In this chapter, we discuss design optimization—one of the mainstream methods in support of engineering design. In design optimization, we minimize (or maximize) an objective function subject to performance constraints by varying a set of design variables, such as part dimensions, material properties, and so on. Usually we deal with one objective function in carrying out the so-called single-objective optimization problem, which is the subject of this chapter. Often, there are more than one objective functions to be minimized simultaneously. Such problems are called multi-objective (or multi-criteria) optimization (MOO) problems, which will be discussed in Chapter 5. The objective function and performance constraints are usually extracted from or defined for a physical problem of a single engineering discipline. For example, if we are designing a load-bearing component, in which we minimize the structural weight and subject to constraints characterizing structural strength performance, we are solving a structural optimization problem that is single disciplinary. In many situations, especially in the context of e-Design, we are dealing with designing a product or a system that involves multiple engineering disciplines, including structural, kinematic and dynamic, manufacturing, product cost, and so forth. Such design problems are called multi-disciplinary optimization (MDO), which will be briefly discussed in Chapter 5, and Projects P5 and S5 through a single-piston engine example.

The existence of optimization methods can be traced to the days of Newton, Lagrange, and Cauchy. The development of differential calculus methods for optimization was possible because of the contributions of Newton and Leibnitz to calculus. The foundations of the calculus of variations, which deals with the minimization of functions, were laid by Bernoulli Euler, Lagrange, and Weistrass. The method of optimization for constrained problems, which involve the addition of unknown multipliers, became known by the name of its inventor, Lagrange. Cauchy made the first application of the steepest descent method to solve unconstrained optimization problems. By the middle of the twentieth century, high-speed digital computers made implementation of the numerical optimization techniques possible and stimulated further research on newer methods. Some major developments include the work of Kuhn and Tucker in the 1950s on the necessary and sufficient conditions for the optimal solution of programming problems, which laid the foundation for later research in nonlinear programming. Early development focused on gradient-based algorithms that employ gradient information of the objective and constraint functions in searching for optimal solutions. In the late twentieth century, non-gradient-based approaches, such as simulated annealing, genetic algorithms, and neural network methods, representing a new class of mathematical programming techniques, came into prominence. Today, optimization techniques are widely employed to support engineering design. The applications for mechanical system or component designs include car suspensions for durability, car bodies for noise/vibration/harshness, pumps and turbines for maximum efficiency, and so forth. There are plenty of applications beyond mechanical designs, such as optimal production planning, controlling, and scheduling; optimum pipeline networks for process industry; controlling the waiting and idle times in production lines to reduce the cost of production; optimum design of control systems; among many others.

This chapter offers an introductory discussion to the single-objective optimization problems. We provide a brief introduction to design optimization for those who are not familiar with the subject. Basic concepts include problem formulation, optimality conditions, and graphical solutions using simple problems for which optimal solutions can be found analytically. In addition, we discuss both linear and nonlinear programming and offer a mathematical basis for design problem formulation and solutions. We include both gradient-based and non-gradient approaches for solving optimization problems. For those who are interested in learning more about optimization or are interested in entering this technical area for research, there are several excellent references in the literature (Arora 2012; Vanderplaats 2005; Rao 2009).
Most of the solution techniques require a large amount of evaluations for the objective and constraint functions. The disciplinary or physical models that are employed for function evaluations are often very complex and must be solved numerically (e.g., using finite element methods). It can take significant amount of computation time for a single function evaluation. As a result, the solution process for an optimization problem can be extremely time-consuming. Therefore, in this chapter, readers should see clearly the limitations of the non-gradient approaches in terms of the computational efforts for large-scale design problems. The gradient-based approaches are more suitable to the typical problems in the context of e-Design.

In addition to offering optimization concept and solution techniques, we use functions provided in MATLAB to solve example problems that are beyond hand calculations. MATLAB scripts developed for numerical examples are provided for reference in Appendix A at the end of this chapter. Moreover, we offer a discussion on the practical aspects of carrying out design optimization for practical engineering problems, followed by a brief review on commercial software tools in hope of ensuring that readers are aware of them and their applicability to engineering applications. We include three case studies to illustrate the technical aspects of performing design optimization using commercial computer-aided design (CAD) and computer-aided engineering (CAE) software. In addition, a simple cantilever beam example, modeled in both Pro/MECHANICA Structure and SolidWorks Simulation, is offered to provide step-by-step details in using the respective software to carry out design optimization. Example files can be found on this book’s companion Website (booksite.elsevier.com/). Detailed instructions on using these models and steps for carrying out optimization are given in Projects P5 and S5.

Overall, the objectives of this chapter are (1) to provide basic knowledge in optimization and help readers understand the concept and solution techniques, (2) to familiarize readers with practical applications of the optimization techniques and be able to apply adequate techniques for solving optimization problems using MATLAB, (3) to introduce readers to popular optimization software that is commercially available and typical practical engineering problems in case studies, and (4) to help readers become familiar with the optimization capabilities provided in Pro/MECHANICA Structure and SolidWorks Simulation for basic applications.

We hope this chapter helps readers to learn the basics of design optimization, become acquainted with the optimization subject in general, and be able to move on to the follow-up chapters to gain an in-depth knowledge of the broad topics of design methods.

### 3.1 INTRODUCTION

Many engineering design problems can be formulated mathematically as single-objective optimization problems, in which one single objective function is to be minimized (or maximized) subject to a set of constraints derived from requirements in, for example, product performance or physical sizes. As a simple example, we design a beer can for a maximum volume with a given amount of surface area, as shown in Figure 3.1a. The geometry of the can is simplified as the cylinder shown in Figure 3.1b with two geometric dimensions, radius \( r \) and height \( h \). The volume and surface area of the can are \( V = \pi r^2 h \) and \( A = 2\pi r(r + h) \), respectively. The can design problem can be formulated mathematically as follows:

Maximize : \[ V(r, h) = \pi r^2 h \]  
Subject to : \[ A(r, h) = \pi r(r + 2h) = A_0 \] \[ 0 < r, \quad 0 < h \]
where $A_0$ is the given amount of surface area. In this case, $V(r, h)$ is the objective function to be maximized, and $A(r, h) = 2\pi r(r + h) = A_0$ is the constraint, or more precisely, an equality constraint to be satisfied. The radius $r$ and height $h$ are design variables. Certainly, both $r$ and $h$ must be greater than 0.

Solving the beer can design problem is straightforward because both the objective and constraint functions are expressed explicitly in terms of the design variables, $r$ and $h$. For example, from Eq. 3.1b, we have

$$r = -h + \sqrt{h^2 + \frac{A_0}{\pi}}$$

Bringing this equation back to Eq. 3.1a, we have

$$V = \pi h \left(-h + \sqrt{h^2 + \frac{A_0}{\pi}}\right)^2$$  \tag{3.2}$$

We hence converted a constrained optimization problem of two design variables to an unconstrained problem of single design variable $h$, which is much easier to solve. How do we solve Eq. 3.2 to find the optimal solution, in this case maximum volume of the beer can? One approach is to graph the volume function in terms of design variable $h$. For example, if the area is given as $A_0 = \pi$, Eq. 3.2 can be graphed as shown in Figure 3.2, for example, using MATLAB. Readers are referred to Appendix A (Script 1) to find the script that graphs the curve. From the graph, we can easily see that when the height $h$ is about 1.2, volume reaches its maxima around 4.8.

The maximum of Eq. 3.2 can also be found by finding the solution of the derivative of Eq. 3.2, $dV/dh = 0$, and checking if the solution $h$ satisfies $d^2V/dh^2 < 0$ (or if the objective function is concave), as we learned in Calculus. As seen in Figure 3.2, at $h = 1.2$, we have $dV/dh = 0$. Also, the function curve is concave in the neighborhood of $h = 1.2$. Hence, the rate of slope change is negative; that is, $d^2V/dh^2 < 0$.

The beer can design problem represents the simplest kind of optimization problem, in which both the objective and constraint are explicit functions of design variables. In most engineering problems, objective and constraint functions are too complex to be expressed in terms of design variables explicitly. In many cases, they have to be evaluated using numerical methods, for example, finite element methods. When the physical model (created in computer) to be solved for the objective and constraint function evaluations is large, the solution process for an optimization problem can be extremely time-consuming. This is especially true for multidisciplinary problems, in which
performance constraints are evaluated through intensive computations of multiple physical models involved in characterizing the physical behavior of the product. As a result, many methods and algorithms are developed to support design optimization by reducing computation time through minimizing the number of function evaluations.

In this chapter, we use simple and analytical examples to illustrate the optimization concept and some of the most popular solution techniques. When you review the concept and solution methods, please keep in mind that the objective and constraint functions are not necessarily expressed in terms of design variables explicitly. We discuss optimization problem formulation in Section 3.2, and then we introduce optimality conditions in Section 3.3. We include three basic solution approaches: the graphical methods in Section 3.4, and the gradient-based methods for constrained and unconstrained problems in Sections 3.5 and 3.6, respectively. Two popular solution techniques using a non-gradient approach, genetic algorithm and simulated annealing, are briefly discussed in Section 3.7. In Section 3.8, we discuss practical aspects of solving engineering optimization problems, followed by a short review on optimization software in Section 3.9. In Section 3.10, we include three case studies and followed by a tutorial example in Section 3.11.

3.2 OPTIMIZATION PROBLEMS

An optimization problem is a problem in which certain parameters (design variables) need to be determined to achieve the best measurable performance (objective function) under given constraints. In Section 3.1, we introduced the basic idea of design optimization using a very simple beer can example. The design problem is straightforward to formulate, analytical expressions are available for the objective and constraint functions, and the optimal solution is obtained graphically. Although the simple problem illustrates the basic concepts of design optimization, in reality, design problems are much more involved in many aspects, including problem formulation, solutions, and results interpretation.
3.2.1 PROBLEM FORMULATION

In general, a single-objective optimization problem can be formulated mathematically as follows:

\[
\begin{align*}
\text{Minimize} & : & f(x) \\
\text{Subject to} & : & g_i(x) \leq 0, & i = 1, m \\
& & h_j(x) = 0, & j = 1, p \\
& & x_k^\ell \leq x_k \leq x_k^u, & k = 1, n
\end{align*}
\]

where \( f(x) \) is the objective function or goal to be minimized (or maximized); \( g_i(x) \) is the \( i \)th inequality constraint; \( m \) is the total number of inequality constraint functions; \( h_j(x) \) is the \( j \)th equality constraint; \( p \) is the total number of equality constraints; \( x \) is the vector of design variables, \( x = [x_1, x_2, \ldots, x_n]^T \); \( n \) is the total number of design variables; and \( x_k^\ell \) and \( x_k^u \) are the lower and upper bounds of the \( k \)th design variable \( x_k \), respectively. Note that Eq. 3.3d is called side constraints. Equation (3.3) can also be written in a shorthand version as

\[
\min_{x \in S} f(x) \tag{3.4}
\]

in which \( S \) is called a feasible set or feasible region, defined as

\[
S = \{ x \in \mathbb{R}^n | g_i(x) \leq 0, & i = 1, m; \ h_j(x) = 0, & j = 1, p; \text{ and } x_k^\ell \leq x_k \leq x_k^u, & k = 1, n \} \tag{3.5}
\]

Note that not all design problems can be formulated mathematically like those in Eq. 3.3 (or Eqs 3.4 and 3.5). However, whenever possible, readers are encouraged to formulate a design problem like the above in order to proceed with solution techniques for solving the design problem.

There are several steps for formulating a design optimization problem. First, the designer or the design team must develop a problem statement. What is the designer trying to accomplish? Is there a clear set of criteria or metrics to determine if the design of the product is successful at the end? Next, the designer or team must collect data and information relevant to the design problem. Is all the information needed to construct and solve the physical models of the design problem available? For example, if the design problem involves structural analysis, is the external load determined accurately?

After the above steps are completed, the designer or team faces two important tasks: design problem formulation and physical modeling.

Design problem formulation transcribes a verbal description (usually qualitative) into a quantitative statement in a mathematical form that defines the optimization problem, like that of Eq. 3.3. This task involves converting qualitative design requirements into quantitative performance measures and identifying objective function (or functions) that determine the performance or outcome of the physical system, such as costs, weight, power output, and so forth. In the meantime, the team identifies the performance constraints that the design must satisfy in accordance with the problem statement or functional and physical requirements identified at the beginning. With the objective and constraint functions identified, the team finds dimensions among other parameters (such as material properties) that largely influence the performance measures and chooses them as design variables with adequate upper or lower bounds. Performance measures are those from which both objective and performance constraints are chosen. Essentially, the goal is to formulate a design problem mathematically, as shown in Eq. 3.3.
The physical modeling involves the construction of mathematical models or equations that describe the physical behavior of the system being designed. Note that, in general, a physical problem is too complex to analyze and must be simplified so that it can be solved either analytically or numerically.

Next, we use the cross bar of the traffic light shown in Figure 3.3 to further illustrate the steps in formulating a design optimization problem. In this project, the goal is to design a structurally strong cross bar for the traffic light with a minimum cost. We have collected information needed for the design of the cross bar, including material to be used and its mechanical properties. Additionally, the length of the cross bar has to be 30 ft. to meet a design requirement. The cross bar can be modeled as a cantilever beam with self-weight and a point load due to the light box at the tip of the beam. Without involving detailed geometric modeling and finite element analysis (FEA), we first simplify the problem by assuming the cross bar as a straight beam with constant cross-section. We further assume the weight of the light box is significantly larger than the weight of the cross bar; therefore, the self-weight of the beam can be neglected. We also assume a solid cross-section of rectangular shape with width $w$ and height $h$. The simplified cross bar is shown schematically in Figure 3.4.

Because we want the beam to be strong and yet as inexpensive as possible, we define the volume of the beam as the objective function to be minimized and we constrain the stress of the beam to be less than its yield strength. The rational is that a beam of lesser volume consumes less material; therefore, it is less expensive. Next, we pick the width and height as design variables. At this point, we need to construct a physical model with equations that govern the behavior of the beam and relate the design variables to the objective and constraint functions. For the objective function, the equation is straightforward; that is, $V(w, h) = whl$. For the stress measure, we use the bending stress equation of
the cantilever beam; that is, \( \sigma(w, h) = \frac{6P\ell}{wh^2} \). Thereafter, the design problem of the cantilever beam example can be formulated mathematically as

\[
\text{Minimize :} \quad V(w, h) = wh \ell \quad (3.6a)
\]

\[
\text{Subject to :} \quad \sigma(w, h) = \frac{6P\ell}{wh^2} - S_y \leq 0 \quad (3.6b)
\]

\[
w > 0, \quad h > 0 \quad (3.6c)
\]

Note that in Eq. 3.6b, \( S_y \) is the material yield strength.

In general, physical modeling is not as straightforward as that shown in the cantilever beam example. Numerical simulations, such as finite element methods, are employed for the evaluation of product performance. For topics in physical modeling using numerical simulations, readers are encouraged to refer to Chang (2013a). Function evaluations that support design optimization are carried out by the analysis of the physical models. If cost is involved in the design problem formulation, the readers are referred to Chang (2013b) or other references, such as Ostwald and McLaren (2004), Ostwald (1991), and Clark and Lorenzoni (1996), for more information.

### 3.2.2 PROBLEM SOLUTIONS

Once the problem is formulated, we need to solve for an optimal solution. The solution process involves selecting a most suitable optimization technique or algorithm to find an optimal solution. In general, an optimization problem is solved numerically, in which it is required that designers understand the basic concept and the pros and cons of various optimization techniques. For problems with two or less design variables, the graphical method is an excellent choice. Some simple problems, where objective and constraint functions are written explicitly in terms of design variables, can be solved by using the necessary and sufficient conditions of the optimality. All of these solution techniques will be discussed in this chapter. Most importantly, once an optimal solution is obtained, designers must analyze, interpret, and validate the solutions before presenting the results to others.

For the cantilever beam example, we use the graphical solution technique because there are only two design variables, width \( w \) and height \( h \) of the beam cross-section. We first graph schematically the stress constraint function \( \sigma \) and side constraints on a plane of two axes \( w \) and \( h \), as shown in Figure 3.5a. All designs \((w, h)\) that satisfy the constraints are called feasible designs. The set that collects all feasible designs is called a feasible set or feasible region. For this example, the feasible region \( S \) can be written as:

\[
S = \{ (w, h) \in R^2 | \sigma(w, h) - S_y \leq 0, \ w > 0, \ \text{and} \ h > 0 \} \quad (3.7)
\]

Next, we plot the objective function \( V(w, h) \) in Figure 3.5b with its iso-lines—in this case, straight lines. The iso-lines of the objective function are decreasing toward the origin of the \( w-h \) plane. It is clear that the minimum of the objective function is when the iso-line reaches the origin, in which the objective function \( V \) is zero. Certainly, this result is impossible physically. Mathematically, such a solution is called infeasible because the design is not in the feasible region. An optimal solution must be sought in the feasible region. Therefore, we graph the objective function in the feasible region as in Figure 3.5c, in which the objective function intersects the boundary of the feasible region at \((w^*, h^*)\) to reach its minimum \( V(w^*, h^*) = w^*h^*\ell \).
3.2.3 CLASSIFICATION OF OPTIMIZATION PROBLEMS

Both the beer can and beam examples involve constraints, either equality (beer can example) or inequality (beam example). Both are called constrained problems. Occasionally, constrained problems can be converted into unconstrained problems; for example, the constrained problem of the beer can in Eq. 3.1 was converted into the unconstrained problem in Eq. 3.2. Solution techniques for constrained and unconstrained problems and other kinds of problems are different.

Optimization problems can be classified in numerous ways. First, a design problem defined in Eq. 3.3 (and as in the beer can and beam examples) is called a single-objective (or single-criterion) problem because there is one single objective function to be optimized. If a design problem involves multiple objective functions, it is called a multiobjective (or multicriterion) problem. In this case, the design goal is to minimize (or maximize) all objective functions simultaneously. We discuss multiobjective optimization in Chapter 5.

As for the constraint functions, as seen in the examples, there are equality and inequality constraints. An optimization problem may have only equality constraints or inequality constraints, or both. Such problems are called constrained optimization problems. If there are no constraints involved, these problems are called unconstrained problems. Furthermore, based on the nature of expressions for objective and constraint functions, optimization problems can be classified as linear, nonlinear, and quadratic programming problems. That is, if all functions are linear, such problems are called linear optimization problems and they are solved by using linear programming techniques. If one of these functions is nonlinear, they are nonlinear problems, and they are solved by nonlinear programming (NLP) techniques. This classification is extremely useful from a computational point of view because there are special methods or algorithms developed for the efficient solution of a particular class of problems. Thus, the first task a designer needs to investigate is the class of problem encountered or formulated. This will, in many cases, dictate the solution techniques to be adopted in solving the problem.

**FIGURE 3.5**
Cantilever beam design problem solved by using the graphical technique: (a) feasible region, (b) iso-lines of the objective function, and (c) optimal solution identified at \((w^*, h^*)\).
As to the types of disciplines of the physical models involved in the objective and constraint functions, there are single-disciplinary and MDO problems. Structural optimization, in which only structural performance measures, such as stress, displacement, buckling load, and natural frequency, are involved in the optimization problems, is in general single disciplinary. We will discuss more on the subject of structural design in Chapter 4. MDO usually involves structural, motion, thermal, fluid, manufacturing, and so on. We include a tutorial example in Projects P5 and S5 to illustrate some of the aspects of the topic.

In the beer can and beam examples, all the design variables are permitted to take any real value (in these cases, positive real values), and the optimization problem is called a real-valued programming problem. In many problems, this may not be the case. If one of the design variables is discrete, the problems are called discrete optimization or integer programming problems. Solving discrete optimization problems is a whole lot different than solving problems with continuous design variables of real numbers. In this book, we assume design variables are continuous real numbers. Those who are interested in discrete optimization problems may refer to Kouvelis and Yu (1997) and Syslo et al. (1983) for more in-depth discussions.

Based on the deterministic nature of the variables involved, optimization problems can be classified as deterministic and stochastic programming problems. A stochastic programming problem is an optimization problem in which some or all of the parameters (design variables and/or preassigned parameters) are expressed probabilistically (nondeterministic or stochastic), such as estimates of the life span of structures that have probabilistic inputs of strength and load capacity. If all design variables are deterministic, we have deterministic optimization problems. In this chapter, we focus on deterministic programming problems. In Chapter 5, we briefly discuss stochastic programming problems.

So far, we have assumed that a single designer or single design team is working on the design problems. On some occasions, there are multiple designers or design groups making respective design decisions for the same product, especially for large-scale and complex systems. In these cases, design methods that employ game theory (discussed in Chapter 2) to aid design decision making (Vincent 1983) is still an open topic, which is continuously being explored by the technical community.

### 3.2.4 SOLUTION TECHNIQUES

In general, solution techniques for optimization problems, constrained or unconstrained, can be categorized into three major groups: optimality criteria methods (also called classical methods), graphical methods, and search methods using numerical algorithms, as shown in Figure 3.6.

The classical methods of differential calculus can be used to find the unconstrained maxima and minima of a function of several variables. These methods assume that the function is differentiable twice with respect to the design variables and the derivatives are continuous. For problems with equality constraints, the Lagrange multiplier method can be used. If the problem has inequality constraints, the Karush–Kuhn–Tucker (KKT) conditions can be used to identify the optimum point. However, these methods lead to a set of nonlinear simultaneous equations that may be difficult to solve. The classical methods of optimization are discussed in Section 3.3.

In Section 3.4, we discuss graphical solutions for solving linear and nonlinear examples. Graphical methods provide a clear picture of feasible region and iso-lines of objective functions that are straightforward in identifying optimal solutions. However, they are effective for up to two design
variables, which substantially limit their applications. Note that neither classical methods nor graphical methods require numerical calculations for solutions.

The mainstream solution techniques for optimization problems are search methods involving numerical calculations that search for optimal solution in an iterative process by starting from an initial design. Some techniques rely on gradient information (i.e., derivatives of objective and constraint functions with respect to design variables) to guide the search process. These methods are called gradient-based approaches. Other techniques follow certain rules for search optimal solutions that do not require gradient information. These are called non-gradient-based approaches. We provide a basic discussion on the gradient-based methods in Section 3.5 and narrow into three major algorithms in Section 3.6, including sequential linear programming (SLP), sequential quadratic programming (SQP), and feasible direction method. We include two key algorithms of non-gradient methods in Section 3.7: genetic algorithms and simulated annealing.

3.3 OPTIMALITY CONDITIONS

A basic knowledge of optimality conditions is important for understanding the performance of the various numerical methods discussed later in the chapter. In this section, we introduce the basic concept of optimality, the necessary and sufficient conditions for the relative maxima and minima of a function, as well as the solution methods based on the optimality conditions. Simple examples are used to explain the underlying concepts. The examples will also show the practical limitations of the methods.

3.3.1 BASIC CONCEPT OF OPTIMALITY

We start by recalling a few basic concepts we learned in Calculus regarding maxima and minima, followed by defining local and global optima; thereafter, we illustrate the concepts using functions of one and multiple variables.

3.3.1.1 Functions of a single variable

This section presents a few definitions for basic terms.

Stationary point: For a continuous and differentiable function \( f(x) \), a stationary point \( x^* \) is a point at which the slope of the function vanishes—that is, \( f'(x) = \frac{df}{dx} = 0 \) at \( x = x^* \), where \( x^* \) belongs to its domain of definition. As illustrated in Figure 3.7, a stationary point can be a minimum if \( f''(x) > 0 \), a maximum if \( f''(x) < 0 \), or an inflection point if \( f''(x) = 0 \) in the neighborhood of \( x^* \).
Global and local minimum: A function \( f(x) \) is said to have a local (or relative) minimum at \( x = x^* \) if \( f(x^*) \leq f(x + \delta) \) for all sufficiently small positive and negative values of \( \delta \), that is, in the neighborhood of the point \( x^* \). A function \( f(x) \) is said to have a global (or absolute) minimum at \( x = x^* \) if \( f(x^*) \leq f(x) \) for all \( x \) in the domain over which \( f(x) \) is defined. Figure 3.8 shows the global and local optimum points of a function \( f(x) \) with a single variable \( x \).

**Necessary condition:** Consider a function \( f(x) \) of single variable defined for \( a < x < b \). To find a point of \( x^* \in (a, b) \) that minimizes \( f(x) \), the first derivative of function \( f(x) \) with respect to \( x \) at \( x = x^* \) must be a stationary point; that is, \( f'(x^*) = 0 \).

**Sufficient condition:** For the same function \( f(x) \) stated above and \( f'(x^*) = 0 \), then it can be said that \( f(x^*) \) is a minimum value of \( f(x) \) if \( f''(x^*) > 0 \), or a maximum value if \( f''(x^*) < 0 \).

**EXAMPLE 3.1**
Find a minimum of the function \( f(x) = x^2 - 2x \), for \( x \in (0, 2) \).

**Solutions**
The first derivative of \( f(x) \) with respect to \( x \) is \( f'(x) = 2x - 2 \). We set \( f'(x) = 0 \), and solve for \( x = 1 \), which is a stationary point. This is the necessary condition for \( x = 1 \) to a minimum of the function \( f(x) \).

We take second derivative of \( f(x) \) with respect to \( x \), \( f''(x) = 2 > 0 \), which satisfies the sufficient condition of the function \( f(x) \) that has a minimum at \( x = 1 \), and the minimum value of the function at \( x = 1 \) is \( f(1) = -1 \).
The concept illustrated above can be easily extended to functions of multiple variables. We use functions of two variables to provide a graphical illustration on the concepts.

### 3.3.1.2 Functions of multiple variables

A function of two variables \( f(x_1, x_2) = -(\cos^2 x_1 + \cos^2 x_2)^2 \) is graphed in Figure 3.9a. Perturbations from point \((x_1, x_2) = (0, 0)\), which is a local minimum, in any direction result in an increase in the function value of \( f(x) \); that is, the slopes of the function with respect to \( x_1 \) and \( x_2 \) are zero at this point of local minimum. Similarly, a function \( f(x_1, x_2) = (\cos^2 x_1 + \cos^2 x_2)^2 \) graphed in Figure 3.9b has a local maximum at \((x_1, x_2) = (0, 0)\). Perturbations from this point in any direction result in a decrease in the function value of \( f(x) \); that is, the slopes of the function with respect to \( x_1 \) and \( x_2 \) are zero at this point of local maximum. The first derivatives of the function with respect to the variables are zero at the minimum or maximum, which again is called a stationary point.

**Necessary condition:** Consider a function \( f(x) \) of multivariables defined for \( x \in \mathbb{R}^n \), where \( n \) is the number of variables. To find a point of \( x^* \in \mathbb{R}^n \) that minimizes \( f(x) \), the gradient of the function \( f(x) \) at \( x = x^* \) must be a stationary point; that is, \( \nabla f(x^*) = 0 \).

The gradient of a function of multivariables is defined as

\[
\nabla f(x) \equiv \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right]^T
\]

(3.8)

Geometrically, the gradient vector is normal to the tangent plane at a given point \( x \), and it points in the direction of maximum increase in the function. These properties are quite important; they will be used in developing optimality conditions and numerical methods for optimum design. In Example 3.2, the gradient vector for a function of two variables is calculated for illustration purpose.

---

**FIGURE 3.9**

Functions of two variables (MATLAB Script 2 can be found in Appendix A): (a) \( f(x_1, x_2) = -(\cos^2 x_1 + \cos^2 x_2)^2 \) with a local minimum at \((0, 0)\) and (b) \( f(x_1, x_2) = (\cos^2 x_1 + \cos^2 x_2)^2 \) with a local maximum at \((0, 0)\).
**Sufficient condition:** For the same function \( f(x) \) stated above, let \( \nabla f(x^*) = 0 \), then \( f(x^*) \) has a minimum value of \( f(x) \) if its Hessian matrix defined in Eq. 3.10 is positive-definite.

\[
\begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{bmatrix}_{n \times n}
\]  

(3.10)

where all derivatives are calculated at the given point \( x^* \). The Hessian matrix is an \( n \times n \) matrix, where \( n \) is the number of variables. It is important to note that each element of the Hessian is a function in

**EXAMPLE 3.2**

A function of two variables is defined as

\[
f(x_1, x_2) = x_2e^{-x_1^2 - x_2^2}
\]

which is graphed in MATLAB shown below (left). The MATLAB script for the graph can be found in Appendix A (Script 3). Calculate the gradient vectors of the function at \((x_1, x_2) = (1, 1)\) and \((x_1, x_2) = (1, -1)\).

**Solutions**

From Eq. 3.8, the gradient vector of the function \( f(x_1, x_2) \) is

\[
\nabla f(x_1, x_2) = \left[ -2x_1x_2e^{-x_1^2 - x_2^2}, e^{-x_1^2 - x_2^2} - 2x_2^2e^{-x_1^2 - x_2^2} \right]^T
\]

(3.9b)

At \((x_1, x_2) = (1, 1)\), \( f(1, 1) = e^{-2} = 0.1353 \), and \( \nabla f(1, 1) = [-2e^{-2}, -e^{-2}]^T \); and at \((x_1, x_2) = (1, -1)\), \( f(1, -1) = -e^{-2} = -0.1353 \), and \( \nabla f(1, -1) = [2e^{-2}, -e^{-2}]^T \). The iso-lines of \( f(1, 1) \) and \( f(1, -1) \) as well as the gradient vectors at \((1, 1)\) and \((1, -1)\) are shown in the figure below (right). In this example, gradient vector at a point \( x \) is perpendicular to the tangent line at \( x \), and the vector points in the direction of maximum increment in the function value. The maximum and minimum of the function are shown for clarity.
itself that is evaluated at the given point $x^*$. Also, because $f(x)$ is assumed to be twice continuously differentiable, the cross partial derivatives are equal; that is,

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}; \quad i, j = 1, n \quad (3.11)$$

Therefore, the Hessian is always a symmetric matrix. The Hessian matrix plays a prominent role in exploring the sufficiency conditions for optimality.

Note that a square matrix is positive-definite if (a) the determinant of the Hessian matrix is positive (i.e., $|H| > 0$) or (b) all its eigenvalues are positive. To calculate the eigenvalues $\lambda$ of a square matrix, the following equation is solved:

$$|H - \lambda I| = 0 \quad (3.12)$$

where $I$ is an identity matrix of $n \times n$.

---

**EXAMPLE 3.3**

A function of three variables is defined as

$$f(x_1, x_2, x_3) = x_1^2 + 2x_1x_2 + 2x_2^2 + x_3^2 - 2x_1 + x_2 + 8 \quad (3.13a)$$

Calculate the gradient vector of the function and determine a stationary point, if it exists. Calculate a Hessian matrix of the function $f$, and determine if the stationary point found gives a minimum value of the function $f$.

**Solutions**

We first calculate the gradient of the function and set it to zero to find the stationary point(s), if any:

$$\nabla f(x_1, x_2, x_3) = [2x_1 + 2x_2 - 2, 2x_1 + 4x_2 + 1, 2x_3]^T \quad (3.13b)$$

Setting Eq. 3.13b to zero, we have $x = [2.5, -1.5, 0]^T$, which is the only stationary point. Now, we calculate the Hessian matrix:

$$H = \nabla^2 f = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.13c)$$

which is positive-definite because

$$|H| = \begin{vmatrix} 2 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 1 \end{vmatrix} = 8 - 4 = 4 > 0 \quad (3.13d)$$

or

$$|H - \lambda I| = \begin{vmatrix} 2 - \lambda & 2 & 0 \\ 2 & 4 - \lambda & 0 \\ 0 & 0 & 1 - \lambda \end{vmatrix} = (2 - \lambda)(4 - \lambda)(1 - \lambda) - 4(1 - \lambda) = 0 \quad (3.13e)$$

Solving Eq. 3.13e, we have $\lambda = 1, 0.7639$ and $5.236$, which are all positive. Hence, the Hessian matrix is positive-definite; therefore, the stationary point $x^* = [2.5, -1.5, 0]^T$ is a minimum point, at which the function value is $f(x^*) = 4.75$. 

---

118 Chapter 3 Design Optimization