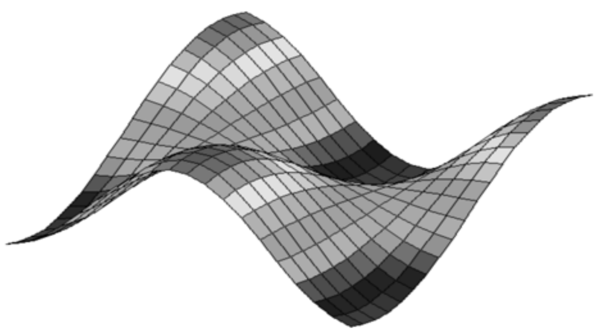
MULTIDISCIPLINE DESIGN OPTIMIZATION



Garret N. Vanderplaats
Founder & Chief Executive Officer
Vanderplaats Research & Development, Inc.
Colorado Springs, CO

CHAPTER

ONE

BASIC CONCEPTS

He Wants Us To Use Optimization!

1-1 INTRODUCTION

The concept of optimization is basic to much of what we do in our daily lives. The desire to run a faster race, win a debate, or increase corporate profit implies a desire to do or be the best in some sense. In engineering, we wish to produce the "best quality of life possible with the resources available." Thus in "creating" new products, we must use design tools which provide the desired results in a timely and economical fashion. Numerical optimization is one of the tools at our disposal.

> ***"Saving even a few pounds of a vehicle's weight... could mean that they would also go faster and consume less fuel. Reducing weight involves reducing materials, which, in turn, means reducing cost as well."***
>
> Henry Ford, 1923

Imagine if the mass of every vehicle in the United States could be reduced by only one percent. This would result in an estimated six tenths of one percent improvement in fuel economy [1]. Now consider that the U.S. uses nearly ten million barrels of gasoline daily and the cost per gallon of gasoline is about $3.00 (in 2007). A 0.6% improvement in fuel economy then represents nearly a three *Billion* Dollar savings by the consumers each year! This does not include the savings in other resources, the reduction in pollutants and other benefits to society.

Now imagine reducing the mass of a 200 passenger commercial aircraft by only 100Kg. This would add a paying passenger for the life of the aircraft. Also, it would reduce per passenger mile pollution by one half of one percent.

These simple examples provide compelling reasons for using design tools that create more efficient designs. These examples are for the reduction of mass but we can make the same argument for improving combustion efficiency or aerodynamic efficiency, improving an injection molding process, etc.

Analysis or Design?

In studying design optimization, it is important to distinguish between analysis and design.

Analysis is the process of determining the response of a specified component or system to its environment. For example, the calculation of stresses in a structure that result from applied loads is referred to here as analysis. Or we may wish to determine the lift, drag and pitching moment of a particular airfoil at one or more flight conditions.

Design, on the other hand, is used to mean the actual process of defining the system. For example, structural design entails defining the sizes and locations of members necessary to support a prescribed set of loads. In computational fluid dynamics (CFD) we may wish to determine the best shape of an airfoil or the pipe diameters in a Chemical Plant. Clearly, analysis is a sub-problem in the design process because analysis is how we evaluate the adequacy of the design.

Much of the design task in engineering is quantifiable, and so we are able to use the computer to analyze alternative designs rapidly. The purpose of numerical optimization is to aid us in rationally searching for the best design to meet our needs.

While the emphasis here is on design, it should be noted that these methods can often be used for analysis as well. Nonlinear structural analysis is an example where optimization can be used to solve a nonlinear energy minimization problem.

Although we may not always think of it this way, design can be defined as the process of finding the minimum or maximum of some parameter which may be called the objective function. For the design to be acceptable, it must also satisfy a certain set of specified requirements called constraints. That is, we wish to find the constrained minimum or maximum of the objective function.

For example, assume we wish to design an internal-combustion engine. The design objective could be to maximize combustion efficiency. The engine may be required to provide a specified power output with an upper limit on the amount of harmful pollutants which can be emitted into the atmosphere. The power requirements and pollution restrictions are therefore constraints on the design.

The Design Process

Various methods can be used to achieve the design goal. If we are designing an internal combustion engine, one approach might be through experimentation where many engines are built and tested. The engine providing maximum economy while satisfying the constraints on the design would then be chosen for production. Clearly this is a very expensive approach with little assurance of obtaining a true optimum design. A second approach might be to define the design process analytically and then to obtain the solution using differential calculus or the calculus of variations. While this is certainly an attractive procedure, it is seldom possible in practical applications to obtain a direct analytical solution because of the complexities of the analysis and design task.

Most design organizations now have computer codes capable of analyzing a design which the engineer considers reasonable. For example, the engineer may have a computer code which, given the compression ratio, airfuel mixture ratio, bore and stroke, camshaft profile and other basic design parameters, can analyze the internal-combustion engine to predict its efficiency, power output, and emissions. The engineer could then change these design variables and rerun the program until an acceptable design is obtained. In other words, the physical experimentation approach where engines are built and tested is replaced by numerical experimentation, recognizing that the final step will still be the construction of one or more prototypes to verify our numerical results.

With the availability of computer codes to analyze the proposed design, the next logical step is to automate the design process itself. In its most basic form, design automation may consist of a series of loops in the computer code which cycle through many combinations of design variables. The combination which provides the best design satisfying the constraints is then termed optimum. This approach, often called the “Try them all” method, has been used with some success and may be quite adequate if the analysis program uses a small amount of computer time and we have only a few design variables. However, the cost of this technique increases dramatically as the number of design variables to be changed increases and as the computer time for a single analysis increases.

Consider, for example, a design problem described by three variables. Assume we wish to investigate the designs for 10 values of each variable.

Assume also that any proposed design can be analyzed in one-tenth of a central processing unit (CPU) second on a digital computer. There are then 10^3 combinations of design variables to be investigated, each requiring one-tenth second for a total of 100 CPU seconds to obtain the desired optimum design. This would probably be considered an economical solution in most design situations. However, now consider a more realistic design problem where 10 variables describe the design. Again, we wish to investigate 10 values of each variable. Also now assume that the analysis of a proposed design requires 100 CPU seconds on the computer. The total CPU time now required to obtain the optimum design is 10^{12} seconds, or roughly 32,000 years of computer time! Clearly, for most practical design problems, a more rational approach to design automation is needed.

Numerical optimization techniques offer a logical approach to design automation, and many algorithms have been proposed over the years. Some of these techniques, such as linear, quadratic, dynamic, and geometric programming algorithms, have been developed to deal with specific classes of optimization problems. A more general category of algorithms referred to as nonlinear programming has evolved for the solution of general optimization problems. Methods for numerical optimization are referred to collectively as mathematical programming techniques.

During the nearly 60 year history of mathematical programming there has been an almost bewildering number of algorithms published for the solution of numerical optimization problems. The author of each algorithm usually has numerical examples which demonstrate the efficiency and accuracy of the method, and the unsuspecting practitioner will often invest a great deal of time and effort in programming an algorithm, only to find that it will not in fact solve the particular optimization problem being attempted. This often leads to disenchantment with these techniques which can be avoided if the user is knowledgeable in the basic concepts of numerical optimization. There is an obvious need, therefore, for a unified, non-theoretical presentation of optimization concepts.

The purpose here is to attempt to bridge the gap between optimization theory and its practical applications. The remainder of this chapter will be devoted to a discussion of the basic concepts of numerical optimization. We will consider the general statement of the nonlinear constrained optimization problem and some (slightly) theoretical aspects regarding the existence and uniqueness of the solution to the optimization problem. Finally, we will consider some practical advantages and limitations to the use of these methods.

Numerical optimization has traditionally been developed in the operations research community. The use of these techniques was introduced to engineering design in 1960 when Schmit [2] applied nonlinear optimization techniques to structural design and coined the phrase “structural synthesis.”

While the work of Ref. 2 was restricted to structural optimization, the concepts presented there offered a fundamentally new approach to engineering design which is applicable to a wide spectrum of design problems. The basic concept is that the purpose of design is the allocation of scarce resources [3]. The purpose of numerical optimization is to provide a computer tool to aid the designer in this task.

This book may be considered to be a modification and extension on an earlier text by this author [4]. Here, we will consider a more limited number of algorithms, focusing on those which provide basic understanding or which have been demonstrated to be powerful and reliable. Beyond this, we will focus on Multidiscipline Design Optimization (MDO) where several separate disciplines are included in the design process. Finally, we will provide more practical examples of how optimization can be used to improve designs.

1-2 MULTIDISCIPLINE SYSTEMS

Almost everything engineers design includes the consideration of multiple disciplines. Most of the time, the disciplines are considered separately. For example, designing the floor panel on an automobile is a structural design problem while the air induction system includes both fluid mechanics and structural considerations. However, even for the air induction system, the two disciplines usually are not considered simultaneously. That is, the shape of the air induction system is defined based on air flow (and perhaps packaging) considerations and the structure is then designed. This is why most engineering organizations are separated into departments, each specializing in a particular discipline such as structures, aerodynamics, thermodynamics, electromechanics, mechanical components, etc. A designer then calls upon these specialists as needed for a particular system being designed.

An example of a design where several disciplines are considered is the turbojet engine shown in Figure 1.1.

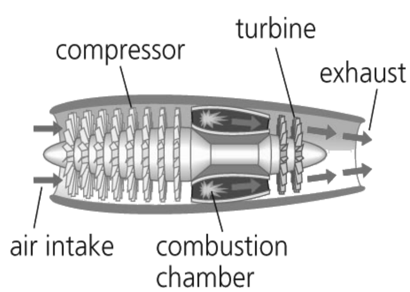


Figure 1.1 Turbojet Engine.

The turbojet is considered to be the simplest of all jet engines, yet it is clear that the interactions between the structure, fluid flow and combustion process are very complex. Even if we just look at a single blade in the turbine, we must consider material, structural, thermal and fluid flow disciplines. Because of these complexities, we rely heavily on computer software to evaluate any proposed designs. Also, it is important to note that the figure is just a very simple portrayal of the real engine. An actual engine includes pumps, pipes, valves, electronic controls and a host of other components.

The turbojet is just one component of a larger system, in this case probably an aircraft. Therefore, we can see that design of real systems is an extremely complex task. However, it is important to remember that we designed very complex systems long before we had computers, computer aided engineering, etc. This observation forms the basis for the design philosophy presented in this text. In other words, our goal will be to present methods that make the design task faster and easier while recognizing that traditional design methods have worked very well. If we use the optimization tools available at every opportunity, we can greatly enhance the traditional design process to design better systems in less time than with any other known methods.

1-3 OPTIMIZATION CONCEPTS

Here we will briefly describe the basic concepts of optimization by means of a physical example and two numerical examples.

Example 1-1 A Physical Optimization Problem

Consider a little boy standing on a hillside, wearing a blindfold as shown in Figure 1.2. He wishes to find the highest point on the hill that is inside the two fences. Now, because of the blindfold, he cannot just look up the hill and go straight to the “optimum” point. He must somehow search for the high point. Consider how he may do this. He may take a small step in the north-south direction and a small step in the east-west direction. From this, he can sense the slope of the hill and then search in the upward direction. In a mathematical sense, what he has done is calculate the direction of steepest ascent by calculating the gradient by finite difference methods.

He can now search in the is steepest ascent direction until he crosses the crown of the hill or encounters a fence.

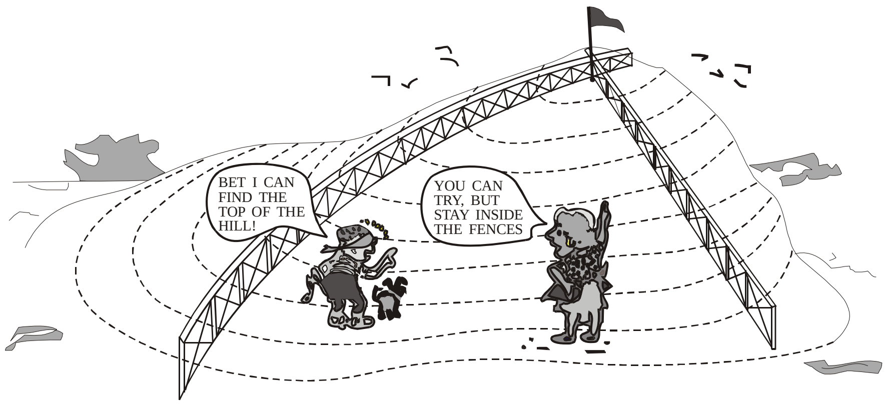


Figure 1.2 The Physical Problem.

If we store his initial location (longitude and latitude) in a vector, $\mathbf{X}^0$, we can call this the vector of "design variables." The steepest ascent direction is also a vector that we can store in a "search direction" vector, $\mathbf{S}^1$. He can now search in this is steepest ascent direction until he crosses the crown of the hill or encounters a fence. In this case, he encounters a fence as shown in Figure 1.3.

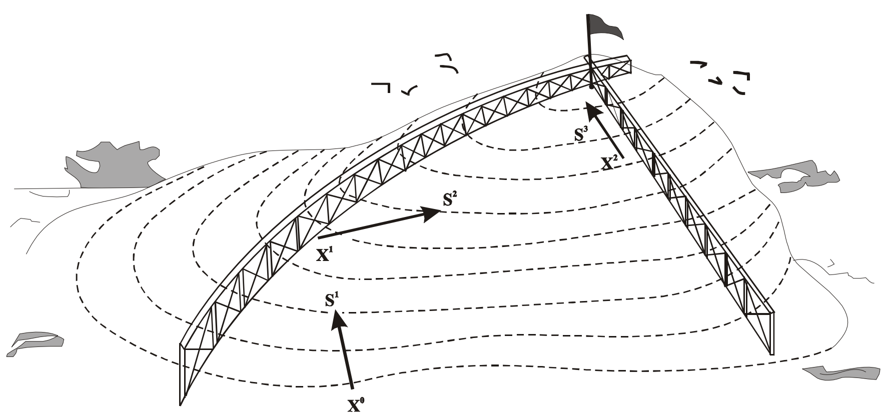


Figure 1.3 The Optimization Process.

We will call his location at the fence, $\mathbf{X}^1$ to indicate that this is the end of the first iteration in our search up the hill.

Note that the design vector, $\mathbf{X}^1$, is given as

$$\mathbf{X}^1 = \mathbf{X}^0 + \alpha^* \mathbf{S}^1$$

where α^* is the number of steps in direction S^1 that he took to reach X^2 and partial steps are allowed.

At $\mathbf{X}^1$ he can repeat the process of taking a small step in the north-south and east-west directions. Now, in addition to calculating the gradient (slope) of the hill, he will calculate the gradient of the fence, which is the outward normal vector to the fence at $\mathbf{X}^1$.

Using this gradient information, he can determine a new search direction, $\mathbf{S}^2$, which will continue to move him up the hill while "pushing" away from the fence. Note that if he tries to follow the curved fence, he cannot move in a straight line without moving outside the fence. Because he wants to move in a straight line as far as possible, he pushes away from the fence.

By finding a direction that moves him up the hill, he has found a "usable" search direction and by finding a direction that stays inside the fences, he has found a "feasible" direction.

He now searches in direction $\mathbf{S}^2$ until he encounters the second fence at design point $\mathbf{X}^2$.

He now repeats the process again to determine a search direction, S^3. Here, we assume he knows the fence is straight (a linear function) so he can search in a direction tangent to the fence. If he did not know the fence is straight, he would push away from it as before and "zig-zag" his way up the hill.

Searching in direction $\mathbf{S}^3$ leads him to the highest point on the hill inside the fences. He can no longer find a search direction that will continue up the hill without going outside the fences and so he concludes that he has reached the highest possible point. In other words, he has found the "constrained optimum."

He can now remove the blindfold and he may see that there are actually higher points elsewhere on the hill and also inside the fences. The search method we used only assures him of achieving a "local" optimum and does not guarantee the "global" optimum. This is often noted as a weakness of gradient based optimization methods used here. However, remember that we have made significant improvement, even if we cannot guarantee the absolute best. This, in fact, models the typical engineering environment where we are pleased to make improvements in the time available. Without optimization, engineers seldom achieve even a local optimum and never even ask about global optimization until some expert tells them to.

In practice, we may begin outside the fences and our first priority will be to find a search direction back inside the fences to the "feasible" region. Indeed, one of the most effective uses of optimization is to find an acceptable design when our current design violates one or more design criteria.

The method described above is a physical interpretation of Zoutendijk's Method of Feasible Directions, first published in 1960 [5]. The concept of finding a "usable-feasible" direction as was done here is fundamental to much of optimization theory. Finally, note that here we maximized the function (elevation). Normally, we will minimize our objective function. If we wish to maximize, we simply minimize the negative of the function.

Example 1-2 Unconstrained Function Minimization

In the example above, we considered an optimization task where fences, or constraints, limit the design. If there are no such "fences, we have an unconstrained problem. This may be thought of as the simplest of optimization tasks because we know from basic calculus that we are seeking the point where the gradient of the function vanishes.

Assume we wish to find the minimum value of the following simple algebraic function.

$$F(\mathbf{X}) = 10X_1^4 - 20X_1^2X_2 + 10X_2^2 + X_1^2 - 2X_1 + 5 \tag{1.1}$$

$F(\mathbf{X})$ is referred to as the objective function which is to be minimized, and we wish to determine the combination of the variables X_1 and X_2 which will achieve this goal. The vector $\mathbf{X}$ contains X_1 and X_2 and we call them the design, or decision, variables. That is,

$$\boldsymbol{X} = \begin{Bmatrix} X_1 \\ X_2 \end{Bmatrix} \tag{1.2}$$

No limits are imposed on the values of X_1 and X_2 and no additional conditions must be met for the "design" to be acceptable. Therefore, $F(\mathbf{X})$ is said to be unconstrained. Figure 1.4 is a graphical representation of the function, where lines of constant value of $F(\mathbf{X})$ are drawn. This function is often referred to as the *banana function* because of its distinctive geometry. Figure 1.4 is referred to as a two-variable function space, or two variable design space, where the design variables X_1

and X_2 correspond to the coordinate axes. In general, a design space will be *n* dimensional, where *n* is the number of design variables of which the objective is a function. The two-variable design space will be used throughout our discussion of optimization techniques to help visualize the various concepts.

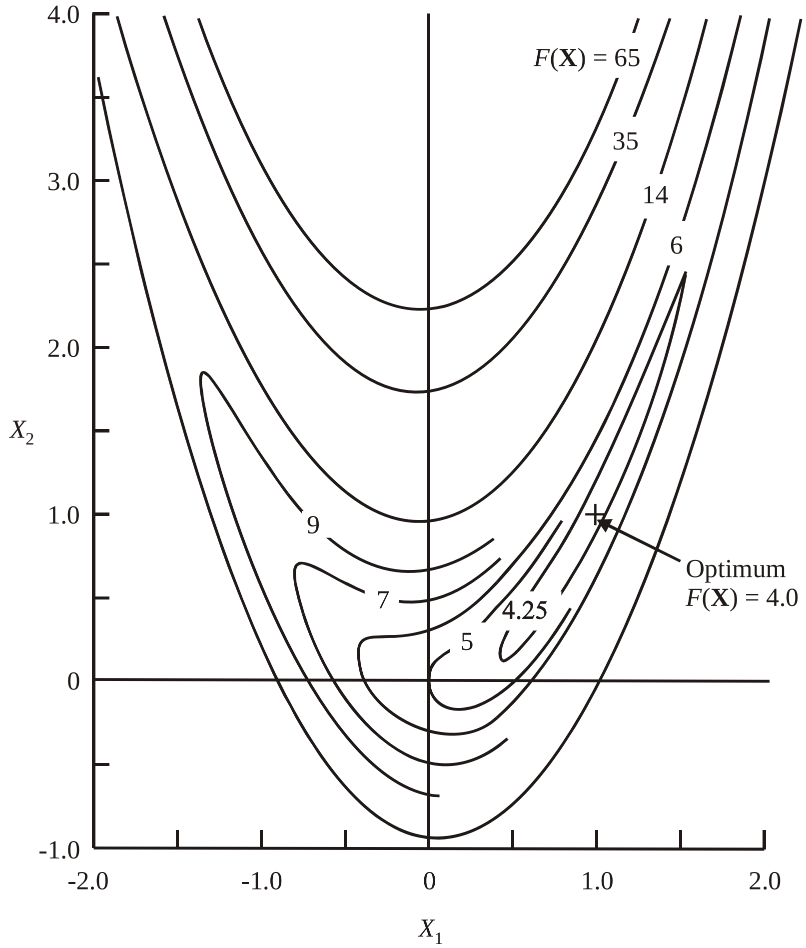


Figure 1.4 Two-variable function space.

From Figure 1.4 we can estimate that the minimum value of $F(X)$ will occur at $X_1^* = 1$ and $X_2^* = 1$. We know also from basic calculus that at the optimum, or minimum, of $F(\mathbf{X})$, the partial derivatives with respect to X_1 and X_2 must vanish. That is

$$\frac{\partial}{\partial X_1}F(\mathbf{X}) = 40X_1^3 - 40X_1X_2 + 2X_1 - 2 = 0 \tag{1.3}$$

$$\frac{\partial}{\partial X_2}F(\mathbf{X}) = -20X_1^2 + 20X_2 = 0 \tag{1.4}$$

Solving for X_1 and X_2, we find that indeed $X_1^* = 1$ and $X_2^* = 1$. We will see later that the vanishing gradient is a necessary but not sufficient condition for finding the minimum.

In this example, we were able to obtain the optimum both graphically and analytically. However, this example is of little engineering value, except for demonstration purposes. In most practical engineering problems the minimum of a function cannot be determined analytically. The problem is further complicated if the decision variables are restricted to values within a specified range or if other conditions are imposed in the minimization problem. Therefore, numerical techniques are needed. We will now consider a simple design example where conditions (constraints) are imposed on the optimization problem.

Example 1-3 Constrained function minimization

Figure 1.5a depicts a tubular column of height h which is required to support a concentrated load P as shown. We wish to find the mean diameter D and the wall thickness t to minimize the weight of the column. The column weight is given by

$$W = \rho A h = \rho \pi D t h \tag{1.5}$$

where A is the cross-sectional area and ρ is the material's unit weight.

We will consider the axial load only, and for simplicity will ignore any eccentricity, lateral loads, or column imperfections. The stress in the column is given by

$$\sigma = \frac{P}{A} = \frac{P}{\pi D t} \tag{1.6}$$

where stress is taken as positive in compression. In order to prevent material failure, this stress must not exceed the allowable stress $\bar{\sigma}$. In addition to preventing material failure, the stress must not exceed that at which Euler buckling or local shell buckling will occur, as shown in Figs. 1.5b and c. The stress at which Euler buckling occurs is given by

$$\sigma_b = \frac{\pi^2 EI}{4Ah^2} = \frac{\pi^2 E(D^2 + t^2)}{32h^2} \tag{1.7}$$

where E = Young's modulus
I = moment of inertia

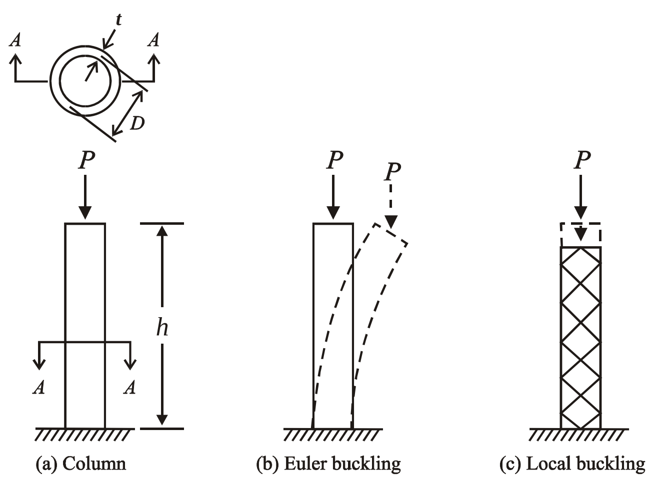


Figure 1.5 Column design for least weight.

The stress at which shell buckling occurs is given by

$$\sigma_s = \frac{2Et}{D\sqrt{3(1-\nu^2)}} \tag{1.8}$$

where ν = Poisson's ratio

The column must now be designed so that the magnitude of the stress is less than the minimum of $\bar{\sigma}$, σ_b, and σ_s. These requirements can be written algebraically as

$$\sigma \leq \bar{\sigma} \tag{1.9}$$

$$\sigma \le \sigma_b \tag{1.10}$$

$$\sigma \le \sigma_s \tag{1.11}$$

In addition to the stress limitations, the design must satisfy the geometric conditions that the mean diameter be greater than the wall thickness and that both the diameter and thickness be positive

$$D \ge t \tag{1.12}$$

$$D \ge 10^{-6} \tag{1.13}$$

$$t \ge 10^{-6} \tag{1.14}$$

Bounds of 10^{-6} are imposed on D and t to ensure that σ in Eq. (1.6) and σ_s in Eq. (1.8) will be finite.

The design problem can now be stated compactly as

Minimize: $$W(D,t) = \rho\pi D t h \tag{1.15}$$

Subject to:

$$g_1(D,t) = \frac{\sigma}{\bar{\sigma}} - 1 \le 0 \tag{1.16a}$$

$$g_2(D,t) = \frac{\sigma}{\sigma_b} - 1 \le 0 \tag{1.16b}$$

$$g_3(D,t) = \frac{\sigma}{\sigma_s} - 1 \le 0 \tag{1.16c}$$

$$g_4(D,t) = t - D \le 0 \tag{1.16d}$$

$$D \ge 10^{-6} \tag{1.17a}$$

$$t \ge 10^{-6} \tag{1.17b}$$

where $\bar{\sigma}$, σ_b, and σ_s are given by Eqs. (1.6), (1.7), and (1.8), respectively. To summarize, Eq. 1.15 defines the objective function and Eqs. 1.16a - 1.16d and 1.17a, 1.17b define the constraints on the design problem. Note that Eq. 1.16a to c is just a normalized form of Eqs. 1.9 to 1.11. The constraints given by Eq. 1.17a and 1.17b are referred to as side constraints because they directly impose bounds on the value of the design variables. Figure 1.6 is the design space associated with the column design problem. In addition to contours of constant objective, the constraint boundaries $[g_j(\mathbf{X}) = 0]$ are also drawn in the design space. That portion of the design space inside the constraint boundaries defined by the hatched lines is referred to as the feasible design space, and all designs in this region are acceptable. Any design which violates these constraint boundaries is unacceptable and is referred to as infeasible. This figure represents a simple example of the general nonlinear constrained optimization problem.

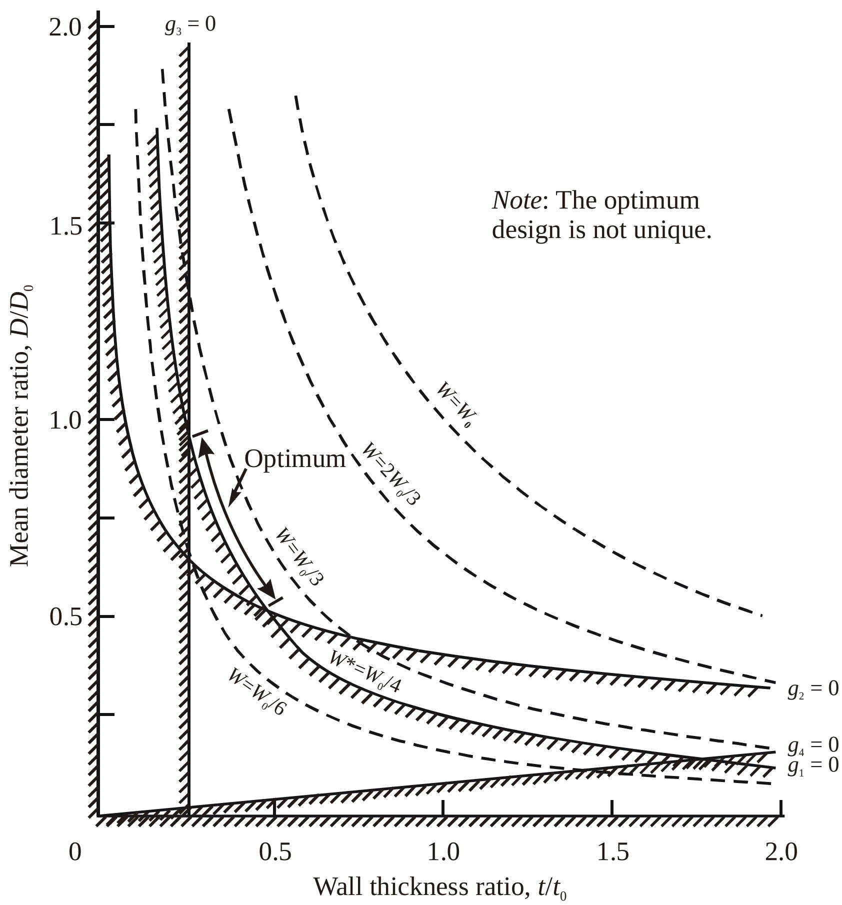


Figure 1.6 Two-variable function space for column.

It is interesting to note that the optimum design is not unique. There are a range of values of D and t that give the same optimum value of the objective function.

1-4 GENERAL PROBLEM STATEMENT

We can now write the nonlinear constrained optimization problem mathematically as follows:

Minimize: $F(\mathbf{X})$ objective function (1.18)

Subject to:

$g_j(\mathbf{X}) \le 0 \quad j=1,m$ inequality constraints (1.19)

$h_k(\mathbf{X}) = 0 \quad k=1,l$ equality constraints (1.20)

$X_i^l \le X_i \le X_i^u \quad i=1,n$ side constraints (1.21)

$$\text{where } \mathbf{X} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ . \\ . \\ . \\ X_n \end{Bmatrix} \quad \text{design variables}$$

The vector **X** is referred to as the vector of design variables. In the column design given above, this vector would contain the two variables *D* and *t*. The objective function $F(\mathbf{X})$ given by Eq. 1.18, as well as the constraint functions defined by Eqs. 1.19 and 1.20 may be linear or nonlinear functions of the design variables **X.** These functions may be explicit or implicit in **X** and may be evaluated by any analytical or numerical techniques we have at our disposal. Indeed, these functions could actually be measured experimentally. However, except for special classes of optimization problems, it is important that these functions be continuous and have continuous first derivatives in **X.**

In the column design example, we considered only inequality constraints of the form given by Eq. 1.19. Additionally, we now include the set of equality constraints $h_k(\mathbf{X})$ as defined by Eq. 1.20. If equality constraints are explicit in **X,** they can often be used to reduce the number of design variables considered. For example, in the column design problem, we may wish to require the thickness be one-tenth the value of the diameter, that is, $t = 0.1D$. This information could be substituted directly into the problem statement to reduce the design problem to one in diameter D only. In general, $h(\mathbf{X})$ may be either a very complicated explicit function of the design variables **X** or may be implicit in **X.**

Equation 1.21 defines bounds on the design variables **X** and so is referred to as side constraints. Although side constraints could be included in the inequality constraint set given by Eq. 1.19, it is usually convenient to treat them separately because they define the region of search for the optimum.

The above form of stating the optimization problem is not unique, and various other statements equivalent to this are presented in the literature. For example, we may wish to state the problem as a maximization problem where we desire to maximize $F(\mathbf{X})$. Similarly, the inequality sign in Eq. 1.19 can be reversed so that $g(\mathbf{X})$ must be greater than or equal to zero. Using our notation, if a particular optimization problem requires maximization, we simply minimize $-F(\mathbf{X})$. The choice of the non-positive inequality sign on the constraints has the geometric significance that, at the optimum, the gradients of the objective and all critical constraints point away from the optimum design.

1-5 THE ITERATIVE OPTIMIZATION PROCEDURE

Most optimization algorithms require that an initial set of design variables, $\mathbf{X}^0$, be specified. Beginning from this starting point, the design is updated iteratively. Probably the most common form of this iterative procedure is given by

$$\mathbf{X}^q = \mathbf{X}^{q-1} + \alpha^* \mathbf{S}^q \tag{1.22}$$

where q is the iteration number and **S** is a vector search direction in the design space. The scalar quantity α^* defines the distance that we wish to move in direction **S**.

Note that $\alpha^* \mathbf{S}^q$ represents a perturbation on **X** so Eq. 1.22 could be written as;

$$\mathbf{X}^q = \mathbf{X}^{q-1} + \delta\mathbf{X} \tag{1.23}$$

Therefore, Eq. 1.22 is very similar to the usual engineering approach of Eq. 1.23 where we perturb an existing design to achieve some improvement.

To see how the iterative relationship given by Eq. 1.22 is applied to the optimization process, consider the two-variable problem shown in Figure 1.7.

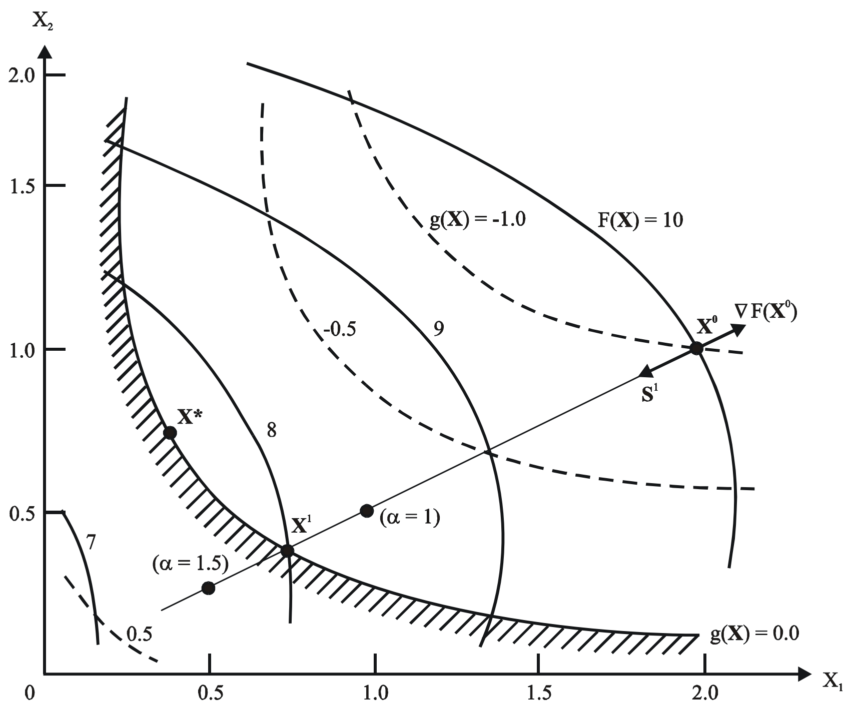


Figure 1.7 Search in direction S.

Assume we begin at point $\mathbf{X}^0$ and we wish to reduce the objective function. We will begin by searching in the direction $\mathbf{S}^1$ given by

$$\mathbf{S}^1 = \begin{Bmatrix} -1.0 \\ -0.5 \end{Bmatrix} \tag{1.24}$$

The choice of **S** is somewhat arbitrary as long as a small move in this direction will reduce the objective function without violating any constraints. In this case, the $\mathbf{S}^1$ vector is approximately the negative of the gradient of the objective function, that is, the direction of steepest descent. It is now necessary to find the scalar α^* in Eq. 1.22 so that the objective is minimized in this direction without violating any constraints.

We now evaluate **X** and the corresponding objective and constraint functions for several values of α to give

$$\alpha = 0 \qquad \mathbf{X} = \begin{Bmatrix} 2.0 \\ 1.0 \end{Bmatrix}$$

$$F(\alpha = 0) = 10.0 \qquad g(\alpha = 0) = -1.0 \tag{1.25a}$$

$$\alpha = 1.0 \qquad \mathbf{X} = \begin{Bmatrix} 2.0 \\ 1.0 \end{Bmatrix} + 1.0 \begin{Bmatrix} -1.0 \\ -0.5 \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 0.5 \end{Bmatrix}$$

$$F(\alpha = 1.0) \approx 8.4 \qquad g(\alpha = 1.0) \approx -0.2 \tag{1.25b}$$

$$\alpha = 1.5 \qquad \mathbf{X} = \begin{Bmatrix} 2.0 \\ 1.0 \end{Bmatrix} + 1.5 \begin{Bmatrix} -1.0 \\ -0.5 \end{Bmatrix} = \begin{Bmatrix} 0.50 \\ 0.25 \end{Bmatrix}$$

$$F(\alpha = 1.5) \approx 7.6 \qquad g(\alpha = 1.5) \approx 0.2 \tag{1.25c}$$

$$\alpha^* = 1.25 \qquad \mathbf{X}^* = \begin{Bmatrix} 2.0 \\ 1.0 \end{Bmatrix} + 1.25 \begin{Bmatrix} -1.0 \\ -0.5 \end{Bmatrix} = \begin{Bmatrix} 0.750 \\ 0.375 \end{Bmatrix}$$

$$F(\alpha^* = 1.25) = 8.0 \qquad g(\alpha^* = 1.25) = 0.0 \tag{1.25d}$$

where the objective and constraint values are estimated using Figure 1.7. In practice, we would evaluate these functions on the computer, and, using several proposed values of α, we would apply a numerical interpolation scheme to estimate α^*. This would provide the minimum $F(\mathbf{X})$ in this search direction which does not violate any constraints. Note that by searching in a specified direction, we have actually converted the problem from one in n variables **X** to one variable α. Thus, we refer to this as a one-dimensional search. At point $\mathbf{X}^1$, we must find a new search direction such that we can continue to reduce the objective without violating constraints. In this way, Eq. 1.22 is used repetitively until no further design improvement can be made.

From this simple example, it is seen that nonlinear optimization algorithms based on Eq. 1.22 can be separated into two basic parts. The first is determination of a direction of search, **S**, which will improve the objective function subject to constraints. The second is determination of the scalar parameter α^* defining the distance of travel in direction **S**. Each of these components plays a fundamental role in the efficiency and reliability of a given optimization algorithm, and each will be discussed in detail in later chapters.

1-6 EXISTENCE AND UNIQUENESS OF AN OPTIMUM SOLUTION

In the application of optimization techniques to design problems of practical interest, it is seldom possible to ensure that the absolute optimum design will be found. This may be because multiple solutions to the optimization problem exist or simply because numerical ill-conditioning in setting up the problem results in poor convergence of the optimization algorithm. From a practical standpoint, the best approach is usually to start the optimization process from several different initial vectors, and if the optimization results in essentially the same final design, we can be reasonably assured that this is the true optimum. It is, however, possible to check mathematically to determine if we at least have a relative minimum. In other words, we can define necessary conditions for an optimum, and we can show that under certain circumstances these necessary conditions are also sufficient to ensure that the solution is the global optimum.

In order to understand why we cannot normally guarantee that an optimum is the absolute best optimum we can find, we need to understand the concept of convexity and the conditions that exist at an optimum.

1-6.1 Convex Sets

We can understand the concept of a convex set by referring to Figure 1.8. The shaded portion of the figures represents the set under consideration. For example, for a two-dimensional space shown in the figures, any combination of the coordinates X_1, X_2 which lie in the shaded area is part of the set.

Now imagine drawing a straight line connecting any two points within the set. Examples of this are shown in Figure 1.9. If every point along this line lies within the set, the set is said to be convex. If any point on the line is outside the set, the set is said to be non convex. In Figure 1.9a, every point on the line is clearly inside the set. It is clear from the figure that, choosing any two points within the set, this is true. On the other hand, the line in Figure 1.9b is partly inside the set and partly outside the set, and so this set is said to be non convex.

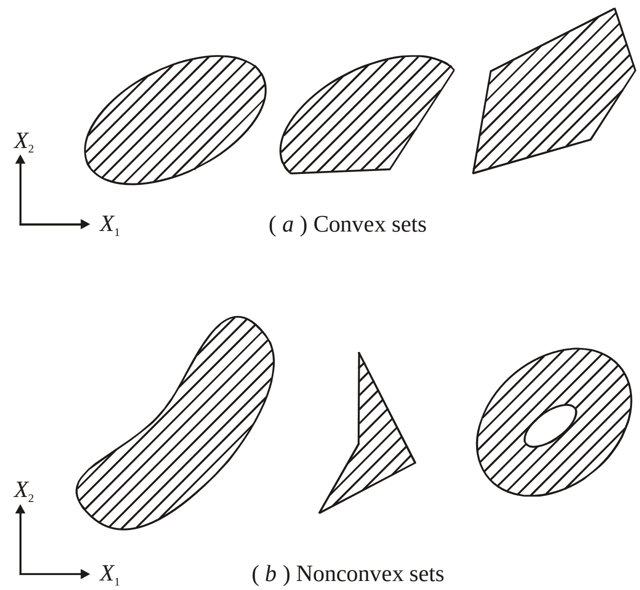


Figure 1.8 Convex and non convex sets.

We can now mathematically define a convex set. Consider any two points $\mathbf{X}^l$ and $\mathbf{X}^2$, each of which lies somewhere within the set or on the boundary. Points on a line connecting $\mathbf{X}^l$ and $\mathbf{X}^2$ are defined by

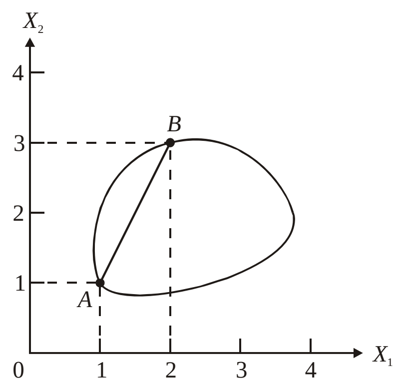
$$\mathbf{w} = \theta\mathbf{X}^1 + (1-\theta)\mathbf{X}^2 \tag{1.26}$$

where θ varies between 0 and l. If all such points **w** lie within the set for any choice of $\mathbf{X}^l$ and $\mathbf{X}^2$ in the set and any value of θ, then the set is convex. For example, in Figure 1.9a, points on the line connecting *A* and *B* are defined by

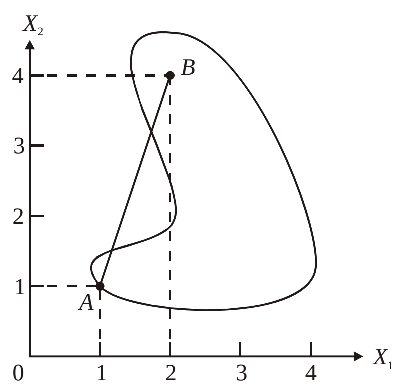
$$\mathbf{w} = \theta\begin{Bmatrix}1\\1\end{Bmatrix} + (1-\theta)\begin{Bmatrix}2\\3\end{Bmatrix} \tag{1.27}$$

Now letting θ = 0.5, for example, we have

$$\mathbf{w} = 0.5\begin{Bmatrix}1\\1\end{Bmatrix} + (1.0-0.5)\begin{Bmatrix}2\\3\end{Bmatrix} = \begin{Bmatrix}1.5\\2.0\end{Bmatrix} \tag{1.28}$$



(a) Convex set



(b) Nonconvex set

Figure 1.9 Mathematical interpretation of convex and nonconvex sets.

Here the point **w** corresponds to $X_1 = 1.5$ and $X_2 = 2.0$ and is clearly within the set.

Now consider a line connecting points *A* and *B* in Figure 1.9b. Points on this line are defined by

$$\mathbf{w} = \theta\begin{Bmatrix}1\\1\end{Bmatrix} + (1-\theta)\begin{Bmatrix}2\\4\end{Bmatrix} \tag{1.29}$$

Again letting θ = 0.5, we have

$$\mathbf{w} = 0.5\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + (1.0 - 0.5)\begin{Bmatrix} 2 \\ 4 \end{Bmatrix} = \begin{Bmatrix} 1.5 \\ 2.5 \end{Bmatrix} \tag{1.30}$$

Here $X_1 = 1.5$ and $X_2 = 2.5$, which is clearly outside the set. On the other hand, if we let θ = 0 in Eq. 1.29, we have point *A*, which is inside the set. However, since all points are not in the set, the set is said to be non convex.

1-6.2 Convex and Concave Functions

Figure 1.10 provides examples of convex functions, concave functions, and functions which are neither convex nor concave.

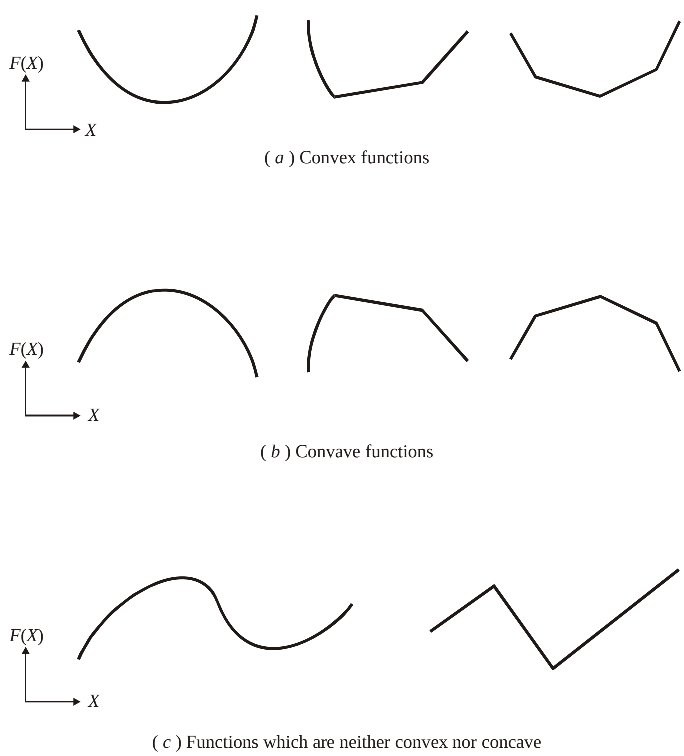


Figure 1.10 Convex and concave functions.

From Figure 1.10a, we can imagine that a convex function is one which will hold water, for example the shape of a bowl. On the other hand, a concave function, as shown in Figure 1.10b, will not hold the water (an upside down bowl). Indeed, mathematically, if a function is convex, the negative of that function is concave. The functions shown in Figure 1.10c are clearly neither convex nor concave. Now consider a convex set. A function F(**X**) bounding the set is defined mathematically as convex if for any two points $\mathbf{X}^l$ and $\mathbf{X}^2$ contained in the set

$$F[\theta\mathbf{X}^1 + (1-\theta)\mathbf{X}^2] \le \theta F(\mathbf{X}^1) + (1-\theta)F(\mathbf{X}^2) \qquad 0 \le \theta \le 1 \tag{1.31}$$

For example, consider the following optimization problem:

$$\text{Minimize:} \quad F(\mathbf{X}) = (X_1 - 0.5)^2 + (X_2 - 0.5)^2 \tag{1.32}$$

Subject to:

$$g(\mathbf{X}) = \frac{1}{X_1} + \frac{1}{X_2} - 2 \le 0 \tag{1.33}$$

The two-variable function space, showing contours of constant objective, and the constraint boundary $g(\mathbf{X}) = 0$ is shown in Figure 1.11.

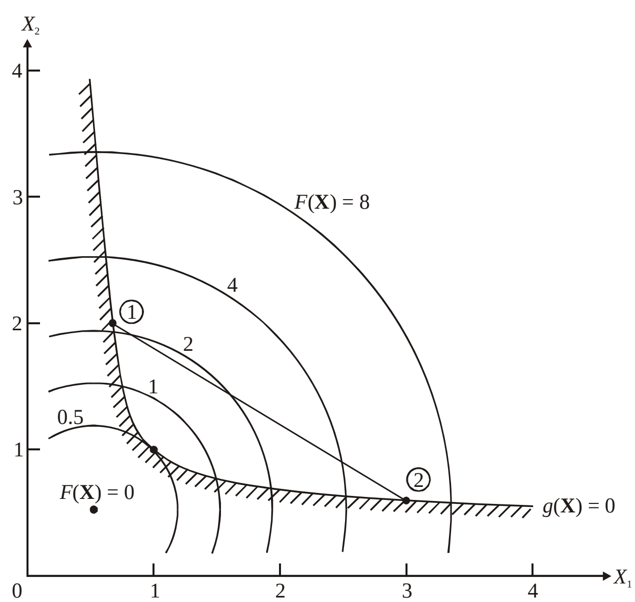


Figure 1.11 A convex design space.

Now consider points 1 and 2 on the constraint boundary defined by

$$\mathbf{X}^1 = \begin{Bmatrix} 0.667 \\ 2.000 \end{Bmatrix} \qquad \mathbf{X}^2 = \begin{Bmatrix} 3.000 \\ 0.600 \end{Bmatrix} \tag{1.34}$$

This constraint function is convex if any point on the line connecting points 1 and 2 corresponds to a value of $g(\mathbf{X})$ less than or equal to zero, where the two points may be anywhere on the constraint surface. In this case, we know that this will be true from our experience in previous examples. To show that this is mathematically true, consider a value of $\theta = 0.5$ in Eq. 1.31. Here we know that $g(\mathbf{X}^1)$ and $g(\mathbf{X}^2)$ both equal zero since they lie on the contour $g(\mathbf{X}) = 0$. Therefore, we need only to evaluate the left-hand side of Eq. 1.31, where the general function $F(\mathbf{X})$ in the equation is replaced by the constraint function $g(\mathbf{X})$. Thus

$$\theta\mathbf{X}^1 + (1-\theta)\mathbf{X}^2 = 0.5\begin{Bmatrix} 0.667 \\ 2.000 \end{Bmatrix} + (1.0 - 0.5)\begin{Bmatrix} 3.000 \\ 0.600 \end{Bmatrix} = \begin{Bmatrix} 1.833 \\ 1.300 \end{Bmatrix} \tag{1.35}$$

and

$$g[\theta\mathbf{X}^1 + (1-\theta)\mathbf{X}^2] = \frac{1}{1.833} + \frac{1}{1.300} - 2 = -0.685 \tag{1.36}$$

Indeed, we find that for any value of θ between 0 and 1, the constraint value will be less than or equal to zero, and so the constraint function is convex.

In a similar manner, consider the objective function defined by Eq. 1.32.

Evaluating $F(\mathbf{X})$ at $\mathbf{X}^1$ and $\mathbf{X}^2$ gives

$$F(\mathbf{X}^1) = 2.278 \tag{1.37a}$$

$$F(\mathbf{X}^2) = 6.260 \tag{1.37b}$$

Again considering a value of $\theta = 0.5$, we have

$$\theta F(\mathbf{X}^1) + (1-\theta)F(\mathbf{X}^2) = 4.269 \tag{1.38}$$

Also, using Eq. 1.32,

$$F[\theta\mathbf{X}^1 + (1-\theta)\mathbf{X}^2] = 2.417 \tag{1.39}$$

From Eqs. 1.38 and 1.39, we see that the inequality of Eq. 1.31 is satisfied (it is satisfied for any $0 < \theta < 1$) and so the objective function satisfies the convexity requirement.

We noted earlier that if the objective and constraint functions can be shown to be convex, then only one optimum exists and this is the global optimum. We now see that if the feasible design space is bounded by convex functions and that if the objective is itself convex, then the design space defines a convex set. While we have not proved this in a mathematical sense, we have offered sufficient geometric interpretation to at least provide insight into the fundamental concepts which underlie much of the theoretical development of nonlinear constrained optimization. The discussion of convexity and concavity given here is somewhat restricted but is sufficient for our purposes. References 6 to 13 provide further details of these concepts.

1-6.3 The Lagrangian and the Kuhn-Tucker Conditions

We can mathematically define conditions where a design is considered to be optimum. These are referred to as the Kuhn-Tucker necessary conditions.

The Kuhn-Tucker conditions define a stationary point of the Lagrangian [7, 13]

$$\mathbf{L}(\mathbf{X}, \lambda) = \boldsymbol{F}(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j \boldsymbol{g_j}(\mathbf{X}) + \sum_{k=1}^{l} \lambda_{m+k} \boldsymbol{h_k}(\mathbf{X}) \tag{1.40}$$

All three conditions are listed here for reference and state simply that if the vector **X*** defines the optimum design, the following conditions must be satisfied:

$$1.\ \mathbf{X}^* \text{ is feasible} \tag{1.41}$$

$$2.\ \lambda_j g_j(\mathbf{X}^*) = 0 \qquad j = 1, m \qquad \lambda_j \geq 0 \tag{1.42}$$

$$3.\ \nabla \boldsymbol{F}(\mathbf{X}^*) + \sum_{j=1}^{m} \lambda_j \nabla \boldsymbol{g_j}(\mathbf{X}^*) + \sum_{k=1}^{l} \lambda_{m+k} \nabla \boldsymbol{h_k}(\mathbf{X}^*) = \mathbf{0} \tag{1.43}$$

$$\lambda_j \geq 0 \tag{1.44}$$

$$\lambda_{m+k} \text{ unrestricted in sign} \tag{1.45}$$

Equation 1.41 is a statement of the obvious requirement that the optimum design must satisfy all constraints. Equation 1.42 imposes the requirement that if the constraint $g_j(\mathbf{X})$ is not precisely satisfied [that is, $g_j(\mathbf{X}) < 0$] then the corresponding Lagrange multiplier must be zero. Equations 1.43 to 1.45 define the vanishing gradient of the Lagrangian. Note that, if there are no constraints, Equation 1.43 simply states that the gradient of the objective function vanishes at the optimum.

The geometric significance of the Kuhn-Tucker conditions can be understood by referring to Figure 1.12, which shows a two-variable minimization problem with three inequality constraints.

We see from the figure that **X*** is feasible so the first condition is met. Constraint $g_3(\mathbf{X}^*)$ is not critical and so, from Eq. 1.42, $\lambda_3 = 0$. Since $g_1(\mathbf{X}^*) = 0$ and $g_2(\mathbf{X}^*) = 0$, the second Kuhn-Tucker condition is satisfied identically with respect to these constraints.

Equation 1.43 requires that, if we multiply the gradient of each critical constraint [$g_1(\mathbf{X}^*)$ and $g_2(\mathbf{X}^*)$] by its corresponding Lagrange multiplier, the vector sum of the result must equal the negative of the gradient of the objective function. Thus

$$\nabla \boldsymbol{F}(\mathbf{X}^*) + \lambda_1 \nabla \boldsymbol{g}_1(\mathbf{X}^*) + \lambda_2 \nabla \boldsymbol{g}_2(\mathbf{X}^*) = \mathbf{0} \tag{1.46a}$$

$$\lambda_1 \geq 0 \qquad \lambda_2 \geq 0 \tag{1.46b}$$

Therefore, each of the Kuhn-Tucker necessary conditions is satisfied.

In the problem of Figure 1.12, the Lagrange multipliers are uniquely determined from the gradients of the objective and the active constraints. However, we can easily imagine situations where this is not so. For example, assume that we have defined another constraint, $g_4(\mathbf{X})$, which happens to be identical to $g_1(\mathbf{X})$ or perhaps a constant times $g_1(\mathbf{X})$. The constraint boundaries $g_1(\mathbf{X}) = 0$ and $g_4(\mathbf{X}) = 0$ would now be the same and the Lagrange multipliers λ_1 and λ_4 can have any combination of values which satisfy the vector addition shown in the figure. Thus, we can say that one of the constraints is redundant. As another example, we can consider a constraint $g_5(\mathbf{X})$ which is independent of the other constraints, but at the optimum in Figure 1.12, constraints $g_1(\mathbf{X})$, $g_2(\mathbf{X})$ and $g_5(\mathbf{X})$ are all critical. Now, we may pick many combinations of λ_1, λ_2, and λ_5 which will satisfy the Kuhn-Tucker conditions so that, while all constraints are independent, the Lagrange multipliers are not unique. These special cases do not detract from the usefulness of the Kuhn-Tucker conditions in optimization theory. It is only necessary that we account for these possibilities when using algorithms that require calculation of the Lagrange multipliers.

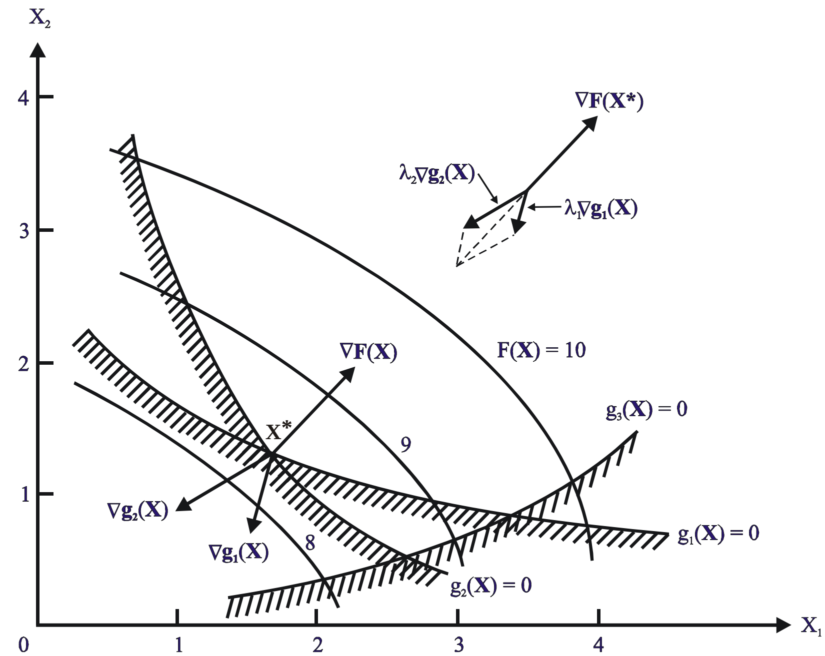


Figure 1.12 Geometric interpretation of the Kuhn-Tucker conditions.

The question now arises, when are the Kuhn-Tucker conditions both necessary and sufficient to define a global optimum? The answer is simply that if the design space is convex, the Kuhn-Tucker conditions define a global optimum. If the design space is not convex, the Kuhn-Tucker conditions only guarantee that a relative optimum has been found.

For a design space to be convex, the Hessian matrix of the objective function and all constraints must be positive definite for all possible combinations of the design variables, where the Hessian matrix is the matrix of second partial derivatives of the function with respect to the design variables,

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 F(\mathbf{X})}{\partial X_1^2} & \frac{\partial^2 F(\mathbf{X})}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 F(\mathbf{X})}{\partial X_1 \partial X_n} \\ \frac{\partial^2 F(\mathbf{X})}{\partial X_2 \partial X_1} & \frac{\partial^2 F(\mathbf{X})}{\partial X_2^2} & \cdots & \frac{\partial^2 F(\mathbf{X})}{\partial X_2 \partial X_n} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial^2 F(\mathbf{X})}{\partial X_n \partial X_1} & \frac{\partial^2 F(\mathbf{X})}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 F(\mathbf{X})}{\partial X_n^2} \end{bmatrix} \quad (1.47)$$

Positive definiteness means that this matrix has all positive eigenvalues. If the gradient is zero and the Hessian matrix is positive definite for a given **X**, this insures that the design is at least a relative minimum, but again it does not insure that the design is a global minimum. The design is only guaranteed to be a global minimum if the Hessian matrix is positive definite for all possible values of the design variables **X.** This can seldom be demonstrated in practical design applications. We must usually be satisfied with starting the design from various initial points to see if we can obtain a consistent optimum and therefore have reasonable assurance that this is the true minimum of the function. However, an understanding of the requirements for a unique optimal solution is important to provide insight into the optimization process. Also, these concepts provide the basis for the development of many of the more powerful algorithms which we will be discussing in later chapters.

1-6.4 Calculating the Lagrange Multipliers

Now consider how we might calculate the values of the Lagrange Multipliers at the optimum. First, we know that if a constraint value is non-zero (within a small tolerance), then from Eq. 1.42, the corresponding Lagrange multiplier is equal to zero. For our purposes here, both inequality and equality constraints are treated the same, so we can treat them all together. It is only important to remember that the equality constraints will always be active at the optimum and that they can have positive or negative Lagrange Multipliers. Also, assuming all constraints are independent, the number of active constraints will be less than or equal to the number of design variables. Thus, Eq. 1.43 often has fewer unknown parameters, λ_j than equations.

Because precise satisfaction of the Kuhn-Tucker conditions may not be reached, we can rewrite Eq. 1.43 as

$$\nabla F(\mathbf{X}^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\mathbf{X}^*) = \mathbf{R} \tag{1.48}$$

where the equality constraints are omitted for brevity and **R** is the vector of residuals.

Now, because we want the residuals as small as possible (if all components of **R** = 0, the Kuhn-Tucker conditions are satisfied precisely), we can minimize the square of the magnitude of **R**.
Let

$$\mathbf{B} = \nabla \mathbf{F}(\mathbf{X}^*) \tag{1.49a}$$

and

$$\mathbf{A} = \left[\nabla g_1(\mathbf{X}^*) \ \nabla g_2(\mathbf{X}^*) \ \ldots \ \nabla g_M(\mathbf{X}^*) \right] \tag{1.49b}$$

where M is the set of active constraints.

Substituting Eqs. 1.49a and 1.49b into Eq. 1.48,

$$\mathbf{B} + \mathbf{A}\lambda = \mathbf{R} \tag{1.50}$$

Now, because we want the residuals as small as possible (if all components of **R** = 0, the Kuhn-Tucker conditions are satisfied precisely), we can minimize the square of the magnitude of **R**.

$$\text{Minimize} \quad \mathbf{R}^T\mathbf{R} = \mathbf{B}^T\mathbf{B} + 2\lambda^T\mathbf{A}^T\mathbf{B} + \lambda^T\mathbf{A}^T\mathbf{A}\lambda \tag{1.51}$$

Differentiating Eq. 1.51 with respect to λ and setting the result to zero gives

$$2\mathbf{A}^T\mathbf{B} + 2\mathbf{A}^T\mathbf{A}\lambda = 0 \tag{1.52}$$

from which

$$\lambda = -[\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{B} \tag{1.53}$$

Now if all components of λ corresponding to inequality constraints are non-negative, we have an acceptable estimate of the Lagrange multipliers. Also, we can substitute Eq. 1.53 into Eq. 1.50 to estimate how precisely the Kuhn-Tucker conditions are met. If all components of the residual vector, **R**, are very near zero, we know that we have reached at least a relative minimum.

1-6.5 Sensitivity of the Optimum to Changes in Constraint Limits

The Lagrange multipliers have particular significance in estimating how sensitive the optimum design is to the active constraints. It can be shown that the derivative of the optimum objective with respect to a constraint is just the value of the Lagrange multiplier of that constraint, so

$$\frac{\partial}{\partial g_j(\mathbf{X}^*)} F(\mathbf{X}^*) = \lambda_j \tag{1.54}$$

or, in more useful form;

$$F(\mathbf{X}^*, \delta g_j) = F(\mathbf{X}^*) + \lambda_j \delta g_j \tag{1.55}$$

If we wish to change the limits on a set of constraints, J, Eq. 1.55 is simply expanded as

$$F(\mathbf{X}^*, \delta g_j) = F(\mathbf{X}^*) + \sum_{j \varepsilon J} \lambda_j \delta g_j \tag{1.56}$$

Remember that Eq. 1.55 is the sensitivity with respect to g_j. In practice, we may want to know the sensitivity with respect to bounds on the response.

Assume we have normalized an upper bound constraint

$$g_j = \frac{R - R_u}{|R_u|} \tag{1.57}$$

where R is the response and R_u is the upper bound.

$$F(\mathbf{X}^*, \delta R_u) = F(\mathbf{X}^*) - \lambda_j \frac{\delta R_u}{|R_u|} \tag{1.58}$$

Similarly, for lower bound constraints

$$F(\mathbf{X}^*, \delta R_l) = F(\mathbf{X}^*) + \lambda_j \frac{\delta R_l}{|R_l|} \tag{1.59}$$

Therefore, the Lagrange multipliers tell us the sensitivity of the optimum with respect to a relative change in the constraint bounds, while the Lagrange multipliers divided by the scaling factor (usually the magnitude of the bound) give us the sensitivity to an absolute change in the bounds.

Example 1-4 Sensitivity of the Optimum

Consider the constrained minimization of a simple quadratic function with a single linear constraint.

Minimize $$F(\mathbf{X}) = X_1^2 + X_2^2 \tag{1.60}$$

Subject to; $$g(\mathbf{X}) = \frac{2 - (X_1 + X_2)}{2} \le 0 \tag{1.61}$$

At the optimum;

$$\mathbf{X}^* = \begin{Bmatrix} 1.0 \\ 1.0 \end{Bmatrix} \qquad \mathrm{F}(\mathbf{X}^*) = 2.0 \qquad g(\mathbf{X}^*) = 0.0 \tag{1.62a}$$

and

$$\nabla F(\mathbf{X}^*) = \begin{Bmatrix} 2.0 \\ 2.0 \end{Bmatrix} \qquad \nabla g(\mathbf{X}^*) = \begin{Bmatrix} -0.5 \\ -0.5 \end{Bmatrix} \qquad \lambda = 4.0 \tag{1.62b}$$

Now, assume we wish to change the lower bound on g from 2.0 to 2.1. From Eq. 1.59 we get

$$F(\mathbf{X}^*, \delta R_l) = 2.0 + 4.0\begin{Bmatrix} \frac{0.1}{2.0} \end{Bmatrix} = 2.20 \tag{1.63}$$

The true optimum for this case is

$$\mathbf{X}^* = \begin{Bmatrix} 1.05 \\ 1.05 \end{Bmatrix} \qquad \mathrm{F}(\mathbf{X}^*, \delta R_l) = 2.205 \tag{1.64}$$

1-7 CONCLUDING REMARKS

In assessing the value of optimization techniques to engineering design, it is worthwhile to review briefly the traditional design approach. The design is often carried out through the use of charts and graphs which have been developed over many years of experience. These methods are usually an efficient means of obtaining a reasonable solution to traditional design problems. However, as the design task becomes more complex, we rely more heavily on the computer for analysis. If we assume that we have a computer code capable of analyzing our proposed design, the output from this program will provide a quantitative indication of the acceptability and optimality of the design. We may change one or more design variables and rerun the computer program to see if any design improvement can be obtained. We then take the results of many computer runs and plot the objective and constraint values versus the various design parameters. From these plots we can interpolate or extrapolate to what we believe to be the optimum design. This is essentially the approach that was used to obtain the optimum constrained minimum of the tubular column shown in Figure 1.5, and this is certainly an efficient and viable approach when the design is a function of only a few variables. However, if the design exceeds three variables, the true optimum may be extremely difficult to obtain graphically. Thus, assuming the computer code exists for the analysis of the proposed design, automation of the design process becomes an attractive alternative. Mathematical programming simply provides a logical framework for carrying out this automated design process. Some advantages and limitations to the use of numerical optimization techniques are listed here.

1-7.1 Advantages of Numerical Optimization

- A major advantage is the reduction in design time – this is especially true when the same computer program can be applied to many design projects.
- Optimization provides a systematized logical design procedure.
- We can deal with a wide variety of design variables and constraints which are difficult to visualize using graphical or tabular methods.
- Optimization virtually always yields some design improvement.
- It is not biased by intuition or experience in engineering. Therefore, the possibility of obtaining improved, nontraditional designs is enhanced.
- Optimization requires a minimal amount of human-machine interaction.

1-7.2 Limitations of Numerical Optimization

- Computational time increases as the number of design variables increases. If one wishes to consider all possible design variables, the cost of automated design may be prohibitive. Also, as the number of design variables increases, these methods tend to become numerically ill-conditioned.
- Optimization techniques have no stored experience or intuition on which to draw. They are limited to the range of applicability of the analysis program.
- If the analysis program is not theoretically precise, the results of optimization may be misleading, and therefore the results should always be checked very carefully. Optimization will invariably take advantage of analysis errors in order to provide mathematical design improvements.
- Most optimization algorithms have difficulty in dealing with discontinuous functions. Also, highly nonlinear problems may converge slowly or not at all. This requires that we be particularly careful in formulating the automated design problem.
- It can seldom be guaranteed that the optimization algorithm will obtain the globally optimum design. Therefore, it may be desirable to restart the optimization process from several different points to provide reasonable assurance of obtaining the global optimum.
- Because many analysis programs were not written with automated design in mind, adaptation of these programs to an optimization code may require significant reprogramming of the analysis routines.

1-7.3 Summary

Optimization techniques, if used effectively, can greatly reduce engineering design time and yield improved, efficient, and economical designs. However, it is important to understand the limitations of optimization techniques and use these methods as only one of many tools at our disposal.

Finally, it is important to recognize that, using numerical optimization techniques, the precise, absolute best design will seldom if ever be achieved. Expectations of achieving the absolute “best” design will invariably lead to “maximum” disappointment. We may better appreciate these techniques by replacing the word “optimization” with “design improvement,” and recognize that a convenient method of improving designs is an extremely valuable tool.

REFERENCES

1. Stodolsky, A. V., Cuenca, R. and Gaines, L., Life-Cycle Energy Savings Potential from Aluminum-Intensive Vehicles, Proc. 1995 Total Life Cycle Conference & Exposition, October 16-19, 1995, Vienna, Austria.
2. Schmit, L. A.: Structural Design by Systematic Synthesis, *Proceedings, 2nd Conference on Electronic Computation, ASCE*, New York, pp. 105 – 122, 1960.
3. Schmit, L. A.: Structural Synthesis – Its Genesis and Development, *AIAA J.*, vol. 10, no. 10, pp. 1249 – 1263, October 1981.
4. Vanderplaats, G. N.: “Numerical Optimization Techniques for Engineering Design,” 4th Edition, Vanderplaats Research & Development, Inc., Colorado Springs, CO, 2005.
5. Zoutendijk, G.: “Methods of Feasible Directions,” Elsevier, Amsterdam, 1960.
6. Himmelblau, D. M.: “Applied Nonlinear Programming,” McGraw-Hill, New York, 1972.
7. Kuhn, H. W. and Tucker, A. W., Nonlinear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, Calif., 1950, pp. 481-492.
8. Gottfried, B. S. and J. Weisman: “Introduction to Optimization Theory,” Prentice-Hall, Englewood Cliffs, N.J., 1973.
9. Wismer, D. A. and R. Chattergy: “Introduction to Nonlinear Optimization,” North Holland, New York, 1978.
10. Zangwill, W. I.: “Nonlinear Programming, A Unified Approach,” Prentice-Hall, Englewood Cliffs, N.J., 1969.
11. Pierre, D. A. and M. J. Lowe: “Mathematical Programming Via Augmented Lagrangians – An Introduction with Computer Programs,” Addison-Wesley, Reading, Mass., 1975.
12. Fox, R. L.: “Optimization Methods for Engineering Design,” Addison-Wesley, Reading, Mass., 1971.
13. Zangwill, W. I.: “Nonlinear Programming: A Unified Approach,” Prentice-Hall, Englewood Cliffs, N.J., 1969.

PROBLEMS

1-1 Consider the 500-N weight hanging by a cable, as shown in Figure 1.13. A horizontal force, $F = 100$ N, is applied to the weight. Under this force, the weight moves from its original position at *A* to a new equilibrium position at *B*. Ignore the cable weight. The equilibrium position is the one at which the total potential energy PE is a minimum, where PE = $WY - FX$.

a. Write an expression for PE in terms of the horizontal displacement X alone.

b. Write an expression for PE in terms of the angle θ alone.

c. Plot a graph of PE versus θ between $\theta = 0°$ and $\theta = 45°$.

d. Find the angle corresponding to the minimum value of PE both graphically and analytically. Prove that this is a minimum.

e. Using statics, verify that the θ at which PE is minimum is indeed the equilibrium position.

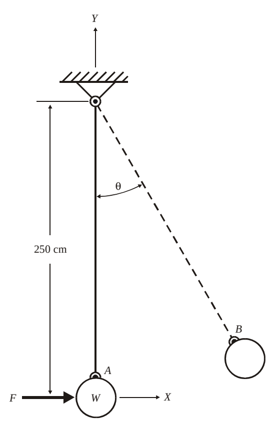


Figure 1.13

1-2 Given the unconstrained function

$$F = X_1 + \frac{1}{X_1} + X_2 + \frac{1}{X_2}$$

a. Calculate the gradient vector and the Hessian matrix.

b. At what combinations of X_1 and X_2 is the gradient equal to zero?

c. For each point identified in part *b*, is the function a minimum, a maximum, or neither?

1-3 Given the unconstrained function,

$$F = X_1^2 + \frac{1}{X_1} + X_2 + \frac{1}{X_2}$$

a. At $X_1 = 2$ and $X_2 = 2$, calculate the gradient of *F*.

b. At $X_1 = 2$ and $X_2 = 2$, calculate the direction of steepest descent.

c. Using the direction of steepest descent calculated in part *b*, update the design by the standard formula

$$\mathbf{X}^1 = \mathbf{X}^0 + \alpha \mathbf{S}^1$$

Evaluate X_1, X_2 and F for $\alpha = 0$, 0.2, 0.5, and 1.0 and plot the curve of F versus α.

d. Write the equation for *F* in terms of α alone. Discuss the character of this function.

e. From part *d*, calculate $dF/d\alpha$ at $\alpha = 0$.

f. Calculate the scalar product $\nabla \mathbf{F} \bullet \mathbf{S}$ using the results of parts *a* and *b* and compare this with the result of part *e*.

1-4 Consider the constrained minimization problem:

Minimize: $F = (X_1 - 1)^2 + (X_2 - 1)^2$

Subject to:

$$X_1 + X_2 \leq 0.5$$

$$X_1 \geq 0.0$$

a. Sketch the two-variable function space showing contours of $F = 0$, 1, and 4 as well as the constraint boundaries.

b. Identify the unconstrained minimum of *F* on the figure.

c. Identify the constrained minimum on the figure.

d. At the constrained minimum, what are the Lagrange multipliers?

1-5 Given the ellipse $(X/2)^2 + Y^2 = 4$, it is desired to find the rectangle of greatest area which will fit inside the ellipse.

a. State this mathematically as a constrained minimization problem. That is, set up the problem for solution using numerical optimization.

b. Analytically determine the optimum dimensions of the rectangle and its corresponding area.

c. Draw the ellipse and the rectangle on the same figure.

1-6 Given the following optimization problem;

Minimize: $F = X_1 + X_2$

Subject to:

$$g_1 = 2 - X_1^2 - X_2 \le 0$$
$$g_2 = 4 - X_1 - 3X_2 \le 0$$
$$g_3 = -30 + X_1 + X_2^4 \le 0$$

The current design is $X_1 = 1$, $X_2 = 1$.

a. Does this design satisfy the Kuhn-Tucker necessary conditions for a constrained optimum? Explain.

b. What are the values of the Lagrange multipliers at this design point?

1-7 Given the following optimization problem:

Minimize: $F = 10 + X_1 + X_2$

Subject to:

$$g_1 = 5 - X_1 - 2X_2 \le 0$$
$$g_2 = \frac{1}{X_1} + \frac{1}{X_2} - 2 \le 0$$
$$X_1 \ge 0.0 \qquad X_2 \ge 0.0$$

a. Plot the two-variable function space showing contours of $F = 10$, 12, and 14 and the constraint boundaries $g_1 = 0$ and $g_2 = 0$.

b. Identify the feasible region.

c. Identify the optimum.