

Part 3

Algorithms and Their Convergence

Chapter 7

The Concept of an Algorithm

In the remainder of the text we describe many algorithms for solving different classes of nonlinear programming problems. This chapter introduces the concept of an algorithm. Algorithms are viewed as point-to-set maps, and the main convergence theorem is proved utilizing the concept of a closed mapping. This theorem will be utilized in the remaining chapters to analyze the convergence of several computational schemes.

The following is an outline of the chapter.

SECTION 7.1: Algorithms and Algorithmic Maps This section presents algorithms as point-to-set maps and introduces the concept of a solution set.

SECTION 7.2: Closed Maps and Convergence We first introduce the concept of a closed map and then prove the main convergence theorem.

SECTION 7.3: Composition of Mappings We establish closedness of composite maps by examining closedness of the individual maps. We then discuss mixed algorithms and give a condition for their convergence.

SECTION 7.4: Comparison Among Algorithms Some practical factors for assessing the efficiency of different algorithms are discussed.

7.1 Algorithms and Algorithmic Maps

Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in S$, where f is the objective function and S is the feasible region. A *solution procedure* or an *algorithm* for solving this problem can be viewed as an iterative process that generates a sequence of points according to a prescribed set of instructions, together with a termination criterion.

The Algorithmic Map

Given a vector \mathbf{x}_k and applying the instructions of the algorithm, we obtain a new point \mathbf{x}_{k+1} . This process can be described by an *algorithmic map* \mathbf{A} . This map is generally a point-to-set map and assigns to each point in the domain X a subset of X . Thus, given the initial point \mathbf{x}_1 , the algorithmic map generates the sequence $\mathbf{x}_1, \mathbf{x}_2, \dots$, where $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$ for each k . The transformation of \mathbf{x}_k into \mathbf{x}_{k+1} through the map constitutes an *iteration* of the algorithm.

7.1.1. Example

Consider the following problem:

$$\begin{array}{ll} \text{Minimize} & x^2 \\ \text{subject to} & x \geq 1 \end{array}$$

whose optimal solution is $\bar{x} = 1$. Let the point-to-point algorithmic map be given by $\mathbf{A}(x) = \frac{1}{2}(x+1)$. It could be easily verified that the sequence obtained by applying the map \mathbf{A} , with any starting point, converges to the optimal solution $\bar{x} = 1$. With $x_1 = 4$, the algorithm generates the sequence $\{4, 2.5, 1.75, 1.375, 1.1875, \dots\}$ as illustrated in Figure 7.1a.

As another example, consider the point-to-set mapping \mathbf{A} , defined by

$$\mathbf{A}(x) = \begin{cases} [1, \frac{1}{2}(x+1)] & \text{if } x \geq 1 \\ [\frac{1}{2}(x+1), 1] & \text{if } x < 1 \end{cases}$$

As shown in Figure 7.1b, the image of any point x is a closed interval, and any point in that interval could be chosen as the successor of x . Starting with any point x_1 , the algorithm converges to $\bar{x} = 1$. With $x_1 = 4$, the sequence $\{4, 2, 1.2, 1.1, 1.02, \dots\}$ is a possible result of the algorithm. Unlike the previous example, other sequences could result from the algorithmic map.

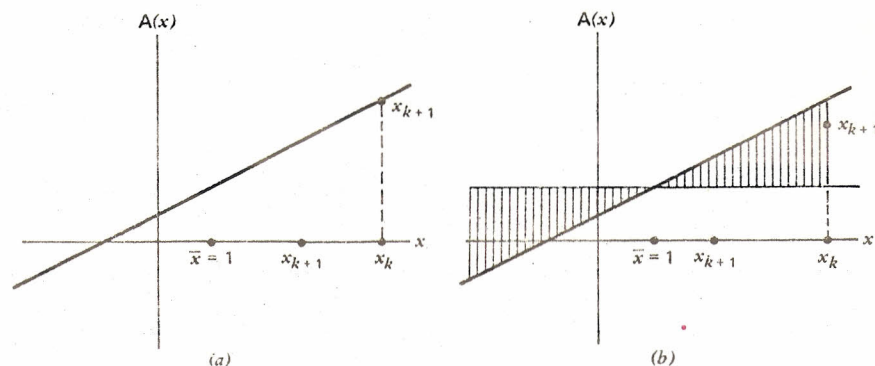


Figure 7.1 Examples of algorithmic maps.

The Solution Set and Convergence of Algorithms

Consider the following nonlinear programming problem

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in S \end{array}$$

A desirable property of an algorithm for solving the above problem is that it generates a sequence of points converging to a global optimal solution. In many cases, however, we may have to be satisfied with less favorable outcomes. In fact, as a result of nonconvexity, problem size, and other difficulties, we may stop the iterative procedure if a point belonging to a prescribed set, which we call the *solution set* Ω , is reached. The following are some typical solution sets for the above mentioned problem.

1. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \text{ is a local optimal solution of the problem}\}.$
2. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) \leq b\},$ where b is an acceptable objective value.
3. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) < LB + \epsilon\},$ where $\epsilon > 0$ is a specified tolerance, and LB is a lower bound on the optimal objective value. A typical lower bound is the objective value of the Lagrangian dual problem.
4. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) - f(\hat{\mathbf{x}}) < \epsilon\},$ where $f(\hat{\mathbf{x}})$ is the global minimum, and $\epsilon > 0$ is specified.
5. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \text{ satisfies the Kuhn-Tucker optimality conditions}\}.$
6. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \text{ satisfies the Fritz John optimality conditions}\}.$

Thus, in general, convergence of algorithms is made in reference to the solution set rather than to the collection of global optimal solutions. In particular, the algorithmic map $\mathbf{A} : X \rightarrow X$ is said to *converge* over $Y \subset X$ if, starting with any initial point $\mathbf{x}_1 \in Y$, the limit of any convergent subsequence of the sequence $\mathbf{x}_1, \mathbf{x}_2, \dots$, generated by the algorithm, belongs to the solution set Ω . Letting Ω be the set of global optimal solutions in Example 7.1.1, it is obvious that the two stated algorithms are convergent over the real line.

7.2 Closed Maps and Convergence

In this section we introduce the notion of closed maps and then prove a convergence theorem. The significance of the concept of closedness will be clear from the following example and the subsequent discussion.

7.2.1 Example

Consider the following problem:

$$\begin{array}{ll} \text{Minimize} & x^2 \\ \text{subject to} & x \geq 1 \end{array}$$

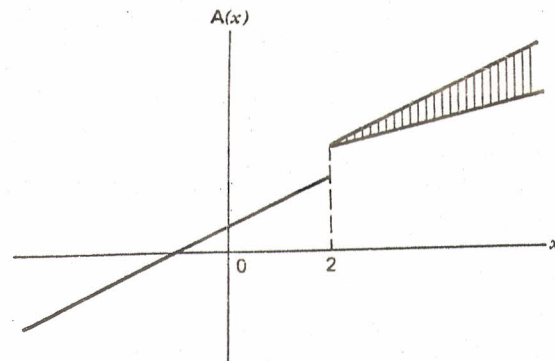


Figure 7.2 An example of a nonconvergent algorithmic map.

Let Ω be the set of global optimal solutions, that is, $\Omega = \{1\}$. Consider the algorithmic map defined by

$$A(x) = \begin{cases} [\frac{3}{2} + \frac{1}{4}x, 1 + \frac{1}{2}x] & \text{if } x \geq 2 \\ \frac{1}{2}(x+1) & \text{if } x < 2 \end{cases}$$

The map A is illustrated in Figure 7.2. Obviously, for any initial point $x_1 \geq 2$, any sequence generated by the map A converges to the point $\hat{x} = 2$. Note that $\hat{x} \notin \Omega$. On the other hand, for $x_1 < 2$, any sequence generated by the algorithm converges to $\bar{x} = 1$. In this example the algorithm converges over the interval $(-\infty, 2)$ but does not converge to a point in the set Ω over the interval $[2, \infty)$.

The above example shows the significance of the initial point x_1 , where convergence to a point in Ω is achieved if $x_1 < 2$ but not realized otherwise. Note that each of the algorithms in Examples 7.1.1 and 7.2.1 satisfy the following conditions:

1. Given a feasible point $x_k \geq 1$, any successor point x_{k+1} is also feasible, that is, $x_{k+1} \geq 1$.
2. Given a feasible point x_k not in the solution set Ω , any successor point x_{k+1} satisfies $f(x_{k+1}) < f(x_k)$, where $f(x) = x^2$. In other words, the objective function strictly decreases.
3. Given a feasible point x_k in the solution set Ω , that is, $x_k = 1$, the successor point is also in Ω , that is, $x_{k+1} = 1$.

Despite the above-mentioned similarities among the algorithms, the two algorithms of Example 7.1.1 converge to $\bar{x} = 1$, while that of Example 7.2.1 does not converge to $\bar{x} = 1$ for any initial point $x_1 \geq 2$. The reason for this is that the algorithmic map of Example 7.2.1 is not closed at $x = 2$. The notion of a closed mapping, which generalizes the notion of a continuous function, is defined below.

Closed Maps

7.2.2 Definition

Let X and Y be nonempty closed sets in E_p and E_q , respectively. Let $A: X \rightarrow Y$ be a point-to-set map. The map A is said to be *closed* at $x \in X$ if

$$\begin{aligned} x_k \in X & \quad x_k \rightarrow x \\ y_k \in A(x_k) & \quad y_k \rightarrow y \end{aligned}$$

imply that $y \in A(x)$. The map A is said to be closed on $Z \subset X$ if it is closed at each point in Z .

Figure 7.2 shows an example of a point-to-set map that is not closed at $x = 2$. In particular, the sequence $\{x_k\}$ with $x_k = 2 - \frac{1}{k}$ converges to $x = 2$, and the sequence $\{y_k\}$ with $y_k = A(x_k) = \frac{3}{2} - \frac{1}{2k}$ converges to $y = \frac{3}{2}$, but $y \notin A(x) = \{2\}$. Figure 7.1 shows two examples of algorithmic maps that are closed everywhere.

The Convergence Theorem

Conditions that ensure convergence of algorithmic maps are stated in Theorem 7.2.3 below. The theorem will be used in the remainder of the text to show convergence of many algorithms.

7.2.3 Theorem

Let X be a nonempty closed set in E_n , and let the nonempty set $\Omega \subset X$ be the solution set. Let $A: X \rightarrow X$ be a point-to-set map. Given $x_1 \in X$, the sequence $\{x_k\}$ is generated iteratively as follows:

If $x_k \in \Omega$ then stop; otherwise, let $x_{k+1} \in A(x_k)$, replace k by $k+1$, and repeat.

Suppose that the sequence x_1, x_2, \dots , produced by the algorithm is contained in a compact subset of X , and suppose that there exists a continuous function α , called the *descent function*, such that $\alpha(y) < \alpha(x)$ if $x \notin \Omega$ and $y \in A(x)$. If the map A is closed over the complement of Ω , then either the algorithm stops in a finite number of steps with a point in Ω or it generates the infinite sequence $\{x_k\}$ such that:

1. Every convergent subsequence of $\{x_k\}$ has a limit in Ω , that is, all accumulation points of $\{x_k\}$ belong to Ω .
2. $\alpha(x_k) \rightarrow \alpha(x)$ for some $x \in \Omega$.

Proof

If at any iteration a point \mathbf{x}_k in Ω is generated, then the algorithm stops. Now suppose that an infinite sequence $\{\mathbf{x}_k\}$ is generated. Let $\{\mathbf{x}_k\}_{\mathcal{K}}$ be any convergent subsequence with limit $\mathbf{x} \in X$. Since α is continuous, then for $k \in \mathcal{K}$, $\alpha(\mathbf{x}_k) \rightarrow \alpha(\mathbf{x})$. Thus, for a given $\varepsilon > 0$, there is a $K \in \mathcal{K}$ such that

$$\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}) < \varepsilon \quad \text{for } k \geq K \text{ with } k \in \mathcal{K}$$

In particular for $k = K$, we get

$$\alpha(\mathbf{x}_K) - \alpha(\mathbf{x}) < \varepsilon \quad (7.1)$$

Now let $k > K$. Since α is a descent function, $\alpha(\mathbf{x}_k) < \alpha(\mathbf{x}_K)$, and from (7.1), we get

$$\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}) = \alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_K) + \alpha(\mathbf{x}_K) - \alpha(\mathbf{x}) < 0 + \varepsilon = \varepsilon$$

Since this is true for every $k > K$, and since $\varepsilon > 0$ was arbitrary, then

$$\lim_{k \rightarrow \infty} \alpha(\mathbf{x}_k) = \alpha(\mathbf{x}) \quad (7.2)$$

We now show that $\mathbf{x} \in \Omega$. By contradiction suppose that $\mathbf{x} \notin \Omega$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}}$. This sequence is contained in a compact subset of X and hence has a convergent subsequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}}$ with limit $\bar{\mathbf{x}}$ in X . Noting (7.2), it is clear that $\alpha(\bar{\mathbf{x}}) = \alpha(\mathbf{x})$. Since \mathbf{A} is closed at \mathbf{x} , and for $k \in \mathcal{K}$, $\mathbf{x}_k \rightarrow \mathbf{x}$, $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, and $\mathbf{x}_{k+1} \rightarrow \bar{\mathbf{x}}$, then $\bar{\mathbf{x}} \in \mathbf{A}(\mathbf{x})$. Therefore $\alpha(\bar{\mathbf{x}}) < \alpha(\mathbf{x})$, contradicting the fact that $\alpha(\bar{\mathbf{x}}) = \alpha(\mathbf{x})$. Thus $\mathbf{x} \in \Omega$ and part 1 of the theorem holds true. This, coupled with (7.2), shows that part 2 of the theorem holds true, and the proof is complete. /

Corollary

Under the assumptions of the theorem, if Ω is the singleton $\{\bar{\mathbf{x}}\}$, then the whole sequence $\{\mathbf{x}_k\}$ converges to $\bar{\mathbf{x}}$.

Proof

Suppose, by contradiction, that there exists an $\varepsilon > 0$ and a sequence $\{\mathbf{x}_k\}_{\mathcal{K}}$ such that

$$\|\mathbf{x}_k - \bar{\mathbf{x}}\| > \varepsilon \quad \text{for } k \in \mathcal{K} \quad (7.3)$$

Note that there exists $\mathcal{K}' \subset \mathcal{K}$ such that $\{\mathbf{x}_k\}_{\mathcal{K}'}$ has a limit \mathbf{x}' . By part 1 of the theorem, $\mathbf{x}' \in \Omega$. But $\Omega = \{\bar{\mathbf{x}}\}$, and thus $\mathbf{x}' = \bar{\mathbf{x}}$. Therefore, $\mathbf{x}_k \rightarrow \bar{\mathbf{x}}$ for $k \in \mathcal{K}'$, violating (7.3). This completes the proof. /

Note that if the point at hand \mathbf{x}_k does not belong to the solution set Ω , then the algorithm generates a new point \mathbf{x}_{k+1} such that $\alpha(\mathbf{x}_{k+1}) < \alpha(\mathbf{x}_k)$. The function α is called a *descent function*. In many cases, α is chosen as the objective function f itself, and thus the algorithm generates a sequence of points with improving objective function values. Other alternative choices of the function α are possible. For instance, if f is differentiable, α could be chosen as $\alpha(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$ for an unconstrained optimization problem.

Terminating the Algorithm

As indicated in Theorem 7.2.3, the algorithm is terminated if we reach a point in the solution set Ω . In most cases, however, convergence to a point in Ω occurs only in a limiting sense, and we must resort to some practical rules for terminating the iterative procedure. The following rules are frequently used to stop a given algorithm. Here, $\varepsilon > 0$ and the positive integer N are prespecified.

$$1. \quad \|\mathbf{x}_{k+N} - \mathbf{x}_k\| < \varepsilon$$

Here, the algorithm is stopped if the distance moved after N applications of the map \mathbf{A} is less than ε .

$$2. \quad \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_k\|} < \varepsilon.$$

Under this criterion, the algorithm is terminated if the relative distance moved during a given iteration is less than ε .

$$3. \quad \alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_{k+N}) < \varepsilon.$$

Here, the algorithm is stopped if the total improvement in the descent function value after N applications of the map \mathbf{A} is less than ε .

$$4. \quad \frac{\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_{k+1})}{|\alpha(\mathbf{x}_k)|} < \varepsilon.$$

If the relative improvement in the descent function value during any given iteration is less than ε , then the termination criterion is realized.

$$5. \quad \alpha(\mathbf{x}_k) - \alpha(\bar{\mathbf{x}}) < \varepsilon, \quad \text{where } \bar{\mathbf{x}} \text{ belongs to } \Omega.$$

This criterion for termination is suitable if $\alpha(\bar{\mathbf{x}})$ is known beforehand; for example, in unconstrained optimization if $\alpha(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$ and $\Omega = \{\bar{\mathbf{x}} : \nabla f(\bar{\mathbf{x}}) = \mathbf{0}\}$, then $\alpha(\bar{\mathbf{x}}) = 0$.

7.3 Composition of Mappings

In most nonlinear programming solution procedures, the algorithmic maps are often composed of several maps. For example, some algorithms first find a direction \mathbf{d}_k to move along and then determine the step size λ_k by solving the one-dimensional problem of minimizing $\alpha(\mathbf{x}_k + \lambda \mathbf{d}_k)$. In this case, the map \mathbf{A} is composed of \mathbf{MD} , where \mathbf{D} finds the direction \mathbf{d}_k , and then \mathbf{M} finds an optimal step size λ_k . It is often easier to prove that the overall map is closed by examining its individual components. In this section, the notion of composite maps is stated precisely, and then a result relating closedness of the overall map to that of its individual components is given. Finally, we discuss mixed algorithms and state conditions under which they converge.

7.3.1 Definition

Let X , Y , and Z be nonempty closed sets in E_n , E_p , and E_q , respectively. Let $\mathbf{B}: X \rightarrow Y$ and $\mathbf{C}: Y \rightarrow Z$ be point-to-set maps. The *composite map* $\mathbf{A} = \mathbf{CB}$ is defined as the point-to-set map $\mathbf{A}: X \rightarrow Z$ with

$$\mathbf{A}(x) = \bigcup \{\mathbf{C}(y) : y \in \mathbf{B}(x)\}.$$

Figure 7.3 illustrates the notion of a composite map, and Theorem 7.3.2 and its corollaries give several sufficient conditions for a composite map to be closed.

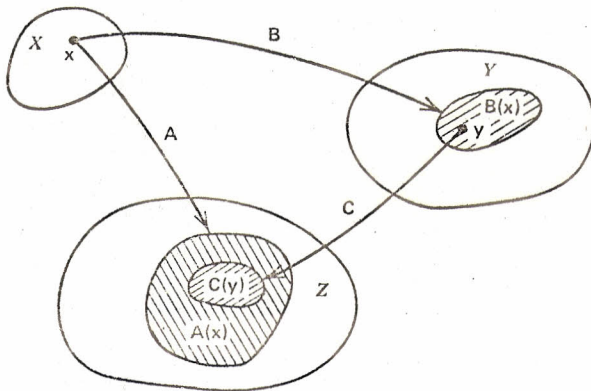


Figure 7.3 Composite maps.

7.3.2 Theorem

Let X , Y , and Z be nonempty closed sets in E_n , E_p , and E_q , respectively. Let $\mathbf{B}: X \rightarrow Y$ and $\mathbf{C}: Y \rightarrow Z$ be point-to-set maps, and consider the composite map $\mathbf{A} = \mathbf{CB}$. Suppose that \mathbf{B} is closed at \mathbf{x} and that \mathbf{C} is closed on $\mathbf{B}(\mathbf{x})$.

Furthermore, suppose that if $\mathbf{x}_k \rightarrow \mathbf{x}$ and $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$, then there is a convergent subsequence of $\{\mathbf{y}_k\}$. Then \mathbf{A} is closed at \mathbf{x} .

Proof

Let $\mathbf{x}_k \rightarrow \mathbf{x}$, $\mathbf{z}_k \in \mathbf{A}(\mathbf{x}_k)$, and $\mathbf{z}_k \rightarrow \mathbf{z}$. We need to show that $\mathbf{z} \in \mathbf{A}(\mathbf{x})$. By definition of \mathbf{A} , for each k , there is a $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$ such that $\mathbf{z}_k \in \mathbf{C}(\mathbf{y}_k)$. By assumption, there is a convergent subsequence $\{\mathbf{y}_k\}_{\mathcal{K}}$ with limit \mathbf{y} . Since \mathbf{B} is closed at \mathbf{x} , then $\mathbf{y} \in \mathbf{B}(\mathbf{x})$. Furthermore, since \mathbf{C} is closed on $\mathbf{B}(\mathbf{x})$ it is closed at \mathbf{y} , and hence $\mathbf{z} \in \mathbf{C}(\mathbf{y})$. Thus, $\mathbf{z} \in \mathbf{C}(\mathbf{y}) \in \mathbf{CB}(\mathbf{x}) = \mathbf{A}(\mathbf{x})$, and hence \mathbf{A} is closed at \mathbf{x} .

Corollary 1

Let X , Y , and Z be nonempty closed sets in E_n , E_p , and E_q , respectively. Let $\mathbf{B}: X \rightarrow Y$ and $\mathbf{C}: Y \rightarrow Z$ be point-to-set maps. Suppose that \mathbf{B} is closed at \mathbf{x} , \mathbf{C} is closed on $\mathbf{B}(\mathbf{x})$, and Y is compact. Then $\mathbf{A} = \mathbf{CB}$ is closed at \mathbf{x} .

Corollary 2

Let X , Y , and Z be nonempty closed sets in E_n , E_p , and E_q , respectively. Let $\mathbf{B}: X \rightarrow Y$ be a function, and let $\mathbf{C}: Y \rightarrow Z$ be a point-to-set map. If \mathbf{B} is continuous at \mathbf{x} , and \mathbf{C} is closed on $\mathbf{B}(\mathbf{x})$, then $\mathbf{A} = \mathbf{CB}$ is closed at \mathbf{x} .

Note the importance of the assumption that a convergent subsequence $\{\mathbf{y}_k\}_{\mathcal{K}}$ exists in Theorem 7.3.2. Without this assumption, even if the maps \mathbf{B} and \mathbf{C} are closed, the composite map $\mathbf{A} = \mathbf{CB}$ is not necessarily closed, as shown by Example 7.3.3 below.

7.3.3 Example

Consider the maps $\mathbf{B}, \mathbf{C}: E_1 \rightarrow E_1$ defined below.

$$\mathbf{B}(x) = \begin{cases} \frac{1}{x} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

$$\mathbf{C}(y) = \{z : |z| \leq |y|\}.$$

Note that both \mathbf{B} and \mathbf{C} are closed everywhere. Now consider the composite map $\mathbf{A} = \mathbf{CB}$. Then, \mathbf{A} is given by $\mathbf{A}(x) = \mathbf{CB}(x) = \{z : |z| \leq |\mathbf{B}(x)|\}$. From the definition of \mathbf{B} , it follows that

$$\mathbf{A}(x) = \begin{cases} \{z : |z| \leq \frac{1}{|x|}\} & \text{if } x \neq 0 \\ \{0\} & \text{if } x = 0 \end{cases}$$

Note that \mathbf{A} is not closed at $x = 0$. In particular, consider the sequence $\{x_k\}$, where $x_k = \frac{1}{k}$. Note that $\mathbf{A}(x_k) = \{z : |z| \leq k\}$, and hence $z_k = 1$ belongs to $\mathbf{A}(x_k)$

for each k . On the other hand, the limit point $z=1$ does not belong to $\mathbf{A}(x)=\{0\}$. Thus the map \mathbf{A} is not closed, even though both \mathbf{B} and \mathbf{C} are closed. Here, Theorem 7.3.2 does not apply, since the sequence $y_k \in \mathbf{B}(x_k)$ for $x_k = \frac{1}{k}$ does not have a convergent subsequence.

Convergence of Algorithms with composite maps

At each iteration, many nonlinear programming algorithms use two maps. One of the maps is usually closed and satisfies the convergence requirements of Theorem 7.2.3. The second map may involve any process as long as the value of the descent function does not increase. As illustrated in Exercise 7.17, the overall map may not be closed, so that Theorem 7.2.3 could not be applied. However, as shown below, such maps do converge.

7.3.4 Theorem

Let X be a nonempty closed set in E_n , and let $\Omega \subset X$ be a nonempty solution set. Let $\alpha: E_n \rightarrow E_1$ be a continuous function, and consider the point-to-set map $\mathbf{C}: X \rightarrow X$ satisfying the following property: Given $\mathbf{x} \in X$, then $\alpha(\mathbf{y}) \leq \alpha(\mathbf{x})$ for $\mathbf{y} \in \mathbf{C}(\mathbf{x})$. Let $\mathbf{B}: X \rightarrow X$ be a point-to-set map that is closed over the complement of Ω and that satisfies $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$ for each $\mathbf{y} \in \mathbf{B}(\mathbf{x})$, if $\mathbf{x} \notin \Omega$. Now consider the algorithm defined by the composite map $\mathbf{A} = \mathbf{CB}$. Given $\mathbf{x}_1 \in X$, suppose that the sequence $\{\mathbf{x}_k\}$ is generated as follows:

If $\mathbf{x}_k \in \Omega$, then stop; otherwise, let $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, replace k by $k+1$, and repeat.

Suppose that the set $\Lambda = \{\mathbf{x}: \alpha(\mathbf{x}) \leq \alpha(\mathbf{x}_1)\}$ is compact. Then either the algorithm stops in a finite number of steps with a point in Ω or all accumulation points of $\{\mathbf{x}_k\}$ belong to Ω .

Proof

If at any iteration $\mathbf{x}_k \in \Omega$, then the algorithm stops. Now suppose that the sequence $\{\mathbf{x}_k\}$ is generated by the algorithm, and let $\{\mathbf{x}_k\}_{\mathcal{K}}$ be a convergent subsequence, with limit \mathbf{x} . Thus $\alpha(\mathbf{x}_k) \rightarrow \alpha(\mathbf{x})$ for $k \in \mathcal{K}$. Using monotonicity of α as in Theorem 7.2.3, it follows that

$$\lim_{k \rightarrow \infty} \alpha(\mathbf{x}_k) = \alpha(\mathbf{x}) \quad (7.4)$$

We want to show that $\mathbf{x} \in \Omega$. By contradiction, suppose that $\mathbf{x} \notin \Omega$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}'}$. By definition of the composite map \mathbf{A} , note that $\mathbf{x}_{k+1} \in \mathbf{C}(\mathbf{y}_k)$, where $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$. Note that $\mathbf{y}_k, \mathbf{x}_{k+1} \in \Lambda$. Since Λ is compact, there exists an index set $\mathcal{K}' \subset \mathcal{K}$ such that $\mathbf{y}_k \rightarrow \mathbf{y}$ and $\mathbf{x}_{k+1} \rightarrow \mathbf{x}'$ for $k \in \mathcal{K}'$. Since \mathbf{B} is closed at $\mathbf{x} \notin \Omega$, then $\mathbf{y} \in \mathbf{B}(\mathbf{x})$, and $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$. Since $\mathbf{x}_{k+1} \in \mathbf{C}(\mathbf{y}_k)$, then

by assumption, $\alpha(\mathbf{x}_{k+1}) \leq \alpha(\mathbf{y}_k)$ for $k \in \mathcal{K}'$, and hence by taking the limit, $\alpha(\mathbf{x}') \leq \alpha(\mathbf{y})$. Since $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$, then $\alpha(\mathbf{x}') < \alpha(\mathbf{x})$. Since $\alpha(\mathbf{x}_{k+1}) \rightarrow \alpha(\mathbf{x}')$ for $k \in \mathcal{K}'$, then $\alpha(\mathbf{x}') < \alpha(\mathbf{x})$ contradicts (7.4). Therefore, $\mathbf{x} \in \Omega$, and the proof is complete. \square

Minimizing Along Independent Directions

We now present a theorem that establishes convergence of a class of algorithms for solving a problem of the form: minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in E_n$. Under mild assumptions, we show that an algorithm that generates n linearly independent search directions, and obtains a new point by sequentially minimizing f along these directions, converges to a stationary point. The theorem also establishes convergence of algorithms using linearly independent and orthogonal search directions.

7.3.5 Theorem

Let $f: E_n \rightarrow E_1$ be differentiable, and consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in E_n$. Consider an algorithm whose map \mathbf{A} is defined as follows. The vector $\mathbf{y} \in \mathbf{A}(\mathbf{x})$ means that \mathbf{y} is obtained by minimizing f sequentially along the directions $\mathbf{d}_1, \dots, \mathbf{d}_n$ starting from \mathbf{x} . Here the search directions $\mathbf{d}_1, \dots, \mathbf{d}_n$ may depend upon \mathbf{x} , and each has norm 1. Suppose that the following properties are true:

1. There exists an $\varepsilon > 0$ such that $\det[\mathbf{D}(\mathbf{x})] \geq \varepsilon$ for each $\mathbf{x} \in E_n$. Here, $\mathbf{D}(\mathbf{x})$ is the $n \times n$ matrix whose columns are the search directions generated by the algorithm, and $\det[\mathbf{D}(\mathbf{x})]$ denotes the determinant of $\mathbf{D}(\mathbf{x})$.
2. The minimum of f along any line in E_n is unique.

Given a starting point \mathbf{x}_1 , suppose that the algorithm generates the sequence $\{\mathbf{x}_k\}$ as follows. If $\nabla f(\mathbf{x}_k) = \mathbf{0}$, then the algorithm stops with \mathbf{x}_k ; otherwise $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, k is replaced by $k+1$, and the process is repeated. If the sequence $\{\mathbf{x}_k\}$ is contained in a compact subset of E_n , then each accumulation point \mathbf{x} of the sequence $\{\mathbf{x}_k\}$ must satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.

Proof

If the sequence $\{\mathbf{x}_k\}$ is finite, then the result is immediate. Now suppose that the algorithm generates the infinite sequence $\{\mathbf{x}_k\}$.

Let \mathcal{K} be an infinite sequence of positive integers, and suppose that the sequence $\{\mathbf{x}_k\}_{\mathcal{K}}$ converges to a point \mathbf{x} . We need to show that $\nabla f(\mathbf{x}) = \mathbf{0}$. Suppose by contradiction that $\nabla f(\mathbf{x}) \neq \mathbf{0}$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}'}$. By assumption, this sequence is contained in a compact subset of E_n , and hence there exists $\mathcal{K}' \subset \mathcal{K}$ such that $\{\mathbf{x}_{k+1}\}_{\mathcal{K}'}$ converges to \mathbf{x}' . We will show that \mathbf{x}'

could be obtained from \mathbf{x} by minimizing f along a set of n linearly independent directions.

Let \mathbf{D}_k be the $n \times n$ matrix whose columns $\mathbf{d}_{1k}, \dots, \mathbf{d}_{nk}$ are the search directions generated at iteration k . Thus, $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{D}_k \boldsymbol{\lambda}_k = \mathbf{x}_k + \sum_{j=1}^n \mathbf{d}_{jk} \lambda_{jk}$, where λ_{jk} is the distance moved along \mathbf{d}_{jk} . In particular, letting $\mathbf{y}_{1k} = \mathbf{x}_k$, $\mathbf{y}_{j+1,k} = \mathbf{y}_{jk} + \lambda_{jk} \mathbf{d}_{jk}$ for $j = 1, \dots, n$ it follows that $\mathbf{x}_{k+1} = \mathbf{y}_{n+1,k}$, and

$$f(\mathbf{y}_{j+1,k}) \leq f(\mathbf{y}_{jk} + \lambda \mathbf{d}_{jk}) \quad \text{for all } \lambda \in E_1 \text{ and } j = 1, \dots, n \quad (7.5)$$

Since $\det[\mathbf{D}_k] \geq \varepsilon > 0$, \mathbf{D}_k is invertible, so that $\boldsymbol{\lambda}_k = \mathbf{D}_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$. Since each column of \mathbf{D}_k has norm 1, there exists $\mathcal{K}'' \subset \mathcal{K}'$ such that $\mathbf{D}_k \rightarrow \mathbf{D}$. Since $\det[\mathbf{D}_k] \geq \varepsilon$ for each k , $\det[\mathbf{D}] \geq \varepsilon$, so that \mathbf{D} is invertible. Now, for $k \in \mathcal{K}''$, $\mathbf{x}_{k+1} \rightarrow \mathbf{x}'$, $\mathbf{x}_k \rightarrow \mathbf{x}$, $\mathbf{D}_k \rightarrow \mathbf{D}$, so that $\boldsymbol{\lambda}_k \rightarrow \boldsymbol{\lambda}$, where $\boldsymbol{\lambda} = \mathbf{D}^{-1}(\mathbf{x}' - \mathbf{x})$. Therefore, $\mathbf{x}' = \mathbf{x} + \mathbf{D}\boldsymbol{\lambda} = \mathbf{x} + \sum_{j=1}^n \mathbf{d}_j \lambda_j$. Let $\mathbf{y}_1 = \mathbf{x}$, and for $j = 1, \dots, n$, let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$, so that $\mathbf{x}' = \mathbf{y}_{n+1}$. To show that \mathbf{x}' is obtained from \mathbf{x} by minimizing f sequentially along $\mathbf{d}_1, \dots, \mathbf{d}_n$, it suffices to show that

$$f(\mathbf{y}_{j+1}) \leq f(\mathbf{y}_j + \lambda \mathbf{d}_j) \quad \text{for all } \lambda \in E_1 \text{ and } j = 1, \dots, n \quad (7.6)$$

Note that $\lambda_{jk} \rightarrow \lambda_j$, $\mathbf{d}_{jk} \rightarrow \mathbf{d}_j$, $\mathbf{x}_k \rightarrow \mathbf{x}$, and $\mathbf{x}_{k+1} \rightarrow \mathbf{x}'$ as $k \in \mathcal{K}''$ approaches ∞ , so that $\mathbf{y}_{jk} \rightarrow \mathbf{y}_j$ for $j = 1, \dots, n+1$ as $k \in \mathcal{K}''$ approaches ∞ . By continuity of f , then, (7.6) follows from (7.5). We have thus shown that \mathbf{x}' is obtained from \mathbf{x} by minimizing f sequentially along the directions $\mathbf{d}_1, \dots, \mathbf{d}_n$.

Obviously, $f(\mathbf{x}') \leq f(\mathbf{x})$. First, consider the case $f(\mathbf{x}') < f(\mathbf{x})$. Since $\{f(\mathbf{x}_k)\}$ is a nonincreasing sequence, and since $f(\mathbf{x}_k) \rightarrow f(\mathbf{x})$ as $k \in \mathcal{K}$ approaches ∞ , $\lim_{k \rightarrow \infty} f(\mathbf{x}_k) = f(\mathbf{x})$. This is impossible, however, in view of the fact that $\mathbf{x}_{k+1} \rightarrow \mathbf{x}'$ as $k \in \mathcal{K}'$ approaches ∞ and the assumption that $f(\mathbf{x}') < f(\mathbf{x})$. Now consider the case $f(\mathbf{x}') = f(\mathbf{x})$. By property 2 of the theorem, and since \mathbf{x}' is obtained from \mathbf{x} by minimizing f along $\mathbf{d}_1, \dots, \mathbf{d}_n$, $\mathbf{x}' = \mathbf{x}$. This further implies that $\nabla f(\mathbf{x}') \mathbf{d}_j = 0$ for $j = 1, \dots, n$. Since $\mathbf{d}_1, \dots, \mathbf{d}_n$ are linearly independent, $\nabla f(\mathbf{x}) = 0$, contradicting our assumption. This completes the proof.

Note that no closedness or continuity assumptions are made on the map providing the search directions. It is only required that the search directions used at each iteration be linearly independent and that as these directions converge, the limiting directions must also be linearly independent. Obviously this holds true if a fixed set of linearly independent search directions are used at every iteration. Alternatively, if the search directions used at each iteration are mutually orthogonal, and each has norm 1, then the search matrix \mathbf{D} satisfies $\mathbf{D}'\mathbf{D} = \mathbf{I}$. Therefore, $\det[\mathbf{D}] = 1$, so that condition 1 of the theorem holds true.

Also note that assumption 2 in the statement of the theorem is used to ensure the following property. If a differentiable function f is minimized along n independent directions starting from a point \mathbf{x} and resulting in \mathbf{x}' , then

$f(\mathbf{x}') < f(\mathbf{x})$, provided that $\nabla f(\mathbf{x}) \neq 0$. Without assumption 2, this is not true, as evidenced by $f(x_1, x_2) = x_2(1 - x_1)$. If $\mathbf{x} = (0, 0)'$, then minimizing f starting from \mathbf{x} along $\mathbf{d}_1 = (1, 0)'$ and then along $\mathbf{d}_2 = (0, 1)'$ could produce the point $\mathbf{x}' = (1, 1)'$, where $f(\mathbf{x}') = f(\mathbf{x}) = 0$, even though $\nabla f(\mathbf{x}) = (0, 1)' \neq (0, 0)'$.

7.4 Comparison Among Algorithms

In the remainder of the text, we discuss several algorithms for solving different classes of nonlinear programming problems. This section discusses some important factors that must be considered when assessing the effectiveness of these algorithms and when comparing them. These factors are (1) generality, reliability, and precision; (2) sensitivity to parameters and data; (3) preparational and computational effort; and (4) convergence.

Generality, Reliability, and Precision

Different algorithms are designed for solving various classes of nonlinear programming problems, such as unconstrained optimization problems, problems with inequality constraints, problems with equality constraints, and problems with both types of constraints. Within each of these classes, different algorithms make specific assumptions about the problem structure. For example, for unconstrained optimization problems, some procedures assume that the objective function is differentiable, whereas other algorithms do not make this assumption and rely primarily on functional evaluations only. For problems with equality constraints, some algorithms can only handle linear constraints, while others can handle nonlinear constraints as well. Thus generality of an algorithm refers to the variety of problems that the algorithm can handle and also to the restrictiveness of the assumptions required by the algorithm.

Another important factor is the reliability, or robustness, of the algorithm. Given any algorithm, it is not difficult to construct a test problem that it cannot solve effectively. Reliability means the ability of the procedure to solve most of the problems in the class for which it is designed with reasonable accuracy. The relationship between reliability of a certain procedure and the problem size and structure cannot be overlooked. Some algorithms are reliable if the number of variables is small or if the constraints are not highly nonlinear, and not reliable otherwise.

As implied by Theorem 7.2.3, convergence of nonlinear programming algorithms usually occurs in a limiting sense, if at all. Thus, we are interested in measuring the quality of the points produced by the algorithm after a reasonable number of iterations. Algorithms that quickly produce feasible solutions with good objective values are preferred. As discussed in Chapter 6 on duality and as will be seen in Chapter 9 on penalty functions, several procedures

generate a sequence of infeasible solutions, where feasibility is achieved only at termination. Hence, at later iterations, it is imperative that the degree of infeasibility be small so that a near-feasible solution will be at hand if the algorithmic process is prematurely terminated.

Sensitivity to Parameters and Data

For most algorithms, the user must set initial values for certain parameters, such as the starting vector, the step size, the acceleration factor, and parameters for terminating the algorithm. Some procedures are quite sensitive to these parameters and to the problem data and may produce different results or stop prematurely, depending on their values. In particular, for a fixed set of selected parameters, the algorithm should solve the problem for a wide range of problem data. Likewise, for a given set of problem data, one would prefer that the algorithm not be very sensitive to the selected values of the parameters.

Preparational and Computational Effort

Another basis for comparing algorithms is the total effort, both preparational and computational, expended for solving problems. The effort of preparing the input data should be taken into consideration when evaluating an algorithm. An algorithm that uses first- or second-order derivatives, especially if the original functions are complicated, requires a considerably larger amount of preparation time than one that only uses functional evaluations. The computational effort of an algorithm is usually assessed by the computer time, the number of iterations, or the number of functional evaluations. However, any of these measures, by itself, is not entirely satisfactory. The computer time needed to execute an algorithm depends not only on its efficiency but also on the type of machine used, the character of the measured time, and the efficiency of coding. Also, the number of iterations cannot be used as the only measure of effectiveness of an algorithm because the effort per iteration may vary considerably from one procedure to another. Finally, the number of functional evaluations can be misleading, since it does not measure other operations, such as matrix multiplication, matrix inversion, and finding suitable directions of movement. In addition, for derivative dependent methods, we have to weigh the evaluation of first- and second-order derivatives against the evaluation of the functions themselves.

Convergence

Theoretical convergence of algorithms to points in the solution set is a highly desirable property. Given two competing algorithms that converge, they could

be compared theoretically on the basis of the order or speed of convergence. This notion is defined below.

7.4.1 Definition

Let the sequence $\{r_k\}$ converge to \bar{r} . The *order of convergence* of the sequence is the supremum of the nonnegative numbers p satisfying

$$\lim_{k \rightarrow \infty} \frac{|r_{k+1} - \bar{r}|}{|r_k - \bar{r}|^p} = \beta < \infty$$

If $p = 1$, the sequence is said to have *linear convergence* if the *convergence ratio* β is less than 1. If $p > 1$, or if $p = 1$ and $\beta = 0$, the sequence is said to have *superlinear convergence*.

If r_k in the above definition represents $\alpha(x_k)$, the value of the descent function at the k th iteration, then the larger the value of p , the faster the convergence of the algorithm. If the limit in Definition 7.4.1 exists, then for large values of k , we asymptotically have $|r_{k+1} - \bar{r}| = \beta |r_k - \bar{r}|^p$, which indicates faster convergence for larger values of p . For the same value of p , the smaller the convergence ratio β , the faster the convergence. It should be noted, however, that the order of convergence and the ratio of convergence must not be solely used for evaluating algorithms that converge, since they represent the progress of the algorithm only as the number of iterations approach infinity.

Another convergence criterion frequently used in comparing algorithms is their ability to effectively minimize quadratic functions. This is used because, near the minimum a linear approximation to a function is poor, while it can be adequately approximated by a quadratic form. Thus, an algorithm that does not perform well for minimizing a quadratic function is unlikely to perform well for a general nonlinear function as we move closer to the optimum.

In order to show that M is not closed, a sequence (x_k, d_k) converging to (x, d) and a sequence $y_k \in M(x_k, d_k)$ converging to y must be exhibited such that $y \notin M(x, d)$. Given that $x_1 = (1, 0)'$, x_{k+1} is the point on the circle $(x_1 - 1)^2 + (x_2 - 1)^2 = 1$ midway between x_k and $(0, 1)'$. The vector d_k is defined by $(x_{k+1} - x_k) / \|x_{k+1} - x_k\|$. Letting $f(x_1, x_2) = (x_1 + 2)^2 + (x_2 - 2)^2$, show that:

- a. The sequence $\{x_k\}$ converges to $x = (0, 1)'$.
- b. The vectors $\{d_k\}$ converge to $d = (0, 1)'$.
- c. The sequence $\{y_k\}$ converges to $y = (0, 1)'$.
- d. The map M is not closed at (x, d) .

7.15 Let $f: E_n \rightarrow E_1$ be a differentiable function. Consider the following direction-finding map $D: E_n \rightarrow E_n \times E_n$ that gives the *deflected negative gradient*. Given $x \geq 0$, then $(x, d) \in D(x)$ means that

$$d_i = \begin{cases} \frac{-\partial f(x)}{\partial x_i} & \text{if } x_i > 0, \text{ or } x_i = 0 \text{ and } \frac{\partial f(x)}{\partial x_i} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

Show that D is not closed.
Hint: Let $f(x_1, x_2) = x_1 - x_2$ and consider the sequence $\{x_k\}$ converging to $(0, 1)'$, where $x_k = (\frac{1}{k}, 1)'$.

7.16 Let $f: E_n \rightarrow E_1$ be a differentiable function. Consider the composite map $A = MD$, where $D: E_n \rightarrow E_n \times E_n$ and $M: E_n \times E_n \rightarrow E_n$ are defined as follows. Given $x \geq 0$, then $(x, d) \in D(x)$ means that

$$d_i = \begin{cases} \frac{-\partial f(x)}{\partial x_i} & \text{if } x_i > 0, \text{ or if } x_i = 0 \text{ and } \frac{\partial f(x)}{\partial x_i} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

The vector $y \in M(x, d)$ means that $y = x + \lambda d$ for some $\lambda \geq 0$, and furthermore, $\bar{\lambda}$ solves the problem to minimize $f(x + \lambda d)$ subject to $x + \lambda d \geq 0, \lambda \geq 0$.

a. Find an optimal solution to the following problem using the Kuhn-Tucker conditions:

Minimize $x_1^2 + x_2^2 - x_1 x_2 + 2x_1 + x_2$
subject to $x_1, x_2 \geq 0$

b. Starting from the point $(2, 1)$, solve the problem in part a using the algorithm defined by the algorithmic map A . Note that the algorithm converges to the optimal solution obtained in part a.

c. Starting from the point $(0, 0.09, 0)$, solve the following problem credited to Wolfe [1972] using the algorithm defined by A .

Minimize $\frac{4}{3}(x_1^2 - x_1 x_2 + x_2^2)^{3/4} - x_3$
subject to $x_1, x_2, x_3 \geq 0$

Note that the sequence generated converges to the point $(0, 0, \bar{x}_3)$, where $\bar{x}_3 = 0.3(1 + 0.5\sqrt{2})$. Using the Kuhn-Tucker conditions, show that this point is not an optimal solution.

Note that the algorithm converges to an optimal solution in part b but not in part c. This is because map A is not closed, as seen in Exercises 7.14 and 7.15.

7.17 This exercise illustrates that a map for a convergent algorithm need not be closed. Consider the following problem.

Minimize x^2
subject to $x \in E_1$

Consider the maps $B, C: E_1 \rightarrow E_1$ defined below.

$$B(x) = \frac{x}{2} \quad \text{for all } x$$
$$C(x) = \begin{cases} x & \text{if } -1 \leq x \leq 1 \\ x + 1 & \text{if } x < -1 \\ x - 1 & \text{if } x > 1 \end{cases}$$

Let the solution set $\Omega = \{0\}$, and let the descent function $\alpha(x) = x^2$.
a. Show that B and C satisfy all the assumptions of Theorem 7.3.4.
b. Verify that the composite map $A = CB$ is as given below, and verify that it is not closed.

$$A(x) = \begin{cases} \frac{x}{2} & \text{if } -2 \leq x \leq 2 \\ \frac{x}{2} + 1 & \text{if } x < -2 \\ \frac{x}{2} - 1 & \text{if } x > 2 \end{cases}$$

c. Despite the fact that A is not closed, show that the algorithm defined by A converges to the point $\bar{x} = 0$, regardless of the starting point.

7.18 In Theorem 7.3.5 we assumed that $\det[D(x)] > \epsilon > 0$. Could this assumption be replaced by the following?

At each point, x_k , generated by the algorithm, the search directions, d_1, \dots, d_n , generated by the algorithm are linearly independent.

7.19 Let X be a closed set in E_n , and let $f: E_n \rightarrow E_1$ and $\beta: E_n \rightarrow E_{m+1}$ be continuous. Show that the point-to-set map $C: E_{m+1} \rightarrow E_n$ defined below is closed.

$$y \in C(w) \text{ if } y \text{ solves the problem to minimize } f(x) + w'\beta(x) \text{ subject to } x \in X.$$

7.20 This exercise introduces a unified approach to the class of *cutting plane methods* that are frequently used in nonlinear programming. We state the algorithm and then give the assumptions under which the algorithm converges. The symbol \mathcal{P} represents the collection of polyhedral sets in E_p , and Ω is the nonempty solution set in E_q .

A General Cutting Plane Algorithm

Initialization Step

Choose a nonempty polyhedral set $Z_1 \subset E_p$, let $k = 1$, and go to the main step.

Main Step

1. Given Z_k , let $\mathbf{w}_k \in \mathbf{B}(Z_k)$, where $\mathbf{B}: \mathcal{S} \rightarrow E_q$. If $\mathbf{w}_k \in \Omega$, stop; otherwise, go to step 2.
2. Let $\mathbf{v}_k \in \mathbf{C}(\mathbf{w}_k)$, where $\mathbf{C}: E_q \rightarrow E_r$. Let $\mathbf{a}: E_r \rightarrow E_1$ and $\mathbf{b}: E_r \rightarrow E_p$ be continuous functions, and let

$$Z_{k+1} = Z_k \cap \{\mathbf{x}: \mathbf{a}(\mathbf{v}_k) + \mathbf{b}'(\mathbf{v}_k)\mathbf{x} \geq 0\}$$

Replace k by $k+1$, and repeat step 1.

Convergence of the Cutting Plane Algorithm

Under the following assumptions, either the algorithm stops in a finite number of steps at a point in Ω or it generates the infinite sequence $\{\mathbf{w}_k\}$ such that all of its accumulation points belong to Ω .

1. $\{\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ are contained in compact sets in E_q and E_r , respectively.
2. For each Z , if $\mathbf{w} \in \mathbf{B}(Z)$, then $\mathbf{w} \in Z$.
3. \mathbf{C} is a closed map.
4. Given $\mathbf{w} \notin \Omega$ and Z , where $\mathbf{w} \in \mathbf{B}(Z)$, then $\mathbf{v} \in \mathbf{C}(\mathbf{w})$ implies that $\mathbf{w} \notin \{\mathbf{x}: \mathbf{a}(\mathbf{v}) + \mathbf{b}'(\mathbf{v})\mathbf{x} \geq 0\}$, and $Z \cap \{\mathbf{x}: \mathbf{a}(\mathbf{v}) + \mathbf{b}'(\mathbf{v})\mathbf{x} \geq 0\} \neq \emptyset$.

Prove the above convergence theorem.

Hint: Let $\{\mathbf{w}_k\}_K$ and $\{\mathbf{v}_k\}_K$ be convergent subsequences with limits \mathbf{w} and \mathbf{v} . First, show that for any k , we must have

$$\mathbf{a}(\mathbf{v}_k) + \mathbf{b}'(\mathbf{v}_k)\mathbf{w}_l \geq 0 \quad \text{for all } l \geq k+1$$

Taking limits, show that $\mathbf{a}(\mathbf{v}) + \mathbf{b}'(\mathbf{v})\mathbf{w} \geq 0$. This inequality, together with assumptions 3 and 4, imply that $\mathbf{w} \in \Omega$, because otherwise a contradiction can be obtained.

- 7.21** Consider the dual cutting plane algorithm described in Section 6.4 for maximizing the dual function.

- a. Show that the dual cutting plane algorithm is a special form of the general cutting plane algorithm discussed in Exercise 7.20.
- b. Verify that the assumptions 1 through 4 of the convergence theorem stated in Exercise 7.20 hold true, so that the dual cutting plane algorithm converges to an optimal solution to the dual problem.

Hint: Referring to Exercise 7.19, note that the map \mathbf{C} is closed.

- 7.22** This exercise describes the cutting plane algorithm of Kelley [1960] for solving a problem of the following form, where g_i for $i = 1, \dots, m$ are convex.

$$\begin{aligned} &\text{Minimize} && \mathbf{c}'\mathbf{x} \\ &\text{subject to} && g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \dots, m \\ &&& \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

Kelley's Cutting Plane Algorithm**Initialization Step**

Let X_1 be a polyhedral set such that $X_1 \supset \{\mathbf{x}: g_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \dots, m\}$. Let $Z_1 = X_1 \cap \{\mathbf{x}: \mathbf{Ax} \leq \mathbf{b}\}$, let $k = 1$, and go to the main step.

Main Step

1. Solve the linear program to minimize $\mathbf{c}'\mathbf{x}$ subject to $\mathbf{x} \in Z_k$. Let \mathbf{x}_k be an optimal solution. If $g_i(\mathbf{x}_k) \leq 0$ for all i , stop; \mathbf{x}_k is an optimal solution. Otherwise, go to step 2.
2. Let $g_i(\mathbf{x}_k) = \text{maximum}_{1 \leq i \leq m} g_i(\mathbf{x}_k)$, and let

$$Z_{k+1} = Z_k \cap \{\mathbf{x}: g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)'(\mathbf{x} - \mathbf{x}_k) \leq 0\}$$

Replace k by $k+1$, and repeat step 1.

(Obviously $\nabla g_i(\mathbf{x}_k) \neq \mathbf{0}$, because otherwise, $g_i(\mathbf{x}) \geq g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)'(\mathbf{x} - \mathbf{x}_k) > 0$ for all \mathbf{x} , implying that the problem is infeasible.)

- a. Apply the above algorithm to solve the following problem:

$$\begin{aligned} &\text{Minimize} && -3x_1 - x_2 \\ &\text{subject to} && x_1^2 + x_2 + 1 \leq 0 \\ &&& x_1 + x_2 \leq 3 \\ &&& x_1, x_2 \geq 0 \end{aligned}$$

- b. Show that Kelley's algorithm is a special case of the general cutting plane algorithm of Exercise 7.20.
- c. Show that the above algorithm converges to an optimal solution using the convergence theorem of Exercise 7.20.
- d. Consider the problem to minimize $f(\mathbf{x})$ subject to $g_i(\mathbf{x}) \leq 0$ for $i = 1, \dots, m$ and $\mathbf{Ax} \leq \mathbf{b}$. Show how the problem could be reformulated so that the above algorithm be applied.

(*Hint:* Consider adding the constraint $f(\mathbf{x}) - z \leq 0$.)

- 7.23** This exercise describes the *supporting hyperplane method* of Veinott [1967] for solving a problem of the following form, where g_i for all i are pseudoconvex and where $g_i(\hat{\mathbf{x}}) < 0$ for $i = 1, \dots, m$ for some point $\hat{\mathbf{x}} \in E_n$.

$$\begin{aligned} &\text{Minimize} && \mathbf{c}'\mathbf{x} \\ &\text{subject to} && g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \dots, m \\ &&& \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

Veinott's Supporting Hyperplane Algorithm**Initialization Step**

Let X_1 be a polyhedral set such that $X_1 \supset \{\mathbf{x}: g_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \dots, m\}$. Let $Z_1 = X_1 \cap \{\mathbf{x}: \mathbf{Ax} \leq \mathbf{b}\}$, let $k = 1$, and go to the main step.

Main Step

1. Solve the linear program to minimize $\mathbf{c}'\mathbf{x}$ subject to $\mathbf{x} \in Z_k$. Let \mathbf{x}_k be an optimal solution. If $g_i(\mathbf{x}_k) \leq 0$ for all i , stop; \mathbf{x}_k is an optimal solution to the original problem. Otherwise, go to step 2.
2. Let $\bar{\mathbf{x}}_k$ be the point on the line segment joining \mathbf{x}_k and $\hat{\mathbf{x}}$, and on the boundary of the region $\{\mathbf{x}: g_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \dots, m\}$. Let $g_i(\bar{\mathbf{x}}_k) = 0$ and let

$$Z_{k+1} = Z_k \cap \{\mathbf{x}: \nabla g_i(\bar{\mathbf{x}}_k)'(\mathbf{x} - \bar{\mathbf{x}}_k) \leq 0\}$$

Replace k by $k+1$, and repeat step 1.