this case can be considered as practically the same as PROBLEM 2.

All three problems were found to be nonconvex problems. Approximation by linearization in PROBLEM 1 was successful, making it possible for the MSLP Code (Modified Sequence of Linear Programming Solutions) to work. However, as the degree of nonconvexity increased in PROBLEMS 2 and 3, in addition to the complexity, the RGM Code (Ricochet Gradient Method) took more computer time and the MSLP Code did not work. Since the MSLP Code does not take too much time to run, it is recommended that it should be tried first and if it fails then the RGM Code is used.

CHAPTER VI

BOUNDARY CONDITIONS, LOADING CONDITIONS, AND BOUNDARY DIMENSIONS

6.1 Introduction

Boundary conditions and loading conditions belong to Step 1 in Chapter II, Section 2.7.

So far the only boundary condition imposed on the grid has been of simple supports. Whereas any other types of boundary condition can be dealt with by the Theory of Plates, at least approximately, two of them are probably most frequently used in practice, namely the simple supports and the clamped (or fixed) supports (Fig 16). The Navier's Method is primarily for a simply supported plate. However, the Timoshenko-Ritz Variational Method, in some features similar to the Navier's Method, can be applied to the clamped supports. By this means, as a result of the use of other functions (e.g., Formulas 6.8 to 6.12) than sines in the series used previously, only the set of parameters A_{mn} which appears in the solution will change. The immediate benefit is that the same formulation can be used with the exception of this modification.

6.2 Timoshenko-Ritz Method

The principle used here is that the total potential energy is stationary,

$$\delta(U - V_W) = 0 \qquad (6.1)$$

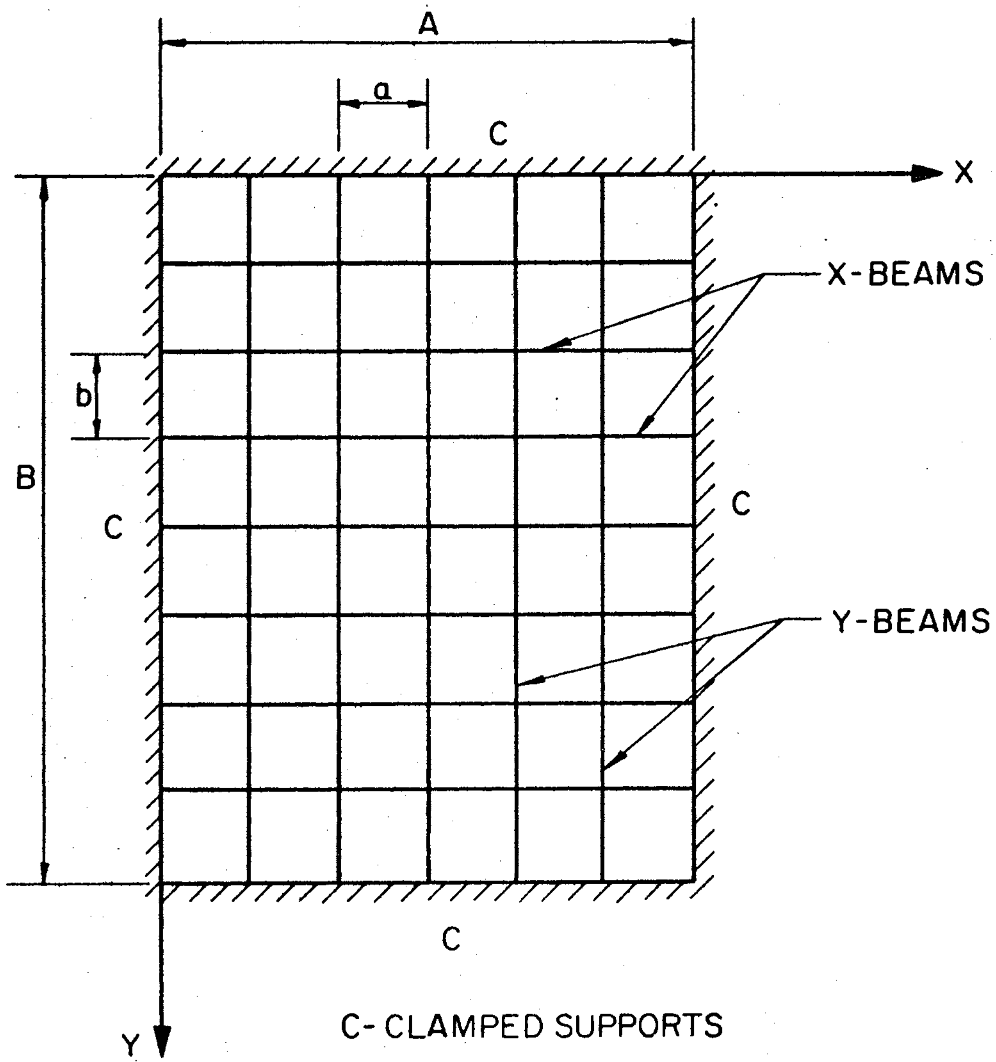


Figure 16. A Grid Clamped on All Four Sides

where U = strain energy of bending, and

V_w = virtual work.

For an orthotropic plate,

$$U = \frac{1}{2}\int_0^A\int_0^B (D_x w_{,xx}^2 + 2H w_{,xx} w_{,yy} + D_y w_{,yy}^2)\, dx\, dy \qquad (6.2)$$

and

$$V_w = \int_0^A\int_0^B q(x,y)\, w\, dx\, dy \qquad (6.3)$$

where q(x, y) = loading function per unit area.

For uniform loading, q(x, y) = p. The assumed solution

$$w(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} A_{mn}\, w(x)\, w(y) \qquad (6.4)$$

must satisfy at least the geometrical boundary conditions. With the origin of the coordinates at the center of the plate,

at $$x = \pm\frac{A}{2},\quad w_x = 0,\quad w_{,x} = 0, \qquad (6.5)$$

and at $$y = \pm\frac{B}{2},\quad w_y = 0,\quad w_{,y} = 0, \qquad (6.6)$$

for the clamped plate. By the principle of stationary potential energy, taking the partial derivatives of the total potential energy with respect to A_{mn},

$$\frac{\partial(U - V_w)}{\partial A_{mn}} = 0, \qquad (6.7)$$

a set of algebraic equations for the unknown coefficients A_{mn} will be obtained.

6.3 A_{mn} for Clamped Supports

Using algebraic feasible functions w(x) and w(y) in Formula (6.4), an assumed solution might take the form

$$w(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} (x^2 - \frac{A^2}{4})^2$$

$$(y^2 - \frac{B^2}{4})^2 x^{m-1} y^{n-1}. \tag{6.8}$$

Other feasible functions might be trigonometric expressions, for instance,

$$w(x) = 1 + \cos \frac{2m\pi x}{A} \quad (m = \text{odd}), \tag{6.9}$$

$$w(x) = 1 - \cos \frac{2m\pi x}{A} \quad (m = \text{even}). \tag{6.10}$$

Similarly,

$$w(y) = 1 + \cos \frac{2n\pi y}{B} \quad (n = \text{odd}), \tag{6.11}$$

$$w(y) = 1 - \cos \frac{2n\pi y}{B} \quad (n = \text{even}). \tag{6.12}$$

Without going into detailed calculations, the value of A_{mn} for the feasible functions, Formulas (6.9) to (6.12), resulted in Formula (6.14). Four-term approximation was used with m = 1, 2 and n = 1, 2. Solution from the trigonometric expressions called for a set of simultaneous algebraic equations in the form of

$$\left[C \right] \left\{ A \right\} = \left\{ D \right\}. \tag{6.13}$$

The column matrix $\left\{ D \right\}$ has four identical elements of (pAB), A and B being the two sides of the grid. The column matrix of the unknown coefficients is

$$\left\{ A \right\} = (A_{11}\ A_{12}\ A_{21}\ A_{22}). \tag{6.14}$$

$$\left[C \right] = \text{a square matrix,} \tag{6.15}$$

with its elements defined as follows:

$$c_{11} = 8\pi^4 \left(\frac{3D_x}{A^3} + \frac{4H}{AB} + \frac{3D_y}{B^3}\right)$$

$$c_{12} = \frac{16\pi^4 D_x}{A^3}$$

$$c_{13} = \frac{16\pi^4 D_y}{B^3}$$

$$c_{14} = 0$$

$$c_{21} = \frac{16\pi^4 D_x}{A^3}$$

$$c_{22} = 8\pi^4 \left(\frac{3D_x}{A^3} + \frac{16}{AB} + \frac{48D_y}{B^3}\right)$$

$$c_{23} = 0$$

$$c_{24} = \frac{256\pi^4 D_y}{B^3}$$

$$c_{31} = \frac{16\pi^4 D_y}{B^3}$$

$$c_{32} = 0$$

$$c_{33} = 8\pi^4 \left(\frac{48D_x}{A^3} + \frac{4}{AB} + \frac{3D_y}{B^3}\right)$$

$$c_{34} = \frac{256\pi^4 D_x}{AB}$$

$$c_{41} = 0$$

$$c_{42} = \frac{256\pi^4 D_y}{B^3}$$

$$c_{43} = \frac{256\pi^4 D_x}{A^3}$$

$$c_{44} = 8\pi^4 \left(\frac{48D_x}{A^3} + \frac{16}{AB} + \frac{48D_y}{B^3}\right).$$

One-term approximation for the trigonometric expressions was also done, resulting in

$$A_{11} = \frac{+p}{8\pi^4} \frac{A^4 B^4}{(4D_x B^4 + HA^2B^2 + 4D_y A^4)} . \qquad (6.16)$$

6.4 Loading Conditions

So far the grid under investigation has been loaded by a uniform load. If the loading conditions are others, the design procedure remains similar. The loading $q(x, y)$ is frequently expressed as a product of two functions, each depending on one direction, or

$$q(x, y) = p\, f(x)\, g(y). \qquad (6.17)$$

The constant uniform load $q(x, y) = p$ is thus replaced by the loading function. Both functions $f(x)$ and $g(y)$ are expanded by the series of a similar type as used in the expression for the deflection w.

Suppose in PROBLEM 3 the loading is a concentrated load P at the point (x_1, y_1) instead of a uniformly distributed load, UDL = p. First, replace the concentrated load P by a local uniform load UDL of the size u by v, with its center at (x_1, y_1), (Fig 17). The deflection is found to be

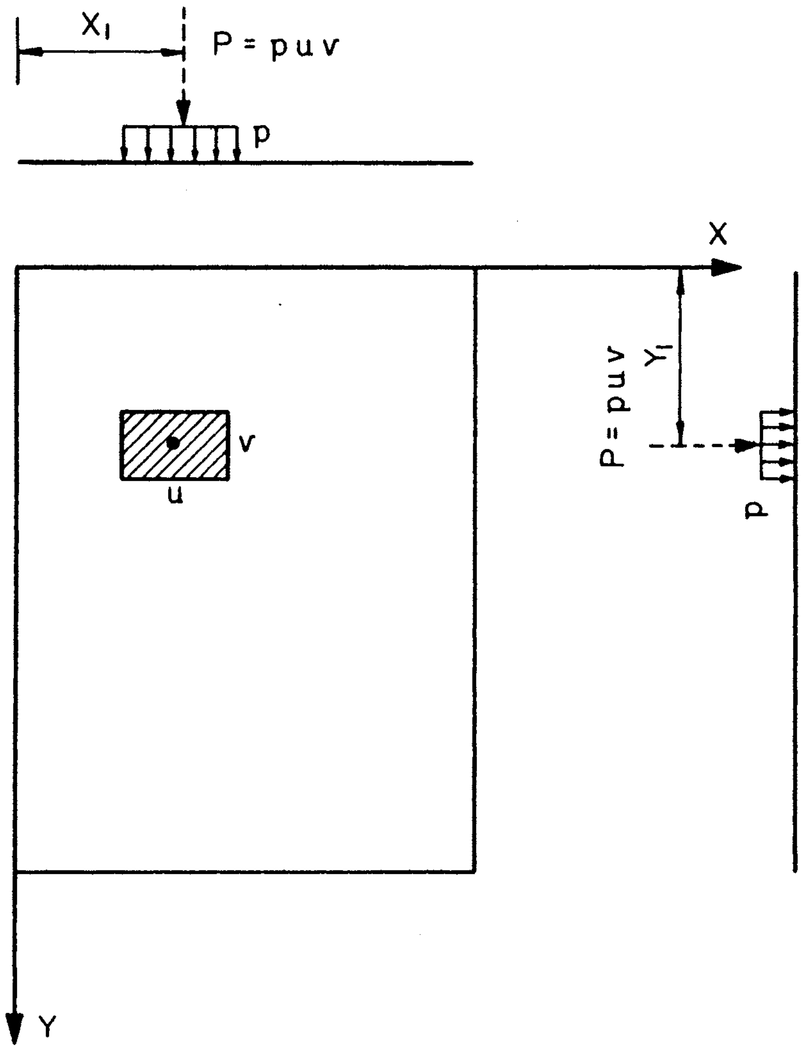


Figure 17. Local Uniform Load Shrinking to a Concentrated Load

$$w(x,y) = \sum_{m=1,3,}^{\infty} \sum_{n=1,3,}^{\infty}$$

$$\frac{16p}{\pi^6 mn\left[(\frac{m}{A})^4 D_x + (\frac{mn}{AB})^2 2H + (\frac{n}{B})^4 D_y\right]}$$

$$\left[\sin\frac{m\pi u}{2A} \sin\frac{n\pi v}{2B} \sin\frac{m\pi x_1}{A} \sin\frac{n\pi y_1}{B}\right.$$

$$\left.\sin\frac{m\pi x}{A} \sin\frac{n\pi y}{B}\right]. \quad (6.18)$$

Replacing p by $\frac{P}{uv}$ and rearranging, Formula (6.18) becomes

$$w(x,y) = \sum_{m=1,3,}^{\infty} \sum_{n=1,3,}^{\infty}$$

$$\frac{16P}{\pi^6 mn\left[(\frac{m}{A})^4 D_x + (\frac{mn}{AB})^2 2H + (\frac{n}{B})^4 D_y\right]}$$

$$\left[\frac{\sin\frac{m\pi x_1}{A} \quad \sin\frac{n\pi y_1}{B}}{\frac{2A}{m\pi} \; \frac{2B}{n\pi}}\right]\left[\frac{\sin\frac{m\pi u}{2A} \quad \sin\frac{n\pi v}{2B}}{\frac{m\pi u}{2A} \; \frac{n\pi v}{2B}}\right]$$

$$\left[\sin\frac{m\pi x}{A} \sin\frac{n\pi y}{B}\right]. \quad (6.19)$$

When the local UDL is shrinking to a point, the limit is taken as

$$\lim_{u,v \to 0} w(x,y). \quad (6.20)$$

Applying the L'Hospital's Rule,

$$\lim_{t \to 0} \frac{\sin t}{t} = \lim_{t \to 0} \frac{\cos t}{1} = 1, \quad (6.21)$$

the second to the last term in brackets of Formula (6.19) becomes unity. The solution is

$$w(x,y) = \sum_{m=1,3,}^{\infty} \sum_{n=1,3,}^{\infty} \frac{4P}{\pi^4 AB\left[(\frac{m}{A})^4 D_x + (\frac{mn}{AB})^2 2H + (\frac{n}{B})^4 D_y\right]} \left[\sin\frac{m\pi x_1}{A} \sin\frac{n\pi y_1}{B} \sin\frac{m\pi x}{A} \sin\frac{n\pi y}{B}\right]. \qquad (6.22)$$

This looks similar to the solution of PROBLEM 3, also with a modification factor of four sine-terms due to this particular case.

6.5 Boundary Dimensions

The boundary dimensions were chosen to be such that the dimension in the x-direction was identical to that in the y-direction because the design variables associated with x were expected to yield solutions identical to those associated with y. This symmetrical solution served as one qualitative check on itself.

This postulation was tested by considering the simplest grid, with one beam in each direction (Fig 18). Let an x-beam with moment of inertia I_x rest on top of a y-beam with a single load P at the intersection and with four ends simply supported. At the intersection, the reaction on the x-beam,

$$R = P\left(\frac{I_y A^3}{I_x B^3 + I_y A^3}\right), \qquad (6.23)$$

$$\max M_x = \frac{PA}{4} - \frac{RA}{4}$$

$$= \frac{PA}{4}\left(\frac{I_x B^3}{I_x B^3 + I_y A^3}\right). \qquad (6.24)$$

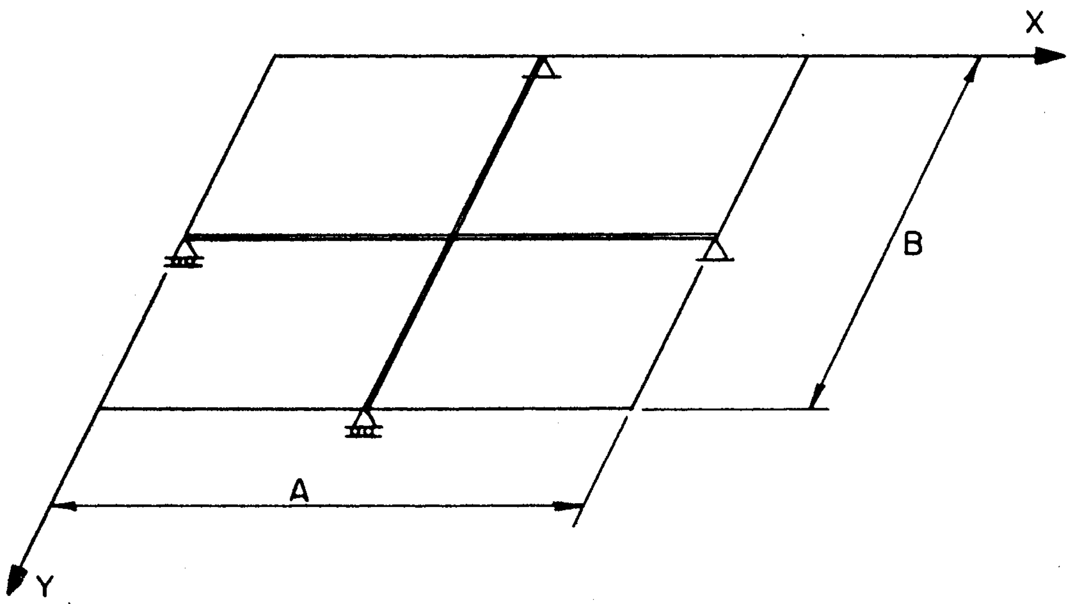


Figure 18. Simplest Grid with One Beam in Each Direction

For the y-beam,

$$\max M_y = \frac{PB}{4}\left(\frac{I_y A^3}{I_x B^3 + I_y A^3}\right) . \tag{6.25}$$

Then, for both beams to be fully stressed to the same amount,

$$M_x = M_y ,$$

$$\frac{PA}{4}\left(\frac{I_x B^3}{I_x B^3 + I_y A^3}\right) = \frac{PB}{4}\left(\frac{I_y A^3}{I_x B^3 + I_y A^3}\right) . \tag{6.26}$$

If A = B, then

$$I_x = I_y . \tag{6.27}$$

Therefore, the postulation was proved, using the simplest grid.

With this postulation, the square grid could have been treated with less design variables, namely using "I" in place of I_x and I_y, and so on. However, this was not done so that the problem could deal with a general rectangular grid.

Usually, the two sides of the rectangular grid are given. However, one side, say A in the x-direction, can be specified and the other side can be expressed as a multiple of A, or

$$B = R_1 A , \tag{6.28}$$

where R_1 = ratio of the two sides.
By introducing this new variable R_1, the optimal solution will reveal whether a square grid or a rectangular grid with a particular ratio of sides is an economical shape of the grid.

Another approach along this line is to make several runs of the problem with

$R_1 = 1, 2, 3, 4, \ldots$

and to plot a graph of R_1 versus Min V. The minimum point of this curve will be the most economical ratio of the two sides.

CHAPTER VII

DISCUSSION OF RESULTS AND CONCLUSION

The three problems were labeled in Chapters III and IV as follows:

PROBLEM 1 --- Beams spanning in one direction,

PROBLEM 2 --- A grid with beams spanning in two orthogonal directions (simple-connection),

PROBLEM 3 --- A grid with beams spanning in two orthogonal directions (rigid-connection).

All of them had a boundary condition of simple supports and a uniformly distributed load.

It should be pointed out again that the formulation was intended to be for a general rectangular grid, as explained in Section 6.5. The square grid was used as a qualitative check on its own solution, where it might seem to suggest superfluous use of design variables.

From the solutions of PROBLEMS 1, 2, and 3 (Tables IV and V), one common feature was evident. The spacings of beams a and b and the web slendernesses λ_x and λ_y reached the maximum as explained in Section 5.4. This is a reasonable result because maximum spacing and maximum web slenderness contribute to economy. However, it should not be concluded that maximum spacings (upper bound) can always be achieved. With increasing magnitude of uniform load (greater than the one used), bigger beams would be required to carry the bigger load until the upper bound of moment of inertia was reached. With even heavier load, the spacings of beams would decrease so that more beams of the same largest section specified could carry this heavy load. Similarly, suppose for some reason that it is desired to use small beams. If all these small beams with upper bounds of spacings cannot carry the load, then more beams of this section will be required, or smaller spacings. Also, it should not be concluded that the upper bound of web slenderness can always be achieved. Within safety limits (constraints) maximum values of spacings and web slenderness would be attained, as explained in Section 5. 4.

As mentioned in Chapter I, the objective function could be a cost function if a unit cost was incorporated with the volume. Also, it is possible to cut holes in the appropriate locations of the web of the beam or to use the castillated beam (Fig 19) in order to reduce the volume further. However, although it is of academic interest, it is not recommended because of increase of fabrication cost unless weight is the prime important factor. Efforts in deriving expressions for this beam and high fabrication costs may not warrant this saving of material.

Another factor of economy is in the choice between plate girders and standard rolled sections. The design was based on I-shaped plate girders. However, for other shapes, section properties could be derived in a similar manner. Their costs vary from company to company and also from locality to locality. Similarly, simply-connection (PROBLEM 2) should be compared with rigid-connection (PROBLEM 3) for economy in fabrication and erection costs since material cost is the same in both cases. Because of the variable nature of different kinds of cost, the volume of material was chosen as the objective function. With known unit cost function the objective function will

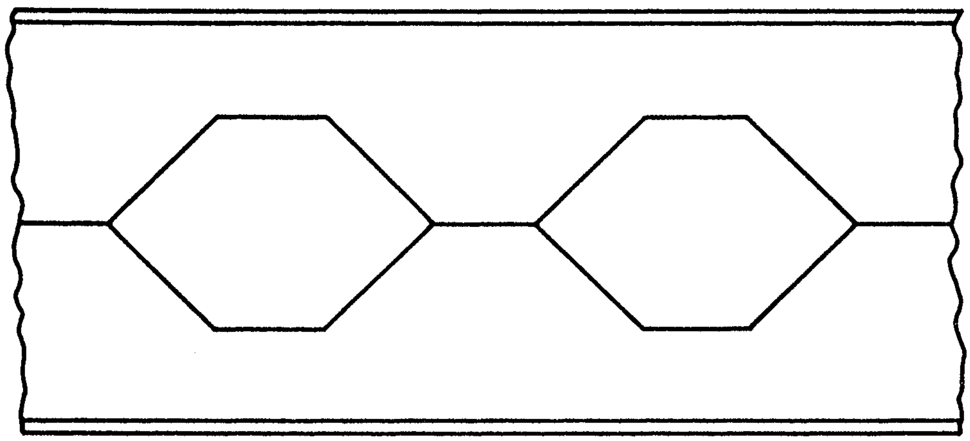


Figure 19. Castillated Beam

yield a minimum cost rather than a minimum volume.

It is concluded as follows:

(1) This approach by Mathematical Programming is feasible in the design of a grid and is superior to the conventional design procedure, because of material economy.

(2) This approach is flexible in the choice of the two essential components: the Structural Analysis method and the Mathematical Programming technique.

(3) The analysis by the Theory of Plates proves to be economical in computer computations.

(4) Exs (5. 1) to (5. 5) illustrated the three cases discussed in Section 3. 5. 2 and demonstrated that it is not always the fully stressed design which is the optimum. PROBLEM 1 was solved by using the analytical approach (Section 3. 5) and the MSLP Code (Modified Sequence of Linear Programming Solutions), coming to a good agreement.

(5) PROBLEM 2 and PROBLEM 3 produced the same solution because the nonvanishing parameter H in the latter was relatively small for the case considered. These two problems were found to be fully stressed designs. However, the web slendernesses reached their upper bounds but the spacings did not.

(6) In a sense, as explained in Section 5. 4, spacings and web slendernesses reached their maximum possible values in all three problems.

(7) All three problems are nonconvex problems. The MSLP Code (Modified Sequence of Linear Programming Solutions)

works only for PROBLEM 1; the RGM Code (Ricochet Gradient Method) works for all three problems.

A few points should be made as suggestions:

(1) As a generalization, the problem is not limited to a Civil Engineering problem but may be one that has been idealized to a model of rectangular grid.

(2) This approach is also flexible in the choice of objective function. Depending on applications, if the weight of the structure is an important factor, the volume will be chosen as the objective function. If the weight is of no concern, then the cost function should be the objective function.

(3) The section properties for other shapes could be derived in a similar manner to the I-shaped welded plate girder used here. Holes may be cut from the web of the beam (Fig 19) to reduce the weight.

(4) As explained in Section 6.5, the ratios of the two sides of the rectangular grid could be regarded as a variable to be optimized.

A SELECTED BIBLIOGRAPHY

(1) Gass, S. I. Linear Programming Methods and Applications. New York: McGraw-Hill Book Company, Third Edition, 1969, ix.

(2) Sheu, C. Y. and Prager, W. "Recent Developments in Optimal Structural Design." Applied Mechanics Reviews, Vol. 21, No. 10 (October, 1968), 985-992.

(3) Wasiutynski, Z. and Brandt, A. "The Present State of Knowledge in the Field of Optimum Design of Structures." Applied Mechanics Reviews, Vol. 16, No. 5 (May, 1963), 341-350.

(4) Schmit. L. A., Jr. "Structural Design by Systematic Synthesis." Second Conference on Electronic Computation, Pittsburgh, Pennsylvania, Journal of Structural Division, ASCE (October, 1960), 105-132.

(5) Goble, G. G. and De Santis, P. V. "Optimum Design of Mixed Steel Composite Girders." Journal of Structural Division, ASCE, Vol. 92, No. St 6 (December, 1966), 25-43.

(6) Razani, R. and Goble, G. G. "Optimum Design of Constant-Depth Plate Girders." Journal of Structural Division, ASCE, Vol. 92, No. St 2 (April, 1966), 253-281.

(7) Mauch, H. R. and Felton, L. P. "Optimum Design of Columns Supported by Tension Ties." Journal of Structural Division ASCE, Vol. 93, No. St 3 (June, 1967), 201-220.

(8) Holt, E. C. and Heithecker, G. L. "Minimum Weight Proportions for Steel Girders." Journal of Structural Division, ASCE, Vol. 95, No. St 10 (October, 1969), 2205-2217.

(9) Seaburg, P. A. and Salmon, C. G. "Minimum Weight Design of Light Gage Steel Members." Civil Engineering Department, University of Wisconsin, Milwaukee, Wisconsin (April, 1970), 1-34.

(10) Pearson, C. E. "Structural Design by High Speed Computing Machines." Proceedings of the First Conference on Electronic Computation, ASCE, New York (1958), 417-436.

(11) Livesley, R. K. "Optimum Design of Structural Frames for Alternative Systems of Loading." Civil Engineering and Public Works Reviews, Vol. 54. No. 636 (June, 1959), 737-740.

(12) Brown, D. M. and Ang, A. "A Nonlinear Programming Approach to the Minimum Weight Elastic Design of Steel Structures." Civil Engineering Studies, Research Series No. 298, University of Illinois (October, 1965), 86-87.

(13) Massachusetts Institute of Technology. ICES (Integrated Civil Engineering System) ICES OPTECH, Research Report R68-65 (June, 1968), 71-91.

(14) Zoutendijk, G. Methods of Feasible Directions. Amsterdam: Elsevier Publishing Company, 1960.

(15) Abadie, J. Integer and Nonlinear Programming. New York: American Elsevier, 1970.

(16) Wolfe, P. Recent Developments in Nonlinear Programming. New York: Academic Press, Inc., 1962.

(17) Private correspondence with Marguerite Sammons, Program Librarian, RAND Corporation, Santa Monica, California, January, 1971.

(18) International Business Machines Corporation. IBM Application Program, H20-0476-1, Mathematical Programming System/360, 1969, Chapter 5, 165-188.

(19) Fiacco, A. V. et al. Programming Under Nonlinear Constraints by Unconstrained Minimization: A Primal-Dual Method. Washington, D. C.: Research Analysis Corporation, September, 1963, AD 423 903.

(20) McCormick, G. P. et al. Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming. McLean, Virginia: Research Analysis Corporation, April, 1965, AD 621 991.

(21) Fiacco, A. V. and McCormick, G. P. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: John Wiley and Sons, Inc., 1968, 39-71.

(22) Kelly, J. E., Jr. "The Cutting Plane Method for Solving Convex Programs." Siam Journal, Vol, 8, No. 4 (December, 1960), 703-712.

(23) Mertz, R. T. "Ricochet Gradient Method." IBM 360 Catalog (October, 1967), 1-110.

(24) Duffin, R. J., Peterson, E. L. and Zener, C. Geometric Programming. New York: John Wiley and Sons, Inc., 1967, Chapters 1 and 3.

(25) Bellman, R. Dynamic Programming. Princeton, N. J.: Princeton University Press, 1957, Chapters 1 and 3.

(26) Bellman, R. and Dreyfus, S. Applied Dynamic Programming. Princeton, N. J.: Princeton University Press, 1962, Chapter 5.

(27) Kuo, S. S. Numerical Methods and Computers. Reading, Mass: Addison-Wesley, 1965, 287-305.

(28) International Business Machines Corporation. IBM Scientific Subroutine Package, RTWI, Form H20-0205-3, 1970, 215-217.

(29) McCalla, T. R. Introduction to Numerical Methods and Fortran Programming. New York: John Wiley and Sons, Inc., 1967, Chapter 3, 72-100.

(30) Wood, C. A. "Numerical Methods Lecture Notes." Oklahoma State University, Stillwater, Oklahoma, 1969.

(31) International Business Machines Corporation, IBM Scientific Subroutine Package, FMFP, Form H20-0205-3, 1970, 221-224.

(32) Fletcher, R. and Powell, M. J. D. "A Rapidly Convergent Descent Method for Minimization." Computer Journal, Vol. 6, iss. 2 (1963), 163-168.

(33) Gass, S. I. Linear Programming Methods and Applications. New York: McGraw-Hill Book Company, Third Edition, 1969, 292-312.

(34) Tochacek, M. and Mehta, C. L. "Economical Design of a Prestressed Plate Girder." Journal of Structural Division, ASCE (to appear).

(35) Beedle, L. S., et al. Structural Steel Design. New York: Roland Press Company, 1964, 545.

(36) U. S. Steel Corporation. U. S. S. Steel Design Manual. Pittsburgh, Pennsylvania, November, 1968, 70-78.

(37) Timoshenko, S. and Gere, J. M. Theory of Elastic Stability. New York: McGraw-Hill Book Company, Inc., Second Edition, 1961, 377.

(38) Timoshenko, S. and Woinowsky-Krieger, S. Theory of Plates and Shells. New York: McGraw-Hill Book Company, Second Edition, 1959, 364-376.

APPENDIX A

COMPUTER PROGRAM LISTING--THE MSLP CODE, (MODIFIED SEQUENCE OF LINEAR PROGRAMMING SOLUTIONS)

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
C------*********************************************************     IP  10
C        ***************************************************         IP 250
C        ***************************************************         IP 240
C  ******************************************************************  IP 130
C  ******************************************************************  IP  80
C
C
C
C
C
C....--------------------------------------------------------------
C--------COMPUTER CODE MSLP
C        (MODIFIED SEQUENCE OF LINEAR PROGRAMMING SOLUTIONS)
C--------PROGRAMMED BY M.S. CHAN AT OKLAHOMA STATE UNIVERSITY,
C        STILLWATER, OKLAHOMA, UNDER THE SUPERVISION OF DR. M.
C        TOCHACEK IN 1971.
C....--------------------------------------------------------------
C.................................................................   IP 300
C  ..........  MAIN  PROGRAM  ..........                             IP 310
C             REQUIRED SUBROUTINES                                   IP 320
C                (1)  SUBROUTINES  PTCON ----- REQ 3                 IP 330
C                (2)  SUBROUTINES  BNEG1                             IP 340
C                (3)  SUBROUTINES  LP2 ----- REQ 1                   IP 350
C.................................................................   IP 360
C
C
C
C
C
      DIMENSION X(12), XM(12), IZERO(12), XP(12)
      DIMENSION  A(65, 65), W(65), L(65)
      DIMENSION          PRDT(45), VALX(65)
      DIMENSION C1(30,12), C2(30,25), C(30,50)
      DIMENSION FX(12),          GC(25), GX(25,12),G(25)
      DIMENSION  GLX(25,12), GLC(25)
      DIMENSION  BL(12), UB(12)
      COMMON MN, LIN, NNL, NPR
    5 FORMAT (I5, E13.4 )
   11 FORMAT ( 10I5)                                                  IP 460
   12 FORMAT (4(8E10.4/))                                             IP 470
   13 FORMAT (4(5X, 8E15.4/))                                         IP 480
   14 FORMAT ( 3I5, F10.2)                                            IP 490
   15 FORMAT ( T5,'ITER**',T20,'RATIO*',T35,'ZVAL**',T50 ,'VMIN ***** IP 500
     1*******************', / 5X, I5, 8E15.4 / )                      IP 510
   16 FORMAT ( 5X, 8E15.4 / )                                         IP 520
   17 FORMAT ( I10, 10F10.2 )                                         IP 530
   22 FORMAT ( 7E11.4 )                                               IP 540
  151 FORMAT ( '1' )                                                  IP 550
  152 FORMAT ( ///5X, ' X(I) ' / )
  154 FORMAT (/////10X, 'PROBLEM ', I3/)                              IP 560
  155 FORMAT ( 5X, ' RATIO  .LE.  0.05 ***** CONVERGENCE *******' /   IP 570
     1 5X, '**********************************' )                     IP 580
  156 FORMAT ( 5X, 'CASE = ', I3 )                                    IP 590
  157 FORMAT ( 5X,I5, 10(1PE12.3 ) )                                  IP 600
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
  207    FORMAT (5X,'FOR .LE.-INEQ, JUST THE UNIT MATRIX ' /                    IP  610
     1   5X, 'FOR .GE.-INEQ, INSERT -1.0 IN THE REAR AND -M ON OBJ FCN'         IP  620
     2   / )                                                                    IP  630
  208  FORMAT(5X, ' INPUT DATA FORM' / )                                        IP  640
  201  FORMAT(5X, ' 1   ' / )                                                   IP  650
  202  FORMAT(5X, ' 4     7 ' / )                                               IP  660
  203  FORMAT(5X, ' 1.0  0    0    0    -G1X1-G1X2-G1X3' /                      IP  670
     1         5X, ' 0    1.0  0    0    -G2X1-G2X2-G2X3' /                      IP  680
     2         5X, ' 0    0    1.0  0    -G3X1-G3X2-G3X3' /                      IP  690
     3         5X, ' 0    0    0    1.0  -G4X1-G4X2-G4X3' / )                    IP  700
  204  FORMAT(5X, ' 0    0    0    0    FX1  FX2  FX3  ' /  )                    IP  710
  205  FORMAT(5X, ' G1C  G2C  G3C  G4C ' / )                                     IP  720
  209    FORMAT ( T5,'X(1)  ',T20,'X(2)  ',T35,'X(3)  ' / , 5X, 8E15.4/)         IP  730
C                                                                               IP  800
C ---------- LINEARIZATION  ----------                                          IP  810
C                                                                               IP  820
  101    FORMAT ( 2X, 10E13.4///)                                               IP  830
  102    FORMAT ( T5,'      ',T20,'      ',T35,'      ',T50 ,'      ',           IP  840
     1           T65,'      ',T80,'      ',T95,'      ',T110,'      ')           IP  850
  103    FORMAT ( T5,'XL    ',T20,'PLL1  ',T35,'SIGA  ',T50,'XK    ',           IP  860
     3           T65,'E     ',T80,'XM    ',T95,'PI    ',T110,'P     ')           IP  870
  104    FORMAT ( T5,'C1    ',T20,'C2    ',T35,'C3    ',T50,'C4    ',           IP  880
     4           T65,'C5    ',T80,'B1    ',T95,'B2    ',T110,'B3    ')           IP  890
  105    FORMAT ( T5,'F     ',T20,'G1    ',T35,'G2    ',T50 ,'G3    ',          IP  900
     1           T65,'G4    ',T80,'G5    ',T95,'G6    ',T110,'G7    ')           IP  910
  106    FORMAT ( T5,'FX1   ',T20,'FX2   ',T35,'FX3   ',T50 ,'G1X1  ',          IP  920
     1           T65,'G1X2  ',T80,'G1X3  ',T95,'      ',T110,'      ')           IP  930
  107    FORMAT ( T5,'G2X1  ',T20,'G2X2  ',T35,'G2X3  ',T50 ,'G3X1  ',          IP  940
     1           T65,'G3X2  ',T80,'G3X3  ',T95,'      ',T110,'      ')           IP  950
  108    FORMAT ( T5,'G4X1  ',T20,'G4X2  ',T35,'G4X3  ',T50 ,'FC    ',          IP  960
     1           T65,'G1C   ',T80,'G2C   ',T95,'G3C   ',T110,'G4C   ')           IP  970
  109    FORMAT ( T5,'FC    ',T20,'FX1   ',T35,'FX2   ',T50 ,'FT    ',          IP  980
     1           T65,'FTMX  ',T80,'FTMIN ',T95,'      ',T110,'      ')           IP  990
  111    FORMAT ( 20X, '0 - EXACT  G3' /                                        IP 1000
     1            20X, '1 - EXACT  G2' )                                        IP 1010
  112    FORMAT ( 20X, '0 - APPROX G3' /                                        IP 1020
     1            20X, '1 - APPROX G2' )                                        IP 1030
  113    FORMAT ( 20X, '0 - EXACT  G3', 10X, '2 - APPROX G3' /                  IP 1040
     1            20X, '1 - EXACT  G2', 10X, '3 - APPROX G2' )                  IP 1050
  114    FORMAT ( 8F10.3 )                                                      IP 1060
  115    FORMAT ( 5X, 3I5, 8E12.4 )                                             IP 1070
  591    FORMAT ( /15X, '   COEF VECTOR ' )                                     IP 1080
  592    FORMAT ( /15X, 'ART-VAR VECTOR ' )                                     IP 1090
  593    FORMAT ( /15X, 'OBJ-FCN VECTOR ' )                                     IP 1100
  594    FORMAT ( /15X, 'VECTOR B (PRDT) ' )                                    IP 1110
  595    FORMAT ( /15X, 'MATRIX C (1ST PART ' )                                 IP 1120
  596    FORMAT ( /15X, 'MATRIX C (2ND PART ' )                                 IP 1130
  597    FORMAT ( /15X, 'MATRIX C (3RD PART ' )                                 IP 1140
C                                                                               IP 1150
C   ----------  READ  IN  DATA  ----------(CON 01--17 )                         IP 1160
C                                                                               IP 1170
C                                                                               IP 1180
  900 READ 11, NPROB                                                            IP  750
      PRINT 151                                                                 IP  760
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      IF (NPROB) 1000, 9999, 1000                                       IP  770
 1000         NPROB = NPROB + 1
      PRINT 154, NPROB
C----------------        1ST RUN                                        IP 1190
C                                                                       IP 1200
              ITER = 0                                                  IP 1210
C     ITAB= 1 FOR ALL TABLES, 0 FOR JUST RESULTS                        IP 1220
C     N= NO. OF CONSTRAINTS, M= NO. OF VARIABLES                        IP 1230
      READ  11, NPR
      PRINT 11, NPR
         READ  14, LIN, NNL, MN, VART                                   IP 1260
         PRINT 14, LIN, NNL, MN, VART                                   IP 1270
              N = LIN + NNL                                             IP 1280
              M = N + MN                                                IP 1290
      PRINT 14,N,M                                                      IP 1300
         READ  22, ( XM(J),J=1, MN )                                    IP 1310
         PRINT 209                                                      IP 1320
         PRINT 101,( XM(J),J=1, MN )                                    IP 1330
         DO 880 I = 1, MN
  880         XP(I) = XM(I)
         CALL        LFCN ( GLX, GLC, BL, UB )
C                                                                       IP 1340
C----------          FOR .LE.-INEQ, JUST THE UNIT MATRIX                IP 1350
C--------      FOR .GE.-INEQ, INSERT -1.0 IN THE REAR AND -M ON OBJ FCN IP 1360
C----------         INPUT DATA FORM                                     IP 1370
C----------                                                             IP 1380
C----------                                                             IP 1390
C----------         1                                                   IP 1400
C----------         3    5                                              IP 1410
C--------               1.0  0    0    -G2X1-G2X2                       IP 1420
C----------             0    1.0  0    -G3X1-G3X2                       IP 1430
C----------             0    0    1.0  -G4X1-G4X2                       IP 1440
C----------             0    0    0    FX1  FX2                         IP 1450
C----------             G2C  G3C  G4C                                   IP 1460
C----------                                                             IP 1470
C----------                                                             IP 1480
C                                                                       IP 1490
         PRINT 207                                                      IP 1500
         PRINT 208                                                      IP 1510
         PRINT 201                                                      IP 1520
         PRINT 202                                                      IP 1530
         PRINT 203                                                      IP 1540
         PRINT 204                                                      IP 1550
         PRINT 205                                                      IP 1560
C                                                                       IP 1570
C   ----------  READ  IN  DATA  ----------(CON 01--17 )                 IP 1580
C                                                                       IP 1590
  705    CONTINUE                                                       IP 1600
              M = N + MN
              IF ( ITER .GT. 5 )  GO TO 9999
         CALL        PTCON ( N,     M,     C1, VALX, PROT, XM, ITER, FC) IP 1610
         CALL        BNEG1( N,      M, C1, PROT, VALX, VART, NC, C, C2,NG) IP 1620
C                                                                       IP 1630
C  ..........  PRINT  MATRIX  C  ..........                             IP 1640
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
     C                                                                       IP 1650
              M = NC                                                         IP 1660
     C-------------------  VALX = COEF OF X  IN OBJ FCN                      IP 1670
     C-------------------  PROT = VECTOR B                                   IP 1680
     C-----    PRINT 11, ITA8
               PRINT 11, N, M                                                IP 1700
     C                                                                       IP 1710
     C---------        SIMPLEX METHOD                                        IP 1720
     C                                                                       IP 1730
                    LP = 2                                                   IP 1740
                    IF ( LP .EQ. 2 ) GO TO 706                               IP 1750
               DO 751 K = 1, NC                                              IP 1760
                    VALX (K) = -VALX(K)                                      IP 1770
       751     CONTINUE                                                      IP 1780
                    LP = 1                                                   IP 1790
               PRINT 209                                                     IP 1800
               PRINT 101, ( X(I), I = 1, MN )                                IP 1810
               GO TO 707                                                     IP 1820
       706     CONTINUE                                                      IP 1830
               CALL        LP2(PROT,A,I,II,JJ,L,X ,C1,C2,N,M,NG,ZVAL,VALX)   IP 1840
       707     CONTINUE                                                      IP 1850
     C                                                                       IP 1860
     C                                                                       IP 1870
     C----------------     2ND RUN  TILL CONVERGENCE                         IP 1880
     C                                                                       IP 1890
     C-----    GO TO 871
       872          NZ = 0
               PRINT 152
               DO 870 I = 1, MN
                    IF ( X(I) .GT. BL(I) )  GO TO 873
                    NZ = NZ + 1
                    X(I) = XP(I)
       873     PRINT 5, I, X(I)
       870     CONTINUE
               PRINT 5, NZ
               CALL        FCNGR ( X,      F, FC, FX, G, GC, GX )            IP 8230
                    VMIN = F
     C------        VMIN = -ZVAL + FC
       871     CONTINUE
                    ITER = ITER + 1                                          IP 1920
                    IF ( ITER  .EQ.  1 )  GO TO 701                          IP 1930
                    AVMIN = ABS(VMIN)                                        IP 1940
                    DIFF = AVMIN - AZMAX                                     IP 1950
                    ADIFF = ABS(DIFF)
                    DIV  = ADIFF / AZMAX
                    RATIO = ABS ( DIV )                                      IP 1970
               PRINT 16, AZMAX , AVMIN, DIFF, RATIO                          IP 1980
       860          IF ( RATIO  .LE.  0.05 )  GO TO 901                      IP 1990
       701     CONTINUE
     C-----    DO 754 I = 1, MN
     C 754          XM(I) = 0.0
                    NZERO = 0                                                IP 2040
               DO 703 I = 1, MN                                              IP 2050
                    XM(I) = X(I)                                             IP 2060
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0217              IF ( XM(I) .NE. 0.0 )  GO TO 703                          IP 2070
0218              NZERO = NZERO + 1                                         IP 2080
0219              IZERO(NZERO) = I                                          IP 2090
0220   703   CONTINUE                                                       IP 2100
0221              IF (NZERO .LE. 1 )  GO TO 752                             IP 2110
0222         PRINT 11, NZERO, ( IZERO(K), K = 1, NZERO )                    IP 2120
0223         GO TO 753                                                      IP 2130
0224   752   PRINT 11, NZERO, IZERO(1)                                      IP 2140
0225   753   CONTINUE                                                       IP 2150
0226         PRINT 16 , FC, F
0227              ZMAX = VMIN
0228              AZMAX = ABS ( ZMAX )                                      IP 2010
0229              RATIO = 0.0
0230   702   PRINT 15, ITER, RATIO, ZVAL, VMIN                              IP 2160
0231              IF ( VMIN .LT. 0.0 )  GO TO 9999                          IP 2170
0232              IF ( NZERO .GT. 0 )  GO TO 9999                           IP 2180
0233         PRINT 151                                                      IP 2190
0234         GO TO 705                                                      IP 2200
0235   901   CONTINUE
0236         PRINT 16 , FC, F
0237         PRINT 15, ITER, RATIO, ZVAL, VMIN
0238         PRINT 155                                                      IP 2220
0239  9999 STOP                                                             IP 2240
0240       END                                                              IP 2250
```

80/80 LIST

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
               SUBROUTINE PTCON ( N,      M,     C1, VALX, PRDT, XM, ITER, FC)   IP 7590
               COMMON MN, LIN, NNL, NPR
               DIMENSION X(12), XM(12), XA(12)
               DIMENSION          PRDT(45), VALX(65)
               DIMENSION             C1(30,25), C(30,50)
               DIMENSION FX(12), G(25), GC(25), GX(25,12)
               DIMENSION  GLX(25,12), GLC(25)
               DIMENSION  BL(12), UB(12)
      C                                                                         IP 7680
      C                                                                         IP 7690
      C.................................................................        IP 7700
      C                 REQUIRED SUBROUTINES                                    IP 7710
      C                     (1)  SUBROUTINES  FCNGR                             IP 7720
      C                     (2)  SUBROUTINES  FPINTP-----REQ 4                  IP 7730
      C                     (3)  SUBROUTINES  LFCN                              IP 7740
      C.................................................................        IP 7750
         16    FORMAT ( 5X, 8E15.4 / )                                          IP 7760
         22    FORMAT ( 7E11.4 )                                                IP 7770
         23    FORMAT ( '  PTCON  ', I5, 8E13.4 / 14X, 8E13.4)
        101    FORMAT ( 2X, 8E15.4 /// )                                        IP 7780
        102    FORMAT ( T5,'       ',T20,'      ',T35,'      ',T50 ,'      ',   IP 7790
          1             T65,'      ',T80,'      ',T95,'      ',T110,'      ')   IP 7800
        103    FORMAT ( T5,'XL     ',T20,'PLL1  ',T35,'SIGA  ',T50,'XK     ',   IP 7810
          3             T65,'E     ',T80,'XM    ',T95,'PI    ',T110,'P     ')   IP 7820
        104    FORMAT ( T5,'C1     ',T20,'C2    ',T35,'C3    ',T50,'C4     ',   IP 7830
          4             T65,'C5    ',T80,'B1    ',T95,'B2    ',T110,'B3    ')   IP 7840
        105    FORMAT ( T5,'F      ',T20,'G1    ',T35,'G2    ',T50 ,'G3    ',   IP 7850
          1             T65,'G4    ',T80,'G5    ',T95,'G6    ',T110,'G7    ')   IP 7860
        106    FORMAT ( T5,'FX1    ',T20,'FX2   ',T35,'FX3   ',T50 ,'G1X1  ',   IP 7870
          1             T65,'G1X2  ',T80,'G1X3  ',T95,'      ',T110,'      ')   IP 7880
        107    FORMAT ( T5,'G2X1   ',T20,'G2X2  ',T35,'G2X3  ',T50 ,'G3X1  ',   IP 7890
          1             T65,'G3X2  ',T80,'G3X3  ',T95,'      ',T110,'      ')   IP 7900
        108    FORMAT ( T5,'G4X1   ',T20,'G4X2  ',T35,'G4X3  ',T50 ,'FC    ',   IP 7910
          1             T65,'G1C   ',T80,'G2C   ',T95,'G3C   ',T110,'G4C   ')   IP 7920
        109    FORMAT ( T5,'FC     ',T20,'FX1   ',T35,'FX2   ',T50 ,'FT    ',   IP 7930
          1             T65,'FTMX  ',T80,'FTMIN ',T95,'      ',T110,'      ')   IP 7940
        114    FORMAT ( 8F10.3 )                                                IP 7950
        200    FORMAT ( 5X, E15.4, 7E15.7 )                                     IP 7960
        202    FORMAT (/5X, ' INFEASIBLE POINT ' / )                            IP 7970
        203    FORMAT (/5X, ' OUTSIDE FEASIBLE REGION, BACK UP ' / )            IP 7980
        204    FORMAT (/5X, ' POINT ON THE CONSTRAINT G1 ' / )                  IP 7990
        205    FORMAT (/5X, ' POINT ON THE CONSTRAINT G2 ' / )                  IP 8000
        206    FORMAT (/5X, ' POINT ON THE CONSTRAINT G3 ' / )                  IP 8010
        207    FORMAT (/5X, ' POINT ON THE CONSTRAINT G', I2, / )               IP 8020
        601  FORMAT (1X,'A',E15.4, 7E15.7 )                                     IP 8030
        602  FORMAT (1X,'B',E15.4, 7E15.7 )                                     IP 8040
        603  FORMAT (1X,'C',E15.4, 7E15.7 )                                     IP 8050
        604  FORMAT (1X,'D',E15.4, 7E15.7 )                                     IP 8060
        605  FORMAT (1X,'E',E15.4, 7E15.7 )                                     IP 8070
        606  FORMAT (1X,'F',E15.4, 7E15.7 )                                     IP 8080
        607  FORMAT (1X,'G',E15.4, 7E15.7 )                                     IP 8090
        608  FORMAT (1X,'H',E15.4, 7E15.7 )                                     IP 8100
        609  FORMAT (1X,'I',E15.4, 7E15.7 )                                     IP 8110
        610  FORMAT (1X,'J',E15.4, 7E15.7 )                                     IP 8120
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
       611  FORMAT (1X,'K',E15.4, 7E15.7 )                                        IP 8130
       612  FORMAT (1X,'L',E15.4, 7E15.7 )                                        IP 8140
       613    FORMAT ( /5X, ' AWAY FROM G1' / )                                   IP 8150
       614    FORMAT ( /5X, ' AWAY FROM G2' / )                                   IP 8160
       615    FORMAT ( /5X, ' AWAY FROM G3' / )                                   IP 8170
       616    FORMAT ( /5X, ' AWAY FROM G', I2, / )                               IP 8180
              DO 218 I = 1, MN                                                    IP 8190
       218         XA(I) = XM(I)                                                  IP 8200
              DO 219 I = 1, MN                                                    IP 8210
       219          X(I) =XA(I)                                                   IP 8220
              PRINT 23 , ITER, ( X(I), I = 1, MN )
              CALL       FCNGR ( X,       F, FC, FX, G, GC, GX )                  IP 8230
                   FCT = FC                                                       IP 8240
              DO 515 I = 1, N                                                     IP 8250
       515         VALX(I) = 0.0                                                  IP 8260
                   K = 0                                                          IP 8270
              DO 516 K = 1, MN                                                    IP 8280
       516         VALX (N+K) = FX(K)                                             IP 8290
              PRINT 16, ( VALX(I), I = 1, M )                                     IP 8300
                   IF ( ITER .NE. 0 ) GO TO 710                                   IP 8310
     C-----------  TEST FOR FEASIBILITY                                           IP 8320
                   IF ( NNL .EQ. 0 )  GO TO 708                                   IP 8330
              PRINT 16,  ( G(K), K = 1, NNL )
              DO 720 K = 1, NNL                                                   IP 8350
                   IF ( G(K) .GE. 0.0 )  GO TO 720                                IP 8360
              PRINT 202                                                           IP 8370
       720    CONTINUE                                                            IP 8380
              PRINT 16,  ( G(K), K = 1, NNL )
       708    CONTINUE                                                            IP 8390
       710    CONTINUE                                                            IP 8400
     C                                                                            IP 8410
     C    ...............................................................         IP 8420
     C                                                                            IP 8430
     C----------          WHEN VECTOR B IS NEGATIVE DO THE FCLLOWING              IP 8440
     C  ----  FOUR CASES                                                          IP 8450
     C 1----  ALL +VE                                                             IP 8460
     C 2----  (1) -VE,/ (2) -VE,/ (3) -VE                                         IP 8470
     C 3----  (1), (2), -VE/  (1) (3) -VE /(2), (3) -VE                           IP 8480
     C 4----  (1), (2), (3) -VE                                                   IP 8490
     C                                                                            IP 8500
     C  ..........   ALL +VE                                                      IP 8510
     C                                                                            IP 8520
                       KCASE = 100                                                IP 8530
     C                                                                            IP 8540
     C    ...............................................................         IP 8550
     C                                                                            IP 8560
     C                                                                            IP 8570
     C----------   NON-LINEAR COND                                                IP 8580
     C  ..........  PUT POINT ON SURFACE  .........                               IP 8590
     C                                                                            IP 8600
                   NEQ = 7
                   NEQ = 0
                   KNN = 0
                   KN = 0
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                 KC = 0
                 IF ( NNL .EQ. 0 )  GO TO 391
    350     CONTINUE
                 KN = KN + 1
                 IF ( KN .GT. NNL ) GO TO 391
                 IF ( ITER .NE. 0 )  GO TO 392
C----------      FOR EQUALITY FROM NEQ ON
                 IF ( NEQ .EQ. 0 )  GO TO 394
                 IF ( KN .LT. NEQ )  GO TO 394
                 KNN = KNN + 1
            DO 395 I = 1, MN
    395          XA(I) = XM(I)
            CALL        FPINTP  ( XA, KNN, X )
            GO TO 392
    394     CONTINUE
            DO 303 I = 1, MN                                                 IP 8680
    303          XA(I) = XM(I)                                               IP 8690
            CALL        FPINTP  ( XA, KN, X )                                IP 8700
    392     CONTINUE                                                         IP 8710
            PRINT 23 , ITER, ( X(I), I = 1, MN )
            CALL        FCNGR ( X,      F, FC, FX, G, GC, GX )               IP 8720
                 PRDT(KN) = GC(KN)                                           IP 8730
C                                                                            IP 8740
C  ..........  FORM MATRIX C1 ROW-WISE  ..........                           IP 8750
C                                                                            IP 8760
            DO 390 J = 1, MN                                                 IP 8770
    390          C1(KN,J) = -GX(KN,J)                                        IP 8780
            GO TO 350                                                        IP 8790
    391     CONTINUE                                                         IP 8800
                 IF ( NNL .NE. 0 )  GO TO 393                                IP 8810
                 KN = 1                                                      IP 8820
    393     CONTINUE                                                         IP 8830
C----------      LINEAR COND                                                 IP 8840
                 IF ( LIN .EQ. 0 )  GO TO 9999                               IP 8850
            CALL        LFCN ( GLX, GLC, BL, UB )
    370     CONTINUE                                                         IP 8870
                 KC = KC + 1                                                 IP 8880
                 IF (KN .GT. (LIN+NNL) ) GO TO 9999                          IP 8890
                 PRDT (KN) = GLC(KC )                                        IP 8900
            DO 360 J = 1, MN                                                 IP 8910
    360          C1 (KN,J) =-GLX(KC,J)                                       IP 8920
                 KN = KN + 1                                                 IP 8930
            GO TO 370                                                        IP 8940
   9999          FC = FCT                                                    IP 8950
            RETURN                                                           IP 8960
            END                                                              IP 8970
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
            SUBROUTINE  FPINTP  ( XA, KN, X )                                   IP 8980
      EXTERNAL FUNCT                                                            IP 8990
            COMMON MN, LIN, NNL, NPR
      DIMENSION X(12 ),G(12 ),H(114)
            DIMENSION  FA(8 ), XA(12), X1(12), X2(12)
C.......................................................................        IP 9030
C                REQUIRED SUBROUTINES                                           IP 9040
C                       (1)  SUBROUTINES  FPM ------ REQ 2                      IP 9050
C                       (2)  SUBROUTINES  INTP                                  IP 9060
C.......................................................................        IP 9070
    1 FORMAT(19H THE MINIMUM VALUE=,E15.5/9H THE DEL=,6E10.4)                  IP 9080
    2 FORMAT(14H OUR POINT IS=,2E15.4)                                         IP 9090
   11       FORMAT ( 10I5)                                                      IP 9100
   14       FORMAT ( 3I5, 3F10.3 )                                              IP 9110
   22       FORMAT ( 7E11.4 )                                                   IP 9120
  100       FORMAT ( '1' )                                                      IP 9130
  101       FORMAT ( 7E11.4 )                                                   IP 9140
  102       FORMAT ( '  FPINTP  ', 10I5 )
  103       FORMAT ( 5X, ' DID NOT REACH G = 0 ' )
  107       FORMAT ( 10I5 )                                                     IP 9150
  500       FORMAT ( '1' )                                                      IP 9160
  802       FORMAT ( 5X, 6E14.4 / )                                             IP 9170
                 NZ= 0
      EST=0.0                                                                   IP 9180
                 EPS = 0.1E-05                                                  IP 9190
      LIMIT= 20
                 NFCN = KN                                                      IP 9210
           DO 303 I = 1, MN                                                     IP 9220
                X1(I) = 0.0                                                     IP 9230
                X2(I) = 0.0                                                     IP 9240
  303      CONTINUE                                                             IP 9250
           DO 203 I = 1, MN                                                     IP 9260
  203           X(I) = XA(I)                                                    IP 9270
           PRINT 102, KN
      CALL FPM (FUNCT,    X, FS, G, EST, EPS, LIMIT, IER, H, NFCN,             IP 9280
     1            FA, X1, X2 )                                                  IP 9290
                 N = 2                                                          IP 9300
                 FAIN = 0.0E 00                                                 IP 9310
           PRINT 101, (X1(K), K = 1, MN )                                       IP 9320
           PRINT 802, FA(1)
           PRINT 101, (X2(K), K = 1, MN )                                       IP 9330
           PRINT 802, FA(2)
           DO 302 K = 1, MN
                 IF ( X2(K) .GT. 0.001 )  GO TO 302
                 NZ  = NZ + 1
           PRINT 103
  302      CONTINUE
                 IF ( NZ .GT. 0 )  CALL EXIT
           DO 201 K = 1, MN                                                     IP 9340
                 IF ( ABS( X1(K) - X2(K) )  .LT. 0.02   )  GO TO 202            IP 9350
                 XA(1) = X1(K)                                                  IP 9360
                 XA(2) = X2(K)                                                  IP 9370
           CALL         INTP ( N, FA, XA, FAIN, YOUT )                          IP 9380
                 X(K) = YOUT                                                    IP 9390
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0055           GO TO 201                                                        IP 9400
0056   202          X(K) = X1(K)                                                IP 9410
0057   201     CONTINUE                                                         IP 9420
0058           PRINT 101, ( X(K), K = 1, MN )                                   IP 9430
0059   301     CONTINUE                                                         IP 9450
0060           RETURN                                                           IP 9460
0061          END                                                               IP 9470
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      SUBROUTINE FPM (FUNCT,  X,F,G,EST,EPS,LIMIT,IER,H, NFCN,              IP    10
     1               FA, X1, X2 )                                           IP    20
         COMMON MN, LIN, NNL, NPR
C        DIMENSIONED DUMMY VARIABLES                                        IP    40
      DIMENSION H(1),X(1),G(1)                                              IP    50
         DIMENSION  FA(8 ),         X1(12), X2(12)
C                                                                           IP    70
C     ..................................................................... IP    80
C                                                                           IP    90
C        SUBROUTINE FPM                                                     IP   100
C                   MODIFIED FMFP                                           IP   110
C                                                                           IP   120
C                                                                           IP   130
C     ..................................................................... IP   140
C                                                                           IP   150
C                                                                           IP   160
C                                                                           IP   170
C....................................................................        IP   180
C             REQUIRED SUBROUTINES                                          IP   190
C                 (1)  SUBROUTINES  FUNCT ----- REQ 1                       IP   200
C....................................................................        IP   210
  101    FORMAT ( 7E11.4 )                                                  IP   220
  105    FORMAT (/'  F1' , 7E11.4 / )                                       IP   230
  102    FORMAT (/'  F2' , 7E11.4 / )                                       IP   240
  103    FORMAT (/'  F3' , 7E11.4 / )                                       IP   250
  104    FORMAT (/'  F4' , 7E11.4 / )                                       IP   260
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT    IP   270
              N = MN                                                        IP   280
              MN = N                                                        IP   290
         CALL    FUNCT(     X, F , G, NFCN, X1, X2, FA )                    IP   300
              IF ( NPR .EQ. 0 )  GO TO 60
         PRINT 105, F
   60    CONTINUE
              IF ( F )  61, 56, 62                                          IP   320
   62         FA(1) = F                                                     IP   330
         DO 64 I = 1, N                                                     IP   340
   64         X1(I) = X(I)                                                  IP   350
              IF ( ABS(F) .GT. 0.2E-02 )  GO TO 67
         DO 66 I = 1, N
   66         X2(I) = X1(I)
         RETURN
   67    CONTINUE
              IF ( NPR .EQ. 0 )  GO TO 68
         PRINT 101, (X1(K), K = 1, MN )
   68    CONTINUE
         GO TO 63                                                           IP   370
   61         FA(2) = F                                                     IP   380
         DO 65 I = 1, N                                                     IP   390
   65         X2(I) = X(I)                                                  IP   400
              IF ( NPR .EQ. 0 )  GO TO 69
         PRINT 101, (X2(K), K = 1, MN )
   69    CONTINUE
         GO TO 56                                                           IP   420
   63    CONTINUE                                                           IP   430
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
      C                                                                  IP   440
      C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX      IP   450
            IER=0                                                        IP   460
            KOUNT=0                                                      IP   470
            N2=N+N                                                       IP   480
            N3=N2+N                                                      IP   490
            N31=N3+1                                                     IP   500
          1 K=N31                                                        IP   510
            DO 4 J=1,N                                                   IP   520
            H(K)=1.                                                      IP   530
            NJ=N-J                                                       IP   540
            IF(NJ)5,5,2                                                  IP   550
          2 DO 3 L=1,NJ                                                  IP   560
            KL=K+L                                                       IP   570
          3 H(KL)=0.                                                     IP   580
          4 K=KL+1                                                       IP   590
      C                                                                  IP   600
      C        START ITERATION LOOP                                      IP   610
          5 KOUNT=KOUNT +1                                               IP   620
      C                                                                  IP   630
      C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR  IP   640
            OLDF=F                                                       IP   650
            DO 9 J=1,N                                                   IP   660
            K=N+J                                                        IP   670
            H(K)=G(J)                                                    IP   680
            K=K+N                                                        IP   690
            H(K)=X(J)                                                    IP   700
      C                                                                  IP   710
      C        DETERMINE DIRECTION VECTOR H                              IP   720
            K=J+N3                                                       IP   730
            T=0.                                                         IP   740
            DO 8 L=1,N                                                   IP   750
            T=T-G(L)*H(K)                                                IP   760
            IF(L-J)6,7,7                                                 IP   770
          6 K=K+N-L                                                      IP   780
            GO TO 8                                                      IP   790
          7 K=K+1                                                        IP   800
          8 CONTINUE                                                     IP   810
          9 H(J)=T                                                       IP   820
      C                                                                  IP   830
      C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.    IP   840
            DY=0.                                                        IP   850
            HNRM=0.                                                      IP   860
            GNRM=0.                                                      IP   870
      C                                                                  IP   880
      C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION IP   890
      C        VECTOR H AND GRADIENT VECTOR G.                           IP   900
            DO 10 J=1,N                                                  IP   910
            HNRM=HNRM+ABS(H(J))                                          IP   920
            GNRM=GNRM+ABS(G(J))                                          IP   930
         10 DY=DY+H(J)*G(J)                                              IP   940
      C                                                                  IP   950
      C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL IP   960
      C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.                IP   970
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      IF(DY)11,51,51                                                    IP  980
C                                                                       IP  990
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION    IP 1000
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.               IP 1010
   11 IF(HNRM/GNRM-EPS)51,51,12                                         IP 1020
C                                                                       IP 1030
C        SEARCH MINIMUM ALONG DIRECTION H                               IP 1040
C                                                                       IP 1050
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE             IP 1060
   12 FY=F                                                              IP 1070
      ALFA=2.*(EST-F)/DY                                                IP 1080
      AMBDA=1.                                                          IP 1090
C                                                                       IP 1100
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN IP 1110
C        1. OTHERWISE TAKE 1. AS STEPSIZE                               IP 1120
      IF(ALFA)15,15,13                                                  IP 1130
   13 IF(ALFA-AMBDA)14,15,15                                            IP 1140
   14 AMBDA=ALFA                                                        IP 1150
   15 ALFA=0.                                                           IP 1160
C                                                                       IP 1170
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT           IP 1180
   16 FX=FY                                                             IP 1190
      DX=DY                                                             IP 1200
C                                                                       IP 1210
C        STEP ARGUMENT ALONG H                                          IP 1220
      DO 17 I=1,N                                                       IP 1230
   17 X(I)=X(I)+AMBDA*H(I)                                              IP 1240
C                                                                       IP 1250
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT           IP 1260
         CALL    FUNCT(     X, F , G, NFCN, X1, X2, FA )                IP 1270
              IF ( NPR .EQ. 0 )  GO TO 70
         PRINT 102, F
   70    CONTINUE
              IF ( F )  71, 56, 72                                      IP 1290
   72         FA(1) = F                                                 IP 1300
         DO 74 I = 1, N                                                 IP 1310
   74         X1(I) = X(I)                                              IP 1320
              IF ( ABS(F) .GT. 0.2E-02 )  GO TO 77
         DO 76 I = 1, N
   76         X2(I) = X1(I)
         RETURN
   77    CONTINUE
              IF ( NPR .EQ. 0 )  GO TO 78
         PRINT 101, (X1(K), K = 1, MN )
   78    CONTINUE
         GO TO 73                                                       IP 1340
   71         FA(2) = F                                                 IP 1350
         DO 75 I = 1, N                                                 IP 1360
   75         X2(I) = X(I)                                              IP 1370
              IF ( NPR .EQ. 0 )  GO TO 79
         PRINT 101, (X2(K), K = 1, MN )
   79    CONTINUE
         GO TO 56                                                       IP 1390
   73    CONTINUE                                                       IP 1400
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      FY=F                                                              IP 1410
C                                                                       IP 1420
C        COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE IP 1430
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND  IP 1440
      DY=0.                                                             IP 1450
      DO 18 I=1,N                                                       IP 1460
   18 DY=DY+G(I)*H(I)                                                   IP 1470
      IF(DY)19,36,22                                                    IP 1480
C                                                                       IP 1490
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT     IP 1500
C        A MINIMUM HAS BEEN PASSED                                      IP 1510
   19 IF(FY-FX)20,22,22                                                 IP 1520
C                                                                       IP 1530
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES         IP 1540
   20 AMBDA=AMBDA+ALFA                                                  IP 1550
      ALFA=AMBDA                                                        IP 1560
C        END OF SEARCH LOOP                                             IP 1570
C                                                                       IP 1580
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE            IP 1590
      IF(HNRM*AMBDA-1.E10)16,16,21                                      IP 1600
C                                                                       IP 1610
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS       IP 1620
   21 IER=2                                                             IP 1630
      RETURN                                                            IP 1640
C                                                                       IP 1650
C        INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH    IP 1660
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION   IP 1670
C        POLYNOMIAL IS MINIMIZED                                        IP 1680
   22 T=0.                                                              IP 1690
   23 IF(AMBDA)24,36,24                                                 IP 1700
   24 Z=3.*(FX-FY)/AMBDA+DX+DY                                          IP 1710
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))                                IP 1720
      DALFA=Z/ALFA                                                      IP 1730
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA                                 IP 1740
      IF(DALFA)51,25,25                                                 IP 1750
   25 W=ALFA*SQRT(DALFA)                                                IP 1760
      ALFA=DY-DX+W+W                                                    IP 1770
      IF(ALFA) 250,251,250                                              IP 1780
  250 ALFA=(DY-Z+W)/ALFA                                                IP 1790
      GO TO 252                                                         IP 1800
  251 ALFA=(Z+DY-W)/(Z+DX+Z+DY)                                         IP 1810
  252 ALFA=ALFA*AMBDA                                                   IP 1820
      DO 26 I=1,N                                                       IP 1830
   26 X(I)=X(I)+(T-ALFA)*H(I)                                           IP 1840
C                                                                       IP 1850
C        TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS    IP 1860
C        THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE IP 1870
C        THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT   IP 1880
C        THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE  IP 1890
C        VALUE OF THE FUNCTION AND ITS GRADIENT AT X                    IP 1900
C                                                                       IP 1910
         CALL    FUNCT(     X, F , G, NFCN, X1, X2, FA )                IP 1920
              IF ( NPR .EQ. 0 )  GO TO 80
         PRINT 103, F
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
       80     CONTINUE
                 IF ( F )  81, 56, 82                                            IP 1940
       82        FA(1) = F                                                       IP 1950
              DO 84 I = 1, N                                                     IP 1960
       84        X1(I) = X(I)                                                    IP 1970
                 IF ( ABS(F) .GT. 0.2E-02 )  GO TO 87
              DO 86 I = 1, N
       86        X2(I) = X1(I)
              RETURN
       87     CONTINUE
                 IF ( NPR .EQ. 0 )  GO TO 88
              PRINT 101, (X1(K), K = 1, MN )
       88     CONTINUE
              GO TO 83                                                           IP 1990
       81        FA(2) = F                                                       IP 2000
              DO 85 I = 1, N                                                     IP 2010
       85        X2(I) = X(I)                                                    IP 2020
                 IF ( NPR .EQ. 0 )  GO TO 89
              PRINT 101, (X2(K), K = 1, MN )
       89     CONTINUE
              GO TO 56                                                           IP 2040
       83     CONTINUE                                                           IP 2050
           IF(F-FX)27,27,28                                                      IP 2060
       27 IF(F-FY)36,36,28                                                       IP 2070
       28 DALFA=0.                                                               IP 2080
          DO 29 I=1,N                                                            IP 2090
       29 DALFA=DALFA+G(I)*H(I)                                                  IP 2100
          IF(DALFA)30,33,33                                                      IP 2110
       30 IF(F-FX)32,31,33                                                       IP 2120
       31 IF(DX-DALFA)32,36,32                                                   IP 2130
       32 FX=F                                                                   IP 2140
          DX=DALFA                                                               IP 2150
          T=ALFA                                                                 IP 2160
          AMBDA=ALFA                                                             IP 2170
          GO TO 23                                                               IP 2180
       33 IF(FY-F)35,34,35                                                       IP 2190
       34 IF(DY-DALFA)35,36,35                                                   IP 2200
       35 FY=F                                                                   IP 2210
          DY=DALFA                                                               IP 2220
          AMBDA=AMBDA-ALFA                                                       IP 2230
          GO TO 22                                                               IP 2240
     C                                                                           IP 2250
     C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION     IP 2260
       36 IF(OLDF-F+EPS)51,38,38                                                 IP 2270
     C                                                                           IP 2280
     C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM           IP 2290
     C        TWO CONSECUTIVE ITERATIONS                                         IP 2300
       38 DO 37 J=1,N                                                            IP 2310
          K=N+J                                                                  IP 2320
          H(K)=G(J)-H(K)                                                         IP 2330
          K=N+K                                                                  IP 2340
       37 H(K)=X(J)-H(K)                                                         IP 2350
     C                                                                           IP 2360
     C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR     IP 2370
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF     IP 2380
C        BOTH ARE LESS THAN  EPS                                        IP 2390
      IER=0                                                             IP 2400
      IF(KOUNT-N)42,39,39                                               IP 2410
   39 T=0.                                                              IP 2420
      Z=0.                                                              IP 2430
      DO 40 J=1,N                                                       IP 2440
      K=N+J                                                             IP 2450
      W=H(K)                                                            IP 2460
      K=K+N                                                             IP 2470
      T=T+ABS(H(K))                                                     IP 2480
   40 Z=Z+W*H(K)                                                        IP 2490
      IF(HNRM-EPS)41,41,42                                              IP 2500
   41 IF(T-EPS)56,56,42                                                 IP 2510
C                                                                       IP 2520
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT         IP 2530
   42 IF(KOUNT-LIMIT)43,50,50                                           IP 2540
C                                                                       IP 2550
C        PREPARE UPDATING OF MATRIX H                                   IP 2560
   43 ALFA=0.                                                           IP 2570
      DO 47 J=1,N                                                       IP 2580
      K=J+N3                                                            IP 2590
      W=0.                                                              IP 2600
      DO 46 L=1,N                                                       IP 2610
      KL=N+L                                                            IP 2620
      W=W+H(KL)*H(K)                                                    IP 2630
      IF(L-J)44,45,45                                                   IP 2640
   44 K=K+N-L                                                           IP 2650
      GO TO 46                                                          IP 2660
   45 K=K+1                                                             IP 2670
   46 CONTINUE                                                          IP 2680
      K=N+J                                                             IP 2690
      ALFA=ALFA+W*H(K)                                                  IP 2700
   47 H(J)=W                                                            IP 2710
C                                                                       IP 2720
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS      IP 2730
C        ARE NOT SATISFACTORY                                           IP 2740
      IF(Z*ALFA)48,1,48                                                 IP 2750
C                                                                       IP 2760
C        UPDATE MATRIX H                                                IP 2770
   48 K=N31                                                             IP 2780
      DO 49 L=1,N                                                       IP 2790
      KL=N2+L                                                           IP 2800
      DO 49 J=L,N                                                       IP 2810
      NJ=N2+J                                                           IP 2820
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA                            IP 2830
   49 K=K+1                                                             IP 2840
      GO TO 5                                                           IP 2850
C        END OF ITERATION LOOP                                          IP 2860
C                                                                       IP 2870
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS                        IP 2880
   50 IER=1                                                             IP 2890
      RETURN                                                            IP 2900
C                                                                       IP 2910
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS                   IP 2920
   51 DO 52 J=1,N                                                       IP 2930
      K=N2+J                                                            IP 2940
   52 X(J)=H(K)                                                         IP 2950
         CALL    FUNCT(     X, F , G, NFCN, X1, X2, FA )                IP 2960
              IF ( NPR .EQ. 0 )  GO TO 90
         PRINT 104, F
   90    CONTINUE
              IF ( F )  91, 56, 92                                      IP 2980
   92         FA(1) = F                                                 IP 2990
         DO 94 I = 1, N                                                 IP 3000
   94         X1(I) = X(I)                                              IP 3010
              IF ( ABS(F) .GT. 0.2E-02 )  GO TO 97
         DO 96 I = 1, N
   96         X2(I) = X1(I)
         RETURN
   97    CONTINUE
              IF ( NPR .EQ. 0 )  GO TO 98
         PRINT 101, (X1(K), K = 1, MN )
   98    CONTINUE
         GO TO 93                                                       IP 3030
   91         FA(2) = F                                                 IP 3040
         DO 95 I = 1, N                                                 IP 3050
   95         X2(I) = X(I)                                              IP 3060
              IF ( NPR .EQ. 0 )  GO TO 99
         PRINT 101, (X2(K), K = 1, MN )
   99    CONTINUE
         GO TO 56                                                       IP 3080
   93    CONTINUE                                                       IP 3090
C                                                                       IP 3100
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE   IP 3110
C        FAILS TO BE SUFFICIENTLY SMALL                                 IP 3120
      IF(GNRM-EPS)55,55,53                                              IP 3130
C                                                                       IP 3140
C        TEST FOR REPEATED FAILURE OF ITERATION                         IP 3150
   53 IF(IER)56,54,54                                                   IP 3160
   54 IER=-1                                                            IP 3170
      GOTO 1                                                            IP 3180
   55 IER=0                                                             IP 3190
   56 RETURN                                                            IP 3200
      END                                                               IP 3210
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      SUBROUTINE FUNCT(     X, FS, G, NFCN, X1, X2, FA )                 IP 3220
         COMMON MN, LIN, NNL, NPR
      DIMENSION GF(15)                                                   IP 3240
         DIMENSION  FA(8 ),         X1(12), X2(12)
      DIMENSION X(12 ),G(12)
         DIMENSION FX(12),        GC(25), GX(25,12)
C                                                                        IP 3300
C                                                                        IP 3310
C                                                                        IP 3360
C                                                                        IP 3370
C.................................................................       IP 3380
C              REQUIRED SUBROUTINES                                      IP 3390
C                  (1)  SUBROUTINES  FCNGR                               IP 3400
C.................................................................       IP 3410
C----------     DEFINE NNL,LF, J, IN SUBR FUNCT                          IP 3320
  790    FORMAT ( 1X, ' FS NEGATIVE ', E15.4 )                           IP 3420
  801    FORMAT ( 5X, 2I5, 7E14.4 / 15X, 7E14.4 / 15X, 7E14.4 / )        IP 3430
  802    FORMAT ( 5X, 6E14.4 / )                                         IP 3440
  804    FORMAT(15X, ' A ', 6E14.4/18X,6E14.4//)
  805    FORMAT ( 5X, 6E14.4    )                                        IP 3460
  806    FORMAT ( 15X, ' FC' , 6E14.4 //)                                IP 3470
  807    FORMAT (/    , ' FS' , 6E14.4 //)                               IP 3480
C                                                                        IP 3490
              LF = 0                                                     IP 3500
         CALL       FCNGR ( X,       F, FC, FX,GF, GC, GX )              IP 3520
C ..................................................................     IP 3530
              J = NFCN                                                   IP 3540
              FS = GF(J)                                                 IP 3560
         DO 850 K = 1, MN                                                IP 3570
  850         G(K) = GX(J,K)                                             IP 3580
C                                                                        IP 3590
C ..................................................................     IP 3600
C-----   PRINT 804, ( G(I), I = 1,MN ), FS
C-----   PRINT 807, FS
 9999    RETURN                                                          IP 3630
         END
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         SUBROUTINE  INTP ( N, X, Y, XIN, YOUT )                        IP 9480
         COMMON MN, LIN, NNL, NPR
         DIMENSION  X(12), Y(12)
C.....................................................................  IP 9500
C             REQUIRED SUBROUTINES                                      IP 9510
C                 NONE                                                  IP 9520
C.....................................................................  IP 9530
  100    FORMAT ( I5/ (2E11.4 ) )                                       IP 9540
  101    FORMAT ( 7E11.4 )                                              IP 9550
  105    FORMAT (/5X, I5 / ( 5X, 2E11.4) )                              IP 9560
  106    FORMAT (/5X,  E11.4//)                                         IP 9570
  107    FORMAT ( 10I5 )                                                IP 9580
  200    FORMAT ( ' LAG POL INTERP AMONG', I3, ' VALUES'  / ' AT X = ', IP 9590
     1          F10.3, ' FCN HAS VALUE OF ', F10.4 )                    IP 9600
  500    FORMAT ( '1' )                                                 IP 9610
              YOUT = 0.0                                                IP 9640
         DO 20 I = 1, N                                                 IP 9650
              TERM = Y(I)                                               IP 9660
         DO 10 J = 1, N                                                 IP 9670
              IF ( I - J )  9, 10, 9                                    IP 9680
    9         TERM = TERM * ( XIN - X(J) ) / ( X(I) - X(J) )            IP 9690
   10    CONTINUE                                                       IP 9700
   20         YOUT = YOUT + TERM                                        IP 9710
              IF ( NPR .EQ. 0 )  GO TO 25
         PRINT 105, N , ( X(I), Y(I), I = 1, N )                        IP 9620
         PRINT 106, XIN                                                 IP 9630
         PRINT 200, N, XIN, YOUT                                        IP 9720
   25    CONTINUE
         RETURN                                                         IP 9730
         END                                                            IP 9740
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      SUBROUTINE BNEGI( N,     M, C1, PRDT, VALX, VART, NC, C, C2,NG) IP 2260
      DIMENSION          PRDT(45), VALX(65)
      DIMENSION C1(30,12), C2(30,25), C(30,50)
      DIMENSION                    UM(30,50),ZC1(30,50),ZC2(30,50)
      COMMON MN, LIN, NNL, NPR
C                                                                       IP 2300
C  ..........  (2)  SUBROUTINE  BNEG  ... HANDLES NEGATIVE ELE IN B  ... IP 2310
C                                                                       IP 2320
C.............................................................        IP 2330
C              REQUIRED SUBROUTINES                                     IP 2340
C                  NONE                                                 IP 2350
C.............................................................        IP 2360
   11     FORMAT ( 10I5)                                                IP 2370
   22     FORMAT ( 7E11.4 )                                             IP 2380
  101     FORMAT ( 2X, 8E15.4 /// )                                     IP 2390
  156     FORMAT ( 5X, 'CASE = ', I3 )                                  IP 2400
  591     FORMAT ( /15X, '   COEF VECTOR ' )                            IP 2410
  592     FORMAT ( /15X, 'ART-VAR VECTOR ' )                            IP 2420
  593     FORMAT ( /15X, 'OBJ-FCN VECTOR ' )                            IP 2430
  594     FORMAT ( /15X, 'VECTOR B (PRDT) ' )                           IP 2440
  595     FORMAT ( /15X, 'MATRIX C (1ST PART ' )                        IP 2450
  596     FORMAT ( /15X, 'MATRIX C (2ND PART ' )                        IP 2460
  597     FORMAT ( /15X, 'MATRIX C (3RD PART ' )                        IP 2470
  598     FORMAT ( /15X, 'MATRIX C ' / )                                IP 2480
C                                                                       IP 2490
C  ..........  TEST FOR NEG ELE IN VECTOR B (PRDT)  ..........          IP 2500
C                                                                       IP 2510
               NPR = 0                                                  IP 2520
               NPR = 1
                    KCASE = 100                                         IP 2530
                      NG = 0                                            IP 2540
          DO 515 J = 1, N                                               IP 2550
               IF ( PRDT(J) .GT. 0.0 ) GO TO 515                        IP 2560
               NG = NG + 1                                              IP 2570
  515     CONTINUE                                                      IP 2580
          PRINT 156, KCASE                                              IP 2590
C                                                                       IP 260C
C  .......... COEF  VECTOR  ..........                                  IP 2610
C                                                                       IP 2620
          DO 520 I = 1, N                                               IP 2630
          DO 520 J = 1, MN                                              IP 2640
               IF (PRDT(I) .LT. 0.0 ) GO TO 521                         IP 2650
                 C1(I,J) = C1(I,J)                                      IP 2660
          GO TO 520                                                     IP 2670
  521            C1(I,J) = -C1(I,J)                                     IP 2680
  520     CONTINUE                                                      IP 2690
          PRINT 591                                                     IP 2700
               IF ( NPR .EQ. 0 ) GO TO 701                              IP 2710
          DO 525 I = 1, N                                               IP 2720
C  ----- PUNCH  22, ( C1(I,J), J = 1, MN )                              IP 2730
  525     PRINT 101, ( C1(I,J), J = 1, MN )                             IP 2740
  701     CONTINUE                                                      IP 2750
C                                                                       IP 2760
C  ..........  ART-VAR  VECTOR  ..........                              IP 2770
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
C                                                                         IP 2780
              PRINT 11 , NG                                               IP 2790
                    IF ( NG .EQ. 0 ) GO TO 544                            IP 2800
                    IF ( NG .GT. 1 ) GO TO 533                            IP 2810
                    J = 1                                                 IP 2820
              DO 536 I = 1, N                                             IP 2830
                    C2(I,J) = 0.0                                         IP 2840
   536        CONTINUE                                                    IP 2850
                    K = 0                                                 IP 2860
              DO 537 I = 1, N                                             IP 2870
                    IF ( PRDT(I) .GT. 0.0 ) GO TO 537                     IP 2880
                       C2(I,J) = -1.0                                     IP 2890
   537        CONTINUE                                                    IP 2900
              PRINT 592                                                   IP 2910
                    IF ( NPR .EQ. 0 ) GO TO 702                           IP 2920
              DO 538 I = 1, N                                             IP 2930
   538        PRINT 101,   C2(I,J)                                        IP 2940
   702        CONTINUE                                                    IP 2950
              GO TO 544                                                   IP 2960
   533        DO 531 J = 1, NG                                            IP 2970
              DO 531 I = 1, N                                             IP 2980
                    C2(I,J) = 0.0                                         IP 2990
   531        CONTINUE                                                    IP 3000
                    K = 0                                                 IP 3010
   532        DO 556 I = 1, N                                             IP 3020
                    IF = 0                                                IP 3030
              DO 557 J = 1, NG                                            IP 3040
                    IF ( IF .EQ. 1 ) GO TO 557                            IP 3050
                    IF ( PRDT(I) .GT. 0.0 ) GO TO 557                     IP 3060
                    K = K + 1                                             IP 3070
                    C2(I,K) = -1.0                                        IP 3080
                    IF = 1                                                IP 3090
   557        CONTINUE                                                    IP 3100
   556        CONTINUE                                                    IP 3110
              PRINT 592                                                   IP 3120
                    IF ( NPR .EQ. 0 ) GO TO 703                           IP 3130
              DO 535 I = 1, N                                             IP 3140
C  -----      PUNCH  22, ( C2(I,J), J = 1, NG )                           IP 3150
   535        PRINT 101, ( C2(I,J), J = 1, NG )                           IP 3160
   703        CONTINUE                                                    IP 3170
C                                                                         IP 3180
C   ..........  OBJ-FCN  VECTOR  ..........                              IP 3190
C                                                                         IP 3200
   544                     NC = N + MN + NG                               IP 3210
              PRINT 11 , NC                                               IP 3220
              DO 540 J = 1, NC                                            IP 3230
                    IF (       J  .GT.   N ) GO TO 542                    IP 3240
                    IF ( PRDT(J) .LT. 0.0 ) VALX(J) = VART                IP 3250
              GO TO 540                                                   IP 3260
   542              IF ( J .GT. N  .AND.  J .LE. M ) GO TO 543            IP 3270
                         VALX(J) = 0.0                                    IP 3280
              GO TO 540                                                   IP 3290
   543              VALX(J) = VALX(J)                                     IP 3300
   540        CONTINUE                                                    IP 3310
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         PRINT 593                                                      IP 3320
              IF ( NPR .EQ. 0 ) GO TO 704                               IP 3330
C  ----- PUNCH  22, ( VALX(J), J = 1, NC )                              IP 3340
         PRINT 101, ( VALX(J), J = 1, NC )                              IP 3350
  704    CONTINUE                                                       IP 3360
C                                                                       IP 3370
C-------------PUT THIS CONTROL CARD BEFORE //-CARD                      IP 3380
C-----------  CHANGE  CC  TO  //                                        IP 3390
CCGO.SYSPUNCH DD SYSOUT=B                                               IP 3400
C                                                                       IP 3410
C                                                                       IP 3420
C  ..........  VECTER  B  (PRDT)  ..........                            IP 3430
C                                                                       IP 3440
         DO 550 J = 1, N                                                IP 3450
              IF ( PRDT(J) .LT. 0.0 ) PRDT(J) = -PRDT(J)                IP 3460
                 PRDT(J) = PRDT(J)                                      IP 3470
  550    CONTINUE                                                       IP 3480
         PRINT 594                                                      IP 3490
              IF ( NPR .EQ. 0 ) GO TO 705                               IP 3500
C  ----- PUNCH  22, ( PRDT(J), J = 1, N  )                              IP 3510
         PRINT 101, ( PRDT(J), J = 1, N  )                              IP 3520
  705    CONTINUE                                                       IP 3530
C                                                                       IP 3540
C  ..........  FORM MATRIX  C  ..........                               IP 3550
C                                                                       IP 3560
C  ..........  1ST PART                                                 IP 3570
C                                                                       IP 3580
         DO 570 I = 1, N                                                IP 3590
         DO 570 J = 1, NC                                               IP 3600
              IF ( I .EQ. J ) GO TO 571                                 IP 3610
                 UM(I,J) = 0.0                                          IP 3620
         GO TO 570                                                      IP 3630
  571            UM(I,J) = 1.0                                          IP 3640
  570    CONTINUE                                                       IP 3650
         PRINT 595                                                      IP 3660
C                                                                       IP 3670
C  ..........  2ND  PRAT  ..........                                    IP 3680
C                                                                       IP 3690
         DO 575 I = 1, N                                                IP 3700
         DO 575 J = 1, NC                                               IP 3710
                ZC1(I,J) = 0.0                                          IP 3720
  575    CONTINUE                                                       IP 3730
         DO 576 I = 1, N                                                IP 3740
         DO 576 J = 1, MN                                               IP 3750
                      K = J + N                                         IP 3760
                ZC1(I,K) = C1(I,J)                                      IP 3770
  576    CONTINUE                                                       IP 3780
         PRINT 596                                                      IP 3790
C                                                                       IP 3800
C  ..........  3RD  PART  ..........                                    IP 3810
C                                                                       IP 3820
         DO 577 I = 1, N                                                IP 3830
         DO 577 J = 1, NC                                               IP 3840
                ZC2(I,J) = 0.0                                          IP 3850
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
    577     CONTINUE                                                      IP 3860
                 IF ( NG .EQ. 0 ) GO TO 582                               IP 3870
                 IF ( NG .GT. 1 ) GO TO 583                               IP 3880
                 J = 1                                                    IP 3890
            DO 584 I = 1, N                                               IP 3900
                         K = J + M                                        IP 3910
                   ZC2(I,K) = C2(I,J)                                     IP 3920
    584     CONTINUE                                                      IP 3930
            GO TO 582                                                     IP 3940
    583     DO 578 I = 1, N                                               IP 3950
            DO 578 J = 1, NG                                              IP 3960
                         K = J + M                                        IP 3970
                   ZC2(I,K) = C2(I,J)                                     IP 3980
    578     CONTINUE                                                      IP 3990
            PRINT 597                                                     IP 4000
    582     DO 580 I = 1, N                                               IP 4010
            DO 580 J = 1, NC                                              IP 4020
                     C(I,J) = UM(I,J) + ZC1(I,J) + ZC2(I,J)               IP 4030
    580     CONTINUE                                                      IP 4040
            PRINT 598                                                     IP 4050
            RETURN                                                        IP 4060
            END                                                           IP 4070
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         SUBROUTINE CALA2(      II, (II, JJ, A, PRDT ,C1,C2,N,M,NG,VALX) IP 6650
         COMMON MN, LIN, NNL, NPR
         DIMENSION          PRDT(45), VALX(65)
         DIMENSION  A(65, 65), DA(65, 65)
         DIMENSION C1(30,12), C2(30,25)
         DIMENSION  UM(30,50)
C-----------    SUBR  CALA2 IS TO TRANSFORM MATRIX C TO MATRIX A         IP 6720
C----           SO THAT SUBR  LP2 CAN BE USED                          IP 6730
C                                                                      IP 6740
C...................................................................... IP 6750
C               REQUIRED SUBROUTINES                                   IP 6760
C                   NONE                                               IP 6770
C...................................................................... IP 6780
    1    FORMAT (10I5)                                                 IP 6790
   13 FORMAT (4(5X, 8E15.4/))                                          IP 6800
   22    FORMAT (7E11.4 )                                              IP 6810
  580    FORMAT (//5X, ' MATRIX A ' / )                                IP 6820
  591    FORMAT ( /15X, '   COEF VECTOR ' )                            IP 6830
  592    FORMAT ( /15X, 'ART-VAR VECTOR ' )                            IP 6840
  593    FORMAT ( /15X, 'OBJ-FCN VECTOR ' )                            IP 6850
  594    FORMAT ( /15X, 'VECTOR B (PRDT) ' )                           IP 6860
              NPR = 1
              NPRT= 1
         DO 502 I = 1, N                                               IP 6890
         DO 502 J = 1, N                                               IP 6900
         IF (I .EQ. J) GO TO 501                                       IP 6910
              UM(I,J) = 0.0                                            IP 6920
         GO TO 502                                                     IP 6930
  501         UM(I,J) = 1.0                                            IP 6940
  502    CONTINUE                                                      IP 6950
         PRINT 591                                                     IP 6960
              IF ( NPRT.EQ. 0 ) GO TO 701
         DO 510 I = 1, N                                               IP 6980
         PRINT 13, ( C1(I,J), J = 1, MN )                              IP 6990
  510    CONTINUE                                                      IP 7000
  701    CONTINUE                                                      IP 7010
         PRINT 592                                                     IP 7020
         PRINT 1, NG                                                   IP 7030
         IF ( NG .LT. 2 ) GO TO 521                                    IP 7040
              IF ( NPRT.EQ. 0 ) GO TO 702
         DO 520 I = 1, N                                               IP 7060
         PRINT 13, ( C2(I,J), J = 1, NG )                              IP 7070
  520    CONTINUE                                                      IP 7080
  702    CONTINUE                                                      IP 7090
         GO TO 522                                                     IP 7100
  521    CONTINUE                                                      IP 7110
              IF ( NG .EQ. 0 )  GO TO 522                              IP 7120
              J = NG                                                   IP 7130
              IF ( NPRT.EQ. 0 ) GO TO 703
         DO 523 I = 1, N                                               IP 7150
  523    PRINT 13, C2(I,J)                                             IP 7160
  703    CONTINUE                                                      IP 7170
  522    CONTINUE                                                      IP 7180
C----------     LP2 HAS THE FORM OF  MAX Z                             IP 7190
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
C----------    -C1*X1 - C2*X2 ......... + Z = 0.0                               IP 7200
               PRINT 593                                                         IP 7210
               DO 526 J = 1, JJ                                                  IP 7220
  526              DA(1,J) = 0.0                                                 IP 7230
               DO 527 J = 1, MN                                                  IP 7240
  527              DA(1,J ) = VALX(J+N)                                          IP 7250
                   IF ( NPRT.EQ. 0 ) GO TO 704
               PRINT 13,    ( DA(1, J), J = 1, JJ )                              IP 7270
  704          CONTINUE                                                          IP 7280
               PRINT 594                                                         IP 7290
                   IF ( NPRT.EQ. 0 ) GO TO 705
               PRINT 13, ( PRDT(J), J = 1, N )                                   IP 7310
  705          CONTINUE                                                          IP 7320
               DO 540 I = 1, N                                                   IP 7330
  540              DA(I+1,JJ)= PRDT(I)                                           IP 7340
               DO 550 I = 1, N                                                   IP 7350
               DO 550 J = 1, MN                                                  IP 7360
  550              DA(I+1,J) = C1(I,J)                                           IP 7370
               DO 560 I = 1, N                                                   IP 7380
               DO 560 J = 1, N                                                   IP 7390
  560              DA(I+1,J+MN) = UM(I,J)                                        IP 7400
               IF ( NG .LT. 2 ) GO TO 621                                        IP 7410
               DO 570 I = 1, N                                                   IP 7420
               DO 570 J = 1, NG                                                  IP 7430
  570              DA(I+1,J+MN+N) = C2(I,J)                                      IP 7440
  621          CONTINUE                                                          IP 7450
                   IF ( NG .EQ. 0 )  GO TO 622                                   IP 7460
                   J = NG                                                        IP 7470
                   IF ( NPRT.EQ. 0 ) GO TO 706
               DO 623 I = 1, N                                                   IP 7490
  623          PRINT 13, C2(I,J)                                                 IP 7500
  706          CONTINUE                                                          IP 7510
  622          CONTINUE                                                          IP 7520
               PRINT 580                                                         IP 7530
               DO 590 I = 1, II                                                  IP 7540
               DO 590 J = 1, JJ                                                  IP 7550
  590              A(I,J) = DA(I,J)                                              IP 7560
               RETURN                                                            IP 7570
               END                                                               IP 7580
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
          SUBROUTINE LP2(PRDT,A,I,II,JJ,L,XT,C1,C2,N,M,NG,ZVAL,VALX)      IP 4080
         DIMENSION  A(65, 65), W(65), L(65), XT(65)
         DIMENSION           PRDT(45), VALX(65)
         DIMENSION C1(30,12), C2(30,25)
         COMMON MN, LIN, NNL, NPR
C        ************************************************************     IP 4130
C...............................................................          IP 4210
C              REQUIRED SUBROUTINES                                        IP 4220
C                 (1)  SUBROUTINES  CALA2                                  IP 4230
C...............................................................          IP 4240
C*********************      M A X     ********************                IP  210
    1    FORMAT (10I5)                                                     IP 4250
C---4    FORMAT (7F10.4)                                                   IP 4260
    4    FORMAT (11F7.3)                                                   IP 4270
    5    FORMAT (I5, E13.4 )                                               IP 4280
    6 FORMAT ( //' FEASIBLE' )                                             IP 4290
    7    FORMAT ( ' VARIABLE         VALUE ' )                             IP 4300
    8    FORMAT ( // ' OBJ. FUNCTION' , E13.4 / )                          IP 4310
   13 FORMAT (4(5X, 8E15.4/))                                              IP 4320
   14    FORMAT (1X, 9E13.4 )                                              IP 4330
  100    FORMAT (  // ' THE FINAL MATRIX ' )                               IP 4340
  101 FORMAT (  / ' ITERATION     OBJ. FUNCTION     NEW BASIC VAR. ' )     IP 4350
  103 FORMAT ( //' FEASIBLE ' )                                            IP 4360
  105    FORMAT (1X, I4, 6X, E13.4, 10X, I4)                               IP 4370
  130    FORMAT(/' UNBOUNDED ******************************** '/)          IP 4380
  150    FORMAT (//35X,' ROW ' , I2/)                                      IP 4390
  151    FORMAT ( ///5X, ' X(I) ' / )                                      IP 4400
  444    FORMAT (16F5.3)                                                   IP 4410
C-----------------------------------                                      IP 4420
 5555    FORMAT (/////10X, 'PROBLEM', I3/)                                 IP 4430
 6666    FORMAT ( '1' )                                                    IP 4440
 7777    FORMAT ( 16I5 )                                                   IP 4450
              NPRT = 0                                                     IP 4460
              II = N + 1                                                   IP 4470
              JJ = M + 1                                                   IP 4480
         PRINT 1, N, M                                                     IP 4490
         PRINT 1,II,JJ                                                     IP 4500
         III = II + 1                                                      IP 4510
         DO 10 I = 1, III                                                  IP 4520
         W (I) = 0.                                                        IP 4530
   10    L (I) = 0                                                         IP 4540
C                                                                          IP 4550
C             READ IN THE ELEMENTS OF THE MATRIX ROW BY ROW                IP 4560
C                                                                          IP 4570
         CALL      CALA2(      II, III, JJ, A, PRDT ,C1,C2,N,M,NG,VALX)    IP 4580
         DO 581 J = 1, JJ                                                  IP 4590
  581         A(III, J) = 0.0                                              IP 4600
              IF ( NPRT .EQ. 0 ) GO TO 310                                 IP 4610
         DO 582 I = 1, II                                                  IP 4620
  582    PRINT 13, ( A(I,J), J = 1,JJ )                                    IP 4630
  583    PRINT 13, ( A(III, J), J = 1, JJ )                                IP 4640
  310    CONTINUE                                                          IP 4650
C                                                                          IP 4660
C   ......... READ IN THE SUBSCRIPT FOR THE SLACK VARIABLE ON ROW I        IP 4670
```

80/80 LIST

```
CARD
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
     C  ..........  WHERE I IS NOT EQUAL TO 1 AND III                           IP 4680
     C                                                                          IP 4690
       402     DO 403 I = 2, II                                                 IP 4700
       403          L(I) = I + MN-1                                             IP 4710
               PRINT 1,( L(I), I = 2, II )                                      IP 4720
     C                                                                          IP 4730
     C              NEXT STATEMENT FOR INITIALIZATION                           IP 4740
     C                                                                          IP 4750
               KKK = 0                                                          IP 4760
     C                                                                          IP 4770
     C              THE NEXT STATEMENTS ARE TO LOOK FOR THE ROW AT WHICH THERE  IP 4780
     C             IS NO SLACK VARIABLE (NOT INCLUDING THE FIRST ROW            IP 4790
     C                                                                          IP 4800
        22     I = 1                                                            IP 4810
        23     I = I + 1                                                        IP 4820
               IF ( I - III ) 23, 40, 40                                        IP 4830
        24     IF ( L(I) ) 23, 25, 23                                           IP 4840
     C                                                                          IP 4850
     C              CALCULATE                                                   IP 4860
     C                                                                          IP 4870
     C                     NEW LAST ROW = LAST ROW WITHOUT SLACK VARUABLE       IP 4880
     C                                                                          IP 4890
        25     DO 27 J = 1, JJ                                                  IP 4900
               IF ( A (I,J ) ) 26, 27,26                                        IP 4910
        26     A(III,J)= A (III,J ) - A (I,J)                                   IP 4920
        27     CONTINUE                                                         IP 4930
               GO TO 23                                                         IP 4940
     C                                                                          IP 4950
     C              NEXT STATEMENTS FOR SEARCHING FOR THE COLUMN AT WHICH THE   IP 4960
     C          MOST NEGATIVE ENTRY APPEARS EITHER IN THE FORST (OBJECTIVE      IP 4970
     C          FUNCTION) OR LAST (FORM P) ROW                                  IP 4980
     C                                                                          IP 4990
        40     K = III                                                          IP 5000
        44     J = 0                                                            IP 5010
               W (K) = 0.                                                       IP 5020
               L ( K ) = 0                                                      IP 5030
        42     J = J + 1                                                        IP 5040
               IF ( J - JJ ) 41, 45, 45                                         IP 5050
        41     IF ( A(K,J) ) 43, 42,42                                          IP 5060
        43     IF ( W (K) - A (K,J ) ) 42,42,47                                 IP 5070
        47     W ( K ) = A (K,J)                                                IP 5080
               L (K) = J                                                        IP 5090
               GO TO 42                                                         IP 5100
     C  ..............................................................          IP 5110
     C                                                                          IP 5120
     C              TEST FOR L(K). IF L(K) IS EQUAL TO ZERO, THAT IS, ALL THE   IP 5130
     C                ENTRIES EXCEPT THE EXTREME RIGHT ONE EITHER IN THE FIRST  IP 5140
     C                OR LAST ROW ARE POSITIVE, GO TO ST. 62 FOR FURTHER EXAM.  IP 5150
     C                                                                          IP 5160
     C  ..............................................................          IP 5170
        45     IF ( L(K) ) 46,62,46                                             IP 5180
     C                                                                          IP 5190
     C              FIND OUT THE PIVOT COLUMN                                   IP 5200
     C                                                                          IP 5210
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
   46     KJ = L(K)                                                        IP 5220
C                                                                          IP 5230
C              TEST EVERY ENTRY IN THE PIVOT COLUMN TO SEE IF IT IS        IP 5240
C                POSITIVE OR NOT .  IF IT IS, GO TO ST. 121 TO COMPUTE     IP 5250
C                RATIO DEFINED IN THE LAST SECTION.                        IP 5260
          DO 120 I = 2, II                                                 IP 5270
          IF (A (I,KJ )) 120,120,121                                       IP 5280
  120     CONTINUE                                                         IP 5290
C                                                                          IP 5300
C         IF ALL THE ENTRIES IN THE PIVOT COLUMN ARE ZERO OR NEGATAVE      IP 5310
C         NUMBERS, 'UNBOUNDED' IS GOING TO BE TYPED                        IP 5320
C                                                                          IP 5330
          PRINT 130                                                        IP 5340
          GO TO 70                                                         IP 5350
C                                                                          IP 5360
C         THE FOLLOWING STATEMENTS ARE FOR COMPUTING THE RATIO DEFINDED    IP 5370
C         IN SECTION 15.4, AND FOR DETERMINING THE LOCATION OF THE PIVOT   IP 5380
C                                                                          IP 5390
  121     I = 1                                                            IP 5400
          JK = 0                                                           IP 5410
   50     I = I + 1                                                        IP 5420
          IF (I - II ) 52, 52,56                                           IP 5430
   52     IF (A(I,KJ)) 50, 50, 51                                          IP 5440
   51     X = A(I,JJ)/A(I,KJ)                                              IP 5450
          IF (JK) 55, 53, 55                                               IP 5460
   55     IF (X - XMIN) 53,50,50                                           IP 5470
   53     XMIN = X                                                         IP 5480
          JK = I                                                           IP 5490
          GO TO 50                                                         IP 5500
C                                                                          IP 5510
C         THE NEXT STATEMENT INDICATES THE PIVOT ELEMENT BEFORE NORMALIZA IP 5520
C                                                                          IP 5530
   56     X = A(JK,KJ)                                                     IP 5540
          L (JK) = KJ                                                      IP 5550
C                                                                          IP 5560
C         NEXT STATEMENTS FOR CALCULATING THE NEW ROWS ABOVE THE PIVOT RO IP 5570
C                                                                          IP 5580
          DO 57 I = 1, III                                                 IP 5590
   57     W (I) = A (I,KJ)                                                 IP 5600
          IJ = JK - 1                                                      IP 5610
          DO 59 I = 1, IJ                                                  IP 5620
          DO 59 J = 1, JJ                                                  IP 5630
          IF (A(JK,J)) 58, 59,58                                           IP 5640
   58     IF (W(I)) 580,59,580                                             IP 5650
  580     A(I,J) = A(I,J) - W(I)*(A(JK,J)/X)                               IP 5660
   59     CONTINUE                                                         IP 5670
C  .............................................................          IP 5680
C                                                                          IP 5690
C         NEXT STATEMENTS FOR CALCULATING NEW ROWS BEFORE THE PIVOT ROW    IP 5700
C                                                                          IP 5710
C  .............................................................          IP 5720
          IJ = JK + 1                                                      IP 5730
          DO 61 I = IJ, III                                                IP 5740
          DO 61 J = 1, JJ                                                  IP 5750
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
               IF ( A(JK,J) ) 60,61,60                                         IP 5760
         60    IF (W (I) ) 600,61,600                                          IP 5770
        600    A (I,J) = A (I,J) - W(I) * (A(JK,J) /X)                         IP 5780
         61    CONTINUE                                                        IP 5790
      C  ............................................................         IP 5800
      C  ............................................................         IP 5810
      C                                                                        IP 5820
      C        NEXT STATEMENTS FOR NORMALIZATION                               IP 5830
      C                                                                        IP 5840
               DO 205 J = 1, JJ                                                IP 5850
        205    A (JK,J) = A(JK,J) / X                                          IP 5860
               KKK = KKK + 1                                                   IP 5870
               PRINT 105, KKK, A(K,JJ), L(JK)                                  IP 5880
                    IF ( NPRT .EQ. 0 ) GO TO 300                               IP 5890
               DO 200 I = 1, II                                                IP 5900
        200    PRINT 13, ( A(I,J), J = 1, JJ )                                 IP 5910
        300    CONTINUE                                                        IP 5920
               GO TO 44                                                        IP 5930
      C  ............................................................         IP 5940
      C                                                                        IP 5950
      C        NEXT STATEMENT FOR TESTING TO SEE IF IT IS THE FIRST ROW ON     IP 5960
      C        WHICH ALL THE ENTRIES ARE POSITIVE EXCEPT EXTREME RIGHT ONE     IP 5970
      C        IF IT IS, THAT MEANS, NO FURTHER IMPROVEMENT ON THE SOLUTION    IP 5980
      C        CAN BE MADE, GO TO ST. 70 AND THE ANSWER WILL BE TYPED OUT      IP 5990
      C                                                                        IP 6000
      C  ............................................................         IP 6010
         62    IF (K - 1 ) 70, 70, 63                                          IP 6020
         63    IJ = JJ - 1                                                     IP 6030
      C                                                                        IP 6040
      C        TEST TO SEE WHETHER ALL THE ELEMENTS ON THE LAST ROW (NOT       IP 6050
      C        INCLUDING THE EXTREME RIGHT ONE)ARE CLSOE TO ZERO.              IP 6060
      C        IT IS DEFINED IN THE NEXT STATEMENTS THAT THE PROBLEM IS        IP 6070
      C        INFEASIBLE IF ONE (OR MORE) OF THEM IS LARGER THAN 0.0001       IP 6080
      C                                                                        IP 6090
               DO 65 J=1, IJ                                                   IP 6100
               IF (A(K,J) - .0001) 65, 65,66                                   IP 6110
         65    CONTINUE                                                        IP 6120
               PRINT 103                                                       IP 6130
               PRINT 101                                                       IP 6140
      C                                                                        IP 6150
      C        IF, AFTER ITERATION, ALL THE ELEMENTS IN THE LAST ROW HAVE      IP 6160
      C        BECOME POSITIVE BUT NEAR ZERO, DEFINI ALL THEM TO BE ZERO       IP 6170
      C                                                                        IP 6180
               DO 140 J = I, JJ                                                IP 6190
        140    A(III,J) = 0.                                                   IP 6200
      C                                                                        IP 6210
      C        IN CASE OF NONARTIFICAL PROBLEMS, DEFINE K=1, AND GO TO ST. 44  IP 6220
      C        FOR SEARCHING GOR THE PIVOT COLUMN                              IP 6230
      C                                                                        IP 6240
               K = 1                                                           IP 6250
               KKK = 0                                                         IP 6260
               GO TO 44                                                        IP 6270
      C  ............................................................         IP 6280
      C                                                                        IP 6290
```

80/80 LIST

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
C        TYPE OUT THE SOLUTION                                          IP 6300
C                                                                       IP 6310
C     ...............................................................   IP 6320
   66    PRINT 6                                                        IP 6330
   70    CONTINUE                                                       IP 6340
              ZVAL = A(1,JJ)                                            IP 6350
         PRINT 8, ZVAL                                                  IP 6360
         PRINT 7                                                        IP 6370
         DO 71 I=2,II                                                   IP 6380
   71    PRINT 5, L(I), A (I,JJ)                                        IP 6390
C                                                                       IP 6400
C                                                                       IP 6410
              IF ( NPRT .EQ. 0 ) GO TO 320                              IP 6420
         PRINT 100                                                      IP 6430
         DO 78 I = 1, III                                               IP 6440
         PRINT 150,I                                                    IP 6450
   78    PRINT 14, (A(I,J), J = 1, JJ )                                 IP 6460
  320    CONTINUE                                                       IP 6470
C     ................................................................  IP 6480
C                                                                       IP 6490
C     ..........  PICK OUT SOLUTICN VECTOR L(I) = 1,6 FOR SUBR LP2  ....... IP 6500
C                                                                       IP 6510
         PRINT 151                                                      IP 6520
              KK = 0                                                    IP 6530
         DO 760 KK = 1, MN                                              IP 6540
  760         XT(KK) = 0.0                                              IP 6550
              K = 0                                                     IP 6560
         DO 750 I = 2, II                                               IP 6570
              IF ( L(I) .GT. MN ) GO TO 750                             IP 6580
              K = L(I)                                                  IP 6590
             XT(K) = A(I,JJ)                                            IP 6600
C-----   PRINT 5, K, XT(K)
  750    CONTINUE                                                       IP 6620
         DO 751 K = 1, MN
  751    PRINT 5, K, XT(K)
         RETURN                                                         IP 6630
         END                                                            IP 6640
```

APPENDIX B

PROBLEM LISTINGS

(1) PROBLEM 1

(2) PROBLEM 3

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      SUBROUTINE FCNGR ( X,       F, FC, FX, G, GC, GX )                   IP 3650
C
C
C
C
C
C....-------------------------------------------------------------
C         PROBLEM 1
C--------PROGRAMMED BY M.S. CHAN AT OKLAHOMA STATE UNIVERSITY,
C         STILLWATER, OKLAHOMA, UNDER THE SUPERVISION OF DR. M.
C         TOCHACEK IN 1971.
C--------PROBLEM 1-----BEAMS SPANNING IN ONE DIRECTION
C....-------------------------------------------------------------
C         PROBLEM 1
C--------NOMENCLATURE
C              LIN = NO. OF LINEAR CONSTRAINTS
C              NNL = NO. OF NONLINEAR CONSTRAINTS
C               MN = NO. OF VARIABLES
C             X(1) = SPACING OF BEAMS ORIENTED IN THE X-DIRECTION
C                    ("X-BEAMS")
C             X(2) = MOMENT OF INERTIA OF THE X-BEAM WITH RESPECT TO THE
C                    HORIZONTAL PRINCIPAL AXIS
C             X(3) = WEB DEPTH-THICKNESS RATIO OF THE X-BEAM (WEB
C                    SLENDERNESS)
C             G(1) = CCNSTRAINT ON FLEXURAL STRENGTH
C             G(2) = CONSTRAINT ON DEFLECTION
C             G(3) = CCNSTRAINT ON WEB BUCKLING
C      BL(1)-(3) = LOWER BOUNDS FOR THE VARIABLES
C      UB(1)-(3) = UPPER BOUNDS FOR THE VARIABLES
C                P = UDL = UNIFORMLY DISTRIBUTED LOAD
C            SIGMA = ALLOWABLE BENDING STRESS
C                A = DIMENSION IN THE X-DIRECTION OF A GIVEN RECTANGULAR
C                    AREA SPANNED BY BEAMS
C                B = DIMENSION IN THE Y-DIRECTION OF A GIVEN RECTANGULAR
C                    AREA SPANNED BY BEAMS
C              XK1 = COEFFICIENT DEPENDING ON THE STRESS DISTRIBUTION
C                E = YOUNG'S MODULUS OF ELASTICITY
C              XMU = POISSON'S RATIO
C               PI = 3.1416
C               C1 = MOMENT COEFFICIENT FOR SIMPLE SUPPORTS AND UNIFORMLY
C                    DISTRIBUTED LOAD
C               C2 = DEFLECTION COEFFICIENT FOR UNIFORMLY DISTRIBUTEC
C                    LOAD
C               C3 = COEFFICIENT FOR DEFLECTION SPECIFIED IN BUILDING
C                    CODES
C               C4 = COEFFICIENT FOR CRITICAL STRESS
C               AX = TOTAL CROSS-SECTIONAL AREA OF X-BEAM
C                F = OBJECTIVE FUNCTION (VOLUME)
C                W = DEFLECTION
C              BMX = BENDING MOMENT PER UNIT WIDTH IN X-BEAM
C              SMX = SECTION MODULUS OF X-BEAM
C             BSTX = BENDING STRESS IN X-BEAM
C             WALL = ALLOWABLE DEFLECTION
C             FCRX = CRITICAL BUCKLING STRESS
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
C
C
C
C
C
C
C
C  ............................................................... IP 3820
C--------      SUBROUTINE FCNGR COMPUTES VALUES OF OBJ FCN & CONSTRAINTS
C              REQUIRED SUBROUTINES                                    IP 3950
C                  NONE                                                IP 3960
C#...........................................................          IP 3970
C#...........................................................          IP 3940
      COMMON MN, LIN, NNL, NPR
      DIMENSION X(12)
      DIMENSION  BL(12), UB(12)
      DIMENSION B3X(12)
      DIMENSION FX(12),         GC(25), GX(25,12),G(25),GCT(25)
      DIMENSION DXX(12), DYX(12), GSUM(12), GSUMX(12,12), XYHX(12)
      DIMENSION GXYHX(12, 12), GXYH(12)
      DIMENSION HX(12),XJX(12),YJX(12)
     DIMENSION  BMXX(12),BMYX(12),BMXYX(12),XSIGX(12),YSIGX(12)
     DIMENSION  TXYX(12),TERM1X(12),TERM2X(12),TERM3X(12)
     DIMENSION  SHEARX(12),SIG1X(12),SIG2X(12)
     DIMENSION  SIGDX(12)
      DIMENSION  SMXX(12), BSTXX(12)
  801 FORMAT ( 5X, 2I5, 7E14.4 / 15X, 7E14.4 / 15X, 7E14.4 / )         IP 3980
  802 FORMAT ( 5X, 6E14.4 / )                                          IP 3990
C                                                                      IP 4000
C----------    PROB 1-1B      A X B LOT          FEB 3 / 71
C -------      PROBLEM 1
               LIN = 6
      NNL = 3
               MN = 3
               N = MN
                   A = 720.0
                   B = 720.0
           XK1 = 23.9
                   E = 30000.0
           XMU = 0.3
                  PI = 3.1416
               P = 0.001
          C1 = 1.0 / 8.0
               C2 = 5.0 / 384.0
     C3=1./360.
          SIGMA = 22.0
     C4=(XK1)*(((PI**2)*E)/(12.*(1.-XMU**2)))
     B6=3.46*A
              B7 =  C1 * P *   A  ** 2.0 / SIGMA
              B8 = ( C2 *P*   A  ** 3.0 ) / E
           BL(1) = 24.0
           BL(2) = 11.3                                                IP 6820
             BL(3) = 100.0
           UB(1) = 120.0
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777777 8
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
              UB(2) = 349000.0
                UB(3) = 350.0
C---------      DELETE IF-STATEMENTS WHEN USED IN RGM
C---------      START OF IF- STAT
                IF ( NPR .EQ. 0 )  GO TO 825
          PRINT 802, (X(I), I = 1, MN )                                  IP 4200
  825     CONTINUE
C                                                                        IP 4290
C  ..........  TO AVOID NEGATIVITY SET                                   IP 4300
C  ..........  LOWER BOUND                                              IP 4310
C                                                                        IP 4320
C                                                                        IP 4330
C                                                                        IP 4350
                NSKBL = MN + 1
                NSKUB = MN + 1
          DO 820 K = 1, MN
                IF ( K .EQ. NSKBL )  GO TO 821
                IF ( X(K) .LT. BL(K) )  X(K) = BL(K)
  821     CONTINUE
                IF ( K .EQ. NSKUB )  GO TO 820
                IF ( X(K) .GT. UB(K) )  X(K) = UB(K)
  820     CONTINUE                                                       IP 4550
                IF ( NPR .EQ. 0 )  GO TO 826
          PRINT 802, (X(I), I = 1, MN )                                  IP 4560
  826     CONTINUE
C---------      END   OF IF- STAT
C
                AX = 3.46 *( X(2) ** 0.5 ) * ( X(3) ** (-0.5) )
                SMX = 1.51 * ( X(2) ** 0.75 ) * ( X(3) ** (-0.25) )
          DO 204 K = 1,MN
                SMXX(K) = 0.0
                SMXX(2) = 0.75*1.51*(X(2)**(-0.25))*(X(3)**(-0.25))
                SMXX(3) = -0.25*1.51*(X(2)**0.75)*(X(3)**(-1.25))
  204     CONTINUE
C  ----------OBJ  FCN  ----------                                       PROB  43
C
                LF = 0
            F = +B6 * (  B * X(1) * * ( -1 ) - 1.0 ) * ( X (2) * * 0.5 )
     1          * ( X (3) * * ( -0.5 ))
          DO 200 K = 1, MN
  200           FX(K) = 0.0
              FX(1) = -B6 * ( X (2) * * 0.5 ) * ( X (3) * * ( - 0.5 ))
     1               *  B * X (1) * * ( -2 )
              FX(2) = +0.5 * B6 * ( B  * X(1) * * ( -1 ) - 1.0 ) * ( X(2)
     1               * * ( -0.5 )) * ( X(3) * * ( -0.5 ))
              FX(3) = -0.5 * B6 * ( B  * X(1) * * ( -1 ) - 1.0 ) *
     1               ( X(2) * * 0.5 )* ( X(3) * * ( -1.5 ))
C
C  ----------  CONSTRAINTS  ---------                                   PROB0047
C
          DO 201 J = 1, NNL
          DO 201 K = 1, MN
  201           GX(J,K) = 0.0
                     G(1) = 1.51 * ( X(2) ** 0.75 ) * ( X(3) ** (-0.25) )
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
     1                 - B7 * X(1)
             GX(1,1) = - B7
             GX(1,2) = 0.75 * 1.51 * ( X(2) ** (-0.25) ) *
     1                 ( X(3) ** (-0.25) )
             GX(1,3) = -0.25 * 1.51 * ( X(2) ** 0.75) * ( X(3) **
     1                 (-1.25) )
           G(2) = C3 - B8 * X (1) * X(2) ** (-1.0 )
            WALL = C3 * A
            W = WALL - G(2)
            GX(2,1) = -B8 * X(2) **(-1.0)
            GX(2,2) = B8 * X(1) * X(2) ** (-2.0)
            GX(2,3) = 0.0
            BSTX = B7*X(1)*SIGMA/SMX
            G(3) = C4 * ( X(3) ** (-2.0))- BSTX
            SIGCR = G(3) + BSTX
      DO 202 K = 1,MN
            BSTXX(K) = 0.0
      IF ( SMXX(K) .EQ. 0.0) GO TO 203
            BSTXX(K) = SIGMA * B7 * X(1) * (-1.0) * ( SMX ** (-2.0)) *
     1                SMXX(K)
  203 CONTINUE
            BSTXX(1) = BSTXX(K)
     1                     + B7 * SIGMA/ SMX
            GX(3,K) = -BSTXX(K)
            GX(3,3) = -BSTXX(K) -2.0 * C4*( X(3) ** (-3.0))
  202 CONTINUE
            IF ( NPR .EQ. 0 )  GO TO 690                                IP 6220
      PRINT 802,F
      PRINT 802,  (FX(K), K=1, MN )
      PRINT 802, X(2), X(3), X(4), AX, SMX
      PRINT 802, SIGMA, BSTX
      PRINT 802, WALL, W
      PRINT 802, SIGCR, BSTX
      DO 688 J = 1, NNL                                                 IP 6290
  688 PRINT 802, G(J)                                                   IP 6300
      DO 689 J = 1, NNL                                                 IP 6310
  689 PRINT 802, ( GX(J,K), K = 1, MN )                                 IP 6320
      PRINT 802, FC                                                     IP 6630
      PRINT 802, F
  690 CONTINUE                                                          IP 6330
            IF ( NNL .NE. 0 )  GO TO 890                                IP 6340
            G(1)     = 0.0                                              IP 6350
            GC(1)    = 0.0                                              IP 6360
            GX(1,1) = 0.0                                               IP 6370
            GX(1,2) = 0.0                                               IP 6380
      GO TO 880                                                         IP 6390
  890 CONTINUE                                                          IP 6400
            IF ( NNL .NE. 1 ) GO TO 870                                 IP 6410
            J = NNL                                                     IP 6420
            GCT(J) = 0.0                                                IP 6430
      DO 670 K = 1, MN                                                  IP 6440
  670       GCT(J) = GCT(J) + GX(J,K) * X(K)                            IP 6450
            GC(J) = G(J) - GCT(J)                                       IP 6460
      GO TO 880                                                         IP 6470
```

80/80 LIST

```
          00000000011111111112222222222333333333344444444445555555555666666666677777777778
          12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
           870    CONTINUE                                                        IP 6480
                  DO 772 J = 1, NNL                                               IP 6490
           772        GCT(J) = 0.0                                                IP 6500
                  DO 770 J = 1, NNL                                               IP 6510
                  DO 770 K = 1, MN                                                IP 6520
           770        GCT(J) = GCT(J) + GX(J,K) * X(K)                            IP 6530
                  DO 771 J = 1, NNL                                               IP 6540
           771        GC(J) = G(J) - GCT(J)                                       IP 6550
           880    CONTINUE                                                        IP 6560
                      SUM = 0.0                                                   IP 6570
                  DO 700 K = 1, MN                                                IP 6580
                      SUM = SUM + FX(K) * X(K)                                    IP 6590
           700    CONTINUE                                                        IP 6600
                      IF ( LF .EQ. 1 )  GO TO 760                                 IP 6610
                      FC = F - SUM                                                IP 6620
                  GO TO 761                                                       IP 6640
           760        FC = 0.0                                                    IP 6650
           761    CONTINUE                                                        IP 6660
                      NPR = 0                                                     IP 6210
                  RETURN                                                          IP 6670
                  END                                                             IP 6680
```

80/80 LIST

```
          00000000011111111112222222222333333333344444444445555555555666666666677777777778
          12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                SUBROUTINE LFCN ( GLX, GLC, BL, UB )
                COMMON MN, LIN, NNL, NPR
                DIMENSION  GLX(25,12), GLC(25)
                DIMENSION  BL(12), UB(12)
          C.................................................................      IP 6730
          C             REQUIRED SUBROUTINES                                      IP 6740
          C                 NCNE                                                  IP 6750
          C.................................................................      IP 6760
          C     PROBLEM 1
          C.................................................................      IP 6760
          C                                                                       IP 6770
          C                                                                       IP 6800
                   BL(1) = 24.0
                   BL(2) = 11.3
                     BL(3) = 100.0
                   UB(1) = 120.0
                   UB(2) = 349000.0
                     UB(3) = 350.0
                     NBL = 3
                     NUB = 3
                DO 401 K = 1, NBL                                                 IP 6950
            401      GLC(K) = -BL(K)                                              IP 6960
                DO 402 K = 1, NUB                                                 IP 6970
            402      GLC(NBL + K) = UB(K)                                         IP 6980
                DO 200 J = 1, LIN                                                 IP 6990
                DO 200 K = 1, MN                                                  IP 7000
            200      GLX(J,K) = 0.0                                               IP 7010
                DO 403 K = 1, NBL                                                 IP 7020
            403      GLX(K, K) = +1.0                                             IP 7030
                DO 404 K = 1, NUB                                                 IP 7040
            404      GLX(NBL + K, K) = -1.0                                       IP 7050
                RETURN                                                            IP 7060
                END                                                               IP 7070
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      SUBROUTINE FCNGR ( X,      F, FC, FX, G, GC, GX )                 FGR 280
C
C
C
C
C
C....-----------------------------------------------------------------
C        PROBLEM 3
C--------PROGRAMMED BY M.S. CHAN AT OKLAHOMA STATE UNIVERSITY,
C        STILLWATER, OKLAHOMA, UNDER THE SUPERVISION OF DR. M.
C        TOCHACEK IN 1971.
C        PROBLEM 3-----A GRID WITH BEAMS SPANNING IN TWO ORTHOGONAL
C                      DIRECTIONS-----RIGID-CONNECTION.
C....-----------------------------------------------------------------
C--------NOMENCLATURE
C              LIN = NO. OF LINEAR CONSTRAINTS
C              NNL = NO. OF NONLINEAR CONSTRAINTS
C               MN = NO. OF VARIABLES
C             X(1) = SPACING OF BEAMS ORIENTED IN THE Y-DIRECTION
C                    ("Y-BEAMS")
C             X(2) = MOMENT OF INERTIA OF THE X-BEAM WITH RESPECT TO THE
C                    HORIZONTAL PRINCIPAL AXIS
C             X(3) = WEB DEPTH-THICKNESS RATIO OF THE X-BEAM (WEB
C                    SLENDERNESS)
C             X(4) = SPACING OF BEAMS ORIENTED IN THE X-DIRECTION
C                    ("X-BEAMS")
C             X(5) = MOMENT OF INERTIA OF THE Y-BEAM WITH RESPECT TO THE
C                    HORIZONTAL PRINCIPAL AXIS
C             X(6) = WEB DEPTH-THICKNESS RATIO OF THE Y-BEAM (WEB
C                    SLENDERNESS)
C             G(1) = CCNSTRAINT ON FLEXURAL STRENGTH IN X-DIRECTION
C             G(2) = CONSTRAINT ON FLEXURAL STRENGTH IN Y-DIRECTION
C             G(3) = CONSTRAINT ON DEFLECTION IN X-DIRECTION
C             G(4) = CCNSTRAINT ON DEFLECTION IN Y-DIRECTION
C             G(5) = CONSTRAINT ON WEB BUCKLING IN X-DIRECTION
C             G(6) = CCNSTRAINT ON WEB BUCKLING IN Y-DIRECTION
C             G(7) = CONSTRAINT ON FLEXURAL STRENGTH AT THE JOINT
C        BL(1)-(6) = LOWER BOUNDS FOR THE VARIABLES
C        UB(1)-(6) = UPPER BOUNDS FOR THE VARIABLES
C                P = UDL = UNIFORMLY DISTRIBUTED LOAD
C            SIGMA = ALLOWABLE BENDING STRESS
C                A = DIMENSION IN THE X-DIRECTION OF A GIVEN RECTANGULAR
C                    AREA SPANNED BY BEAMS
C                B = DIMENSION IN THE Y-DIRECTION OF A GIVEN RECTANGULAR
C                    AREA SPANNED BY BEAMS
C              XK1 = COEFFICIENT DEPENDING ON THE STRESS DISTRIBUTION
C                E = YOUNG'S MODULUS OF ELASTICITY
C              XMU = POISSON'S RATIO
C               PI = 3.1416
C               C1 = MOMENT COEFFICIENT FOR SIMPLE SUPPORTS AND UNIFORMLY
C                    DISTRIBUTED LOAD
C               C2 = DEFLECTION COEFFICIENT FOR UNIFCRMLY DISTRIBUTEC
C                    LOAD
C               C3 = COEFFICIENT FOR DEFLECTION SPECIFIED IN BUILDING
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
C               CODES
C            C4 = COEFFICIENT FOR CRITICAL STRESS
C            B1 = RATIO OF FLANGE THICKNESS TO WEB THICKNESS
C            B2 = RATIO OF FLANGE AREA TO WEB AREA
C            XG = SHEAR MODULUS
C            DX = UNIT FLEXURAL RIGIDITY OF X-BEAM
C            DY = UNIT FLEXURAL RIGIDITY OF Y-BEAM
C            AX = TOTAL CROSS-SECTIONAL AREA OF X-BEAM
C            AY = TOTAL CROSS-SECTIONAL AREA OF Y-BEAM
C           AWX = WEB CROSS-SECTIONAL AREA OF X-BEAM
C           AWY = WEB CROSS-SECTIONAL AREA OF Y-BEAM
C          DELX = WEB THICKNESS OF X-BEAM
C          DELY = WEB THICKNESS OF Y-BEAM
C             H = WEB DEPTH
C             F = OBJECTIVE FUNCTION (VOLUME)
C            AA = PARAMETERS ASSOCIATED WITH THE ASSUMED SOLUTION
C             W = DEFLECTION
C           WXX = 2ND PARTIAL DERIVATIVE OF DEFLECTION WITH RESPECT
C                 TO X
C           BMX = BENDING MOMENT PER UNIT WIDTH IN X-BEAM
C           WYY = 2ND PARTIAL DERIVATIVE OF DEFLECTION WITH RESPECT
C                 TO Y
C           BMY = BENDING MOMENT PER UNIT WIDTH IN Y-BEAM
C           WXY = 2ND MIXED PARTIAL DERIVATIVE OF DEFLECTION WITH
C                 RESPECT TO X AND Y
C          BMXY = TORSIONAL MOMENT PER UNIT WIDTH IN X-BEAM
C           SMX = SECTION MODULUS OF X-BEAM
C           SMY = SECTION MODULUS OF Y-BEAM
C          BSTX = BENDING STRESS IN X-BEAM
C          BSTY = BENDING STRESS IN Y-BEAM
C          XSIG = BENDING STRESS IN X-BEAM
C          YSIG = BENDING STRESS IN Y-BEAM
C           TXY = SHEARING STRESS
C           WXA = ALLOWABLE DEFLECTION OF X-BEAM
C           WYA = ALLOWABLE DEFLECTION OF Y-BEAM
C          FCRX = CRITICAL BUCKLING STRESS IN X-BEAM
C          FCRY = CRITICAL BUCKLING STRESS IN Y-BEAM
C          SIGD = COMBINED STRESS AT A JOINT
C.............................................................
C
C
C
C
C
C..........................................................
C--------      SUBROUTINE FCNGR COMPUTES VALUES OF OBJ FCN & CONSTRAINTS
C              REQUIRED SUBROUTINES                                     FGR 560
C                  NONE                                                 FGR 570
C  ............................................................. FGR 540
C.............................................................     FGR 550
C---------     PROBLEM 3
C.............................................................     FGR 580
         COMMON MN, LIN, NNL, NPR                                       FGR 290
         DIMENSION X(12)                                                FGR 300
```

80/80 LIST

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
              DIMENSION  BL(12), UB(12)                                         FGR 310
              DIMENSION FX(12),        GC(25), GX(25,12),G(25),GCT(25)          FGR 320
              DIMENSION DXX( 8), DYX( 8)                                        FGR 330
              DIMENSION HX( 8),XJX( 8),YJX( 8)                                  FGR 340
         DIMENSION  BMXX( 8),BMYX( 8),BMXYX( 8),XSIGX( 8),YSIGX( 8)             FGR 350
         DIMENSION  TXYX( 8),TERM1X( 8),TERM2X( 8),TERM3X( 8)                   FGR 360
         DIMENSION  SHEARX( 8),SIG1X( 8),SIG2X( 8)                              FGR 370
         DIMENSION  SIGDX( 8)                                                   FGR 380
              DIMENSION                          FCRXX( 8), FAXX( 8)            FGR 390
              DIMENSION                          FCRYX( 8), FAYX( 8)            FGR 400
              DIMENSION  WX(8), WSUMX(8), WXXX(8)                               FGR 410
              DIMENSION          WSUMY(8), WYYK(8), WXYX(8)                     FGR 420
              DIMENSION AA(5,5) , D1(5,5) , D2(5,5) , D3(5,5) , D4(5,5) ,       FGR 430
        1            WW(5,5) , WWXX(5,5) , WWYY(5,5) , WWXY(5,5) , DAA(5,5)     FGR 440
              DIMENSION AAX(5,5,8) , WWX(5,5,8) , DAAX(5,5,8)                   FGR 450
              DIMENSION WWXXX(5,5,8) , WWYYK(5,5,8) , WWXYX(5,5,8)              FGR 460
              DIMENSION SMXX( 8), SMYX( 8)                                      FGR 470
         DIMENSION BSTXX( 8),BSTYX( 8),BSTXYX( 8)                               FGR 480
              DIMENSION WX3K(8), WYYXK(8), QXK(8)                               FGR 490
              DIMENSION WWX3(5,5), WWYYX(5,5)                                   FGR 500
              DIMENSION WWX3K(5,5,8), WWYYXK(5,5,8)                             FGR 510
              DIMENSION  DELXK(8), DELYK(8)
    801       FORMAT ( 5X, 2I5, 7E14.4 / 15X, 7E14.4 / 15X, 7E14.4 / )          FGR 590
    802       FORMAT ( 5X, 9E14.4 / )                                           FGR 600
C--------------     PROBLEM 302          WEB BUCKLING          APRIL 19 / 71 FGR 610
                      LIN = 12
                      NNL = 7                                                   FGR 630
                      MN = 6
                      N = MN                                                    FGR 650
              NTEMP = N                                                         FGR 660
                      SFB = 1.0                                                 FGR 670
                              P = 0.001                                         FGR 680
                      YIELD = 36.0                                              FGR 690
                           SIGMA = 22.0                                         FGR 700
                      SIGSH = 14.3                                              FGR 710
                              A = 720.0                                         FGR 720
                              B = 720.0                                         FGR 730
                      XK1 = 23.9
                      K2 = 0                                                    FGR 750
                              E = 30000.0                                       FGR 760
                            XMU = 0.3                                           FGR 770
                             PI = 3.1416                                        FGR 780
                             C1 =(16.0 * P / ( PI ** 6.0 )) * E / 24.0          FGR 790
                      C2 = 360.0                                                FGR 800
                             C4 = XK1* ( PI ** 2.0 ) * E /                      FGR 810
        1                         ( 12.0 * (1.0 - XMU ** 2.0) )                 FGR 820
                             C5 =(P * A / 2.0 ) * SFB                           FGR 830
                             C6 =(P * B / 2.0 ) * SFB                           FGR 840
                             C7 = 16.0 * P / ( PI ** 6.0 )                      FGR 850
                      B1 = 1.59                                                 FGR 860
                             B2 = 0.54
                             B3 = (2.0 * B2 * (B1 ** 2.0) + 1.0) / 3.0          FGR 880
                             XG = 12000.0                                       FGR 890
                             B4= (16.0 * P / ( PI ** 6.0 )) * XG/ 24.0          FGR 900
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                 B5 = K2 * C4 /XK1
     C----------   LOWER BOUNDS                                                FGR 920
                BL(1) = 24.0                                                   FGR 930
                BL(2) = 11.3                                                   FGR 940
                BL(3) = 100.0                                                  FGR 950
                BL(4) = BL(1)                                                  FGR 960
                BL(5) = BL(2)                                                  FGR 970
                BL(6) = BL(3)                                                  FGR 980
     C----------   UPPER BOUNDS                                                FGR1010
                UB(1) = 120.0                                                  FGR1020
                UB(2) = 349000.0                                               FGR1030
                  UB(3) = 350.0                                                FGR1040
                UB(4) = UB(1)                                                  FGR1050
                UB(5) = UB(2)                                                  FGR1060
                UB(6) = UB(3)                                                  FGR1070
     C---------    DELETE IF-STATEMENTS WHEN USED IN RGM                       FGR1100
     C---------    START OF IF- STAT                                           FGR1110
                   IF ( NPR .EQ. 0 )  GO TO 825                                FGR1120
              PRINT 802, (X(I), I = 1, MN )                                    FGR1130
       825    CONTINUE                                                         FGR1140
     C                                                                         FGR1150
     C  ..........  TO AVOID NEGATIVITY SET                                     FGR1160
     C  ..........  LOWER BOUND                                                FGR1170
     C                                                                         FGR1180
     C                                                                         FGR1190
     C                                                                         FGR1200
                   NSKBL = MN + 1                                              FGR1210
                   NSKUB = MN + 1                                              FGR1220
              DO 820 K = 1, MN                                                 FGR1230
                   IF ( K .EQ. NSKBL )  GO TO 821                              FGR1240
                   IF ( X(K) .LT. BL(K) )  X(K) = BL(K)                        FGR1250
       821    CONTINUE                                                         FGR1260
                   IF ( K .EQ. NSKUB )  GO TO 820                              FGR1270
                   IF ( X(K) .GT. UB(K) )  X(K) = UB(K)                        FGR1280
       820    CONTINUE                                                         FGR1290
                   IF ( NPR .EQ. 0 )  GO TO 826                                FGR1300
              PRINT 802, (X(I), I = 1, MN )                                    FGR1310
       826    CONTINUE                                                         FGR1320
     C---------    END   OF IF- STAT                                           FGR1330
                          DX = E * X(2) / X(4)                                 FGR1340
                          DY = E * X(5) / X(1)                                 FGR1350
              DO 672 K = 1, MN                                                 FGR1360
                   DELXK(K) = 0.0
                   DELYK(K) = 0.0
                   DXX(K) = 0.0                                                FGR1370
       672         DYX(K) = 0.0                                                FGR1380
                      DXX(2) = E / X(4)                                        FGR1390
                      DXX(4) = -E * X(2) * X(4) ** (-2.0)                      FGR1400
                      DYX(1) =  - E * X(5) * X(1) ** (-2.0)                    FGR1410
                      DYX(5) = E / X(1)                                        FGR1420
                   AX = 3.46 *( X(2) ** 0.5 ) * ( X(3) ** (-0.5) )             FGR4440
                   AY = 3.46 *( X(5) ** 0.5 ) * ( X(6) ** (-0.5) )             FGR4450
                   AWX = AX / ( 2.0 * B2 + 1.0 )
                   AWY = AY / ( 2.0 * B2 + 1.0 )
```

80/80 LIST

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                     DELX = ( AWX / X(3) ) ** 0.5
                     DELY = ( AWY / X(6) ) ** 0.5
     C---------      ALTERNATELY
     C---------      DELX = 1.29 * ( X(2) ** 0.25 ) * ( X(3) ** 0.75 )
     C---------      DELY = 1.29 * ( X(5) ** 0.25 ) * ( X(6) ** 0.75 )
                     DELXK(2) = 0.25 * 1.29 * ( X(2) ** (-0.75) ) *
          1                    ( X(3) ** 0.75 )
                     DELXK(3) = 0.75 * 1.29 * ( X(2) ** 0.25 ) *
          1                    ( X(3) ** (-0.25) )
                     DELYK(5) = 0.25 * 1.29 * ( X(5) ** (-0.75) ) *
          1                    ( X(6) ** 0.75 )
                     DELYK(6) = 0.75 * 1.29 * ( X(5) ** 0.25 ) *
          1                    ( X(6) ** (-0.25) )
                        XJ = B3 * X(3) * DELX **4.0
                        YJ = B3 * X(6) * DELY **4.0
                         H = 0.5 * XG* (XJ / X(4) + YJ / X(1))                  FGR1450
                     XJG = XJ * XG                                              FGR1460
                     YJG = YJ * XG                                              FGR1470
                     XEI = X(2) * E                                             FGR1480
                     YEI = X(5) * E                                             FGR1490
                     DO 501 K = 1,MN                                            FGR1500
                          XJX(K) = 0.0                                          FGR1510
                          YJX(K) = 0.0                                          FGR1520
                           HX(K) = 0.0                                          FGR1530
       501 CONTINUE                                                             FGR1540
                DO 1501 K = 1, MN
                     XJX(K) = 4.0 * B3 * X(3) * ( DELX ** 3.0 ) * DELXK(K)
                     YJX(K) = 4.0 * B3 * X(6) * ( DELY ** 3.0 ) * DELYK(K)
      1501      CONTINUE
                        XJX(3) = B3 * DELX ** 4.0 +
          1                      B3 * X(3) * 4.0 * ( DELX ** 3.0 ) * DELXK(3)
                        YJX(6) = B3 * DELY ** 4.0 +
          1                0     B3 * X(6) * 4.0 * ( DELY ** 3.0 ) * DELYK(6)
                DO 502 K = 1,MN                                                 FGR1590
                HX(K) = 0.5 * XG* (XJX(K) / X(4) + YJX(K)/X(1))                 FGR1620
       502      CONTINUE                                                        FGR1630
                     HX(1) = 0.5 * XG* (XJX(1) / X(4) + YJX(1) / X(1)-YJ        FGR1640
          1          * X(1) ** (-2.0))                                          FGR1650
                     HX(4) = 0.5 * XG* (XJX(4) / X(4) -XJ * X(4) ** (-2.0)      FGR1660
          1               + YJX(4) / X(1))                                      FGR1670
     C                                                                          FGR1680
     C  ..........  OBJ  FCN  ..........                                        FGR1690
     C---------      MIN F                                                      FGR1700
     C                                                                          FGR1710
                     LF = 0                                                     FGR1720
                DO 671 K = 1, MN                                                FGR1730
       671           FX(K) = 0.0                                                FGR1740
                              F = 3.46 * A * ( B / X(4) - 1.0 ) * ( X(2) ** 0.5) FGR1750
          1                     * ( X(3) ** (-0.5) ) + 3.46 * B * ( A / X(1) - FGR1760
          2                     1.0 ) * ( X(5) ** 0.5 ) * ( X(6) ** (-0.5) )    FGR1770
                         FX(1) = - A * ( X(1) ** (-2.0) ) * 3.46 * B *          FGR1780
          1                    ( X(5) ** 0.5 ) * ( X (6) ** (-0.5) )            FGR1790
                         FX(2) = 0.5 * ( X(2) ** (-0.5) ) * 3.46 * A *          FGR1800
          1                     ( B / X(4) -1.0 ) * ( X(3) ** (-0.5) )          FGR1810
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
                  FX(3) = -0.5 * ( X(3) ** (-1.5) ) * 3.46 * A *        FGR1820
     1                    ( B / X(4) - 1.0) * ( X(2) ** 0.5 )           FGR1830
                  FX(4) = - B * ( X(4) ** (-2.0) ) * 3.46 * A * ( X(2)** FGR1840
     1                    0.5 ) * ( X(3) ** (-0.5) )                    FGR1850
                  FX(5) = 0.5 * ( X(5) ** (-0.5) ) * 3.46 * B *         FGR1860
     1                    ( A / X(1) - 1.0 ) * ( X(6) ** (-0.5) )       FGR1870
                  FX(6) = -0.5 * ( X(6) ** (-1.5) ) * 3.46 * B *        FGR1880
     1                    ( A / X(1) - 1.0 ) * ( X(5) ** 0.5 )          FGR1890
C                                                                       FGR1900
C  ..........  CONSTRAINTS  ..........                                  FGR1910
C                                                                       FGR1920
              NT = 5                                                    FGR1930
              MT = 5                                                    FGR1940
              MI = 2                                                    FGR1950
              NI = 2                                                    FGR1960
              DO 810 M = 1 , MT , MI                                    FGR1970
              DO 810 N = 1 , NT , NI                                    FGR1980
         DO 810 K = 1, MN                                               FGR1990
                   D1(M,N) = 0.0                                        FGR2000
                   D2(M,N) = 0.0                                        FGR2010
                   D3(M,N) = 0.0                                        FGR2020
                   D4(M,N) = 0.0                                        FGR2030
                 DAA(M,N) = 0.0                                         FGR2040
                   AA(M,N) = 0.0                                        FGR2050
                   WW(M,N) = 0.0                                        FGR2060
                WWXX(M,N) = 0.0                                         FGR2070
                WWYY(M,N) = 0.0                                         FGR2080
                WWXY(M,N) = 0.0                                         FGR2090
               DAAX(M,N,K) = 0.0                                        FGR2100
                 AAX(M,N,K) = 0.0                                       FGR2110
                 WWX(M,N,K) = 0.0                                       FGR2120
              WWXXX(M,N,K) = 0.0                                        FGR2130
              WWYYK0M,N,K) = 0.0                                        FGR2140
              WWXYX ( M,N,K ) = 0.0                                     FGR2150
              WWX3K(M,N,K) = 0.0                                        FGR2160
              D1 ( M,N ) = ( M / A ) ** 4.0                             FGR2170
              D2 ( M,N ) = 2.0 * (( M * N ) / ( A * B )) ** 2.0         FGR2180
              D3 ( M,N ) = ( N / B ) ** 4.0                             FGR2190
              D4 ( M,N ) = 16.0 / ( M * N * PI ** 6.0 )                 FGR2200
  810    CONTINUE                                                       FGR2210
C--------------------------------    SIMPLE-SUPP          MAR 27, 71    FGR2220
C---------------------- ORIGIN AT NW CORNER (  0.0 , 0.0  )             FGR2230
C----------------------         AT    CENTER ( A/2.0,B/2.0 )            FGR2240
C----------------------         ON X-BOUNDARY( A/2.0, 0.0  )            FGR2250
C----------------------         ON Y-BOUNDARY(  0.0 ,B/2.0 )            FGR2260
              XXT = A / 2.0                                             FGR2270
              YYT = B / 2.0                                             FGR2280
              XX = XXT                                                  FGR2290
              YY = YYT                                                  FGR2300
                    W = 0.0                                             FGR2310
              DO 811 M = 1 , MT , MI                                    FGR2320
              DO 811 N = 1 , NT , NI                                    FGR2330
              DAA(M,N) = D1(M,N) * DX + D2(M,N) * H + D3(M,N) * DY      FGR2340
                    AA(M,N) = D4(M,N) * P/DAA(M,N)                      FGR2350
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
               IF ( XX .EQ. 0.0 )  GO TO 531
               CX = XX * XX
          GO TO 533
   531         CX = 0.0
   533         IF ( YY .EQ. 0.0 )  GO TO 532
               CY = YY * YY
          GO TO 534
   532         CY = 0.0
   534    CONTINUE
          CWW = SIN ( M * PI * XX / A ) * SIN ( N * PI * YY / B )
                    WW(M,N) = AA(M,N) * CWW
                    W = W + WW(M,N)
C--------           PARTIALS
          DO 859 K = 1,MN
   859      WX(K) = 0.0
          DO 851  K = 1,MN
               DAAX(M,N,K) = D1(M,N) * DXX(K) + D2(M,N) * HX(K) +
     1              D3(M,N) * DYX(K)
               AAX(M,N,K) = -D4(M,N) * P * (DAA(M,N) **(-2.0)) *
     1                        DAAX(M,N,K)
               WWX(M,N,K) = AAX(M,N,K) * CWW
               WX(K) = WX(K) + WWX(M,N,K)
   851 CONTINUE
   811 CONTINUE
C----------------- BENDING MOMENT IN X, BMX
               XX = XXT
                  WSUM = 0.0
                   WXX = 0.0
               WX3 = 0.0
          DO 857 K = 1, MN
                 WSUMX(K) = 0.0
                  WXXX(K) = 0.0
               WX3K(K) = 0.0
   857    CONTINUE
          DO 813 I = 1,3
               XX = XXT
               YY = YYT
          IF(I-2) 814,815,816
   814         YY = -X(4) + YY
          GO TO 817
   815         YY = YY
          GO TO 817
   816         YY = X(4) + YY
   817    DO 812 M = 1 , MT , MI
          DO 812 N = 1 , NT , NI
               IF ( XX .EQ. 0.0 )  GO TO 541
               CX = XX * XX
          GO TO 543
   541         CX = 0.0
   543         IF ( YY .EQ. 0.0 )  GO TO 542
               CY = YY * YY
          GO TO 544
   542         CY = 0.0
   544    CONTINUE
```

80/80 LIST

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                  CWWXX = - (( M * PI / A ) ** 2.0 ) * SIN ( M * PI * XX / A ) *  FGR2900
             1          SIN ( N * PI * YY / B )                                  FGR2910
                              WWXX(M,N) = AA(M,N) * CWWXX                        FGR2920
                        CWWX3 = - ( ( M * PI / A ) ** 3.0 ) * COS ( M * PI*XX / A) FGR2930
             1                      * SIN ( N * PI * YY / B )                    FGR2940
                        WWX3(M,N) = AA(M,N) * CWWX3                              FGR2950
                  DO 852 K = 1, MN                                               FGR2960
                              WWXXX(M,N,K) = AAX(M,N,K) * CWWXX                  FGR2970
                        WWX3K(M,N,K) = AAX(M,N,K) * CWWX3                        FGR2980
           852    CONTINUE                                                       FGR2990
                  IF(I-2) 818,819,818                                            FGR3000
           818    WSUM = WSUM + WWXX(M,N)                                        FGR3010
                  DO 853 K = 1, MN                                               FGR3020
           853          WSUMX(K) = WSUMX(K) + WWXXX(M,N,K)                       FGR3030
                  GO TO 1812                                                     FGR3040
           819    WSUM = WSUM + 22.0 * WWXX(M,N)                                 FGR3050
                        WX3 = WX3 + WWX3(M,N)                                    FGR3060
                  DO 854 K = 1, MN                                               FGR3070
                        WX3K(K) = WX3K(K) + WWX3K(M,N,K)                         FGR3080
           854          WSUMX(K) = WSUMX(K) + WWXXX(M,N,K) * 22.0                FGR3090
          1812    CONTINUE                                                       FGR3100
           812 CONTINUE                                                          FGR3110
                  WXX = WXX + WSUM                                               FGR3120
           813 CONTINUE                                                          FGR3130
                BMX = -(E * X(2) / 24.0) * WXX                                   FGR3140
                DO 856 K = 1, MN                                                 FGR3150
           856 BMXX(K) = -(E * X(2)/24.0) * WXXX(K)                              FGR3160
               BMXX(2) = -(E * X(2)/24.0) * WXXX(2)                              FGR3170
             1              - ( E / 24.0 ) * WXX                                 FGR3180
         C-----GO TO 9000                                                        FGR3190
         C-------------         BENDING MOMENT IN Y, BMY                         FGR3200
                        YY = YYT                                                 FGR3210
                           WSUM = 0.0                                            FGR3220
                            WYY = 0.0                                            FGR3230
                        WYYX = 0.0                                               FGR3240
                  DO 867 K = 1, MN                                               FGR3250
                           WYYK(K) = 0.0                                         FGR3260
                          WSUMY(K) = 0.0                                         FGR3270
                        WYYXK(K) = 0.0                                           FGR3280
           867    CONTINUE                                                       FGR3290
                  DO 833 I = 1,3                                                 FGR3300
                        XX = XXT                                                 FGR3310
                        YY = YYT                                                 FGR3320
                        IF(I-2) 834,835,836                                      FGR3330
           834    XX = - X(1) + XX                                               FGR3340
               GO TO 837                                                         FGR3350
           835          XX = XX                                                  FGR3360
               GO TO 837                                                         FGR3370
           836    XX = X(1) + XX                                                 FGR3380
           837          DO 832 M = 1 , MT , MI                                   FGR3390
                        DO 832 N = 1 , NT , NI                                   FGR3400
                        IF ( XX .EQ. 0.0 )  GO TO 551                            FGR3410
                        CX = XX * XX                                             FGR3420
                  GO TO 553                                                      FGR3430
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
 551        CX = 0.0                                                    FGR3440
 553        IF ( YY .EQ. 0.0 )  GO TO 552                               FGR3450
            CY = YY * YY                                                FGR3460
         GO TO 554                                                      FGR3470
 552        CY = 0.0                                                    FGR3480
 554     CONTINUE                                                       FGR3490
         CWWYY = - (( N * PI / B ) ** 2.0 ) * SIN ( M * PI * XX / A ) * FGR3500
    1       SIN ( N * PI * YY / B )                                     FGR3510
               WWYY ( M,N ) = AA ( M,N ) * CWWYY                        FGR3520
            CWYYX = - ( ( N * PI / B ) ** 2.0 ) * (M * PI / A ) *       FGR3530
    2                  COS( M * PI * XX / A ) * SIN( N * PI * YY / B )  FGR3540
            WWYYX(M,N) = AA(M,N) * CWYYX                                FGR3550
         DO 862 K = 1, MN                                               FGR3560
            WWYYK( M,N,K ) = AAX ( M,N,K ) * CWWYY                      FGR3570
 862        WWYYXK(M,N,K) = AAX(M,N,K) * CWYYX                          FGR3580
            IF (I-2) 838,839,838                                        FGR3590
 838      WSUM = WSUM + WWYY(M,N)                                       FGR3600
         DO 863 K = 1, MN                                               FGR3610
 863        WSUMY(K) = WSUMY(K) + WWYYK(M,N,K)                          FGR3620
         GO TO 1832                                                     FGR3630
 839      WSUM = WSUM + 22.0 * WWYY(M,N)                                FGR3640
            WYYX = WYYX + WWYYX(M,N)                                    FGR3650
         DO 864 K = 1, MN                                               FGR3660
            WYYXK(K) = WYYXK(K) + WWYYXK(M,N,K)                         FGR3670
 864        WSUMY(K) = WSUMY(K) + WWYYK(M,N,K) * 22.0                   FGR3680
1832     CONTINUE                                                       FGR3690
 832 CONTINUE                                                           FGR3700
         WYY = WYY + WSUM                                               FGR3710
         DO 865 K = 1, MN                                               FGR3720
            WYYK(K) = WYYK(K) + WSUMY(K)                                FGR3730
 865     CONTINUE                                                       FGR3740
 833 CONTINUE                                                           FGR3750
         BMY = -(E * X(5) / 24.0) * WYY                                 FGR3760
         DO 866 K = 1, MN                                               FGR3770
 866 BMYX(K) = -(E * X(5) / 24.0) * WYYK(K)                             FGR3780
     BMYX(5) = -(E * X(5) / 24.0) * WYYK(5)                             FGR3790
    1           - ( E / 24.0 ) * WYY                                    FGR3800
C----------------  TORSIONAL MOMENT, BMXY                               FGR3810
            XX = XXT                                                    FGR3820
              WSUM = 0.0                                                FGR3830
               WXY = 0.0                                                FGR3840
         DO 877 K = 1, MN                                               FGR3850
              WXYX(K) = 0.0                                             FGR3860
             WSUMX(K) = 0.0                                             FGR3870
 877     CONTINUE                                                       FGR3880
         DO 843 I = 1,3                                                 FGR3890
            XX = XXT                                                    FGR3900
            YY = YYT                                                    FGR3910
         IF(I-2) 844,845,846                                            FGR3920
 844     YY = X(4) + YY                                                 FGR3930
      GO TO 847                                                         FGR3940
 845     YY = YY                                                        FGR3950
      GO TO 847                                                         FGR3960
 846     YY = X(4) + YY                                                 FGR3970
```

80/80 LIST

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
 847          DO 842 M = 1 , MT , MI                                         FGR3980
              DO 842 N = 1 , NT , NI                                         FGR3990
              IF ( XX .EQ. 0.0 )  GO TO 571                                  FGR4000
              CX = XX * XX * XX                                              FGR4010
         GO TO 573                                                           FGR4020
 571          CX = 0.0                                                       FGR4030
 573          IF ( YY .EQ. 0.0 )  GO TO 572                                  FGR4040
              CY = YY * YY * YY                                              FGR4050
         GO TO 574                                                           FGR4060
 572          CY = 0.0                                                       FGR4070
 574     CONTINUE                                                            FGR4080
         CWWXY = ( M * PI / A ) * ( N * PI / B ) *                           FGR4090
     1        COS ( M * PI * XX / A ) * COS ( N * PI * YY / B )              FGR4100
         DO 872 K = 1, MN                                                    FGR4110
              IF ( XX .EQ. 0.0 )  GO TO 561                                  FGR4120
              CX = XX * XX                                                   FGR4130
         GO TO 563                                                           FGR4140
 561          CX = 0.0                                                       FGR4150
 563          IF ( YY .EQ. 0.0 )  GO TO 562                                  FGR4160
              CY = YY * YY * YY                                              FGR4170
         GO TO 564                                                           FGR4180
 872     WWXYX( M,N,K ) = AAX( M,N,K ) * CWWXY                               FGR4190
 562          CY = 0.0                                                       FGR4200
 564     CONTINUE                                                            FGR4210
         IF(I-2) 848,849,848                                                 FGR4220
 848          WSUM = WSUM + WWXY(M,N)                                        FGR4230
         DO 873 K = 1, MN                                                    FGR4240
 873          WSUMX(K) = WSUMX(K) + WWXYX(M,N,K)                             FGR4250
      GO TO 842                                                              FGR4260
 849          WSUM = WSUM + 22.0*WWXY(M,N)                                   FGR4270
         DO 874 K = 1, MN                                                    FGR4280
 874          WSUMX(K) = WSUMX(K) + WWXYX(M,N,K) * 22.0                      FGR4290
 842  CONTINUE                                                               FGR4300
         WXY = WXY + WSUM                                                    FGR4310
         DO 875 K = 1, MN                                                    FGR4320
 875          WXYX(K) = WXYX(K) + WSUMX(K)                                   FGR4330
 843  CONTINUE                                                               FGR4340
         BMXY =(XJG / 24.0 ) * WXY                                           FGR4350
         DO 876 K = 1, MN                                                    FGR4360
 876  BMXYX(K) = (XJG/24.0) * WXYX(K)                                        FGR4370
     1         +(XG*XJX(K) / 24.0 ) * WXY                                    FGR4380
              QX = DX * WX3 + H * WYYX                                       FGR4390
         DO 3100 K = 1, MN                                                   FGR4400
3100          QXK(K) = DXX(K) * WX3 + DX * WX3K(K) + HX(K) * WYYX            FGR4410
     1                  + H * WYYXK(K)                                       FGR4420
C----------   N = NTEMP                                                      FGR4430
              SMX = 1.51 * ( X(2) ** 0.75 ) * ( X(3) ** (-0.25) )            FGR4460
              SMY = 1.51 * ( X(5) ** 0.75 ) * ( X(6) ** (-0.25) )            FGR4470
                    WXA = A / C2                                             FGR4480
                    WYA = B / C2                                             FGR4490
              FCRX = 0.0                                                     FGR4500
              FCRY = 0.0                                                     FGR4510
              FAX = 0.0                                                      FGR4520
              FAY = 0.0                                                      FGR4530
```

80/80 LIST

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
                        BMX  = ABS(BMX)                                                 FGR4540
                        BMY  = ABS(BMY)                                                 FGR4550
                        BMXY = ABS(BMXY)                                                FGR4560
                             BSTX = BMX * SMX ** (-1.0)                                 FGR4600
                             BSTY = BMY * SMY ** (-1.0)                                 FGR4610
                        XSIG = BSTX                                                     FGR4630
                        YSIG = BSTY                                                     FGR4640
                        TXY = BMXY * DELX * XJ ** (-1.0)
                  TERM1 = 0.5 * (XSIG+ YSIG)                                            FGR4650
                  TERM2 = (0.5 * (XSIG - YSIG)) * (0.5 * (XSIG - YSIG))                 FGR4660
                  TERM3 = TXY **2.0                                                     FGR4670
                  SHEAR = SQRT(TERM2 + TERM3)                                           FGR4680
                  SIG1 = TERM1 + SHEAR                                                  FGR4690
                  SIG2 =    (TERM1 - SHEAR)                                             FGR4700
                        SIGD = SQRT(SIG1**2.0-SIG1*SIG2+SIG2**2.0)                      FGR4720
                        FCRX = C4 * X(3) ** (-2.0)                                      FGR4730
                        FAX =BMX                                                        FGR4740
                        FCRY = C4 * X(6) ** (-2.0)                                      FGR4750
                        FAY = BMX                                                       FGR4760
         C----------    NONLINEAR CONTRAINTS                                            FGR4770
         C---------     G1 TO G6 > 0.0                                                  FGR4780
                        G(1) = SIGMA - BSTX                                             FGR4790
                        G(2) = SIGMA - BSTY                                             FGR4800
                        G(3) = WXA - W                                                  FGR4810
                        G(4) = WYA - W                                                  FGR4820
                        G(5) = FCRX - BSTX                                              FGR4830
                        G(6) = FCRY - BSTY                                              FGR4840
                        G(7) = SIGMA - SIGD                                             FGR4850
                  DO 791 J = 1, NNL                                                     FGR4870
                  DO 791 K = 1, MN                                                      FGR4880
            791         GX(J,K) = 0.0                                                   FGR4890
                  DO 891 K = 1 , MN                                                     FGR4900
                        FCRXX(K) = 0.0                                                  FGR4910
                        SMXX(K) = 0.0                                                   FGR4920
                        SMXX(2) = 0.75 *1.51 * ( X(2) ** (-0.25) ) *                    FGR4930
              1              ( X(3) ** (-0.25) )                                        FGR4940
                        SMXX(3) = -0.25 *1.51 * ( X(2) **0.75 ) *                       FGR4950
              1              ( X(3) ** (-1.25) )                                        FGR4960
                        FCRXX(3) = - 2.0 * C4 * X(3) ** (-3.0)                          FGR4970
                        FAXX(K) = BMXX(K)                                               FGR4980
                        FCRYX(K) = 0.0                                                  FGR4990
                        SMYX(K) = 0.0                                                   FGR5000
                        SMYX(5) = 0.75 * 1.51 * ( X(5) ** (-0.25) ) *                   FGR5010
              1              ( X(6) ** (-0.25) )                                        FGR5020
                        SMXX(3) = -0.25 *1.51 * ( X(5) **0.75 ) *                       FGR5030
              1              ( X(6) ** (-1.25) )                                        FGR5040
                        FCRYX(6) = - 2.0 * C4 * X(6) ** (-3.0)                          FGR5050
                        FAYX(K) = BMYX(K)                                               FGR5060
            891   CONTINUE                                                              FGR5070
                        DO 520 K = 1,MN                                                 FGR5080
                             BSTXX(K) = BMXX(K) * SMX ** (-1.0)                         FGR5120
              1                   - BMX * ( SMX ** (-2.0) ) * SMXX(K)                   FGR5130
                             BSTYX(K) = BMYX(K) * SMY ** (-1.0)                         FGR5140
              1                   - BMY * ( SMY ** (-2.0) ) * SMYX(K)                   FGR5150
```

80/80 LIST

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
                   TXYX(K) = BMXYX(K) * DELX * XJ ** (-1.0) +
     1                    BMXY * DELXK(K)* XJ ** (-1.0) -
     2                    BMXY * DELX * ( XJ ** (-2.0) ) * XJX(K)
  520         CONTINUE                                                  FGR5190
              DO 3000 K = 1,MN                                          FGR5340
                   XSIGX(K) = BSTXX(K)                                  FGR5350
                   YSIGX(K) = BSTYX(K)                                  FGR5360
                        TERM1X(K) = 0.5 * (XSIGX(K) + YSIGX(K))         FGR5370
                        TERM2X(K) = 2.0 * (0.5 * (XSIG - YSIG)) * 0.5 * FGR5380
     1                       (XSIGX(K) - YSIGX(K))                      FGR5390
                        TERM3X(K) = 2.0 * TXY * TXYX(K)                 FGR5400
                        SHEARX(K) = 0.5 * (TERM2 + TERM3) * (TERM2X(K) + FGR5410
     1                              TERM3X(K))                          FGR5420
                        SIG1X(K) = TERM1X(K) + SHEARX(K)                FGR5430
                        SIG2X(K) = TERM1X(K) - SHEARX(K)                FGR5440
                   SIGDX(K) = 0.5 * ((SIG1**2.0-SIG1*SIG2+SIG2**2.0)**(-0.5)) FGR5450
     1                     * ( 2.0 * SIG1 * SIG1X(K) - SIG2 )           FGR5460
C---------         GRADIENTS                                            FGR5470
                        GX(1,K) = -BSTXX(K)                             FGR5480
                        GX(2,K) = -BSTYX(K)                             FGR5490
                        GX(3,K) =  - WX(K)                              FGR5500
                        GX(4,K) =  - WX(K)                              FGR5510
                        GX(5,K) = FCRXX(K) - BSTXX(K)                   FGR5520
                        GX(6,K) = FCRYX(K) - BSTYX(K)                   FGR5530
                        GX(7,K) = - SIGDX(K)                            FGR5540
 3000         CONTINUE                                                  FGR5550
C9000 CONTINUE                                                          FGR5560
                   N = NTEMP                                            FGR5570
C----------        CUT OFF HERE WHEN USED IN RGM                        FGR5580
                   IF ( NPR .EQ. 0 )  GO TO 690                         FGR5590
              DO 881 M = 1, MT , MI                                     FGR5600
  881         PRINT 802, (D1(M,N), N = 1, NT , NI )                     FGR5610
              DO 882 M = 1, MT , MI                                     FGR5620
  882         PRINT 802, (D2(M,N), N = 1, NT , NI )                     FGR5630
              DO 883 M = 1, MT , MI                                     FGR5640
  883         PRINT 802, (D3(M,N), N = 1, NT , NI )                     FGR5650
              DO 884 M = 1, MT , MI                                     FGR5660
  884         PRINT 802, (D4(M,N), N = 1, NT , NI )                     FGR5670
              DO 899 M = 1, MT, MI                                      FGR5680
  899         PRINT 802,(DAA(M,N), N = 1, NT , NI )                     FGR5690
              DO 892 M = 1, MT, MI                                      FGR5700
  892         PRINT 802, (AA(M,N), N = 1, NT , NI )                     FGR5710
              PRINT 802,F                                               FGR5720
              PRINT 802,  (FX(K), K=1, MN )                             FGR5730
              PRINT 802, X(2), X(3), DELX, AX, SMX
              PRINT 802, X(5), X(6), DELY, AY, SMY
              PRINT 802, XEI, YEI                                       FGR5760
              PRINT 802, XJ, YJ                                         FGR5770
              PRINT 802, XJG, YJG, H                                    FGR5780
              PRINT 802, BMX, BMY, BMXY                                 FGR5790
              PRINT 802, DX, WX3, H, WYYX, QX                           FGR5800
C-----RETURN                                                            FGR5810
              PRINT 802, XSIG, YSIG, TXY                                FGR5820
              PRINT 802, TERM1, TERM2,   TERM3                          FGR5830
```

80/80 LIST

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
CARD
            PRINT 802, SHEAR, SIG1, SIG2                                      FGR5840
            PRINT 802, SIGMA, BSTX                                            FGR5850
            PRINT 802, SIGMA, BSTY                                            FGR5860
            PRINT 802, WXA, W                                                 FGR5870
            PRINT 802, WYA, W                                                 FGR5880
            PRINT 802, FCRX, BSTX                                             FGR5890
            PRINT 802, FCRY, BSTY                                             FGR5900
            PRINT 802, SIGMA, SIGD                                            FGR5910
            PRINT 802, SIGMA, BSTX, WXA, W, FCRX, BSTX                        FGR5920
            PRINT 802, SIGMA, BSTY, WYA, W, FCRY, BSTY, SIGMA, SIGD           FGR5930
            DO 688 J = 1, NNL                                                 FGR5940
      688   PRINT 802, G(J)                                                   FGR5950
            DO 689 J = 1, NNL                                                 FGR5960
      689   PRINT 802, ( GX(J,K), K = 1, MN )                                 FGR5970
      690   CONTINUE                                                          FGR5980
                 IF ( NNL .NE. 0 )  GO TO 890                                 FGR5990
                 G(1)    = 0.0                                                FGR6000
                 GC(1)   = 0.0                                                FGR6010
                 GX(1,1) = 0.0                                                FGR6020
                 GX(1,2) = 0.0                                                FGR6030
            GO TO 880                                                         FGR6040
      890   CONTINUE                                                          FGR6050
                 IF ( NNL .NE. 1 ) GO TO 870                                  FGR6060
                 J = NNL                                                      FGR6070
                 GCT(J) = 0.0                                                 FGR6080
            DO 670 K = 1, MN                                                  FGR6090
      670        GCT(J) = GCT(J) + GX(J,K) * X(K)                             FGR6100
                 GC(J) = G(J) - GCT(J)                                        FGR6110
            GO TO 880                                                         FGR6120
      870   CONTINUE                                                          FGR6130
            DO 772 J = 1, NNL                                                 FGR6140
      772        GCT(J) = 0.0                                                 FGR6150
            DO 770 J = 1, NNL                                                 FGR6160
            DO 770 K = 1, MN                                                  FGR6170
      770        GCT(J) = GCT(J) + GX(J,K) * X(K)                             FGR6180
            DO 771 J = 1, NNL                                                 FGR6190
      771        GC(J) = G(J) - GCT(J)                                        FGR6200
      880   CONTINUE                                                          FGR6210
                 SUM = 0.0                                                    FGR6220
            DO 700 K = 1, MN                                                  FGR6230
                 SUM = SUM + FX(K) * X(K)                                     FGR6240
      700   CONTINUE                                                          FGR6250
                 IF ( LF .EQ. 1 )  GO TO 760                                  FGR6260
                 FC = F - SUM                                                 FGR6270
                 IF ( NPR .EQ. 0 )  GO TO 827                                 FGR6280
            PRINT 802, FC, F, SUM                                             FGR6290
      827   CONTINUE                                                          FGR6300
            GO TO 761                                                         FGR6310
      760        FC = 0.0                                                     FGR6320
      761   CONTINUE                                                          FGR6330
                 NPR = 0                                                      FGR6340
            RETURN                                                            FGR6350
            END                                                               FGR6360
```

80/80 LIST

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
               SUBROUTINE LFCN ( GLX, GLC, BL, UB )                             FGR3900
         C..............................................................        FGR3940
         C--------        SUBROUTINE LFCN COMPUTES VALUES OF LINEAR FCN
         C                REQUIRED SUBROUTINES                                  FGR3950
         C                   NONE                                               FGR3960
         C..............................................................        FGR3970
         C     PROBLEM 3
         C..............................................................        F
               COMMON MN, LIN, NNL, NPR                                         FGR3910
               DIMENSION  GLX(25,12), GLC(25)                                   FGR3920
               DIMENSION  BL(12), UB(12)                                        FGR3930
         C                                                                      FGR4010
                 BL(1) = 24.0                                                   FGR4020
                 BL(2) = 11.3                                                   FGR4030
                 BL(3) = 100.0                                                  FGR4040
                 BL(4) = BL(1)                                                  FGR4050
                 BL(5) = BL(2)                                                  FGR4060
                 BL(6) = BL(3)                                                  FGR4070
                 UB(1) = 120.0                                                  FGR4100
                 UB(2) = 349000.0                                               FGR 630
                   UB(3) = 350.0
                 UB(4) = UB(1)                                                  FGR4130
                 UB(5) = UB(2)                                                  FGR4140
                 UB(6) = UB(3)                                                  FGR4150
                   NBL = 6                                                      FGR4180
                   NUB = 6                                                      FGR4190
               DO 401 K = 1, NBL                                                FGR4200
           401     GLC(K) = -BL(K)                                              FGR4210
               DO 402 K = 1, NUB                                                FGR4220
           402     GLC(NBL + K) = UB(K)                                         FGR4230
               DO 200 J = 1, LIN                                                FGR4240
               DO 200 K = 1, MN                                                 FGR4250
           200     GLX(J,K) = 0.0                                               FGR4260
               DO 403 K = 1, NBL                                                FGR4270
           403     GLX(K, K) = +1.0                                             FGR4280
               DO 404 K = 1, NUB                                                FGR4290
           404     GLX(NBL + K, K) = -1.0                                       FGR4300
         C---------       OTHER CONSTRAINTS                                     FGR4310
               RETURN                                                           FGR4320
               END                                                              FGR4330
```

APPENDIX C

DETERMINATION OF $C = C_{11}$ FROM DEFLECTION CONSIDERATION FOR CASE (2)--FORMULA (3.46) WITH THE EQUALITY SIGN

From deflection consideration, C can be determined as follows:

Letting

$$C_5 = \frac{C_2}{C_1} \tag{C1}$$

and noting Formula (3.16), Formula (3.21) can be rewritten as

$$C_5 \frac{MA^2}{EI} \leq C_3 A, \tag{C2}$$

$$\frac{M}{I} \leq \frac{C_3 E}{C_5 A} . \tag{C3}$$

Let

$$C_9 = \frac{C_3 E}{C_5 A}$$

then

$$\frac{M}{I} \leq C_9 . \tag{C4}$$

From strength consideration, Formula (3.35) leads to

$$\frac{M}{I} = \frac{2\sigma_{all}^{1.25}}{0.5^{0.5} C_4^{0.25}} \frac{C^{1.25}}{A_t^{0.5}} . \tag{C5}$$

Let

$$C_{10} = \frac{2\sigma_{all}^{1.25}}{0.5^{0.5}\, C_4^{0.25}} \qquad (C6)$$

then

$$\frac{M}{I} = C_{10} \frac{C^{1.25}}{A_t^{0.5}} . \qquad (C7)$$

Combining Formulas (C7) and (C4),

$$C_{10} \frac{C^{1.25}}{A_t^{0.5}} \leq C_9 \qquad (C8)$$

$$C \leq (\frac{C_9}{C_{10}})^{0.8} A_t^{0.4} . \qquad (C9)$$

Let

$$C_6 = (\frac{C_9}{C_{10}})^{0.8} \qquad (C10)$$

then

$$C \leq C_6 A_t^{0.4}. \qquad (C11)$$

A_t can be expressed from strength condition (Formula 3.35),

$$\frac{M}{C\,\sigma_{all}} = S = 0.235\,(\lambda A_t^{3})^{0.5} \qquad (C12)$$

$$(\frac{M}{0.235\,\sigma_{all}})^4 = C^4 \lambda^2 A_t^{6}. \qquad (C13)$$

Let

$$C_7 = (\frac{M}{0.235\,\sigma_{all}})^4 \qquad (C14)$$

then

$$C_7 = C^4 \lambda^2 A_t^{6}. \qquad (C15)$$

Using λ from Formula (3.49),

$$C_7 = C^3 \frac{C_4}{\sigma_{all}} A_t^{\,6} \qquad (C16)$$

or

$$A_t = \frac{1}{C^{0.5}} \left(\frac{C_7 \sigma_{all}}{C_4}\right)^{\frac{1}{6}} . \qquad (C17)$$

Let

$$C_8 = \left(\frac{C_7 \sigma_{all}}{C_4}\right)^{\frac{1}{6}} \qquad (C18)$$

then

$$A_t = \frac{1}{C^{0.5}} C_8 . \qquad (C19)$$

Substituting Formula (C19) into Formula (C11),

$$C \leq C_6^{\,0.833} C_8^{\,0.333} . \qquad (C20)$$

Let

$$C_{11} = C_6^{\,0.833} C_8^{\,0.333} , \qquad (C21)$$

then

$$C \leq C_{11} . \qquad (C22)$$

Substitution results in C_{11} expressed by all constants, and

$$C_{11} = \left(\frac{C_1^{\frac{13}{18}} C_4^{\frac{1}{9}}}{A^{\frac{5}{9}} \sigma_{all}}\right) \left(\frac{0.354 E C_3}{C_2}\right)^{\frac{2}{3}} \left(\frac{pb}{0.00305}\right)^{\frac{1}{18}} \qquad (C23)$$

APPENDIX D

DETERMINATION OF C_{12} AND THE MOMENT OF INERTIA, I, FOR CASE (3)

Case (3) implies that G_1 and G_2 carry equality signs. Rewriting G_1 (Formula 3.26 with equality sign) with

$$S = \frac{2I}{h} \tag{D1}$$

instead, G_1 becomes

$$\frac{h C_1 p b A^2}{2I} = \sigma_{all} \, . \tag{D2}$$

Dividing Formula (D2) by G_2 (Formula 3.27 with equality sign) and rearranging, then

$$\frac{h}{A} = \frac{2 C_2 \sigma_{all}}{C_1 C_3 E} \, . \tag{D3}$$

Defining

$$C_{12} = \frac{2 C_2 \sigma_{all}}{C_1 C_3 E} \, , \tag{D4}$$

then

$$h = C_{12} A . \tag{D5}$$

Substituting Formula (D5) into Formula (D2) and rearranging, the moment of inertia

$$I = \frac{C_1 p b A^3 C_{12}}{2 \sigma_{all}} \, . \tag{D6}$$

VITA

Meng Sam Chan

Candidate for the Degree of

Doctor of Philosophy

Thesis: OPTIMIZATION OF THE SIZE AND THE LOCATION OF STEEL MEMBERS OF A GRIDWORK BY MATHEMATICAL PROGRAMMING

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Macao, Portuguese Colony, April 17, 1936, son of Mr. and Mrs. Lap Man Chan.

Education: Graduated from Kwong Dai High School, Macao, in 1952; joined Mount Allison University, Sackville, N. B., Canada, in 1958 and received the Engineering Diploma in 1961; joined Heald Engineering College, San Francisco, California, with a major in Civil Engineering, in 1963 and received the Bachelor of Science degree in 1964; received the Master of Science degree from the University of New Mexico, Albuquerque, New Mexico, in 1966; joined Oklahoma State University in 1968 and completed requirements for the Doctor of Philosophy degree at Oklahoma State University in 1971.

Professional Experience: Concrete Quality control, Tiverton Nuclear Power Generating Station, Ontario Hydro, Canada, 1961; Power line footing layout, North Bay section, Ontario Hydro, Canada, 1962; Design drafting, Glassi Precast Concrete Company, South San Francisco, California, 1964; Graduate assistant, University of New Mexico, 1964-1966.

Membership in Organization:

Associate Member of Engineering Institute of Canada; Student Member of American Concrete Institute.