INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106
PLECNIK, Joseph Matthew, 1945-

OPTIMIZATION OF ENGINEERING SYSTEMS USING EXTENDED GEOMETRIC PROGRAMMING.

The Ohio State University, Ph.D., 1974
Engineering, civil

University Microfilms, A XEROX Company, Ann Arbor, Michigan

74-17,799

This dissertation has been microfilmed exactly as received.
OPTIMIZATION OF ENGINEERING SYSTEMS
USING EXTENDED GEOMETRIC PROGRAMMING

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By

Joseph Matthew Plecnik, B.E., M.S.

* * * * *

The Ohio State University

1974

Reading Committee:

Alfred G. Bishara
Albert B. Bishop
T. Y. Chen
Ralph W. Swain

Approved By

S. Y. Chen
Adviser
Department of
Civil Engineering
ACKNOWLEDGEMENTS

The author acknowledges his gratitude to Professor T.Y. Chen for suggesting this dissertation topic and providing sound advice during the past four years of graduate studies. Many helpful suggestions in the field of Geometric Programming were also provided by Professor Richard E. Wendell of Carnegie-Mellon University in Pittsburg, Pa. Several other individuals contributed to the theoretical developments within this dissertation and their aid is also greatly appreciated.

The author's graduate research and studies were financially supported by the National Science Foundation. The encouragement given to the author by his parents will always be a source of lasting memories. Lastly, I wish to extend eternal gratitude to my wife, Anne Marie, who not only typed this dissertation, but also provided guidance in the development of the format utilized in the following chapters.
VITA

Dec. 28, 1945

Born - Hotedersica, Yugoslavia

1969

B.S.C.E., Youngstown State University, Youngstown, Ohio

1970

M.S., The Ohio State University, Columbus, Ohio

1970-1971

Research Engineer, USAMERDC, Fort Belvoir, Va.

1973-

Assistant Professor, California State University, Long Beach

PUBLICATIONS


"Reinforcing Bar-Concrete Interaction in Simply Supported Concrete Slabs," M.S. Thesis, The Ohio State University, Columbus, Ohio, March, 1970.


FIELDS OF STUDY

Major Field: Structural Engineering

Studies in Structural Analysis. Professor T.Y. Chen
Studies in Structural Design. Professor Morris Ojavo
Studies in Geometric Programming. Professor R. Wendell
Studies in Nonlinear Programming. Professor A. Bishop
FIELDS OF STUDY (Cont.)

Studies in Plasticity Theory. Professor N. Perrone
Studies in Elasticity. Professor Paul F. Graham
Studies in Applied Mathematics. Professor H.D. Colson
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>1. PRELIMINARY DISCUSSION</td>
<td>1</td>
</tr>
<tr>
<td>Section 1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Section 1.2 Geometric Programming (GP)</td>
<td>3</td>
</tr>
<tr>
<td>Section 1.3 Objectives of this Dissertation</td>
<td>4</td>
</tr>
<tr>
<td>2. ANALYSIS OF TORSIONAL SYSTEMS</td>
<td>6</td>
</tr>
<tr>
<td>Section 2.1 Optimization in Structural Engineering</td>
<td>6</td>
</tr>
<tr>
<td>Section 2.2 Analysis for a Simple Monocoque Structure</td>
<td>7</td>
</tr>
<tr>
<td>Section 2.3 Analysis of Large Torsional Systems</td>
<td>16</td>
</tr>
<tr>
<td>3. DEVELOPMENT OF ALGORITHMS</td>
<td>22</td>
</tr>
<tr>
<td>Chapter</td>
<td>Section</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>4.</td>
</tr>
<tr>
<td></td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>5.</td>
</tr>
<tr>
<td></td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>6.</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Section 6.1 Advantages of Algorithm EGPLP</td>
<td>87</td>
</tr>
<tr>
<td>Section 6.2 Disadvantages of Algorithm EGPLP</td>
<td>87</td>
</tr>
<tr>
<td>Section 6.3 Final Remarks</td>
<td>88</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>89</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 5.1.1 ............................................ 83
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2</td>
<td>17</td>
</tr>
<tr>
<td>4.1.1</td>
<td>52</td>
</tr>
<tr>
<td>4.2.1</td>
<td>64</td>
</tr>
<tr>
<td>4.3.1</td>
<td>74</td>
</tr>
</tbody>
</table>
CHAPTER 1  PRELIMINARY DISCUSSION

Section 1.1  Introduction

Throughout history man has sought to maximize, either knowingly or unknowingly, his profits, minimize his costs, and develop the most efficient systems and processes. Therefore, it would be logical to assume that the art and science of optimization is as old as man himself. The universality of the optimum principle was best stated by an outstanding mathematician, Leonhard Euler: "Nothing happens in this world in which some reason of maximum or minimum would not come to light." This statement may today be an oversimplification of human and natural laws. Nevertheless, every engineer and scientist can quickly recite the many optimum principles that seem to govern both human and natural behavior.

Rudimentary aspects of optimization were known both to Greek and Roman civilizations and, perhaps, much earlier. But the word "optimum" was first employed by Leibniz in 1710. The development of calculus during the late seventeenth century by Leibniz and Newton mathematically defined the concepts of maximum and minimum of functions. In 1696, John Bernoulli proposed the brachistochrone problem which initiated the development of variational calculus. These early concepts of the optimum conditions led to a rapid development of the
many minimum principles in classical mechanics eventually summarized by the well-known generalized Hamiltonian Principle in 1827. These and other classical optimization methods are quite useful in theoretical proofs. However, only simple, practical problems can be solved by such techniques. Due to computational difficulties, large problems remained unsolvable until the development of the electronic digital computer in 1946 (Ref. 1).

With advancements in computer technology and a desire for economy, Mathematical Programming developed into a long list of practical but highly specialized techniques based primarily on algebra, vector calculus, and the sequence convergence theory. The first and still the most important single technique in Mathematical Programming is the well-known simplex algorithm developed by G.B. Dantzig in 1947. The simplex technique is directly applicable only to linear programming problems. However, approximation by linearization of nonlinear equations is often employed to solve nonlinear programming problems via simplex or the revised simplex methods. Although the simplex algorithm can be used to solve any well-defined linear programming problem, an efficient solution of large nonlinear problems requires essentially independent analysis for each problem. Accordingly, this custom job aspect of nonlinear problems has resulted in a variety of optimization techniques, including quadratic programming, convex programming, geometric programming,
etc. Since geometric programming will be the primary technique utilized in this research paper, a brief background of this new and valuable technique will be given in Sections 1.2 and 3.2.

Section 1.2 Geometric Programming (GP)

In 1961, Clarence Zener (Westinghouse, Pittsburgh) observed that many problems in engineering consisted of optimizing the sum of component costs of a system, where each cost is a function of the product of variables, and raising each variable to a known power, i.e., the total cost is the sum of posynomial terms.\(^1\) Furthermore, if the number of posynomial terms in the cost function exceeded the number of variables by one, Zener noted that the global optimum solution is uniquely obtained by solving a system of linear equations. In 1962, Duffin and Peterson of Carnegie-Mellon University joined Zener and extended his technique to problems with posynomial inequality constraints. Further research by these three individuals resulted in a comprehensive theory for posynomial GP, fully described in Ref. 2 and briefly

\[^1\text{A function } P(X) = \sum_{j=1}^{N} C_j \prod_{i=1}^{N} X_i^{a_{ij}} \text{ is defined as a signomial if } C_j \text{ and } a_{ij} \text{ are real numbers. If all } C_j \text{ coefficients are positive real numbers, } P(X) \text{ is said to be a posynomial.}\]
summarized in Sec.3.2. Although posynomial GP has considerable practical application, the bulk of engineering problems contain signomial terms.

Signomial GP has received extensive attention since 1967 resulting in the development of many optimization techniques for both general and specific problems. Nearly all of these techniques approximate signomials with posynomials and then solve the dual GP. However, at the present time, there is no efficient technique (Ref. 12) for the solution of the dual for many GP problems. As will be indicated in Chapter 3, when the number of posynomial terms exceeds the number of variables by more than five or six, the solution of the dual may become a difficult task. Realizing such hardships, this research paper will present two optimization algorithms designed to solve specific large systems efficiently.

Section 1.3 Objectives of this Dissertation

Strictly speaking, signomial and posynomial GP deals only with inequality constraints. Wilde (Ref. 5) first considered equality constraints in GP by utilizing Lagrangian multipliers, a technique which is complicated and inefficient. Most, if not all, signomial GP techniques generate a sequence of feasible solutions when equality constraints are not present. Since equality constraints do appear in problems of this research paper, a different procedure had to be developed, as indicated in Sec. 3.5.
The two algorithms in Chapter 3 generate a sequence of solutions, $X^{(i)}$, which satisfy the actual inequality but not necessarily the equality constraints. The solutions generated within each iteration are therefore infeasible, and if convergence does occur, then only the final optimal solution will be necessarily feasible. The disadvantage of infeasible intermediate solutions is partially offset by a rapid convergence to the optimal solution when convergence is possible. Additional discussion of this topic is given in Sec. 5.2, following the development of general equations for torsional problems in Chapter 2.
CHAPTER 2 ANALYSIS OF TORSIONAL SYSTEMS

Section 2.1 Optimization in Structural Engineering

Classical optimization techniques were applied in Structural Engineering as early as 1904 when A.G.M. Michell wrote a monumental paper on material economy in framed structures (Ref. 6). With the development of linear programming and plastic design of framed structures, mathematical programming obtained a firm foundation in structural design. Additional applications were discovered in many different areas of Civil Engineering, especially for linear systems. Large nonlinear systems have received much less attention due to complexity of analysis and a lack of interest by the construction industry and consulting firms. The rapid development of the finite element method has not as yet produced a significant impact on optimization of large systems in Civil Engineering.

One area of Structural Engineering that has adapted a considerable amount of mathematical programming, both linear and nonlinear, is the weight-conscious aerospace industry. Due to complexity and aerodynamical considerations, the optimal design of wing and fuselage structures is generally performed by assuming the values
of some variables and using finite element methods for analysis and optimization of as yet undefined variables. The procedure given in the next section for monocoque structures may be used for optimal design of wing and fuselage structures when a simplified mathematical model is defined. The primary Civil Engineering application of the procedure and equations derived in Sec. 2.2 lies in the design of framework or other lattice structures composed partially or entirely of thin-walled, closed box members such as bridges. In particular, this procedure is directly applicable to design of lightweight mobile military assault bridges utilized extensively by the United States Army.

Section 2.2 Analysis for a Simple Monocoque Structure

For the sake of illustration, a simple two-celled problem will first be presented, then followed by a more generalized problem. A thin-walled monocoque structure is shown in Fig. 2.2.1, with the dotted lines representing the centerlines of the wall thicknesses. To avoid unnecessary complications, only torsion is considered in this chapter although solved examples in Chapter 4 will include shear loads. Let \( A_1 \) and \( A_2 \) denote the respective cell areas enclosed within the dotted lines, and assume that all wall members are composed of the same material. Enclosed areas \( A_1 \) and \( A_2 \) may or may not be independent of member lengths \( l_1 \). \( A_1 \) and \( A_2 \) will be functions of \( l_1 \) when
definite cross-sectional shapes such as rectangles are desired.

\[ l_i = \text{member lengths} \]
\[ q_1, q_2 = \text{shear flows} \]
\[ t_i = \text{wall thicknesses corresponding to } l_i \]

All wall thicknesses assumed to be "thin".

Fig. 2.2.1

The objective for this problem is to minimize the weight per unit length of the cross-section subject to various constraints. Note that all variables listed in Fig. 2.2.1 are constrained to be positive -- this requirement immediately suggests GP as a possible optimization technique. Let \( B \) be the weight per unit volume of the material, the objective function is

\[
Z = \text{Min} (\text{Weight}) = B(\text{Min}(l_1t_1 + \ldots + l_7t_7)) \quad (2.2.1)
\]
The constraints on wall thicknesses may consist of lower and/or upper bounds. The lower bound may be the minimum thickness, $t_i^\text{min}$, commercially available while the upper bound, $t_i^\text{max}$, may represent the maximum desirable thickness. Experience and the form of the objective function in Fig. 2.2.1 have indicated that monocoque structures subjected to torsion have optimal solutions at or near the lower bound thickness, $t_i^\text{min}$, and, consequently, the upper bounds on $t_i^\text{max}$ can be neglected. Hence, the only constraints, in addition to positivity, required on wall thickness for Fig. 2.2.1 are

$$t_i^\text{min} \geq t_i > 0, \quad i=1,2,\ldots,7. \tag{2.2.2}$$

Variable wall thicknesses may be approximated by segments with constant thickness.

Many constraints on wall lengths, $l_i$, and cell areas, $A_k$, are possible although at least a lower bound, $l_i^\text{min}$, should be provided for each $l_i$ to avoid some $l_i \rightarrow 0^+$, i.e.,

$$l_i \geq l_i^\text{min} > 0, \quad i=1,\ldots,7 \tag{2.2.3}$$

Likewise, if cell areas, $A_k$, are independent of wall lengths, $l_i$, desirable upper and/or lower bounds should be provided on cell areas to prevent $A_k \rightarrow 0^+$ or $A_k \rightarrow +\infty$. In most practical monocoque structures such as airplane wings and bridges, the cell areas, $A_k$, are well-defined in terms of lengths, $l_i$, and constraints on
$A_k$ would then be redundant.

The most uncontrollable and indefinite variables are the shear flows, $q_k$, since the objective function is explicitly independent of these variables. A lower bound, in addition to positivity condition, $q_k > 0$, need not be imposed on $q_k$. Imposition of such lower bounds would probably result in a more constrained and thus a heavier structure. The upper bounds on $q_k$ are also unnecessary due to the presence of the yielding constraint (Mises-Hencky yield criterion was chosen in this research paper).

\[ \frac{q_k}{t_1} \leq \frac{f_y}{\sqrt{3}}, \quad f_y = \text{yield strength} \quad (2.2.4) \]

By specifying a lower bound on $t_1$, an upper bound on $q_k$ is then implicitly given by the last equation. If the yield constraints are stated as equalities in (2.2.4), $q_k$ could easily be solved in terms of $t_1$, using the implicit function theorem resulting in a reduction of constraint equations and number of variables. Equality yield constraints represent a fully stressed structure.

A structural system is defined to be fully stressed if every independent structural element or part of it has attained the yielding state followed by possible plastic flow. In the study of large, fully stressed structural systems, the analysis and solution for optimality conditions can often be rather simple, since the fully
stressed structure may degenerate into a linear programming problem, as in the case of plastic design of framed structures. However, the necessary and sufficient conditions required to show that fully stressed structures do indeed form an optimal design are very rare. In fact, such conditions exist for few practical problems (Ref. 7). Furthermore, there are cases where a fully stressed structure does not produce an optimal design with regard to weight or cost (Ref. 7), a situation that may occur in torsional analysis of monocoque structures. The following paragraph returns to the primary task of this chapter, i.e., the development of constraint equations.

So far, the upper and lower bounds for all the variables have been considered in addition to yield constraints defined by (2.2.4). Additional equality and inequality constraints will now be developed from elasticity equations. It is assumed that if yielding is attained, i.e., equality holds in (2.2.4) at the optimal solution, plastic flow will not occur and elastic solution is still valid (If plastic flow is initiated in any cell, local failure will occur due to limited thickness of the cell walls resulting in excessive cell rotations). Let \( T, T_0 \) be the torque capacity and design torque respectively; from Fig. 2.2.1 (see Ref. 8, Sec. 2.3 for elasticity theory),

\[
T = 2 \sum q_1 A_1 = 2(q_1 A_1 + q_2 A_2) \geq T_0. \tag{2.2.5}
\]

This constraint may also be written as \( T = T_0 \), however,
the inequality constraint represents a more general problem. The remaining constraints required for a complete definition of this torsional problem deal with rotation of cells under torque $T_o$. Rotations of Cell 1 and Cell 2 are $\theta_1$ and $\theta_2$ respectively, as given by

$$2G\theta_1 = \frac{q_1}{A_1} \left[ \frac{l_1}{t_1} + \frac{l_2}{t_2} + \frac{l_3}{t_3} + \frac{l_4}{t_4} \right] - \frac{q_2 l_4}{A_1 t_4}$$

$$2G\theta_2 = \frac{q_2}{A_2} \left[ \frac{l_4}{t_4} + \frac{l_5}{t_5} + \frac{l_6}{t_6} + \frac{l_7}{t_7} \right] - \frac{q_1 l_4}{A_2 t_4} \quad (2.2.6)$$

From compatibility conditions,

$$\theta_1 - \theta_2 = 0. \quad (2.2.7)$$

Hence,

$$\frac{q_1}{A_1} \left[ \frac{l_1}{t_1} + \frac{l_2}{t_2} + \frac{l_3}{t_3} + \frac{l_4}{t_4} \right] + \frac{q_1 l_4}{A_2 t_4}$$

$$- \frac{q_2}{A_2} \left[ \frac{l_4}{t_4} + \frac{l_5}{t_5} + \frac{l_6}{t_6} + \frac{l_7}{t_7} \right] - \frac{q_2 l_4}{A_1 t_4} = 0 \quad (2.2.8)$$

It is now obvious that the most complex constraints for such torsional problems are due to equality of cell rotations.

Additional constraints may be imposed on this problem if so desired. For example, if the cross-section were rectangular, the total length or width may be constrained within certain limits. Such constraints would probably increase the weight of the structure and should
not be included unless they are required. The objective function and the least number of constraints required for complete definition of the two-celled cross-section, given in Fig. 2.2.1, are now summarized.

\[ Z = \text{Min (weight)} = B \text{Min } (t_1l_1 + t_2l_2 + \ldots + t_7l_7) \quad (2.2.9) \]

subject to

\[
\begin{align*}
& t_i \geq t_i, \quad i=1,2,\ldots,7 \quad (2.2.10) \\
& l_i \geq l_i, \quad i=1,2,\ldots,7 \quad (2.2.11) \\
& T \geq T_0 \\
& \frac{q_k}{t_i} \leq \frac{f_v}{\sqrt{3}} \quad \text{for all } k=1,2; \quad i=1,\ldots,7 \quad (2.2.12) \\
& \theta_1 - \theta_2 = 0 \quad (2.2.13) \\
& t_i, l_i > 0; \quad i=1,\ldots,7 \quad (2.2.14) \\
& q_1, q_2 > 0 
\end{align*}
\]

There are a total of 46 constraints in this simple two-celled structure if lower bounds and positivity constraints are included. This astounding total for such a simple system would indicate that much larger torsional systems of this type would be virtually unsolvable. However, by employing UP, all explicit positivity constraints in (2.2.14) become implicit constraints and need not be considered. Lower bounds given by (2.2.10) and (2.2.11) will also become implicit constraints by re-definition of variables as indicated in Sec. 3.7. Hence,
by using GP as the optimization technique, the number of constraints has been reduced from 46 to 16 without performing a single optimization operation. Experience and intuition can further reduce the number of variables and constraints through analysis of symmetry of the structures and insight of the inter-relationship between objective function and constraints.

Define lower bounds as follows:

\[ t_i = t_0; \quad i = 1, 2, \ldots, 7 \]  
\[ l_i = l_0; \quad i = 1, 2, \ldots, 7. \]  

(2.2.15)

If no distinction is made between the two cells, then an optimal design requires that the two cells be identical. This observation results in only three independent variables defined as

\[ t = t_i, \quad l = l_i; \quad i = 1, \ldots, 7 \]  
\[ q_1 = q_2 = q \]  

(2.2.16)

The equality constraint \( \theta_1 - \theta_2 = 0 \) is now satisfied for all \( t, l, \) and \( q \). Hence, the rather complex system given by (2.2.9)-(2.2.14) is now reduced to

\[ \min Z = 7B \quad \text{Min}(tl) \]  
subject to

\[ T \geq T_0 \]  
\[ \frac{q}{t} \leq \frac{FV}{\sqrt{3}} \]  
\[ t, l, q > 0 \]  

(2.2.17)
Hence, by assuming the lower bounds given in (2.2.15), the resulting optimization problem of (2.2.17) is a rather simple signomial GP problem with three variables and two inequality constraints.

To reiterate an old cliche, the last paragraph vividly indicates why optimization is both a science and an art. For larger systems, the art aspect of optimization may mean the difference between solving and not solving a problem. Furthermore, each system must be analyzed individually in order to generate an efficient technique for optimization. The succeeding section develops general equations for monocoque structures subjected to torsion (Ref. 9, Chpt. 1).
Section 2.3 Analysis of Large Torsional Systems

The notation introduced in this section is self-explanatory and is illustrated in Figures 2.3.1 and 2.3.2. This somewhat laborious notation will allow for a rapid mechanical development of equations for any structure of this type. The cross-sectional shapes of the cells are quite arbitrary since the areas $A_{ij}$ are considered as independent variables.
Note the convention used to identify the outside region of the box structure. Fig. 2.3.2 gives the enlargement for cell \((i, j)\).

The objective function for the weight per unit length (hence, the minimum cross-sectional area) of the structure may now be written as

\[
Z = \text{Min} \left( \text{cross-sectional area} \right)
\]

\[
= \text{Min} \sum_{i=1}^{I_0+1} \sum_{j=1}^{J_0} t_{i, j/i-1, j/i-1, j}
\]

(2.3.1)

\[
+ \sum_{i=1}^{I_0} \sum_{j=1}^{J_0+1} t_{i, j/i, j-1/i, j/i, j-1}
\]
Lower bounds on length variables \( l \) are as follows.

\[
\begin{align*}
    l_{i,j/i-1,j} & \geq l_{i,j/i-1,j} > 0 & i=1,\ldots,I_0+1, \\
    j=1,\ldots,J_0 \\
    l_{i,j/i,j-1} & \geq l_{i,j/i,j-1} > 0 & i=1,\ldots,I_0, \\
    j=1,\ldots,J_0+1
\end{align*}
\] (2.3.2)

Replacing \( l \) and \( l \) by \( t \) and \( t \) respectively in the last set of equations, the lower bounds on the thickness variable, \( t \), may be defined. This notational convention results in indicial symmetry of the form

\[
\begin{align*}
    t_{i,j/m,n} & = t_{m,n/i,j} \\
    l_{i,j/m,n} & = l_{m,n/i,j}
\end{align*}
\] (2.3.3)

Utilizing this known symmetry, considerable simplification of most equations is possible.

Lower bounds on cell areas, \( A_{ij} \), are necessary to prevent an optimum design with \( A_{ij} \to 0^+ \). Hence,

\[
A_{ij} \geq A_{ij} > 0 \quad i=1,\ldots,I_0, \\
     j=1,\ldots,J_0 \] (2.3.4)

Obviously, if all cells have a known shape, then \( A_{ij} \) will no longer be independent variables and lower bounds \( A_{ij} \) are not necessary.

Bounds for shear flow variables \( q_{ij} \) are likewise not required since yield constraints will now be defined. From Fig. 2.3.1 and using Mises-Hencky yield criterion with yield stress, \( f_y \),
Where,

\[ \begin{align*}
q_{i-1,1} - q_{i-1,1} & \leq f_y \\
t_{i,j/i-1,j} & \sqrt{3} \\
q_{i-1,1} & = q_{i,j+1} = q_{j+1} = 0, \quad i=1, \ldots, I_0 \\
q_{i,j+1} & = 0, \quad j=1, \ldots, J_0+1
\end{align*} \]

(2.3.5)

An equivalent set of equations consists of the following:

\[ \begin{align*}
q_{i-1,1} - q_{i,1} & \leq f_y \\
t_{i,j/i-1,j} & \sqrt{3} \\
q_{i-1,1} & = q_{i,j+1} = q_{j+1} = 0, \quad i=1, \ldots, I_0 \\
q_{i,j+1} & = 0, \quad j=1, \ldots, J_0+1
\end{align*} \]

(2.3.6)

The number of equations represented by (2.3.6) is nearly unimaginable. At most, only one half of the inequalities given by (2.3.6) will govern the design. Hence, a judicious choice of the governing constraints in (2.3.6) will result in fewer constraints without enlarging the feasible region. One additional inequality constraint is due to a requirement that the design torque \( T_0 \) must be equal to or less than the torque capacity of the entire
structural system.

\[ T = 2 \sum_{i=1}^{I_o} \sum_{j=1}^{J_o} (q_{i,j} A_{i,j}) \geq T_0 \quad (2.3.7) \]

The remaining constraints are due to equality of rotation for all cells. The rotation of cell \((i,j)\) is given by

\[ \theta_{i,j} = \frac{1}{2GA_{i,j}} \int dS \frac{\partial \phi}{\partial t} \]

\[ \theta_{i,j} = \frac{1}{2GA_{i,j}} \left[ q_{i,j} \left( \frac{l_{i,j,i-1,j}}{t_{i,j,i-1,j}} + \frac{l_{i,j,i+1,j+1}}{t_{i,j,i+1,j+1}} + \frac{l_{i,j,i+1,j}}{t_{i,j,i+1,j}} \right) + \frac{l_{i,j,i-1,j}}{t_{i,j,i-1,j}} \right] \quad (2.3.8) \]

\[ -q_{i-1,j} \left( \frac{l_{i,j,i-1,j}}{t_{i,j,i-1,j}} \right) - q_{i,j+1} \left( \frac{l_{i,j,i+1,j+1}}{t_{i,j,i+1,j+1}} \right) \]

\[ -q_{i+1,j} \left( \frac{l_{i,j,i+1,j}}{t_{i,j,i+1,j}} \right) - q_{i,j-1} \left( \frac{l_{i,j,i-1,j-1}}{t_{i,j,i-1,j-1}} \right) \]

The most efficient procedure for equating cell rotations is to choose reference cells in some arbitrary row, \(m\), and develop equations of the form

\[ \theta_{m_1} = \theta_{m_j} \quad j=2, \ldots, J_0. \quad (2.3.9) \]

Then, using \(\theta_{m_j}\) as the reference cell in column \(j\),
Without any form of symmetry, the total number of rotation equations given by (2.3.9) and (2.3.10) is equal to $I_o \times J_o - 1$. These equality constraints complete the set of necessary constraints required to solve these torsional problems. Additional constraints on the shape of the entire box structure and/or individual cells may also be desired. Although the general notation used in previous equations is rather laborious, it does allow for internal generation of the objective function and all constraint equations by the computer, a task which otherwise would have to be performed by an individual.

The general equations developed in this section are applicable to a variety of problems dealing with optimal design of monocoque structures. Chapter 4 will analyze two large problems utilizing the equations derived in this section. To optimize such large structural systems, the next chapter will briefly develop two optimization techniques based primarily on geometric programming. Due to a rather sophisticated mathematical background of geometric programming, much of the presentation in Chapter 3 will not be reinforced with complete proofs.
CHAPTER 3  DEVELOPMENT OF ALGORITHMS

Section 3.1  General Considerations on Optimization

From observation of equations derived in Chapter 2, such optimization problems are nonlinear with both equality and inequality constraints. Due to the complexity and the large number of variables, such problems are difficult to solve and, if solution is attained, the computer time can be quite extensive. Decomposition techniques may not be used for the torsional problems of Chapter 2 due to the form of the constraint equations. Furthermore, such well-known nonlinear methods as dynamic programming, quadratic programming and convex programming are not directly applicable. The highly useful computer program SUMT was initially used to solve several large problems with semi-satisfactory results. Computer overflow and underflow errors may present difficulties when SUMT (Ref. 11) is used to solve large problems. Further consideration of various nonlinear optimization techniques indicated that the optimization problems of Chapter 2 may be solved by Geometric Programming (GP). The following sections of this chapter will develop the techniques used to transform many types of nonlinear problems into standard GP problems defined by Duffin, Peterson
and Zener (Ref. 2).

Since problems in Chapter 2 are not of a convex nature, the solutions obtained in this research paper will be at best only local rather than global optima. No attempt will be made to verify the nature of the optimal solutions since such time-consuming methods as exhaustive search or calculus (Kuhn-Tucker conditions, saddle-point conditions, etc.) would have to be utilized.

Section 3.2 Description of Geometric Programming

GP as developed in Ref. 2 is applicable to primal problems of the form

\[(PQP) \text{ Min } g_o(X) \quad (3.2.1)\]

subject to:

\[g_k(X) \leq 1 \quad k=1, \ldots, M_o \quad (3.2.2)\]

\[X = (X_1, \ldots, X_N)\]

\[X_i > 0 \quad (3.2.3)\]

Where

\[g_k(X) = \sum_{j=1}^{\theta(k)} \left\{ c_{jk} \prod_{i=1}^{N} \left[ X_i \right]^{a_{ijk}} \right\} \quad k=0, 1, \ldots, M_o\]

\[\theta(k) = \text{ number of posynomial terms in } k \text{ th constraint}\]

\[X_i \geq 0\]

\[X_i \neq 0\]

This is not a grave restriction since if \(X_i \geq 0\), it may be possible to define \(X_i = Y_i - C_i\); where \(C_i\) is sufficiently large to assure \(Y_i > 0\) for all feasible \(X_i\).
The exponential coefficients, \( a_{ijk} \), are real numbers, while \( c_{jk} \) coefficients must be positive real numbers. This restriction on \( c_{jk} \) guarantees that posynomial GP problems yield global optimum solutions since the transformed primal problem is convex (See Ref. 2, p. 83). A quick glance into Chapter 2 indicates that torsional problems, and in fact most engineering problems, are defined by signomial objective function and signomial constraints.

DGP will denote the dual problem of PGP. Each dual variable, \( W_{jk} \), corresponds to a coefficient \( c_{jk} \) in the primal problem.

\[
(DGP) \quad v(W) = \max \left\{ \sum_{k=0}^{M_0} \left[ \sum_{j=1}^{Q} \left( \frac{c_{jk}}{W_{jk}} \right) W_{jk} \right] \lambda_k \right\}
\]

Where

\[
\lambda_k = \sum_{j=1}^{Q} W_{jk} \quad k=0,1,\ldots,M_0
\]

Constraints on dual variables \( W_{jk} \) are as follows.

\[
W_{jk} \geq 0 \quad (3.2.5)
\]

\[
\lambda_0 = 1 \quad (3.2.6)
\]

\[
\sum_{k=0}^{M_0} \sum_{j=1}^{Q} a_{ijk} W_{jk} = 0 \quad i=1,\ldots,N \quad (3.2.7)
\]
Equations (3.2.6) and (3.2.7) yield N+1 independent linear equations with \( \sum_{k=0}^{M_0} \phi(k) \) variables. If \( \sum_{k=0}^{M_0} \phi(k) = N + 1 \), the dual variables, \( W_{jk} \), have a unique solution obtained by solving the linear equations in (3.2.6) and (3.2.7). When \( \sum_{k=0}^{M_0} \phi(k) < N + 1 \), dual constraints cannot be satisfied and no optimum exists; the primal problem is inconsistent and must be reformulated. If \( \sum_{k=0}^{M_0} \phi(k) > N + 1 \), the dual problem is said to have a degree of difficulty \( DOD = \sum_{k=0}^{M_0} \phi(k) - (N + 1) \) and the dual variables no longer have a unique solution. Since the complexity of solving the dual problem is minimized when \( DOD = 0 \), this research paper will concentrate on the transformations that reduce \( DOD \) to zero or a low value.

The duality theory in Ref. 2 states that for all feasible solution vectors \( X \) and \( W \),

\[
\mathbf{g}_o(X) \geq \mathbf{v}(W) \quad (3.2.8)
\]

Furthermore, at the optimal solutions, denoted by \( X^* \) and \( W^* \),

\[
\mathbf{g}_o(X^*) = \mathbf{v}(W^*) \quad (3.2.9)
\]

The relationship between \( X^* \) and \( W^* \) is
\[ \sum_{j=1}^{N} (x_i^*)^{a_{1j}} = w_{jo}^* v(w^*) \quad j=1, \ldots, \beta(0) \]

\[ \sum_{j=1}^{N} (x_i^*)^{a_{1jk}} = \frac{w_{jk}^*}{\lambda_k(w^*)} \quad j=1, \ldots, \beta(k) \quad k=1, \ldots, N_0 \]

However,

\[ \sum_{j=1}^{N} (x_i^*)^{a_{1jk}} > 0, \]

hence (3.2.10) is valid only if all \( w_{jk}^* > 0 \). When all \( w_{jk}^* > 0 \), the GP problem is said to be canonical, otherwise it is defined to be degenerate.

Many additional concepts concerning posynoimal GP can be found in Ref. 2. A computer program, named GEOPT, used for solution of the dual (DGP) was obtained from J.N. Sidall (Ref. 3) and a number of small problems were solved efficiently. However, as the degree of difficulty increased beyond five or six, convergence of GEOPT to the known optimum solution was difficult to attain and generally required a considerable amount of computer time. With such unfavorable results, it became apparent that large systems, such as those of Chapter 2, can be solved efficiently by GP techniques only if degree of difficulty is reduced to the lowest possible level, preferably zero. The remaining sections of this chapter will concentrate on various techniques that will approx-
imate signomial problems by posynomial GP with low degree of difficulty so that the dual problem may be solved efficiently. Otherwise, a primal algorithm described in Sec. 3.7 utilizing linear programming will be much more efficient.

Section 3.3 Signomials

Although a few practical engineering problems can be solved by posynomial GP, the majority of mathematical models will contain both positive and negative \( c_{jk} \) coefficients. Hence, considerable work has been directed to possible extensions of GP to include signomial terms.

Primal Extended Geometric Programming (PEGP) refers to problems of the form

\[
\text{(PEGP)} \quad \text{Min } \left[ g_0(x) \right] \quad \text{(3.3.1)}
\]

subject to:

\[
g_k(x) \leq 0 \quad k = 1, \ldots, M_0 \quad \text{(3.3.2)}
\]

\[ x > 0, \]

where

\[
g_k(x) = \sum_{j=1}^{J(k)} \bar{c}_{jk} \prod_{i=1}^{N} \bar{a}_{ijk} x_i^{\bar{a}_{ijk}} \quad k = 0, 1, \ldots, M_0 \quad \text{(3.3.3)}
\]

\( J(k) = \) total number of signomial terms in \( k \) th equation.

In PEGP, \( \bar{a}_{ijk} \) and \( \bar{c}_{jk} \) are restricted to the space of all real numbers. Equality constraints
can be transformed into two inequality constraints of the form

\[ h_k(x) \leq 1 \quad (3.3.5) \]

and

\[ \frac{1}{h_k(x)} \leq 1. \quad (3.3.6) \]

Such transformations of equality constraints are impractical from the viewpoint of reducing the degree of difficulty, hence it will be shown that each equality constraint can be approximated by a single term inequality constraint. Signomial inequality constraints may be transformed to posynomial inequality constraints by the following technique.

Section 3.4 Signomial Inequality Constraints

A useful technique for transformation of signomial inequality constraints to posynomial type was developed by Avriel and Williams (Ref. 4). This technique utilizes the cornerstone of GP, namely the inequality relating geometric and arithmetic mean. Briefly, this inequality states that for real numbers, \( d_i \) and \( e_i \), such that,

\[ e_i \geq 0 \quad i = 1, \ldots, M \]

\[ \sum_{i=1}^{M} e_i = 1 \]
Then,

\[ \sum_{i=1}^{M} d_i \geq \prod_{i=1}^{M} \left( \frac{d_i}{e_i} \right)^{e_i} \quad (3.4.1) \]

where

\[ \lim_{e_i \to 0^+} (e_i e_i) = 1. \]

Equality in (3.4.1) will be achieved if and only if there exists a constant, \( C \), such that

\[ \frac{d_i}{e_i} = C \quad \text{for} \quad i=1, \ldots, M. \]

From equations of Chapter 2, it can be seen that every signomial inequality constraint can be written as the difference of two posynomials. For example, let the \( k \) th signomial inequality constraint be defined as

\[ g_k(x) = \sum_{j=1}^{J(k)} \left[ \overline{c}_{jk} \prod_{i=1}^{N} (x_i)^{\overline{a}_{ijk}} \right] \leq 0, \quad (3.4.2) \]

where \( \overline{c}_{jk} \) may be any real number except zero. This last equation may be rewritten as

\[ g_k(x) = u_k(x) - v_k(x) \leq 0. \quad (3.4.3) \]
$U_k(X)$ and $V_k(X)$ are posynomials given by

$$U_k(X) = \sum_{j=1}^{\phi(k)} \left[ c_{jk} \prod_{i=1}^{N} (x_i) a_{ijk} \right]$$

$$V_k(X) = \sum_{j=1}^{\psi(k)} \left[ d_{jk} \prod_{i=1}^{N} (x_i) b_{ijk} \right]$$

The relationship between $a_{ijk}$ and $c_{jk}$ of (3.4.2) and $a_{ijk}$, $b_{ijk}$, $c_{jk}$, and $d_{jk}$ of (3.4.4) is self evident.

From (3.4.3), conclude that since $V_k(X)$ is a posynomial, then $V_k(X) \geq 0$, hence

$$\frac{U_k(X)}{V_k(X)} \leq 1.$$  \hspace{1cm} (3.4.5)

Since $U_k(X)$ is a posynomial with $\phi(k)$ terms, if $V_k(X)$ can be approximated at $\bar{X}$ by a single term posynomial $V_k(X, \bar{X})$, then $U_k(X)/V_k(X, \bar{X})$ will be a posynomial with $\phi(k)$ terms. The $N$ dimensional vector, $\bar{X}$, can be any feasible solution for inequality constraints, although for large problems with many inequality constraints, such feasible solutions may not be readily obtained without a Phase I procedure.

The posynomial approximation $V_k(X, \bar{X})$ of $V_k(X)$ at $\bar{X}$ is obtained via arithmetic-geometric inequality.

Define $v_{jk}(X)$ as
For a fixed \( k \), let

\[
  v_k(x) = \sum_{j=1}^{\Psi(k)} v_{jk}(x) = \sum_{j=1}^{\Psi(k)} \left[ d_{jk} \prod_{i=1}^{N} (x_i)^{b_{ijk}} \right] \tag{3.4.6}
\]

\[
  d_j = v_{jk}(x) \tag{3.4.7}
\]

\[
  e_j = \frac{v_{jk}(x)}{v_k(x)} \tag{3.4.8}
\]

From the inequality given by Eq. 3.4.1 for all finite \( x > 0 \) and \( \overline{x} > 0 \) regardless of feasibility,

\[
  v_k(x) = \sum_{j=1}^{\Psi(k)} v_{jk}(x) \geq \prod_{j=1}^{\Psi(k)} \left\{ \frac{v_{jk}(x)}{v_k(x)} \frac{v_{jk}(\overline{x})}{v_k(\overline{x})} \right\} \tag{3.4.9}
\]

Substitution from (3.4.6) for \( v_{jk}(x) \) and \( v_{jk}(\overline{x}) \) within parenthesis of the last equation,

\[
  v_k(x) \geq v_k(\overline{x}) \prod_{j=1}^{\Psi(k)} \left[ \frac{\prod_{i=1}^{N} (x_i)^{b_{ijk}}}{\prod_{i=1}^{N} (\overline{x}_i)^{b_{ijk}}} \right] \frac{v_{jk}(\overline{x})}{v_k(\overline{x})} \tag{3.4.10}
\]

Further simplification of the last equation yields the following result.

\[
  v_k(x) \geq v_k(\overline{x}) \prod_{i=1}^{N} \left( \frac{x_i}{\overline{x}_i} \right) \sum_{j=1}^{\Psi(k)} \left[ \frac{b_{ijk} v_{jk}(\overline{x})}{v_k(\overline{x})} \right] = v_k(x, \overline{x})
\]
Note that $V_k(X,X)$ is a single term posynomial and the inequality expressed by the last equation is valid for all finite $X > 0$ and $\bar{X} > 0$. If and only if $X = \bar{X}$, then

$$V_k(X) = V_k(X, \bar{X}).$$

In the trivial case when $V_k(X)$ is a single term posynomial, (3.4.10) readily shows that for all $\bar{X}$

$$V_k(X) = V_k(X, \bar{X}). \quad (3.4.11)$$

From inequalities of (3.4.5) and (3.4.10),

$$\frac{U_k(X)}{V_k(X)} \leq \frac{U_k(X)}{V_k(X, \bar{X})}. \quad (3.4.12)$$

Since equality in this last equation will hold if and only if $X = \bar{X}$, define the following,

$$G_k(X, \bar{X}) = \frac{U_k(X)}{V_k(X, \bar{X})} \leq 1 \quad k=1, \ldots, M_0 \quad (3.4.12)$$

From the last two constraint equations for all $X$ and $\bar{X}$,

$$\frac{U_k(X)}{V_k(X)} \leq \frac{U_k(X)}{V_k(X, \bar{X})} = G_k(X, \bar{X}) \leq 1. \quad (3.4.13)$$

Hence, all signomial inequality constraints of the form $U_k(X)/V_k(X) \leq 1$ will be replaced at $\bar{X}$ by the approximate posynomial inequality $U_k(X)/V_k(X, \bar{X}) \leq 1$ and as $X \to \bar{X}$,

$$U_k(X)/V_k(X, \bar{X}) \to U_k(X)/V_k(X).$$
The number of posynomial terms in each approximate inequality constraint, \( \frac{U_k(X)}{V_k(X,X)} \leq 1 \), will be equal to the number of posynomial terms in \( U_k(X) \) designated as \( \phi(k) \) in (3.4.4) regardless of the number of posynomial terms in \( V_k(X) \). This lack of dependency on the number of \( V_k(X) \) terms is extremely useful in decreasing computational work since the ultimate effect will be a lower degree of difficulty in the solution of the dual. For example, it will be shown in the next section that for purposes of calculation every equality constraint can be transformed to a single term inequality constraint. Hence, the inequality

\[ \frac{U_k(X)}{V_k(X,X)} \leq 1 \quad \text{(3.4.14)} \]

may be reduced to a single term inequality constraint and an equality constraint by introduction of an additional variable, \( y > 0 \), such that

\[ \frac{y}{V_k(X,X)} \leq 1 \]

and

\[ y = U_k(X). \quad \text{(3.4.15)} \]

Equations given in (3.4.15) are equivalent to the constraint in (3.4.14). Computationally, however, the latter contains only two posynomial terms after appropriate
transformation of the equality constraint while the former has $\phi(k)$ posynomial terms. These and similar techniques can be extremely useful in simplifying inequality constraints and reducing the degree of difficulty. However, equality constraints, to be considered next, provide even greater simplification.

Section 3.5 Equality Constraints

The techniques developed in the last section will now be extended to equality constraints. Proofs for convergence to local optimal solutions have not been derived; however, a brief correlation with theoretical results given in Ref. 4 will be included at the end of Chapter 5. The best proof (from the engineering viewpoint) for the following results lies in effective solution of large problems defined in Chapter 2.

Consider the equality constraints defined by (3.3.4) consisting of signomial terms. As in the case of inequality constraints (see (3.4.3)), every equality constraint can be written as

$$h_k(X) = P_k(X) - Q_k(X) = 0.$$  \hspace{1cm} (3.5.1)

$P_k(X)$ and $Q_k(X)$ are posynomials, hence

$$\frac{P_k(X)}{Q_k(X)} = 1.$$  \hspace{1cm} (3.5.2)

Define $Q_k(X,\overline{X})$ similarly to $V_k(X,\overline{X})$ of (3.4.10), then
\[ Q_k(x) \geq Q_k(\overline{x}) \frac{N}{1} \left[ \sum_{i=1}^{\phi(k)} \left( \frac{x_i}{x_i} \right) \left\{ b_{i,j,k} q_{i,j,k}(\overline{x}) \right\} \right] \]

Combining (3.5.2) and (3.5.3),

\[ 1 = \frac{P_k(x)}{Q_k(x)} \leq \frac{P_k(x)}{Q_k(x, \overline{x})} \]

Therefore,

\[ \frac{Q_k(x, \overline{x})}{P_k(x)} \leq 1. \] (3.5.4)

Corresponding to \( Q_k(x, \overline{x}) \), \( P_k(x, \overline{x}) \) may be defined as

\[ P_k(x) \geq P_k(\overline{x}) \frac{N}{1} \left[ \sum_{i=1}^{\phi(k)} \left( \frac{x_i}{x_i} \right) \left\{ a_{i,j,k} p_{i,j,k}(\overline{x}) \right\} \right] \]

\[ = P_k(x, \overline{x}) \] (3.5.5)

From (3.5.4) and (3.5.5), define \( G_k(x, \overline{x}) \) as

\[ \frac{Q_k(x, \overline{x})}{P_k(x)} \leq \frac{Q_k(x, \overline{x})}{P_k(x, \overline{x})} = G_k(x, \overline{x}) \leq 1. \] (3.5.6)

Note that by constructing \( G_k(x, \overline{x}) \leq 1 \), the inequality in (3.5.4) is automatically satisfied. Equality will hold in (3.5.6) if and only if \( x = \overline{x} \). Thus, if con-
vergence to optimal solution occurs, that is, if
\[ x \rightarrow \bar{x} \rightarrow x^*, \]
then
\[ h_k(x) \rightarrow h_k(\bar{x}) \rightarrow h_k(x^*) = 0. \]

In summary, equality constraints of the form
\[ P_k(x) - Q_k(x) = 0 \]
will be replaced by a single term inequality constraint,
\[ G_k(x, \bar{x}) = \frac{Q_k(x, \bar{x})}{P_k(x, \bar{x})} \leq 1. \quad (3.5.7) \]

Since \( Q_k(x, \bar{x}) \) and \( P_k(x, \bar{x}) \) are both single term posynomials, then \( G_k(x, \bar{x}) \) in (3.5.7) gives a single term posynomial inequality constraint as an approximation to the corresponding equality constraint at \( x = \bar{x} \). Hence, each equality constraint \( P_k(x) - Q_k(x) = 0 \) will increase the degree of difficulty by only one when solving the dual problem. This observation illustrates one reason why the transformation given in (3.4.15) is highly useful. The results derived in the last two sections will be combined in the following sections in development of two alternative algorithms.
Section 3.6  Algorithm DUALGP for Solution of Dual

A general primal extended geometric programming (PEGP) problem consisting of signomial terms is defined by

\[(\text{PEGP}) \quad \text{Min} \ (g_0(X)) \]

subject to:
\[g_k(X) \leq 0 \quad k=1, \ldots, M_0 \quad (3.6.1)\]
\[g_k(X) = 0 \quad k=M_0+1, \ldots, (M_0+M_1=M_2)\]
\[X > 0 \]

and
\[g_k(X) = \sum_{j=1}^{\phi(k)} c_{jk} \prod_{i=1}^{N} (x_i)^{a_{ijk}} \]
\[\Psi(k) = \sum_{j=1}^{\psi(k)} d_{jk} \prod_{i=1}^{N} (x_i)^{b_{ijk}} \quad k=0, 1, \ldots, M_2 . \]

Without loss of generality, \( g_o(X) \) is assumed to be a posynomial. If \( g_o(X) \) is not a posynomial, then

\[Z_1 = \text{Min} \ (g_o(X)) \iff \text{Min} \ (y) \]
\[X > 0 \quad g_o(X) \leq y \quad (3.6.2)\]
\[y > 0; \quad X > 0. \]

Likewise,

\[Z_1 = \text{Min} \ (g_o(X)) \iff \text{Min} \ (y) \]
\[X > 0 \quad g_o(X) = y \quad (3.6.3)\]
\[y > 0; \quad X > 0. \]
If \( Z_1 \leq 0 \) in (3.6.2) and (3.6.3), assume a sufficiently large constant, \( C \), such that for all feasible \( X \), \( g_0(X) + C > 0 \) and define an equivalent objective function

\[
Z_1 = \min (g_0(X)) = \left[ -C + \min (g_0(X) + C) \right]
\]

For a feasible \( \bar{X} \) using (3.4.13) and (3.5.6), the optimization problem given by (3.6.1) may be replaced by a Primal Complementary Geometric Program (PCGP).

\[
\text{(PCGP) } \min (g_0(X)) \quad \text{subject to:} \\
G_K(X, \bar{X}) \leq 1 \quad k=1, \ldots, M_2 \quad (3.6.4) \\
X > 0; \quad \bar{X} > 0
\]

where

\[
G_0(X, \bar{X}) = g_0(X, \bar{X}) \quad (3.6.5)
\]

\[
G_k(X, \bar{X}) = \sum_{j=1}^{\Theta(k)} C_{jk} \prod_{i=1}^{N} (X_i)^{e_{ijk}} \quad k=0, 1, \ldots, M_2 \quad (3.6.6)
\]

\[
\Theta(k) = \begin{cases} 
\emptyset(k), & k=0 \\
\emptyset(k), & k=1, \ldots, M_0 \\
1, & k=M_0+1, \ldots, M_2
\end{cases} \quad (3.6.7)
\]

\( C_{jk} \) and \( e_{ijk} \) are all functions of \( \bar{X} \) and were defined in the last two sections. Since all \( C_{jk} \) are positive constants for a given \( \bar{X} \) vector, the optimization problem PCGP defined by (3.6.4) is of the form PGP defined in Section 3.2, hence, there exists a corresponding
dual program DCGP.

\[
(DCGP) \quad v(w) = \max \left\{ M_2 \left[ \frac{\theta(k)}{w_{jk}} \left( \frac{c_{ijk}}{w_{jk}} \right)^{w_{jk}} \right] \lambda_k \right\} \quad (3.6.8)
\]

subject to:

\[
w_{jk} \geq 0 \quad (3.6.9)
\]

\[
\lambda_0 = 1 \quad (3.6.10)
\]

\[
M_2 \sum_{k=0}^{M_2} \sum_{j=1}^{\theta(k)} (a_{ijk}w_{jk}) = 0 \quad i=1, \ldots, N \quad (3.6.11)
\]

where

\[
\lambda_k = \sum_{j=1}^{\theta(k)} w_{jk} \quad k=0, 1, \ldots, M_2
\]

In solving the dual problem, the degree of difficulty, DOD, for DCGP is given by

\[
DOD = \sum_{k=0}^{M_0} \theta(k) + M_1 - (N+1). \quad (3.6.12)
\]

The numerical value for DOD can be varied considerably through transformation of the objective function and inequality constraints. For example, consider \( k \) th inequality constraint in \( (3.6.4) \) with \( \theta(k) \) posynomial terms and define \( g_{jk}(x, \bar{x}) \) as follows.
\[ G_k(X, \bar{X}) = \sum_{j=1}^{\Theta(k)} g_{jk}(X, \bar{X}) \leq 1 \]  \hspace{1cm} (3.6.13)

\( G_k(X, \bar{X}) \leq 1 \) can be replaced by inequality and equality constraints of the form

\[ \sum_{j=1}^{L(k)} g_{jk}(X, \bar{X}) + y \leq 1 \]  \hspace{1cm} (3.6.14)

\[ y - \sum_{j=L(k)+1}^{\Theta(k)} g_{jk}(X, \bar{X}) = 0 \]

where

\[ 0 < L(k) < \Theta(k) \]

\( y = \) new primal variable; \( y > 0 \).

The original constraint in (3.6.13) increases the degree of difficulty by a value of \( \Theta(k) \); the equivalent constraints of (3.6.14) increase degree of difficulty by only \( L(k) + 2 \). This flexibility allows for the reduction of many problems to a zero or a low degree of difficulty, thus allowing for an efficient solution of the dual problem. In torsional problems of Chapter 2, many inequality constraints contain only a single term in PCGP, hence, only the objective function can be transformed in such a manner to allow for a reduction in degree of difficulty. Accordingly, many problems of this type may be quite complex and difficult to solve by
Algorithm DUALGP. For such problems, an efficient algorithm utilizing linear programming will be given in the next section.

A brief summary of the dual algorithm named DUALGP will now be given. The names of the subroutines used to perform the indicated operations are also included.

**Algorithm DUALGP**

1) Read data for objective function using SUBROUTINE READOB.

2) Read data for inequality constraints using SUBROUTINE READLE.

3) Read data for equality constraints using SUBROUTINE READEQ.

4) Choose $X = X^{(1)}$ as any feasible solution for inequality constraints as indicated in Section 3.4.

5) For $i$ th iteration use $X = X^{(i)}$ and transform signomial inequality constraints to posynomial inequality constraints using SUBROUTINE EXPGLE.

6) Using $X = X^{(i)}$, transform signomial equality constraints to single term posynomial inequality constraints via SUBROUTINE EXPGEX.

7a) Steps 5 and 6 define the Primal Complementary Geometric Program (PCGP) and a corresponding dual program DCGP. SUBROUTINE GEOM is used to solve DCGP yielding optimal solution $X^*(i)$.

7b) If some dual variables $W_{ij} = 0$ (i.e., a degenerate case), use SUBROUTINE GEOCOR to remove the corresponding inactive constraints and reorganize the remaining active constraints into a canonical problem. Use SUBROUTINE GEOM to obtain optimal solution $X^*(i)$.

8) If $|x_j^{*(i)} - \bar{X}_j^{(i)}| < \varepsilon x_j^{*(i)}$, $j=1, \ldots, N$.

Optimal solution attained. STOP.

$\varepsilon$ = allowable error.
9) If \( |x_j^{*}(1) - \bar{x}_j^{(1)}| \geq \varepsilon x_j^{*}(1) \) for any \( j=1,\ldots,N \), set \( \bar{x}^{(i+1)} = x^{*}(1) \) and repeat steps 5 through 9.

As stated previously, the efficiency of this algorithm depends primarily upon the degree of difficulty of the canonical problem solved by SUBROUTINE GEOM. In large problems, it is not always possible to reduce the degree of difficulty to zero or a low value, therefore, an alternate algorithm utilizing linear programming will be described in the following section.

Section 3.7 Extended Geometric Programming Using LP

Algorithm EGPLP

Algorithm EGPLP can be utilized in the solution of many large engineering problems with considerable efficiency and simplicity. Since both the primal and dual problems can be solved by essentially the same procedure, a choice can be made for each problem as to the technique (primal vs. dual) which may require the least computer time.

The primary reason for the development of this algorithm may be found in the rather useful and well-known observation that if the objective function and all constraints contain a single posynomial term, then both the primal and the dual can be transformed into a Linear Programming (LP) problem. Hence, the key to the solution of signomial problems given in (3.6.1) by LP methods lies in the possibility of approximation of signomials.
by single term posynomials. The groundwork for such approximations was developed in the previous sections of this chapter and will not be presently repeated. For the sake of quick reference, primal problem \( \text{PEGP} \) given by (3.6.1) is here repeated as (3.7.1).

\[
(\text{PEGP}) \quad \text{Min } (g_0(X)) \\
\text{subject to:} \\
g_k(X) \leq 0 \quad k=1,\ldots,M \quad (3.7.1) \\
g_k(X) = 0 \quad k=M+1,\ldots,(M+M=M_2) \\
X > 0
\]

and

\[
g_k(X) = \sum_{j=1}^{\Theta(k)} c_{jk} \prod_{i=1}^{N} (X_i)^{a_{ijk}} - \sum_{j=1}^{\Sigma(k)} d_{jk} \prod_{i=1}^{N} (X_i)^{b_{ijk}}
\]

\(c_{jk}\) and \(d_{jk}\) are positive real numbers.

Without loss of generality, \(g_0(X)\) is assumed to be a single term posynomial (see (3.6.2)). In Section 3.5, it was shown that for a given \(X\) vector, each equality constraint may be replaced by an approximate single term posynomial constraint \(g_K(X) \leq 1\) (see (3.5.6)). Hence, only the inequality constraints in (3.7.1) need to be reduced to single term posynomials.
For the $k$th inequality constraint from Section 3.4,

$$
\varepsilon_k(x) = u_k(x) - v_k(x) \leq 0 \tag{3.7.2}
$$

$$
u_k(x) = \phi(k) \sum_{j=1}^{N} c_{jk} \prod_{i=1}^{M} (x_i)^{a_{ijk}} \quad k=1, \ldots, M_0 \tag{3.7.3}
$$

If $u_k(x)$ has only a single posynomial term, i.e., if $\phi(k) = 1$, then the approximate posynomial

$$
G_k(x, \overline{x}) = u_k(x)/v_k(x, \overline{x}) \leq 1
$$

will be a single term posynomial as desired. When $\phi(k) > 1$, define a new variable $y_k > 0$ such that

$$
y_k = u_k(x). \tag{3.7.4}
$$

Therefore, every inequality constraint $u_k(x) - v_k(x) \leq 0$, with $\phi(k) > 1$ may be replaced by

$$
\frac{y_k}{v_k(x, \overline{x})} \leq 1 \tag{3.7.5}
$$

$$
y_k - u_k(x) = 0 \tag{3.7.6}
$$

Note that each of these last two constraints can be approximated by a single term posynomial. The introduction of an additional variable is the price paid for simplification of the inequality constraint; in general, this procedure appears to be beneficial. Assume that $M_3$ is the number of inequality constraints with $\phi(k) > 1$. The optimization problem PEGP defined in (3.7.1) may be replaced for a given $\overline{x}$ by the following
set of single term posynomials.

\[(\text{PSGP}) \quad \text{Min} \ (G_o(x, \overline{x})) \quad \text{(3.7.7)}\]

subject to: \[G_k(x, \overline{x}) \leq 1, \ k = 1, \ldots, M_4\]

\[G_k(x, \overline{x}) = \sum_{i=1}^{NN} c_{ik} x_i^{e_{ik}} \quad k = 0, \ldots, M_4 \quad \text{(3.7.8)}\]

\[\overline{x} = (\overline{x}_1, \ldots, \overline{x}_N) > 0\]

\[x = (x_1, \ldots, x_N, x_{N+1}, \ldots, x_{NN}) > 0\]

\[NN = N + M_3\]

\[M_4 = M_2 + M_3\]

The variables \((x_{N+1}, \ldots, x_{NN})\) must be introduced as indicated in (3.7.4). \(c_k\) and \(e_{ik}\) are constants dependent upon the \(\overline{x}\) vector. To transform the posynomial program PSGP into a linear programming problem, only a simple variable transformation is required. Since most engineering problems contain variables with upper and/or lower bounds (in addition to positivity condition \(x > 0\)), the following definitions will satisfy one of the bounds. Let \(R = (R_1, \ldots, R_{NN})\) be the upper or lower bounds on primal variables \(x\) and define

\[x_1 = \begin{cases} 1, & \text{if } R_1 \text{ is lower bound} \\ -1, & \text{if } R_1 \text{ is upper bound} \end{cases} \quad \text{(3.7.9)}\]

\((R_{N+1}, \ldots, R_{NN})\) should be the lower bounds on \((x_{N+1}, \ldots, x_{NN})\) obtained by consideration of \((R_1, \ldots, R_N)\) and the definition of \((x_1, \ldots, x_N)\) in the following
Let

\[ T = (t_1, \ldots, t_{NN}) \geq 0 \]

be a set of primal auxiliary variables defined by the following exponential equation.

\[ X_i = R_i \exp (r_i t_i) \quad i = 1, \ldots, NN. \]

Substitute for \( X_i \) in (3.7.7), then

\[ Z_0 = \min \left\{ \sum_{i=1}^{NN} C_i (X_i)^{e_{io}} \right\} \]

\[ Z_0 = \min \left\{ \sum_{i=1}^{NN} \left[ R_i \exp(r_i t_i) \right]^{e_{io}} \right\} \]

\[ Z_0 = \left[ \sum_{i=1}^{NN} (R_i)^{e_{io}} \right] \min \left\{ \exp \left( \sum_{i=1}^{NN} r_i e_{io t_i} \right) \right\} \]

\[ e_{io} = 0, \quad i = N+1, \ldots, NN \]

The exponential function is a strictly monotone increasing function, hence, the objective function becomes

\[ Z_0 = \min \left\{ G_0(X(T)) \right\} \]

\[ = \sum_{i=1}^{NN} (R_i)^{e_{io}} \exp \left( \min \left( \sum_{i=1}^{NN} r_i e_{io t_i} \right) \right) \]  (3.7.10)

Note that the function to be minimized is linear with
respect to primal auxiliary variables \( T \). Next, consider the single term posynomial constraints in (3.7.8).

\[
G_k(X) = \prod_{i=1}^{NN} (X_i)_{eik} \leq 1 \quad k = 1, \ldots, M_4
\]

\[
G_k(T) = \exp \left\{ \sum_{i=1}^{NN} (r_i e_{ik} t_i) + \ln \left[ \prod_{i=1}^{NN} \left( R_i \right)_{eik} \right] \right\} \leq 1
\]

\( k = 1, \ldots, M_4 \).

Due to monotonicity of exponential function, the last inequality will be satisfied if and only if the following set of linear inequalities in \( T \) variables is satisfied.

\[
\sum_{i=1}^{NN} (r_i e_{ik} t_i) \leq -\ln \left[ \prod_{i=1}^{NN} \left( R_i \right)_{eik} \right] \quad (3.7.11)
\]

\( k = 1, \ldots, M_4 \)

In summary, the primal optimization problem in \( X \) variables given by (3.7.7) and (3.7.8) may be transformed into a primal linear programming problem named PLP in \( T \) variables with the following format.

\[
(PLP) \quad Z_1 = \text{Min} \left[ \sum_{i=1}^{NN} (r_i e_{i0} t_i) \right]
\]

subject to: for \( k = 1, \ldots, M_4 \),

\[
\sum_{i=1}^{NN} (r_i e_{ik} t_i) \leq -\ln \left[ \prod_{i=1}^{NN} \left( R_i \right)_{eik} \right]; \quad T \geq 0 \quad (3.7.12)
\]
Let \( T^* = (t_1^*, \ldots, t_{NN}^*) \) be the optimal solution to (3.7.12), the \( X \) variables and the objective function become

\[
X_i^*(\bar{X}) = R_i \exp(r_i t_i^*) \quad i = 1, \ldots, N \tag*{(3.7.13)}
\]

\[
Z_o^*(\bar{X}) = \left[ \sum_{i=1}^{NN} \frac{C_i}{C_{i1}} (R_i)^{e_{i0}} \right] \exp(Z_1) \tag*{(3.7.14)}
\]

The notation, \( X_i^* = X_i^*(\bar{X}) \) and \( Z_o = Z_o^*(\bar{X}) \), is here used to indicate the dependency of \( X_i^* \) and \( Z_o \) on the chosen \( \bar{X} \) vector due to approximation of signomials by posynomials.

The dual to PIP, designated as DLP, can be derived via the standard geometric dual given in Sec. 3.2 or by utilizing the linear program PIP in (3.7.12). In either procedure, the dual will also be a linear program, however, the formulation of the dual from (3.7.12) is somewhat more direct.

Let \( W = (W_1, \ldots, W_{M_4}) \geq 0 \) be the dual variables. The dual program named DLP is of the form

\[
\text{(DLP)} \quad v(W) = \max \left[ \sum_{k=1}^{M_4} \left\{ \ln \left[ \sum_{i=1}^{NN} (R_i)^{e_{ik}} \right] w_k \right\} \right]
\]

subject to:

\[
\sum_{k=1}^{M_4} (-e_{ik} w_k) \leq e_{i0} \quad i = 1, \ldots, NN
\]

\( W \geq 0 \)

The solution of the dual program DLP may perhaps be more
efficient in terms of computer time since the number of primal constraints in PLP is greater than the number of dual constraints in DLP, i.e., $M_d > N_d$. If the dual, rather than the primal, program is solved, the optimal primal variables $X^* = X^*(\bar{X})$ may be obtained by the Duality Theorem of Linear Programming.

A brief summary of Algorithm EGPLP is now included for the sake of completeness. Revised simplex method is used in the solution of linear programming problems. The names of the subroutines used to perform the indicated operations are also included.

**ALGORITHM EGPLP**

1) Rewrite objective function and inequality constraints and define $(X_{N+1}, \ldots, X_{NN})$ as indicated at the beginning of this section.

2) Read $r$ vector as defined in (3.7.9) and $R$ vector (upper or lower bounds) in MAIN PROGRAM.

3) Read data for objective function using SUBROUTINE READOB.

4) Read data for inequality constraints using SUBROUTINE READLE.

5) Read data for equality constraints using SUBROUTINE READEQ.

6) Choose $\bar{X}^{(1)} = (\bar{X}_1^{(1)}, \ldots, \bar{X}_N^{(1)})$ as any feasible solution for inequality constraints as indicated in Section 3.4.

7) For $k$ th iteration, use $\bar{X} = \bar{X}^{(k)}$ and transform signomial inequality constraints into single term posynomial inequality constraints via SUBROUTINE EXPGLP.
8) Using $\bar{X} = X^{(k)}$, transform signomial equality constraints into single term posynomial inequality constraint via SUBROUTINE EXPGEX.

9) Steps 7 and 8 define primal problem PSGP in (3.7.7) and (3.7.8). Use SUBROUTINE LPFOEF to determine LP coefficients given by (3.7.12) or (3.7.15).

10) Use SUBROUTINE SIMPLE to solve primal problem PIP or dual DIP and obtain optimal solution, $T^*(k) = (t_1^*(k), \ldots, t_N^*(k))$.

11) Use (3.7.13) and (3.7.14) to obtain $Z_0$ and $(X_1^*(k), \ldots, X_N^*(k))$. The variables, $(X_{N+1}^*(k), \ldots, X_{NN}^*(k))$, need not be defined.

12) If $|X_j^*(k) - \bar{X}_j^*(k)| < \epsilon |X_j^*(k)|$, for all $j=1,\ldots,N$, optimal solution attained. STOP. $\epsilon = $ allowable error.

13) If $|X_j^*(k) - \bar{X}_j^*(k)| \geq \epsilon |X_j^*(k)|$ for any $j=1,\ldots,N$, set $\bar{X}_j^{(k+1)} = X_j^*(k)$ for all $j=1,\ldots,N$. Repeat steps 7 to 13.

The ultimate advantage of this algorithm lies in the usage of LP within each iteration to find optimal solution. Since LP may be utilized for extremely large problems, this algorithm places no restriction on the number of constraints and variables (computer storage and time do restrict problem size). Hence, the problems described in Chapter 2 may be solved by this last algorithm with considerable efficiency as indicated in the following chapter.
In this chapter, a local optimal solution will be found for the three problems described. Each problem illustrates a particular aspect of nonlinear optimization in order to provide a deeper understanding of the iteration technique utilized by Algorithm EGPLP of Sec. 3.7. Algorithm DUALGP of Sec. 3.6 could be used to solve the first problem since degree of difficulty $DOD = 2$ for PROBLEM A. However, DUALGP will not solve efficiently PROBLEMS B and C due to large DOD values. Hence, for comparison purposes, all problems will be solved via Algorithm EGPLP. PROBLEM A will now illustrate the procedure for analysis by geometric programming of optimization problems with unsigned variables, i.e., $X_j$ may have positive and negative values.
Section 4.1 Description and Solution of PROBLEM A

Applied torsional loads are defined below.

\[ M_1 = -u_1 \phi_1 \quad M_2 = -u_2 \phi_2 \quad M_3 = -u_3 \phi_3 \]

**FIGURE 4.1.1**

Fig. 4.1.1 illustrates a non-uniform circular bar with three elastic supports subjected to torsional loads. Vector notation is used to denote all moments and support rotations. Define the following terms:

- \( u_i \) = moment per unit rotation at elastic support \( i \).
- \( \phi_i \) = rotation of support \( i \).
- \( a_i \) = radius of first member.
- \( a_2 \) = radius of second member.
- \( l_i \) = length of member \( i \).
- \( J_i \) = polar moment of inertia for member \( i \).
- \( T_0 \) = applied concentrated torque at center of \( 1 \).
- \( m_2 \) = uniform torsional load on \( 2 \).
From equilibrium equation for each support,

\[-u_1 \phi_1 + T_{11} = 0 \]
\[-u_2 \phi_2 + T_{22} - T_{21} = 0 \]
\[-u_3 \phi_3 - T_{32} = 0 \]

Let \( \phi_{11} \) = rotation per unit torque applied at the ends of the torsional member.

\[ \phi_{11} = \frac{2l_1}{\pi G a_1} \]
\[ \phi_{22} = \frac{2l_2}{\pi G a_2} \]

\( T^f_{ki} \) = fixed torsional moment at support \( k \) due to all torsional loads in member \( i \). The inter-relationship between torsional moments and rotations are as follows.

\[ T_{11} = T^f_{11} - \frac{\phi_1}{\phi_{11}} + \frac{\phi_2}{\phi_{11}} \]
\[ T_{21} = T^f_{21} - \frac{\phi_1}{\phi_{11}} + \frac{\phi_2}{\phi_{11}} \]
\[ T_{22} = T^f_{22} - \frac{\phi_2}{\phi_{22}} + \frac{\phi_3}{\phi_{22}} \]
\[ T_{32} = T^f_{32} - \frac{\phi_2}{\phi_{22}} + \frac{\phi_3}{\phi_{22}} \]

To avoid plasticity, the following yield constraints will be required (Mises-Hencky yield criterion will again be utilized). For the torsional loads applied in
this problem, the maximum shear stresses will occur near the supports, hence, the yielding constraints are given by

\[ |\tau| = \frac{|a_\tau|}{J} \leq \frac{f_Y}{\sqrt{3}}. \]

The absolute value sign in the last equation may be removed by the following procedure. On the left and right sides of the second support, yield constraints are given by the following set of equations.

\[
\begin{align*}
\frac{a_{11}^T_{21}}{J_1} &\leq \frac{f_Y}{\sqrt{3}} \\
\frac{-a_{11}^T_{21}}{J_1} &\leq \frac{f_Y}{\sqrt{3}} \\
\frac{a_{22}^T_{22}}{J_2} &\leq \frac{f_Y}{\sqrt{3}} \\
\frac{-a_{22}^T_{22}}{J_2} &\leq \frac{f_Y}{\sqrt{3}}
\end{align*}
\] (4.1.9)

For the torsional loads considered in this problem, \( T_{11} \) will always be positive, hence, the yield constraint near the first support is given by

\[ \frac{a_{11}^T_{11}}{J_1} \leq \frac{f_Y}{\sqrt{3}}. \] (4.1.11)

Near the third support, \( T_{32} \leq 0 \), thus

\[ \frac{-a_{22}^T_{32}}{J_2} \leq \frac{f_Y}{\sqrt{3}}. \] (4.1.12)

Most of the variables \( T_{ij} \) and \( \phi_i \) may have either positive or negative values, therefore, additional
analysis is necessary. Assume that the following bounds are desirable:

\[ l_1 \geq 10 \text{ in.} \quad 1_2 \geq 15 \text{ in.} \]
\[ a_1 \leq 5 \text{ in.} \quad a_2 \leq 3 \text{ in.} \]

From \((4.1.9)\) and \((4.1.13)\) with \(f_y/\sqrt{3} = 20 \text{ ksi, and}
\[ J_1 = \frac{\pi}{2} a_1^4, \]

\[ -T_{21} \leq 20 \left( \frac{J_1}{a_1} \right) = 1250 \text{ in.} \]
\[ T_{21} \geq -1250 \text{ in.} > -3920. \]

Similar inequalities may be obtained for the remaining torsional moments. Although \(T_{11} > 0\), for the sake of uniformity, consider the following:

\[ T_{11} > -3920 \]
\[ T_{21} > -3920 \]
\[ T_{22} > -850 \]
\[ T_{32} > -850 \]

\((4.1.15)\)

Define a new set of positive torsional moments \(S_{ij}\):

\[ S_{11} = T_{11} + 3920 > 0 \]
\[ S_{21} = T_{21} + 3920 > 0 \]
\[ S_{22} = T_{22} + 850 > 0 \]
\[ S_{32} = T_{32} + 850 > 0 \]

\((4.1.16)\)

A similar procedure is employed for rotational variables.
through definition of positive rotational variables \( b_i \).

\[
\begin{align*}
    b_1 &= \phi_1 + 0.315 > 0 \\
    b_2 &= \phi_2 + 0.25 > 0 \\
    b_3 &= \phi_3 + 0.21 > 0
\end{align*}
\]

The inequality constraints are obtained by substitution of (4.1.16) into (4.1.9) - (4.1.12).

\[
\begin{align*}
    S_{11} - (3920 + 31.4 a_1^3) &\leq 0 \\
    S_{21} - (3920 + 31.4 a_1^3) &\leq 0 \\
    3920 - (S_{21} + 31.4 a_1^3) &\leq 0 \\
    S_{22} - (850 + 31.4 a_2^3) &\leq 0 \\
    850 - (S_{22} + 31.4 a_2^3) &\leq 0 \\
    850 - (S_{32} + 31.4 a_2^3) &\leq 0
\end{align*}
\]  
(4.1.17)

Equality constraints given by (4.1.1) - (4.1.3) may now be rewritten as

\[
\begin{align*}
    S_{21} + 850 + u_2 b_2 - (3920 + S_{22} + 0.25 u_2) &= 0 \\
    S_{11} + 0.315 u_1 - (3920 + u_1 b_1) &= 0 \\
    S_{32} + u_3 b_3 - (850 + 0.21 u_3) &= 0
\end{align*}
\]  
(4.1.18)

Similarly, equality constraints defined in (4.1.5) - (4.1.8) may be rewritten as

\[
\begin{align*}
    S_{11} + \frac{\pi G a_1^4}{2l_1} (b_1 - 0.315) - \left[ 3920 + T_{11}^f + \frac{\pi G a_1^4}{2l_1} (b_2 - 0.25) \right] &= 0
\end{align*}
\]  
(4.1.19)
cont.
\[ S_{21} + \frac{\pi G a_1^4}{2L_1} (b_1 - .315) - \left[ 3920 + T_{21}^f + \frac{\pi G a_1^4}{2L_1} (b_2 - .25) \right] = 0 \]

\[ S_{22} + \frac{\pi G a_2^4}{2L_2} (b_2 - .25) - \left[ 850 + T_{22}^f + \frac{\pi G a_2^4}{2L_2} (b_3 - .21) \right] = 0 \]

\[ S_{32} + \frac{\pi G a_2^4}{2L_2} (b_2 - .25) - \left[ 850 + T_{32}^f + \frac{\pi G a_2^4}{2L_2} (b_3 - .21) \right] = 0 \]

Fixed end torsional moments due to applied torsional loads shown in Fig. 4.1.1 are as follows:

\[ T_{11}^f = \frac{T_0}{2} \]
\[ T_{21}^f = -\frac{T_0}{2} \]
\[ T_{22}^f = \frac{m_2 l_2}{2} \]
\[ T_{32}^f = -\frac{m_2 l_2}{2} \]

For the sake of numerical computation, consider the following data:

\[ G = 12 \times 10^3 \text{ ksi} \]
\[ T_0 = 100 \text{ kip in.} \]
\[ m_2 = 5 \frac{\text{kip in.}}{\text{in.}} \]
\[ u_1 = 800 \frac{\text{kip in.}}{\text{rad.}} \]
\[ u_2 = 1000 \frac{\text{kip in.}}{\text{rad.}} \]
\[ u_3 = 1200 \frac{\text{kip in.}}{\text{rad.}} \]
Furthermore, define the $X$ vector as follows:

\[
\begin{align*}
X_1 &= l_1 & X_7 &= b_3 \\
X_2 &= l_2 & X_8 &= s_{11} \\
X_3 &= a_1 & X_9 &= s_{21} \\
X_4 &= a_2 & X_{10} &= s_{22} \\
X_5 &= b_1 & X_{11} &= s_{32} \\
X_6 &= b_2
\end{align*}
\]  

(4.1.20)

The objective for this problem is to minimize the weight and therefore, the volume of the system. Hence,

\[
Z = \text{Min}(\text{Volume}) = \text{Min} [\prod (X_1X_3^2 + X_2X_4^2)]
\]

Since the objective function must contain only a single posynomial term, define an additional variable $X_{12}$ such that

\[
Z = \text{Min} (X_{12})
\]

subject to:

\[
X_{12} = 3.14(X_1X_3^2 + X_2X_4^2)
\]

Based on the given data, a complete summary of PROBLEM A is now included.

\[
Z = \text{Min} (X_{12})
\]  

(4.1.21)

The inequality constraints are obtained from (3.1.17).
\( g_1(x) = x_8 - \left[ 3920 + 31.4x_3^3 \right] \leq 0 \)
\( g_2(x) = x_9 - \left[ 3920 + 31.4x_3^3 \right] \leq 0 \)
\( g_3(x) = 3920 - \left[ x_9 + 31.4x_3^3 \right] \leq 0 \)
\( g_4(x) = x_{10} - \left[ 850 + 31.4x_4^3 \right] \leq 0 \)
\( g_5(x) = 850 - \left[ x_{10} + 31.4x_4^3 \right] \leq 0 \)
\( g_6(x) = 850 - \left[ x_{11} + 31.4x_4^3 \right] \leq 0 \) (4.1.22)

The equality constraints are obtained from (4.1.18) and (4.1.19).

\( g_7(x) = x_9 + 1000x_6 - \left[ 3320 + x_{10} \right] = 0 \) (4.1.23)
\( g_8(x) = x_8 - \left[ 3668 + 800x_5 \right] = 0 \)
\( g_9(x) = x_{11} + 1200x_7 - 1102 = 0 \)
\( g_{10}(x) = x_8 + 18.8 \times 10^3 \left( \frac{x_4^3 x_5}{x_1} \right) \)
\( - \left[ 3970 + 1.23 \times 10^3 \left( \frac{x_3^4}{x_1} \right) + 18.8 \times 10^3 \left( \frac{x_4^3 x_6}{x_1} \right) \right] = 0 \)
\( g_{11}(x) = \left[ x_9 + 18.8 \times 10^3 \left( \frac{x_2^4 x_5}{x_1} \right) \right] \)
\( - \left[ 3870 + 18.8 \times 10^3 \left( \frac{x_2^4 x_6}{x_1} \right) + 1.22 \times 10^3 \left( \frac{x_2^4}{x_1} \right) \right] = 0 \)
\( g_{12}(x) = \left[ x_{10} + 18.8 \times 10^3 \left( \frac{x_4^4 x_6}{x_2} \right) \right] \)
\( - \left[ 850 + 2.5x_2 + 18.8 \times 10^3 \left( \frac{x_4^4 x_7}{x_2} \right) + 754 \left( \frac{x_4^4}{x_2} \right) \right] = 0 \)
\[ g_{13}(X) = \left[ X_{11} + 18.8 \times 10^3 \left( \frac{x_4 x_6}{x_2} \right) + 2.5x_2 \right] \quad (4.1.23) \]

\[ - \left[ 850 + 18.8 \times 10^3 \left( \frac{x_4 x_7}{x_2} \right) + 754 \left( \frac{x_4^2}{x_2} \right) \right] = 0 \]

\[ g_{14}(X) = x_{12} - \left[ 3.14 x_1 x_3^2 + 3.14 x_2 x_4^2 \right] = 0 \]

The bounds on X variables are denoted by the vector R; the distinction between upper and lower bounds is determined from the r vector as defined in \((3.7.9)\). The starting vector, \(\bar{X}\), must also be defined as indicated in Algorithm EGPLP, Step 6. A practical set of values for W, r, and \(\bar{X}\) vectors is here illustrated:

\[
\begin{align*}
R_1 &= 10 & r_1 &= 1 & \bar{X}_1^{(1)} &= 11. \\
R_2 &= 15 & r_2 &= 1 & \bar{X}_2^{(1)} &= 17. \\
R_3 &= 5 & r_3 &= -1 & \bar{X}_3^{(1)} &= 4. \\
R_4 &= 3 & r_4 &= -1 & \bar{X}_4^{(1)} &= 3. \\
R_5 &= 0.63 & r_5 &= -1 & \bar{X}_5^{(1)} &= 0.4. \\
R_6 &= 0.5 & r_6 &= -1 & \bar{X}_6^{(1)} &= 0.2 \quad (4.1.24) \\
R_7 &= 0.42 & r_7 &= -1 & \bar{X}_7^{(1)} &= 0.2. \\
R_8 &= 7800 & r_8 &= -1 & \bar{X}_8^{(1)} &= 3920. \\
R_9 &= 7800 & r_9 &= -1 & \bar{X}_9^{(1)} &= 3920. \\
R_{10} &= 1700 & r_{10} &= -1 & \bar{X}_{10}^{(1)} &= 850. \\
R_{11} &= 1700 & r_{11} &= -1 & \bar{X}_{11}^{(1)} &= 850. \\
R_{12} &= 1 & r_{12} &= 1 & \bar{X}_{12}^{(1)} &= 5. \\
\end{align*}
\]
The data given in (4.1.21) - (4.1.24) was read into the Algorithm EGILP and seven iterations were required to attain local optimum solution. Only the final solution will now be given. \( X^* \) indicates the values of \( X \) variables at the local optimum solution.

\[
\begin{align*}
X_1^* &= 10.0 \\
X_2^* &= 15.0 \\
X_3^* &= 1.18 \\
X_4^* &= 1.27 \\
X_5^* &= 0.378 \\
X_6^* &= 0.312 \\
X_7^* &= 0.264 \\
X_8^* &= 3.970 \\
X_9^* &= 3.868 \\
X_{10}^* &= 8.604 \\
X_{11}^* &= 7.854 \\
X_{12}^* &= 12.01 \\
Z^* &= \text{Min(Volume)} = 120.1 \text{ in.}^3
\end{align*}
\]

From (4.1.20) and (4.1.25), the locally optimal dimensions for the torsional bars shown in Fig. 4.1.1 are as follows:

\[
\begin{align*}
L_1^* &= 10.0 \text{ in.} \\
L_2^* &= 15.0 \text{ in.} \\
a_1^* &= 1.18 \text{ in.} \\
a_2^* &= 1.27 \text{ in.}
\end{align*}
\]

The moments and rotations developed at supports for this optimal design under the loads specified in Fig. 4.1.1 may also be obtained from (4.1.25) and the previous definitions of the variables.
\( \varphi_1^* = 0.063 \text{ radians} \quad T_{11}^* = 50 \text{ kip in.} \\
\varphi_2^* = 0.062 \text{ radians} \quad T_{21}^* = -52 \text{ kip in.} \\
\varphi_3^* = 0.054 \text{ radians} \quad T_{22}^* = 10.4 \text{ kip in.} \\
T_{32}^* = -64.6 \text{ kip in.} \\

Evaluation of inequality constraints given by (4.1.22) at the optimal solution yields the following results.

\[
\begin{align*}
g_1(x^*) &= -1.7 \leq 0 \\
g_2(x^*) &= -103.7 \leq 0 \\
g_3(x^*) &= 0 \leq 0 \\
g_4(x^*) &= -54.2 \leq 0 \\
g_5(x^*) &= -75.0 \leq 0 \\
g_6(x^*) &= 0 \leq 0
\end{align*}
\]

Since the yield constraints given by \( g_1(x^*) \), \( g_3(x^*) \), and \( g_6(x^*) \) are nearly zero, this implies that the most stressed fibers in both bars are on the verge of yielding at the optimal solution. The solution vector, \( x^* \), also satisfies the equality constraints of (4.1.23), hence, it is indeed a feasible solution. To show that \( x^* \) is a locally optimal solution would require extensive amount of computational work. Hence, as in the case of most nonlinear optimization problems, the local optimality of \( x^* \) is assumed.

**PROBLEM A** was purposely designed unsymmetrically, i.e., all components of the \( x^* \) were of different magnitude.
However, if symmetric solutions are feasible, experience has indicated that the optimal solution will also be symmetric. An efficient algorithm should be able to take advantage of symmetry and thus converge to the locally optimum solution in fewer iterations. The following example, PROBLEM B, was designed symmetrically in order to test the efficiency of Algorithm EGPLP.
Section 4.2 PROBLEM B

This problem studies the optimal design of a multicellular thin-walled, monocoque structure subjected to both torsional and shear loads as indicated in Fig. 4.2.1. The equations derived in Sec. 2.3 are directly applicable for this problem and will be used in the subsequent design.

\[ V_1, V_2 = \text{vertical and horizontal shear loads} \]
\[ T_0 = \text{torsional load} \]

Fig. 4.2.1
From Fig. 4.2.1, the following notation is observed.

\[ X_1, X_2, \ldots, X_{12} \text{ - shear flows} \]
\[ X_{13} \text{ - height and width of all cells} \]
\[ X_{14} \text{ - uniform thickness for all cell walls} \]

The objective function may now be obtained either from (2.3.1) or directly from Fig. 4.2.1.

\[ Z = \min(\text{Cross-sectional area}) \]
\[ Z = \min(36X_{13}X_{14}) \quad (4.2.1) \]

The first inequality constraint is derived from (2.3.7) stating that the maximum allowable torsional load must be greater than or equal to the applied load, \( T_0 \). Therefore,

\[ T_0 - 2X_{13}^2(X_1 + X_2 + \ldots + X_{12}) \leq 0. \quad (4.2.2) \]

Utilizing simple shear formula, the following constraints are also necessary.

\[ V_1 = \tau_v A_v \]
\[ V_2 = \tau_h A_h \quad (4.2.3) \]

Due to symmetry, \( A_v \) and \( A_h \) are defined by the following equations.

\[ A_v = \text{vertical web area} = 18X_{13}X_{14} \]
\[ A_h = \text{horizontal web area} = 18X_{13}X_{14} \]
Assume $f_y/\sqrt{3} = 20$, and define the following:

$$\tau_h = \frac{f_y}{\sqrt{3}} - x_{15} = 20 - x_{15}$$
$$\tau_v = \frac{f_y}{\sqrt{3}} - x_{16} = 20 - x_{16}$$

Substitution in (4.2.3) for $A_v$, $A_h$, $\tau_h$, and $\tau_v$ gives the following equality constraints.

$$V_1 = -18x_{13}x_{14}x_{16} + 360x_{13}x_{14}$$
$$V_2 = -18x_{13}x_{14}x_{15} + 360x_{13}x_{14} \quad (4.2.4)$$

Next, the yielding constraints will be developed for cell 1. From (2.3.5) with inclusion of vertical and horizontal shears $\tau_v$ and $\tau_h$, the following yield constraints are required for cell 1.

$$\frac{x_1}{x_{14}} + \tau_v \leq 20, \text{ thus } \frac{x_1}{x_{14}} - x_{16} \leq 0$$
$$\frac{x_1}{x_{14}} + \tau_h \leq 20, \text{ thus } \frac{x_1}{x_{14}} - x_{15} \leq 0 \quad (4.2.5)$$

$$\frac{(x_2 - x_1)}{x_{14}} + \tau_v \leq 20, \text{ thus } x_2 - (x_1 + x_{14}x_{16}) \leq 0$$

Similar constraints may be developed for the remaining eleven cells; all the yield constraints are summarized in (4.2.7). This completes the set of required inequality constraints. As mentioned in Sec. 2.3, other
inequality constraints may be imposed upon the problem, however, the inclusion of additional constraints may result in a heavier or a more expensive design.

Having completed the development of inequality constraints, our attention must next be focused on equality constraints. In addition to the equality constraints given in (4.2.4), other equality constraints arise from (2.3.10) which defines the equality of angular rotations for all cells. Utilizing (2.3.8), a set of eleven equations may be developed and these are given as $g_{28}(X)$ to $g_{38}(X)$ in (4.2.8). For the sake of quick reference, a complete summary of the derived equations is listed below.

Objective function:

$$ Z = \text{Min}(36X_{13}X_{14}) $$

Summary of inequality constraints:

$$
\begin{align*}
g_1(x) &= t_0 - 2x_{13}^2(x_1 + x_2 + \ldots + x_{12}) \leq 0 \\
g_2(x) &= x_1 - x_{14}x_{15} \leq 0 \\
g_3(x) &= x_1 - x_{14}x_{16} \leq 0 \\
g_4(x) &= x_2 - (x_1 + x_{14}x_{16}) \leq 0 \\
g_5(x) &= x_2 - x_{14}x_{15} \leq 0 \\
g_6(x) &= x_3 - (x_2 + x_{14}x_{16}) \leq 0 \\
g_7(x) &= x_3 - x_{14}x_{15} \leq 0 \\
g_8(x) &= x_4 - (x_3 + x_{14}x_{16}) \leq 0 \\
g_9(x) &= x_4 - x_{14}x_{15} \leq 0
\end{align*}
$$
\[ \varepsilon_{10}(x) = x_5 - (x_4 + x_{14}x_{15}) \leq 0 \]
\[ \varepsilon_{11}(x) = x_5 - x_{14}x_{16} \leq 0 \]
\[ \varepsilon_{12}(x) = x_6 - (x_5 + x_{14}x_{15}) \leq 0 \]
\[ \varepsilon_{13}(x) = x_6 - x_{14}x_{16} \leq 0 \]
\[ \varepsilon_{14}(x) = x_7 - (x_6 + x_{14}x_{15}) \leq 0 \]
\[ \varepsilon_{15}(x) = (x_{15} + \frac{x_7}{x_{14}}) - 40 \leq 0^3 \]
\[ \varepsilon_{16}(x) = (x_{16} + \frac{x_7}{x_{14}}) - 40 \leq 0^3 \]
\[ \varepsilon_{17}(x) = x_7 - (x_8 + x_{14}x_{16}) \leq 0 \]
\[ \varepsilon_{18}(x) = x_8 - x_{14}x_{15} \leq 0 \]
\[ \varepsilon_{19}(x) = x_8 - (x_9 + x_{14}x_{16}) \leq 0 \]
\[ \varepsilon_{20}(x) = x_9 - x_{14}x_{15} \leq 0 \]
\[ \varepsilon_{21}(x) = x_9 - (x_{10} + x_{14}x_{16}) \leq 0 \]
\[ \varepsilon_{22}(x) = x_{10} - x_{14}x_{16} \leq 0 \]
\[ \varepsilon_{23}(x) = x_{10} - (x_{11} + x_{14}x_{15}) \leq 0 \]
\[ \varepsilon_{24}(x) = x_{11} - x_{14}x_{16} \leq 0 \]
\[ \varepsilon_{25}(x) = x_{11} - (x_{12} + x_{14}x_{15}) \leq 0 \]
\[ \varepsilon_{26}(x) = x_{12} - x_{14}x_{16} \leq 0 \]
\[ \varepsilon_{27}(x) = x_{12} - (x_1 + x_{14}x_{15}) \leq 0 \]

Since each of these equations has two positive terms, the procedure defined in Sec. 3.7 must be followed if Algorithm EGPIR is to be used.
Summary of equality constraints:

\[ g_{28}(x) = (5x_1 + x_3) - (5x_2 + x_{12}) = 0 \]  
\[ g_{29}(x) = (4x_1 + x_4) - (x_{12} + 4x_3) = 0 \]  
\[ g_{30}(x) = (4x_1 + x_3 + x_5) - (x_2 + x_{12} + 4x_4) = 0 \]  
\[ g_{31}(x) = (4x_1 + x_4 + x_6) - (x_2 + x_{12} + 4x_5) = 0 \]  
\[ g_{32}(x) = (4x_1 + x_7 + x_5) - (x_2 + x_{12} + 4x_6) = 0 \]  
\[ g_{33}(x) = (4x_1 + x_8 + x_6) - (x_2 + x_{12} + 4x_7) = 0 \]  
\[ g_{34}(x) = (4x_1 + x_9 + x_7) - (x_2 + x_{12} + 4x_8) = 0 \]  
\[ g_{35}(x) = (4x_1 + x_8 + x_{10}) - (x_2 + x_{12} + 4x_9) = 0 \]  
\[ g_{36}(x) = (4x_1 + x_9 + x_{11}) - (x_2 + x_{12} + 4x_{10}) = 0 \]  
\[ g_{37}(x) = (4x_1 + x_{10} + x_{12}) - (x_2 + x_{12} + 4x_{11}) = 0 \]  
\[ g_{38}(x) = (5x_1 + x_{11}) - (x_2 + 5x_{12}) = 0 \]  
\[ g_{39}(x) = (V_1 + 18x_{13}x_{14}x_{16}) - 360x_{13}x_{14} = 0 \]  
\[ g_{40}(x) = (V_2 + 18x_{13}x_{14}x_{15}) - 360x_{13}x_{14} = 0 \]  

This problem consists of sixteen variables, 27 nonlinear inequality constraints, eleven linear equality constraints, and two nonlinear equality constraints. For computational purposes, assume the following data.

\[ T_0 = 1000 \text{ kip in.} \]  
\[ V_1 = 50 \text{ kips} \]  
\[ V_2 = 100 \text{ kips} \]  

The bounded values for \( x \) vector represented by the \( R \) vector, the \( r \) vector, and the initial vector \( \bar{x}^{(1)} \) are listed below. \( \bar{x}_i^{(1)} \), \( i=1,\ldots,12 \), were intentionally assumed to be unequal, although symmetry assures their
equality.

\[
\begin{align*}
R_1 &= 6 & r_1 &= -1 & \bar{x}_1^{(1)} &= 6.0 \\
R_2 &= 6 & r_2 &= -1 & \bar{x}_2^{(1)} &= 5.5 \\
R_3 &= 6 & r_3 &= -1 & \bar{x}_3^{(1)} &= 5.0 \\
R_4 &= 6 & r_4 &= -1 & \bar{x}_4^{(1)} &= 4.5 \\
R_5 &= 6 & r_5 &= -1 & \bar{x}_5^{(1)} &= 4.0 \\
R_6 &= 6 & r_6 &= -1 & \bar{x}_6^{(1)} &= 3.5 \\
R_7 &= 6 & r_7 &= -1 & \bar{x}_7^{(1)} &= 3.7 \\
R_8 &= 6 & r_8 &= -1 & \bar{x}_8^{(1)} &= 4.2 \\
R_9 &= 6 & r_9 &= -1 & \bar{x}_9^{(1)} &= 5.2 \\
R_{10} &= 6 & r_{10} &= -1 & \bar{x}_{10}^{(1)} &= 5.7 \\
R_{11} &= 6 & r_{11} &= -1 & \bar{x}_{11}^{(1)} &= 3.2 \\
R_{12} &= 6 & r_{12} &= -1 & \bar{x}_{12}^{(1)} &= 3.0 \\
R_{13} &= 10 & r_{13} &= 1 & \bar{x}_{13}^{(1)} &= 13. \\
R_{14} &= .05 & r_{14} &= 1 & \bar{x}_{14}^{(1)} &= 0.1 \\
R_{15} &= 20. & r_{15} &= -1 & \bar{x}_{15}^{(1)} &= 15. \\
R_{16} &= 20. & r_{16} &= -1 & \bar{x}_{16}^{(1)} &= 17. \\
\end{align*}
\]

(4.2.10)

The data given in (4.2.6) - (4.2.10) was read into Algorithm EGPLP and only four iterations were required to attain a locally optimum solution, due to intended symmetry of variables, \(X_1, X_2, \ldots, X_{12}\). Only the final solution, i.e., the fourth iteration, will now be summarized. \(X^*\) indicates the locally optimal solution vector.
The minimal cross-sectional area, i.e., the value of the objective function is

\[ Z = 18.0 \text{ in.}^2 \]

The solution vector \( \mathbf{x}^* \) satisfies all equality constraints and, upon evaluation of the inequality constraints, the following values are obtained.

\[
\begin{align*}
\varepsilon_1(\mathbf{x}^*) &= -33.3 \leq 0 \\
\varepsilon_2(\mathbf{x}^*) &= 0 \\
\varepsilon_3(\mathbf{x}^*) &= -5.5 \leq 0 \\
\varepsilon_4(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_5(\mathbf{x}^*) &= 0 \\
\varepsilon_6(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_7(\mathbf{x}^*) &= 0 \\
\varepsilon_8(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_9(\mathbf{x}^*) &= 0 \\
\varepsilon_{10}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{11}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{12}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{13}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{14}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{15}(\mathbf{x}^*) &= -22.2 \leq 0 \\
\varepsilon_{16}(\mathbf{x}^*) &= -16.7 \leq 0 \\
\varepsilon_{17}(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_{18}(\mathbf{x}^*) &= 0 \\
\varepsilon_{19}(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_{20}(\mathbf{x}^*) &= 0 \\
\varepsilon_{21}(\mathbf{x}^*) &= -0.7 \leq 0 \\
\varepsilon_{22}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{23}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{24}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{25}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{26}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{27}(\mathbf{x}^*) &= -0.4 \leq 0 \\
\varepsilon_{28}(\mathbf{x}^*) &= -0.3 \leq 0 \\
\varepsilon_{29}(\mathbf{x}^*) &= -0.4 \leq 0
\end{align*}
\]

Since all constraint equations are satisfied, the \( \mathbf{x}^* \)
vector is a feasible solution to the problem. As indicated previously, no attempt will be made to theoretically verify the optimality of the solutions.

The primary purpose of this problem was to check the efficiency of Algorithm EGPLP when symmetry is present and common values for a set of variables are to be expected. From Fig. 4.2.1 and constraint equations, it is obvious that symmetry exists for all shear flows, \( X_i \), where \( i=1,2,\ldots,12 \), a fact supported by a common optimal solution of 0.444 kips/in. for this set of variables. Since only four iterations were required to solve such a large nonlinear problem, it may be concluded that Algorithm EGPLP will not only recognize symmetric problems, but it will take advantage of symmetry and generate the optimal solution efficiently. The next problem will further test the efficiency of Algorithm EGPLP through analysis of a bridge deck which generates a more complex set of nonlinear constraints.
Section 4.3  PROBLEM C

The first two problems were designed to test the efficiency and versatility of Algorithm EGPLP as discussed in the first two sections of this chapter. The final problem to be discussed in this chapter is concerned with the optimal design of a bridge decking system as illustrated by Fig. 4.3.1. This problem is of a practical nature, since both shear and torsional effects are considered. By inclusion of one additional constraint, bending may be incorporated into this problem if only horizontal members are considered effective, i.e., the webs are neglected for bending purposes, and an approximate value is utilized for the moment of inertia. More efficient techniques may consist of a two-step procedure that would first analyze shear and torsional loads only, and then check the bending requirements. If the bending requirements are satisfied, the optimal design is satisfactory; if bending is not satisfied, the lower bounds of the thickness of horizontal members are simply increased and the analysis for shear and torsional loads is repeated.

The following presentation will therefore consider only shear and torsional loads as illustrated in Fig. 4.3.1. Since the analysis for this problem is very similar to the presentation in Sec. 4.2, only the objective function and the final constraint equations will be presented. The nomenclature and the positive direction
The objective for this problem is to minimize the weight and, thus, the cross-sectional area of the bridge decking. Hence,

\[
Z = \text{Min} (\text{Cross-sectional area})
\]

\[
= \text{Min} \left[ 6.83x_8x_{17} + 4x_9x_{17} + 4x_{10}x_{16} + 4x_{10}x_{15} \\
+ 2x_{11}x_{16} + 2x_{11}x_{15} + 2x_{12}x_{15} + 2x_{12}x_{16} \\
+ 2x_{13}x_{15} + 2x_{13}x_{16} + 2x_{18}x_{20} + 4x_{14}x_{18} \\
+ 6x_{14}x_{19} - 2x_{14}x_{17} \right] (4.3.1)
\]
A minimum set of seventeen inequality constraints is required to adequately define this problem. These constraints are as follows:

\[ g_1(x) = T_o - (4x_1x_8x_9 + 2x_1^2x_8 + 4x_2x_10x_14 \\
+ 4x_3x_10x_14 + 4x_4x_11x_14 + 4x_5x_12x_14 \\
+ 2x_6x_13x_14 + 2x_7x_13x_14) \leq 0 \]

\[ g_2(x) = 140 - (2x_{10} + x_{11} + x_{12} + x_{13}) \leq 0 \]

\[ g_3(x) = x_1 - x_{17}x_{23} \leq 0 \]

\[ g_4(x) = x_1 - (x_2 + x_9x_{23}) \leq 0 \]

\[ g_5(x) = x_2 - 20x_{16} \leq 0 \]

\[ g_6(x) = x_2 - (x_3 + x_9x_{23}) \leq 0 \]

\[ g_7(x) = x_3 - 20x_{16} \leq 0 \]

\[ g_8(x) = x_3 - (x_4 + x_9x_{23}) \leq 0 \]

\[ g_9(x) = x_4 - 20x_{16} \leq 0 \]

\[ g_{10}(x) = x_4 - (x_5 + x_9x_{23}) \leq 0 \]

\[ g_{11}(x) = x_5 - 20x_{16} \leq 0 \]

\[ g_{12}(x) = x_5 - (x_6 + x_9x_{23}) \leq 0 \]

\[ g_{13}(x) = x_6 - 20x_{15} \leq 0 \]

\[ g_{14}(x) = x_6 - (x_7 + x_9x_{23}) \leq 0 \]

\[ g_{15}(x) = x_7 - 20x_{16} \leq 0 \]

\[ g_{16}(x) = x_7 - (x_6 + x_9x_{23}) \leq 0 \]

\[ g_{17}(x) = x_{16} - x_{15} \leq 0 \]

The last inequality constraint stipulates that the upper flange thickness will be larger than or equal to the lower flange thickness.
Eleven equality constraints are required for this problem. The first four equality constraints involve definition of new variables required to simplify inequality constraints; the last seven equality constraints assure equality of rotation for all cells.

\[ g_{18}(x) = x_{23} + x_{22} - 20 = 0 \]
\[ g_{19}(x) = x_{21} - (x_8x_9 + .5x_8^2) = 0 \]
\[ g_{20}(x) = x_{20}^2 - (x_{13}^2 + x_{14}^2) = 0 \]
\[ g_{21}(x) = x_8 + x_9 - (8 + x_{14}) = 0 \]

(4.3.3)

\[ g_{22}(x) = \left[ A + \frac{x_1x_{14}}{x_{17}x_{21}} + \frac{x_2x_{14}}{x_{19}x_{21}} \right] - \left[ B + \frac{3.41x_1x_8}{x_{17}x_{21}} + \frac{2x_1x_9}{x_{17}x_{21}} + \frac{x_1x_{14}}{x_{19}x_{21}} \right] = 0 \]

\[ g_{23}(x) = \left[ A + \frac{x_1}{x_{10}x_{19}} + \frac{x_3}{x_{10}x_{19}} \right] - \left[ B + \frac{2x_2}{x_{10}x_{19}} + \frac{x_2}{x_{14}x_{16}} + \frac{x_2}{x_{14}x_{15}} \right] = 0 \]

\[ g_{24}(x) = \left[ A + \frac{x_2}{x_{10}x_{19}} + \frac{x_4}{x_{10}x_{19}} \right] - \left[ B + \frac{2x_3}{x_{10}x_{19}} + \frac{x_3}{x_{14}x_{16}} + \frac{x_3}{x_{14}x_{15}} \right] = 0 \]
\[ e_{25}(x) = \left[ A + \frac{x_3}{x_{11}x_{19}} + \frac{x_5}{x_{11}x_{18}} \right] \]
\[ - \left[ B + \frac{x_4}{x_{11}x_{19}} + \frac{x_4}{x_{14}x_{16}} + \frac{x_4}{x_{11}x_{18}} + \frac{x_4}{x_{14}x_{15}} \right] = 0 \]

\[ e_{26}(x) = \left[ A + \frac{x_4}{x_{12}x_{18}} + \frac{x_6}{x_{12}x_{18}} \right] \]
\[ - \left[ B + \frac{2x_5}{x_{12}x_{18}} + \frac{x_5}{x_{14}x_{16}} + \frac{x_5}{x_{14}x_{15}} \right] = 0 \]

\[ e_{27}(x) = \left[ \frac{2x_7}{x_{14}x_{16}} + \frac{2x_5}{x_{13}x_{18}} + \frac{4x_7x_{20}}{x_{13}x_{14}x_{18}} \right] \]
\[ - \left[ B + \frac{2x_6}{x_{13}x_{18}} + \frac{2x_6x_{20}}{x_{13}x_{14}x_{18}} + \frac{2x_6}{x_{14}x_{15}} \right] = 0 \]

\[ e_{28}(x) = \left[ v_0 + 2x_{14}x_{17}x_{22} + 4x_{14}x_{19}x_{22} + 6x_{14}x_{19}x_{22} + 6x_{14}x_{18}x_{22} \right] = 0 \]

\[ A = \frac{2x_{17}x_{20}}{x_{13}x_{14}x_{18}} + \frac{2x_7}{x_{14}x_{16}} \]
\[ B = \frac{2x_6x_{20}}{x_{13}x_{14}x_{18}} \]

These equality constraints are highly nonlinear; hence, many optimization techniques would probably converge to the local optimum solution with considerable difficulty. Algorithm ECPLP, however, approximates all constraints by a single term posynomial. Thus the
complexity of constraints has very little influence on convergence within each iterative step. Unfortunately, as with most algorithms, the number of iterations required to attain a local optimal solution is generally dependent upon the nature and stability of the constraint equations. For computational purposes, assume the following data:

\[ T_0 = \text{applied torsional load} = 25,000 \text{ in.-kips} \]
\[ V_0 = \text{applied shear load} = 200 \text{ kips} \]

To illustrate the versatility of Algorithm EGPLP, the torsional moment constraint, \( T \geq T_0 \) (\( T = \text{torsional capacity of the cross-section} \)), is included with inequality constraints, i.e., \( g_1(X) \) in (4.3.2). For this problem, assume that equality is desired, i.e., \( T = T_0 \). It is not necessary to move the \( g_1(X) \) constraint into the equality constraint set, but rather a large positive cost coefficient is assumed for the slack variable corresponding to \( g_1(X) \) in the linear programming algorithm, subroutine SIMPLE. This procedure was followed in the solution of this problem.

The bounded values for \( X \) vector represented by the \( R \) vector, the \( r \) vector, and the initial vector \( X^{(1)} \) are listed below.

\[
\begin{align*}
R_1 &= 6.0 & r_1 &= -1 & X_1^{(1)} &= 2.5 \text{ in.}
\end{align*}
\]
\[
\begin{align*}
R_2 &= 6.0 & r_2 &= -1 & X_2^{(1)} &= 2.5
\end{align*}
\]
The data given by (4.3.1)-(4.3.4) was read into the Algorithm EGPIIP, and twelve iterations were required to attain a locally optimum solution. For the sake of brevity, only the final solution, i.e., the results of the twelfth iteration, will now be summarized. \( x^* \) in-
The optimal value of the objective function which represents the minimal cross-sectional area is

\[ z^* = 225.35 \text{ in.}^2 \]

The optimal solution vector \( x^* \) satisfies all equality constraints and, upon evaluation of the inequality constraints, the following values are obtained.

\[
\begin{align*}
\varepsilon_1(x^*) &= -0.21 \\
\varepsilon_2(x^*) &= -0.06 \\
\varepsilon_3(x^*) &= -14.27 \\
\varepsilon_4(x^*) &= -3.10 \\
\varepsilon_5(x^*) &= -16.11 \\
\varepsilon_6(x^*) &= -2.88 \\
\varepsilon_7(x^*) &= -15.22 \\
\varepsilon_8(x^*) &= -2.78 \\
\varepsilon_9(x^*) &= -14.82 \\
\varepsilon_{10}(x^*) &= -3.64 \\
\varepsilon_{11}(x^*) &= -14.63 \\
\varepsilon_{12}(x^*) &= -3.61 \\
\varepsilon_{13}(x^*) &= -14.57 \\
\varepsilon_{14}(x^*) &= -3.60 \\
\varepsilon_{15}(x^*) &= -14.56 \\
\varepsilon_{16}(x^*) &= -3.59 \\
\varepsilon_{17}(x^*) &= 0
\end{align*}
\]
The $X^*$ vector satisfies all constraint equations; hence, it is a feasible solution. As in the case of the first two problems of this chapter, no attempt will be made to theoretically verify the optimality of this solution.

The primary purpose of this problem was to check the ability and efficiency of Algorithm EGPLP in the solution of problems with complex constraint equations. Although twelve iterations were required to attain the solution given on the previous page, such a rate of convergence appears to be quite satisfactory when due regard is given to the complexity of this problem. Additional discussion on convergence and efficiency of Algorithm EGPLP is given in the following chapter.
Chapter 5 DISCUSSION OF CONVERGENCE

Section 5.1 Convergence Rates for Problems in Chapter 4

The problems discussed in the last chapter all converged to an assumed locally optimal solution. In order to illustrate the efficiency and the rate of convergence of the Algorithm EGPLP, Table 5.1.1, given on the following page, summarizes the most important characteristics of each problem solved in Chapter 4. PROBLEM B required only four iterations because of the extensive symmetry of the structure as shown in Fig. 4.2.1. From this table, it can easily be concluded that the number of iterations required for convergence is a function of not only the numerical values for N, NIC, and NEC, but also of such immeasurable quantities as the complexity of equations and the extent of symmetry present within each problem. Analysis of results in the above table and from about fifteen other problems could not be integrated into a single formula that would accurately predict the number of iterations required for convergence to a locally optimal solution. However, when extensive familiarity with the Algorithm EGPLP is attained, a good estimation of the number of iterations required for convergence can generally be given prior to actual solution.
### TABLE 5.1.1

<table>
<thead>
<tr>
<th></th>
<th>PROBLEM A</th>
<th>PROBLEM B</th>
<th>PROBLEM C</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = number of variables</td>
<td>12</td>
<td>18</td>
<td>24</td>
</tr>
<tr>
<td>NIC = number of inequality constraints</td>
<td>6</td>
<td>27</td>
<td>17</td>
</tr>
<tr>
<td>NEC = number of equality constraints</td>
<td>8</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>ITERATIONS REQUIRED a</td>
<td>7</td>
<td>4</td>
<td>12</td>
</tr>
</tbody>
</table>

aIteration procedure terminated according to Step 12 on page 50 with \( \varepsilon = 0.015 \).

bCPU TIME = Computational time on IBM 360/175.

The efficiency of Algorithm EGPLP is best illustrated by the computer time in Table 5.1.1 required for convergence. Since no other optimization technique was utilized to solve PROBLEMS A, B, and C, a direct comparison of EGPLP with other nonlinear optimization techniques is not possible. Note that the computer time for PROBLEM C is only about twice the amount required for PROBLEM A.
Section 5.2 Convergence of Algorithm EGPLP

If the optimization problem is devoid of equality constraints and, if all inequality constraints are regular, then it can be shown that both Algorithms EGPLP and DUALGP converge to a locally optimal solution by a proof quite similar to the convergence theorem given in Ref. 4 (Solution procedure given in Ref. 4 is identical to that given for Algorithm DUALGP if equality constraints are not present). However, the regularity condition cannot be readily verified for large nonlinear optimization problems, hence, the proof given in Ref. 4 for convergence is of little value for large problems. Since the problems discussed in the previous chapters contained both equality and inequality constraints, the convergence theory in Ref. 4 is not directly applicable. With this in mind, the following paragraphs will briefly describe the convergence procedure for Algorithm EGPLP.

The most general problem discussed in the previous chapters consisted of the following set of relationships.

\[
\begin{align*}
\text{Min } g_0(X) \\
\text{subject to: } X &= \text{real} \\
g_k(X) &\leq 0 \quad k=1, \ldots, M_0 \\
g_k(X) &= 0 \quad k=M_0+1, \ldots, (M_0+M_1=M_2).
\end{align*}
\]

An equivalent problem which is considered by Algorithm EGPLP is defined as follows.
\[
\begin{align*}
\text{Min } & (x_{N+1}) \\
\text{subject to: } & X = \text{real} \\
g_0(x) - x_{N+1} & \leq 0 \quad (5.2.2) \\
g_k(x) & \leq 0 \quad k = 1, \ldots, M_0 \\
g_k(x) & = 0 \quad k = M_0 + 1, \ldots, (M_0 + M_1 = M_2)
\end{align*}
\]

Let \( X = (X_1, \ldots, X_N, X_{N+1}) \), and choose \( X^{(1)} \) vector as indicated in Step 6 of Algorithm EGPLP. The sequence of values for \( x_{N+1}^{(k)} \) may be monotone decreasing if so desired, however, this is not a necessity. The uniqueness of Algorithm EGPLP lies in the fact that at each iteration step, the feasible region is approximated by a set of single term posynomials. The approximate feasible region is such that the original inequality constraints are satisfied at each iteration step, but the equality constraints are generally not satisfied, except at the last iteration step if convergence is attained. Hence, the results at all iteration steps except the last may be infeasible. This represents a disadvantage that is also common among the penalty methods (Ref. 10, Chp. 12). The convergence proofs for such infeasible methods are often quite difficult to develop, especially for such general problems as defined in (5.2.2).

With regard to Algorithm EGPLP, no formal proof for convergence will be given. At the \( i \) th iteration, this algorithm approximates all relationships at a specific point \( X^{(i)} \) with a set of single term posynomials. The
new approximate posynomial optimization problem will have a global optimal solution whose relationship to a locally optimal solution of the original problem will depend primarily upon the accuracy of the approximation. Accordingly, the rate of convergence will depend primarily upon the size of the optimization problem and the extent of smoothness in the objective function and the equality and inequality constraints. Solution of problems, such as those in the previous chapters, have indicated that this is indeed the case.

Algorithm EGPLP was used to solve nearly twenty nonlinear optimization problems and convergence was attained—in all cases. These results are by no means a proof for convergence of Algorithm EGPLP, nevertheless, from these satisfactory results it may be concluded that if the objective function and all constraint equations are "well defined" and "sufficiently smooth", convergence will probably occur in a limited number of iteration steps as illustrated in Table 5.1.1. As stated in the previous sections of this chapter, convergence rates are difficult to predict prior to actual solution due to the rather complex solution procedure inherent in Algorithm EGPLP. As with most aspects of engineering, only experience in the usage of Algorithm EGPLP will bring out the advantages, disadvantages, and other related aspects of this new Algorithm.
CHAPTER 6 CONCLUSION

In the process of development of new concepts, the advantages are accentuated while the disadvantages may often be neglected. However, by stating the disadvantages, a greater insight may often be obtained into the advantages. Therefore, a brief list of both the positive and negative aspects of EGPLP will be outlined below without extensive explanation.

Section 6.1 Advantages of Algorithm EGPLP

a) Both nonlinear inequality and equality constraints can be analyzed.

b) Solution procedure is independent of the complexity (extent of nonlinearity) of the objective function and constraint equations.

c) EGPLP seems to be quite reliable in solving many nonlinear problems.

d) Large nonlinear optimization problems may be solved by EGPLP without additional computational difficulties as compared to small and medium size problems.

Section 6.2 Disadvantages of Algorithm EGPLP

a) Extensive and efficient usage of EGPLP requires understanding of the solution procedure within EGPLP.

b) Small nonlinear problems are not solved efficiently by EGPLP.

c) Constraint equations which are not smooth may present difficulties.

d) As with most other algorithms, the maximum size of nonlinear problems that may be solved by EGPLP is limited by computer storage.
The first disadvantage listed above may be a source of discouragement to possible users of EGPLP. Awareness of the solution procedure as outlined in Chapter 3 and an understanding of the nature of the functions involved in an optimization problem will enable a user of EGPLP to organize his problem so as to reduce computational time and increase the rate of convergence. Furthermore, the applicability of EGPLP can be extended to problems other than those consisting of signomial equations if the solution procedure is understood.

Section 6.3 Final Remarks

The development of Algorithm EGPLP resulted from a desire to use the many advantages of Geometric Programming in solution of large nonlinear optimization problems. A proof for convergence of EGPLP and DUALGP could not be developed as explained in Chapter 5. Problems B and C in Chapter 4 illustrated the solution of practical engineering problems employing Algorithm EGPLP while Chapter 5 illustrated the efficiency of this Algorithm.

Anyone interested in using EGPLP should first read Chapters 3, 4, and 5. A listing of the EGPLP program will not be given in this research paper. However, anyone interested in obtaining the program may write to the author, Civil Engineering Department, California State University at Long Beach, Long Beach, California 90840.
REFERENCES


REFERENCES (Cont.)