Chapter 1

LINEAR COMPLEMENTARITY PROBLEM, ITS GEOMETRY, AND APPLICATIONS

1.1 THE LINEAR COMPLEMENTARITY PROBLEM AND ITS GEOMETRY

The **Linear Complementarity Problem** (abbreviated as LCP) is a general problem which unifies linear and quadratic programs and bimatrix games. The study of LCP has led to many far reaching benefits. For example, an algorithm known as the **complementary pivot algorithm** first developed for solving LCPs, has been generalized in a direct manner to yield efficient algorithms for computing Brouwer and Kakutani fixed points, for computing economic equilibria, and for solving systems of nonlinear equations and nonlinear programming problems. Also, iterative methods developed for solving LCPs hold great promise for handling very large scale linear programs which cannot be tackled with the well known simplex method because of their large size and the consequent numerical difficulties. For these reasons the study of LCP offers rich rewards for people learning or doing research in optimization or engaged in practical applications of optimization. In this book we discuss the LCP in all its depth.

Let M be a given square matrix of order n and q a column vector in $\mathbf{R}^n$. Throughout this book we will use the symbols $w_1, \dots, w_n$; $z_1, \dots, z_n$ to denote the variables in the problem. **In an** LCP **there is no objective function to be optimized**. The problem is: find $w = (w_1, \dots, w_n)^T$, $z = (z_1, \dots, z_n)^T$ satisfying

$$
\begin{aligned}
& w - Mz = q \\
& w \geqq 0,\ z \geqq 0 \quad \text{and} \quad w_i z_i = 0 \quad \text{for all } i
\end{aligned}
\tag{1.1}
$$

The only data in the problem is the column vector q and the square matrix M. So we will denote the LCP of finding $w \in \mathbf{R}^n$, $z \in \mathbf{R}^n$ satisfying (1.1) by the symbol (q, M). It is said to be an LCP of **order** n. In an LCP of order n there are $2n$ variables. As a specific example, let $n = 2$, $M = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, $q = \begin{pmatrix} -5 \\ -6 \end{pmatrix}$. This leads to the LCP

$$\begin{aligned} w_1 \qquad - 2z_1 - \ z_2 &= -5 \\ w_2 - \ z_1 - 2z_2 &= -6. \\ w_1, w_2, z_1, z_2 \geqq 0 \quad \text{and} \quad w_1 z_1 &= w_2 z_2 = 0. \end{aligned} \tag{1.2}$$

The problem (1.2) can be expressed in the form of a vector equation as

$$w_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + w_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + z_1 \begin{pmatrix} -2 \\ -1 \end{pmatrix} + z_2 \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} -5 \\ -6 \end{pmatrix} \tag{1.3}$$

$$w_1, w_2, z_1, z_2 \geqq 0 \quad \text{and} \quad w_1 z_1 = w_2 z_2 = 0 \tag{1.4}$$

In any solution satisfying (1.4), at least one of the variables in each pair (w_j, z_j), has to equal zero. One approach for solving this problem is to pick one variable from each of the pairs (w_1, z_1), (w_2, z_2) and to fix them at zero value in (1.3). The remaining variables in the system may be called **usable variables**. After eliminating the zero variables from (1.3), if the remaining system has a solution in which the usable variables are nonnegative, that would provide a solution to (1.3) and (1.4).

Pick w_1, w_2 as the zero-valued variables. After setting w_1, w_2 equal to 0 in (1.3), the remaining system is

$$\begin{gathered} z_1 \begin{pmatrix} -2 \\ -1 \end{pmatrix} + z_2 \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} -5 \\ -6 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = q \\ z_1 \geqq 0, \quad z_2 \geqq 0 \end{gathered} \tag{1.5}$$

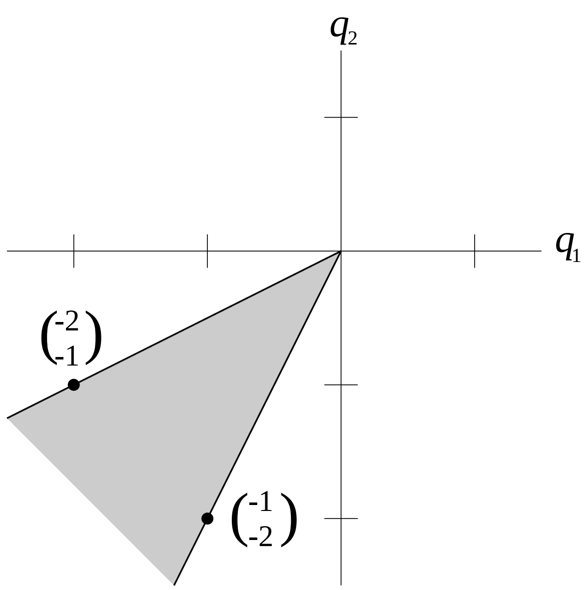


Figure 1.1 A Complementary Cone

Equation (1.5) has a solution iff the vector q can be expressed as a nonnegative linear combination of the vectors $(-2, -1)^T$ and $(-1, -2)^T$. The set of all nonnegative linear combinations of $(-2,-1)^T$ and $(-1,-2)^T$ is a cone in the q_1, q_2-space as in Figure 1.1. Only if the given vector $q = (-5,-6)^T$ lies in this cone, does the LCP (1.2) have a solution in which the usable variables are z_1, z_2. We verify that the point $(-5,-6)^T$ does lie in the cone, that the solution of (1.5) is $(z_1, z_2) = (4/3, 7/3)$ and, hence, a solution for (1.2) is $(w_1, w_2; z_1, z_2) = (0, 0; 4/3, 7/3)$. The cone in Figure 1.1 is known as a **complementary cone** associated with the LCP (1.2). Complementary cones are generalizations of the well-known class of quadrants or orthants.

1.1.1 Notation

The symbol I usually denotes the unit matrix. If we want to emphasize its order, we denote the unit matrix of order n by the symbol I_n.

We will use the abbreviation LP for "Linear Program" and BFS for "Basic Feasible Solution". See [1.28, 2.26]. LCP is the abbreviation for "Linear Complementarity Problem" and NLP is the abbreviation for "Nonlinear Program".

Column and Row Vectors of a Matrix

If $A = (a_{ij})$ is a matrix of order $m \times n$ say, we will denote its jth column vector $(a_{1j}, \dots, a_{mj})^T$ by the symbol $A_{.j}$, and its ith row vector $(a_{i1}, \dots, a_{in})$ by $A_{i.}$.

Nonnegative, Semipositive, Positive Vectors

Let $x = (x_1, \dots, x_n)^T \in \mathbf{R}^n$. $x \geqq 0$, that is nonnegative, if $x_j \geqq 0$ for all j. Clearly, $0 \geqq 0$. x is said to be semipositive, denoted by $x \geq 0$, if $x_j \geqq 0$ for all j and at least one $x_j > 0$. Notice the distinction in the symbols for denoting nonnegative ($\geqq$ with two lines under the $>$) and semipositive ($\geq$ with only a single line under the $>$). $0 \ngeq 0$, the zero vector is the only nonnegative vector which is not semipositive. Also, if $x \geq 0$, $\sum_{j=1}^{n} x_j > 0$. The vector $x > 0$, strictly positive, if $x_j > 0$ for all j. Given two vectors $x, y \in \mathbf{R}^n$; we write $x \geqq y$, if $x - y \geqq 0$, $x \geq y$ if $x - y \geq 0$, and $x > y$ if $x - y > 0$.

Pos Cones

If $\{x^1, \dots, x^r\} \subset \mathbf{R}^n$, the cone $\{x : x = \alpha_1 x^1 + \ldots + \alpha_r x^r, \alpha_1, \dots, \alpha_r \geqq 0\}$ is denoted by $\text{Pos}\{x^1, \dots, x^r\}$. Given the matrix A of order $m \times n$, $\text{Pos}(A)$ denotes the cone $\text{Pos}\{A_{.1}, \dots, A_{.n}\} = \{x : x = A\alpha \text{ for } \alpha = (\alpha_1, \dots, \alpha_n)^T \geqq 0\}$.

Directions, Rays, Half-Lines, and Step Length

Any point $y \in \mathbf{R}^n$, $y \neq 0$, defines a direction in $\mathbf{R}^n$. Given the direction y, it's ray is the half-line obtained by joining the origin 0 to y and continuing indefinitely in the same direction, it is the set of points $\{\alpha y : \alpha \geqq 0\}$. Given $\overline{x} \in \mathbf{R}^n$, by moving from $\overline{x}$ in the direction y we get points of the form $\overline{x} + \alpha y$ where $\alpha \geqq 0$, and the set of all such points $\{\overline{x} + \alpha y : \alpha \geqq 0\}$ is the **halfline** or **ray** through $\overline{x}$ in the direction y. The point $\overline{x} + \alpha y$ for $\alpha > 0$ is said to have been obtained by moving from $\overline{x}$ in the direction y a step length of α. As an example, if $y = (1, 1)^T \in \mathbf{R}^n$, the ray of y is the set of all points of the form $\{(\alpha, \alpha)^T : \alpha \geqq 0\}$. In addition, if, $\overline{x} = (1, -1)^T$, the halfline through $\overline{x}$ in the direction y is the set of all points of the form $\{(1 + \alpha, -1 + \alpha)^T : \alpha \geqq 0\}$. See Figure 1.2. In this half-line, letting $\alpha = 9$, we get the point $(10, 8)^T$, and this point is obtained by taking a step of length 9 from $\overline{x} = (1, -1)^T$ in the direction $y = (1, 1)^T$.

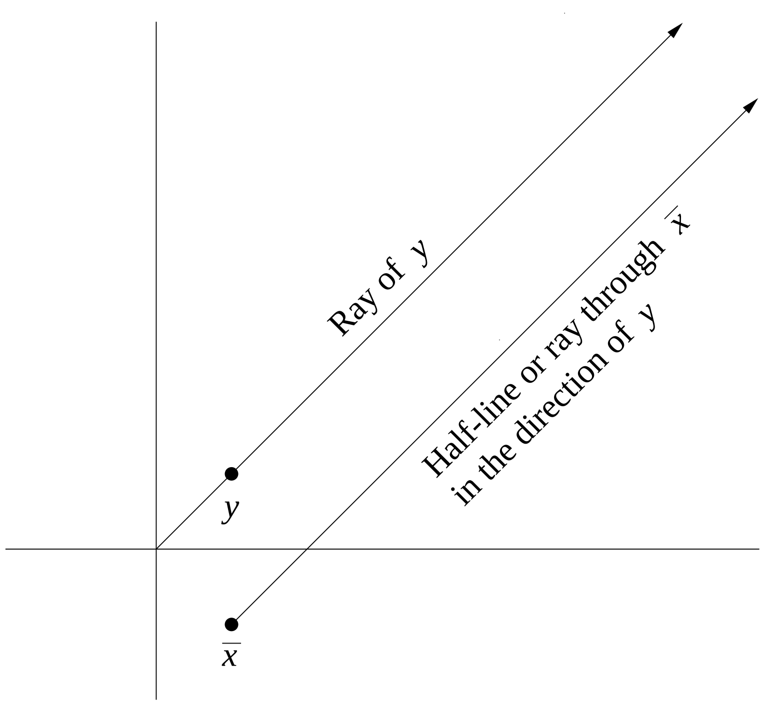


Figure 1.2 Rays and Half-Lines

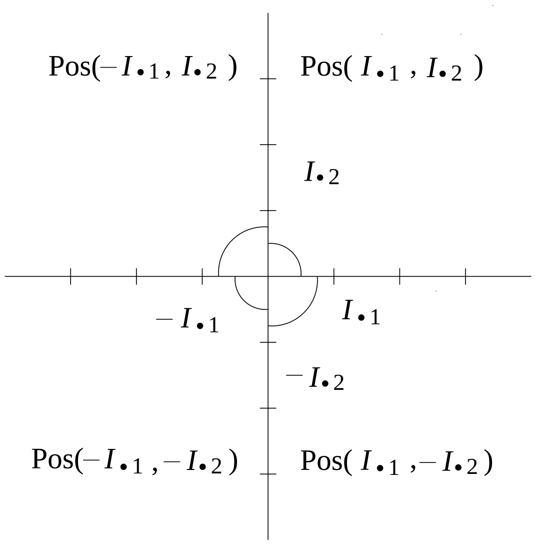
1.1.2 Complementary Cones

In the LCP (q, M), the complementary cones are defined by the matrix M. The point q does not play any role in the definition of complementary cones.

Let M be a given square matrix of order n. For obtaining $\mathcal{C}(M)$, the class of complementary cones corresponding to M, the pair of column vectors $(I_{.j}, -M_{.j})$ is known as the jth **complementary pair of vectors**, $1 \leqq j \leqq n$. Pick a vector from the pair $(I_{\cdot j}, -M_{\cdot j})$ and denote it by $A_{\cdot j}$. The ordered set of vectors $(A_{\cdot 1}, \ldots, A_{\cdot n})$ is known as a **complementary set of vectors**. The cone $\text{Pos}(A_{\cdot 1}, \ldots, A_{\cdot n}) = \{y : y = \alpha_1 A_{\cdot 1} + \ldots + \alpha_n A_{\cdot n}; \alpha_1 \geqq 0, \ldots, \alpha_n \geqq 0\}$ is known as a **complementary cone** in the class $\mathcal{C}(M)$. Clearly there are $2^{\mathbf{n}}$ complementary cones.

Example 1.1

Let $n = 2$ and $M = I$. In this case, the class $\mathcal{C}(I)$ is just the class of orthants in $\mathbf{R}^2$. In general for any n, $\mathcal{C}(I)$ is the class of orthants in $\mathbf{R}^n$. Thus the class of complementary cones is a generalization of the class of orthants. See Figure 1.3. Figures 1.4 and 1.5 provide some more examples of complementary cones. In the example in Figure 1.5 since $\{I_{\cdot 1}, -M_{\cdot 2}\}$ is a linearly dependent set, the cone $\text{Pos}(I_{\cdot 1}, -M_{\cdot 2})$ has an empty interior. It consists of all the points on the horizontal axis in Figure 1.6 (the thick axis). The remaining three complementary cones have nonempty interiors.



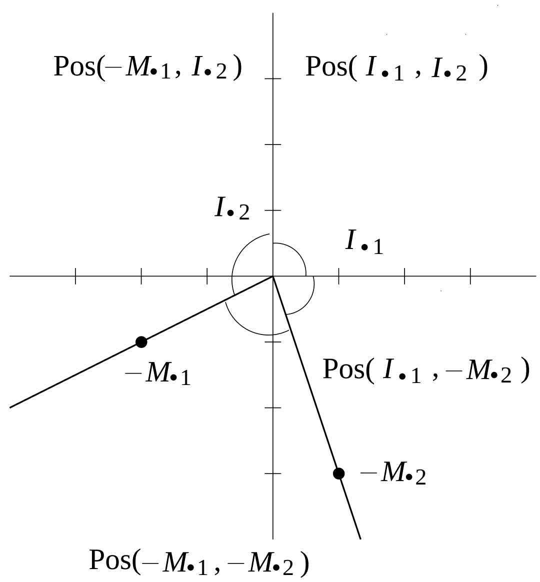
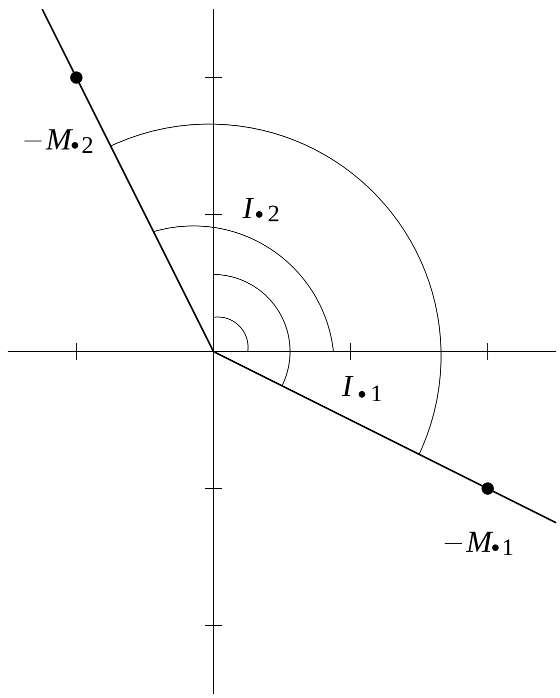


Figure 1.3 When $M = I$, the Complementarity Cones are the Orthants.

Figure 1.4 Complementary Cones when $M = \begin{pmatrix} 2 & -1 \\ 1 & 3 \end{pmatrix}$.

Degenerate, Nondegenerate Complementary Cones

Let $\text{Pos}(A_{\cdot 1}, \dots, A_{\cdot n})$ be a complementary cone in $\mathcal{C}(M)$. This cone is said to be a nondegenerate complementary cone if it has a nonempty interior, that is if $\{A_{\cdot 1}, \dots, A_{\cdot n}\}$ is a linearly independent set; degenerate complementary cone if its interior is empty, which happens when $\{A_{\cdot 1}, \dots, A_{\cdot n}\}$ is a linearly dependent set. As examples, all the complementary cones in Figures 1.3, 1.4, 1.5, are nondegenerate. In Figure 1.6 the complementary cone $\text{Pos}(I_{\cdot 1}, -M_{\cdot 2})$ is degenerate, the remaining three complementary cones are nondegenerate.



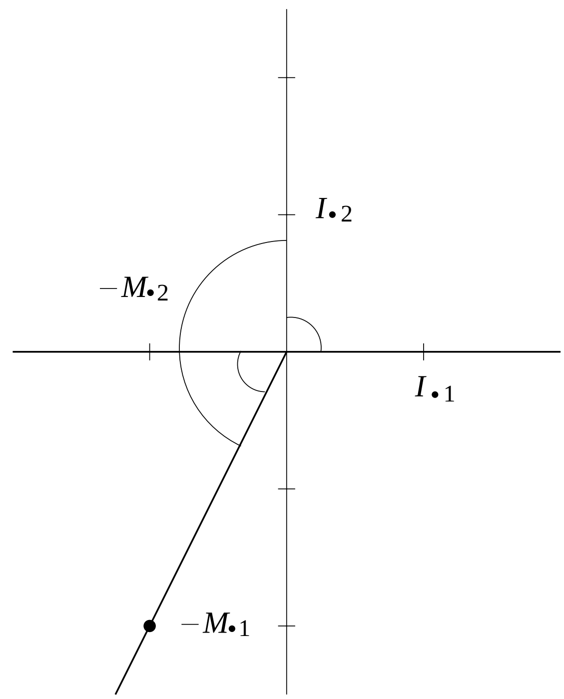


Figure 1.5 Complementary Cones when $M = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$.

Figure 1.6 Complementary Cones when $M = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$.

1.1.3 The Linear Complementarity Problem

Given the square matrix M of order n and the column vector $q \in \mathbf{R}^n$, the LCP (q, M), is equivalent to the problem of finding a complementary cone in $\mathcal{C}(M)$ that contains the point q, that is, to find a complementary set of column vectors $(A_{\cdot 1}, \dots, A_{\cdot n})$ such that

(i) $A_{\cdot j} \in \{I_{\cdot j}, -M_{\cdot j}\}$ for $1 \leqq j \leqq n$

(ii) q can be expressed as a nonnegative linear combination of $(A_{\cdot 1}, \dots, A_{\cdot n})$

where I is the identity matrix of order n and $I_{.j}$ is its jth column vector. This is equivalent to finding $w \in \mathbf{R}^n$, $z \in \mathbf{R}^n$ satisfying $\sum_{j=1}^{n} I_{.j} w_j - \sum_{j=1}^{n} M_{.j} z_j = q$, $w_j \geqq 0$, $z_j \geqq 0$ for all j, and either $w_j = 0$ or $z_j = 0$ for all j. In matrix notation this is

$$w - Mz = q \tag{1.6}$$

$$w \geqq 0 \quad z \geqq 0 \tag{1.7}$$

$$w_j z_j = 0 \quad \text{for all } j. \tag{1.8}$$

Because of (1.7), the condition (1.8) is equivalent to $\sum_{j=1}^{n} w_j z_j = w^T z = 0$; this condition is known as the **complementarity constraint**. In any solution of the LCP (q, M), if one of the variables in the pair (w_j, z_j) is positive, the other should be zero. Hence, the pair (w_j, z_j) is known as the jth **complementary pair of variables** and each variable in this pair is the **complement** of the other. In (1.6) the column vector corresponding to w_j is $I_{.j}$, and the column vector corresponding to z_j is $-M_{.j}$. For $j = 1$ to n, the pair $(I_{.j}, -M_{.j})$ is the jth complementary pair of column vectors in the LCP (q, M). For $j = 1$ to n, let $y_j \in \{w_j, z_j\}$ and let $A_{.j}$ be the column vector corresponding to y_j in (1.6). So $A_{.j} \in \{I_{.j} - M_{.j}\}$. Then $y = (y_1, \ldots, y_n)$ is a **complementary vector of variables** in this LCP, the ordered set $(A_{.1}, \ldots, A_{.n})$ is the **complementary set of column vectors corresponding to it** and the matrix A with its column vectors as $A_{.1}, \ldots, A_{.n}$ in that order is known as the **complementary matrix** corresponding to it. If $\{A_{.1}, \ldots, A_{.n}\}$ is linearly independent, y is a **complementary basic vector of variables** in this LCP, and the complementary matrix A whose column vectors are $A_{.1}, \ldots, A_{.n}$ in that order, is known as the **complementary basis** for (1.6) corresponding to the complementary basic vector y. The cone $\text{Pos}(A_{.1}, \ldots, A_{.n}) = \{x : x = \alpha_1 A_{.1} + \ldots + \alpha_n A_{.n},\ \alpha_1 \geqq 0, \ldots, \alpha_n \geqq 0\}$ is the complementary cone in the class $\mathcal{C}(M)$ corresponding to the complementary set of column vectors $(A_{.1}, \ldots, A_{.n})$, or the associated complementary vector of variables y. A **solution of the LCP** (q, M), always means a $(w; z)$ satisfying all the constraints (1.6), (1.7), (1.8).

A **complementary feasible basic vector** for this LCP is a complementary basic vector satisfying the property that q can be expressed as a nonnegative combination of column vectors in the corresponding complementary basis. Thus each complementary feasible basic vector leads to a solution of the LCP.

The union of all the complementary cones associated with the square matrix M is denoted by the symbol $\mathbf{K}(M)$. $\mathbf{K}(M)$ is clearly the set of all vectors q for which the LCP (q, M) has at least one solution.

We will say that the vector $\overline{z}$ **leads to a solution** of the LCP (q, M) iff $(\overline{w} = M\overline{z} + q, \overline{z})$ is a solution of this LCP.

As an illustration, here are all the complementary vectors of variables and the corresponding complementary matrices for (1.2), an LCP of order 2.

Complementary vector of variables	The corresponding complementary matrix
(w_1, w_2)	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
(w_1, z_2)	$\begin{pmatrix} 1 & -1 \\ 0 & -2 \end{pmatrix}$
(z_1, w_2)	$\begin{pmatrix} -2 & 0 \\ -1 & 1 \end{pmatrix}$
(z_1, z_2)	$\begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix}$

Since each of these complementary matrices is nonsingular, all the complementary vectors are complementary basic vectors, and all the complementary matrices are complementary bases, in this LCP. Since $q = (-5, -6)^T$ in (1.2) can be expressed as a nonnegative combination of the complementary matrix corresponding to (z_1, z_2); (z_1, z_2) is a complementary feasible basic vector for this LCP. The reader should draw all the complementary cones corresponding to this LCP on the two dimensional Cartesian plane, and verify that for this LCP, their union, the set $\mathbf{K}(M) = \mathbf{R}^2$.

The Total Enumeration Method for the LCP

Consider the LCP (q, M) of order n. The complementarity constraint (1.8) implies that in any solution (w, z) of this LCP, for each $j = 1$ to n, we must have

$$\begin{aligned} \text{either } & w_j = 0 \\ \text{or } & z_j = 0. \end{aligned}$$

This gives the LCP a combinatorial, rather than nonlinear flavour. It automatically leads to an enumeration method for the LCP.

There are exactly $2^{\mathbf{n}}$ complementary vectors of variables. Let

$$y^r = (y_1^r, \dots, y_n^r), \quad r = 1 \text{ to } 2^{\mathbf{n}}$$

where $y_j^r \in \{w_j, z_j\}$ for each $j = 1$ to n, be all the complementary vectors of variables. Let A_r be the complementary matrix corresponding to y^r, $r = 1$ to $2^{\mathbf{n}}$. Solve the following system (P_r).

$$\begin{aligned} A_r y^r &= q \\ y^r &\geqq 0\ . \end{aligned} \qquad (P_r)$$

This system can be solved by Phase I of the simplex method for LP, or by other methods for solving linear equality and inequality systems. If this system has a feasible solution, $\overline{y}^r$, say, then

$$y^r = \overline{y}^r$$

all variables not in y^r, equal to zero is a solution of LCP (q, M). If the complementary matrix A_r is singular, the system (P_r) may have no feasible solution, or have one or an infinite number of feasible solutions. Each feasible solution of (P_r) leads to a solution of the LCP (q, M) as discussed above. When this is repeated for $r = 1$ to $2^{\mathbf{n}}$, all solutions of the LCP (q, M) can be obtained. The method discussed at the beginning of Section 1.1 to solve an LCP of order 2 is exactly this enumeration method.

This enumeration method is convenient to use only when $n = 2$, since $2^{\mathbf{2}} = 4$ is small; and to check whether the system (P_r) has a solution for any r, we can draw the corresponding complementary cone in the two dimensional Cartesian plane and check whether it contains q. When $n > 2$, particularly for large n, this enumeration method becomes impractical since $2^{\mathbf{n}}$ grows very rapidly. In Chapter 2 and later chapters we discuss efficient pivotal and other methods for solving special classes of LCPs that arise in several practical applications. In Section 8.7 we show that the general LCP is a hard problem. At the moment, the only known algorithms which are guaranteed to solve the general LCP are enumerative methods, see Section 11.3.

1.2 APPLICATION TO LINEAR PROGRAMMING

In a general LP there may be some inequality constraints, equality constraints, sign restricted variables and unrestricted variables. Transform each lower bounded variable, say $x_j \geqq l_j$, into a nonnegative variable by substituting $x_j = l_j + y_j$ where $y_j \geqq 0$. Transform each sign restricted variable of the form $x_j \leqq 0$ into a nonnegative variable by substituting $x_j = -y_j$ where $y_j \geqq 0$. Eliminate the unrestricted variables one after the other, using the equality constraints (see Chapter 2 of [1.28 or 2.26]). In the resulting system, if there is still an equality constraint left, eliminate a nonnegative variable from the system using it, thereby transforming the constraint into an inequality constraint in the remaining variables. Repeat this process until there are no more equality constraints. In the resulting system, transform any inequality constraint of the "$\leqq$" form into one of "$\geqq$" form, by multiplying both sides of it by '-1'. If the objetive function is to be maximized, replace it by its negative which should be minimized, and eliminate any constant terms in it. When all this work is completed, the original LP is transformed into:

$$\begin{array}{ll} \text{Minimize} & cx \\ \text{Subject to} & Ax \geqq b \\ & x \geqq 0 \end{array} \tag{1.9}$$

which is in **symmetric form**. Here, suppose A is of order $m \times N$. If x is an optimum feasible solution of (1.9), by the results of the duality theory of linear programming (see [1.28, 2.26]) there exists a dual vector $y \in \mathbf{R}^m$, primal slack vector $v \in \mathbf{R}^m$, and dual slack vector $u \in \mathbf{R}^N$ which together satisfy

$$\begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c^T \\ -b \end{pmatrix}$$
$$\begin{pmatrix} u \\ v \end{pmatrix} \geqq 0 \quad \begin{pmatrix} x \\ y \end{pmatrix} \geqq 0 \quad \text{and} \quad \begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} = 0 \,. \tag{1.10}$$

Conversely, if u, v, x, y together satisfy all the conditions in (1.10), x is an optimum solution of (1.9). In (1.10) all the vectors and matrices are written in **partitioned form**. For example, $\binom{u}{v}$ is the vector $(u_1, \ldots, u_N, v_1, \ldots, v_m)^T$. If $n = m + N$,

$$w = \begin{pmatrix} u \\ v \end{pmatrix}, \; z = \begin{pmatrix} x \\ y \end{pmatrix}, \; M = \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}, \; q = \begin{pmatrix} c^T \\ -b \end{pmatrix},$$

(1.10) is seen to be an LCP of order n of the type (1.6) to (1.8). Solving the LP (1.9) can be achieved by solving the LCP (1.10).

Also, the various complementary pairs of variables in the LCP (1.10) are exactly those in the pair of primal, dual LPs (1.9) and its dual. As an example consider the following LP.

$$\begin{array}{lrl} \text{Minimize} & -13x_1 + 42x_2 & \\ \text{Subject to} & 8x_1 - \;\; x_2 + \;\; 3x_2 & \geqq -16 \\ & -\;3x_1 + \;2x_2 - 13x_3 & \geqq \;\; 12 \\ & x_j \geqq 0, \; j = & \;\; 1, 2, 3. \end{array}$$

Let (v_1, y_1), (v_2, y_2) denote the nonnegative slack variable, dual variable respectively, associated with the two primal constraints in that order. Let u_1, u_2, u_3 denote the nonnegative dual slack variable associated with the dual constraint corresponding to the primal variable x_1, x_2, x_3, in that order. Then the primal and dual systems together with the complementary slackness conditions for optimality are

$$\begin{array}{rcl} 8x_1 - \;\; x_2 + \;\; 3x_3 - v_1 & = & -16 \\ -3x_1 + \;2x_2 - 13x_3 \qquad - v_2 & = & 12 \\ 8y_1 - \;\; 3y_2 \qquad + u_1 & = & -13 \\ -\; y_1 + \;2y_2 \qquad + u_2 & = & 42 \\ 3y_1 - 13y_2 \qquad + u_3 & = & 0. \end{array}$$

$$x_j, u_j, y_i, v_i \geqq 0 \;\; \text{for all } i, j.$$
$$x_j u_j = y_i v_i = 0 \;\; \text{for all } i, j.$$

This is exactly the following LCP.

u_1	u_2	u_3	v_1	v_2	x_1	x_2	x_3	y_1	y_2	
1	0	0	0	0	0	0	0	8	–3	–13
0	1	0	0	0	0	0	0	–1	2	42
0	0	1	0	0	0	0	0	3	–13	0
0	0	0	1	0	–8	1	–3	0	0	16
0	0	0	0	1	3	–2	13	0	0	–12

All variables $\geqq 0$. $u_1x_1 = u_2x_2 = u_3x_3 = v_1y_1 = v_2y_2 = 0$.

1.3 QUADRATIC PROGRAMMING

Using the methods discussed in Section 1.2 any problem in which a quadratic objective function has to be optimized subject to linear equality and inequality constraints can be transformed into a problem of the form

$$\begin{array}{lrl} \text{Minimize} & Q(x) = & cx + \frac{1}{2}x^TDx \\ \text{Subject to} & Ax \geqq & b \\ & x \geqq & 0 \end{array} \tag{1.11}$$

where A is a matrix of order $m \times N$, and D is a **square symmetric matrix of order** N. There is no loss of generality in assuming that D is a symmetric matrix, because if it is not symmetric replacing D by $(D + D^T)/2$ (which is a symmetric matrix) leaves $Q(x)$ unchanged. **We assume that D is symmetric**.

1.3.1 Review on Positive Semidefinite Matrices

A square matrix $F = (f_{ij})$ of order n, whether it is symmetric or not, is said to be a **positive semidefinite matrix** if $y^TFy \geqq 0$ for all $y \in \mathbf{R}^n$. It is said to be a **positive definite matrix** if $y^TFy > 0$ for all $y \neq 0$. We will use the abbreviations PSD, PD for "positive semidefinite" and "positive definite", respectively.

Principal Submatrices, Principal Subdeterminants

Let $F = (f_{ij})$ be a square matrix of order n. Let $\{i_1, \ldots, i_r\} \subset \{1, \ldots, n\}$ with its elements arranged in increasing order. Erase all the entries in F in row i and column i for each $i \notin \{i_1, \ldots, i_r\}$. What remains is a square submatrix of F of order r:

$$\begin{pmatrix} f_{i_1,i_1} & \dots & f_{i_1,i_r} \\ \vdots & & \vdots \\ f_{i_r,i_1} & \dots & f_{i_r,i_r} \end{pmatrix}.$$

This submatrix is known as the **principal submatrix** of F determined by the subset $\{i_1, \dots, i_r\}$. Denoting the subset $\{i_1, \dots, i_r\}$ by $\mathbf{J}$, we denote this principal submatrix by the symbol $F_{\mathbf{JJ}}$. It is $(f_{ij} : i \in \mathbf{J}, j \in \mathbf{J})$. The determinant of this principal submatrix is called the principal subdeterminant of F determined by the subset $\mathbf{J}$. The principal submatrix of F determined by ϕ, the empty set, is the empty matrix which has no entries. Its determinant is defined by convention to be equal to 1. The principal submatrix of F determined by $\{1, \dots, n\}$ is F itself. The principal submatrices of F determined by nonempty subsets of $\{1, \dots, n\}$ are **nonempty principal submatrices** of F. Since the number of distinct nonempty subsets of $\{1, \dots, n\}$ is $2^{\mathbf{n}} - 1$, there are $2^{\mathbf{n}} - 1$ nonempty principal submatrices of F. The principal submatrices of F determined by proper subsets of $\{1, \dots, n\}$ are known as **proper principal submatrices** of F. So each proper principal submatrix of F is of order $\leqq n - 1$.

Example 1.2

Let

$$F = \begin{pmatrix} 0 & -1 & 2 \\ 1 & 3 & 4 \\ 1 & 5 & -3 \end{pmatrix}.$$

The principal submatrix corresponding to the subset $\{1, 3\}$ is $\begin{pmatrix} 0 & 2 \\ 1 & -3 \end{pmatrix}$. The principal submatrix corresponding to the subset $\{2\}$ is 3, the second element in the principal diagonal of F.

Several results useful in studying P(S)D matrices will now be discussed.

Results on P(S)D *Matrices*

Result 1.1 If $B = (b_{11})$ is a matrix of order 1×1, it is PD iff $b_{11} > 0$, and it is PSD iff $b_{11} \geqq 0$.

Proof. Let $y = (y_1) \in \mathbf{R}^1$. Then $y^T B y = b_{11} y_1^{\mathbf{2}}$. So $y^T B y > 0$ for all $y \in \mathbf{R}^1$, $y \neq 0$, iff $b_{11} > 0$, and hence B is PD iff $b_{11} > 0$. Also $y^T B y \geqq 0$ for all $y \in \mathbf{R}^1$, iff $b_{11} \geqq 0$, and hence B is PSD iff $b_{11} \geqq 0$.

□

Result 1.2 If F is a PD matrix all its principal submatrices must also be PD.

Proof. Consider the principal submatrix, G, generated by the subset $\{1, 2\}$.

$$G = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix}. \quad \text{Let } t = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

Pick $y = (y_1, y_2, 0, 0, \dots, 0)^T$. Then $y^T F y = t^T G t$. However, since F is PD, $y^T F y > 0$ for all $y \neq 0$. So $t^T G t > 0$ for all $t \neq 0$. Hence, G is PD too. A similar argument can be used to prove that every principal submatrix of F is also PD.

□

Result 1.3 If F is PD, $f_{ii} > 0$ for all i. This follows as a corollary of Result 1.2.

Result 1.4 If F is a PSD matrix, all principal submatrices of F are also PSD. This is proved using arguments similar to those in Result 1.2.

Result 1.5 If F is PSD matrix, $f_{ii} \geqq 0$ for all i. This follows from Result 1.4.

Result 1.6 Suppose F is a PSD matrix. If $f_{ii} = 0$, then $f_{ij} + f_{ji} = 0$ for all j.

Proof. To be specific let f_{11} be 0 and suppose that $f_{12} + f_{21} \neq 0$. By Result 1.4 the principal submatrix

$$\begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} = \begin{pmatrix} 0 & f_{12} \\ f_{21} & f_{22} \end{pmatrix}$$

must be PSD. Hence $f_{22} y_2^2 + (f_{12} + f_{21}) y_1 y_2 \geqq 0$ for all y_1, y_2. Since $f_{12} + f_{21} \neq 0$, take $y_1 = (-f_{22} - 1)/(f_{12} + f_{21})$ and $y_2 = 1$. The above inequality is violated since the left-hand side becomes equal to -1, leading to a contradiction.

□

Result 1.7 If D is a symmetric PSD matrix and $d_{ii} = 0$, then $D_{.i} = D_{i.} = 0$. This follows from Result 1.6.

Definition: The Gaussian Pivot Step

Let $A = (a_{ij})$ be a matrix of order $m \times n$. A Gaussian pivot step on A, with row r as the pivot row and column s as the pivot column can only be carried out if the element lying in both of them, a_{rs}, is nonzero. This element a_{rs} is known as the pivot element for this pivot step. The pivot step subtracts suitable multiples of the pivot row from each row i for $i > r$ so as to transform the entry in this row and the pivot column into zero. Thus this pivot step transforms

$$A = \begin{pmatrix} a_{11} & \dots & a_{1s} & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{r1} & \dots & a_{rs} & \dots & a_{rn} \\ a_{r+1,1} & \dots & a_{r+1,s} & \dots & a_{r+1,n} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \dots & a_{ms} & \dots & a_{mn} \end{pmatrix}$$

$$\text{into} \quad \begin{pmatrix} a_{11} & \dots & a_{1s} & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{r1} & \dots & a_{rs} & \dots & a_{rn} \\ a'_{r+1,1} & \dots & 0 & \dots & a'_{r+1,n} \\ \vdots & & \vdots & & \vdots \\ a'_{m1} & \dots & 0 & \dots & a'_{mn} \end{pmatrix}$$

where $a'_{ij} = a_{ij} - (a_{rj}a_{is})/a_{rs}$, for $i = r+1$ to m, $j = 1$ to n. As an example consider the Gaussian pivot step in the following matrix with row 2 as the pivot row and column 3 as the pivot column. The pivot element is inside a box.

$$\begin{pmatrix} 1 & -2 & 10 & -4 & -1 \\ 4 & 6 & \boxed{2} & -8 & -4 \\ -3 & 1 & -1 & 2 & 3 \\ 1 & -4 & 2 & 3 & 0 \end{pmatrix}$$

This Gaussian pivot step transforms this matrix into

$$\begin{pmatrix} 1 & -2 & 10 & -4 & -1 \\ 4 & 6 & 2 & -8 & -4 \\ -1 & 4 & 0 & -2 & 1 \\ -3 & -10 & 0 & 11 & 4 \end{pmatrix}$$

Result 1.8 Let D be a square symmetric matrix of order $n \geqq 2$. Suppose D is PD. Subtract suitable multiples of row 1 from each of the other rows so that all the entries in column 1 except the first is transformed into zero. That is, transform

$$D = \begin{bmatrix} d_{11} & \dots & d_{1n} \\ d_{21} & \dots & d_{2n} \\ \vdots & & \vdots \\ d_{n1} & \dots & d_{nn} \end{bmatrix} \quad \text{into} \quad D_1 = \begin{bmatrix} d_{11} & & \dots & d_{1n} \\ 0 & \tilde{d}_{22} & \dots & \tilde{d}_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & \tilde{d}_{n2} & \dots & \tilde{d}_{nn} \end{bmatrix}$$

by a Gaussian pivot step with row 1 as pivot row and column 1 as pivot column, clearly $\tilde{d}_{ij} = d_{ij} - d_{1j}d_{i1}/d_{11}$ for all $i, j \geqq 2$. E_1, the matrix obtained by striking off the first row and the first column from D_1, is also symmetric and PD.

Also, if D is an arbitrary square symmetric matrix, it is PD iff $d_{11} > 0$ and the matrix E_1 obtained as above is PD.

Proof. Since D is symmetric $d_{ij} = d_{ji}$ for all i, j. Therefore,

$$\begin{aligned} y^T D y &= \sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j d_{ij} = d_{11}y_1^2 + 2y_1 \sum_{j=2}^{n} d_{1j}y_j + \sum_{i,j \geqq 2} y_i y_j d_{ij} \\ &= d_{11}\Big(y_1 + \Big(\sum_{j=2}^{n} d_{1j}y_j\Big)/d_{11}\Big)^2 + \sum_{i,j\geqq 2} y_i \tilde{d}_{ij} y_j \,. \end{aligned}$$

Letting $y_1 = -(\sum_{j=2}^{n} d_{1j}y_j)/d_{11}$, we verify that if D is PD, then $\sum_{i,j\geqq 2} y_i \tilde{d}_{ij} y_j > 0$ for all $(y_2, \dots, y_n) \neq 0$, which implies that E_1 is PD. The fact that E_1 is also symmetric is clear since $\tilde{d}_{ij} = d_{ij} - d_{1j}d_{i1}/d_{11} = \tilde{d}_{ji}$ by the symmetry of D. If D is an arbitrary symmetric matrix, the above equation clearly implies that D is PD iff $d_{11} > 0$ and E_1 is PD.

∎

Result 1.9 A square matrix F is PD (or PSD) iff $F + F^T$ is PD (or PSD).

Proof. This follows because $x^T(F + F^T)x = 2x^T Fx$.

□

Result 1.10 Let F be a square matrix of order n and E a matrix of order $m \times n$. The square matrix $A = \begin{pmatrix} F & -E^T \\ E & 0 \end{pmatrix}$ of order $(m + n)$ is PSD iff F is PSD.

Proof. Let $\xi = (y_1, \ldots, y_n, t_1, \ldots, t_m)^T \in \mathbf{R}^{n+m}$ and $y = (y_1, \ldots, y_n)^T$. For all ξ, we have $\xi^T A\xi = y^T Fy$. So $\xi^T A\xi \geqq 0$ for all $\xi \in \mathbf{R}^{n+m}$ iff $y^T Fy \geqq 0$ for all $y \in \mathbf{R}^n$. That is, A is PSD iff F is PSD.

□

Result 1.11 If B is a square nonsingular matrix of order n, $D = B^T B$ is PD and symmetric.

Proof. The symmetry follows because $D^T = D$. For any $y \in \mathbf{R}^n$, $y \neq 0$, $y^T Dy = y^T B^T By = \|yB\|^2 > 0$ since $yB \neq 0$ (because B is nonsingular, $y \neq 0$ implies $yB \neq 0$). So D is PD.

□

Result 1.12 If A is any matrix of order $m \times n$, $A^T A$ is PSD and symmetric.

Proof. Similar to the proof of Result 1.11.

□

Principal Subdeterminants of PD, PSD *Matrices*

We will need the following theorem from elementary calculus.

Theorem 1.1 Intermediate value theorem: *Let $f(\lambda)$ be a continuous real valued function defined on the closed interval $\lambda_0 \leqq \lambda \leqq \lambda_1$ where $\lambda_0 < \lambda_1$. Let $\overline{f}$ be a real number strictly between $f(\lambda_0)$ and $f(\lambda_1)$. Then there exists a $\overline{\lambda}$ satisfying $\lambda_0 < \overline{\lambda} < \lambda_1$, and $f(\overline{\lambda}) = \overline{f}$.*

□

For a proof of Theorem 1.1 see books on calculus, for example, W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, second edition, 1964, p. 81. Theorem 1.1 states that a continuous real valued function defined on a closed interval, assumes all intermediate values between its initial and final values in this interval.

Now we will resume our discussion of PD, PSD matrices.

Theorem 1.2 *If F is a* PD *matrix, whether it is symmetric or not, the determinant of F is strictly positive.*

Proof. Let F be of order n. Let I be the identity matrix of order n. If the determinant of F is zero, F is singular, and hence there exists a nonzero column vector $x \in \mathbf{R}^n$ such that $x^T F = 0$, which implies that $x^T Fx = 0$, a contradiction to the hypothesis that F

is PD. So the determinant of F is nonzero. In a similar manner we conclude that the determinant of any PD-matrix is nonzero. For $0 < \lambda < 1$, define $F(\lambda) = \lambda F + (1-\lambda)I$, and $f(\lambda) =$ determinant of $F(\lambda)$.

Obviously $f(\lambda)$ is a polynomial in λ, and hence $f(\lambda)$ is a real valued continuous function defined on the interval $0 \leqq \lambda \leqq 1$. Given a column vector $x \in \mathbf{R}^n$, $x \neq 0$, $x^T F(\lambda)x = \lambda x^T F x + (1-\lambda)x^T x > 0$ for all $0 \leqq \lambda \leqq 1$ because F is PD. So $F(\lambda)$ is a PD matrix for all $0 \leqq \lambda \leqq 1$. So from the above argument $f(\lambda) \neq 0$ for any λ satisfying $0 \leqq \lambda \leqq 1$. Clearly, $f(0) = 1$, and $f(1) =$ determinant of F. If $f(1) < 0$ by Theorem 1.1 there exists a $\overline{\lambda}$ satisfying $0 < \overline{\lambda} < 1$ and $f(\overline{\lambda}) = 0$, a contradiction. Hence $f(1) \nless 0$. Hence the determinant of F cannot be negative. Also it is nonzero. Hence the determinant of F is strictly positive.

□

Theorem 1.3 *If F is a* PD *matrix, whether it is symmetric or not, all principal subdeterminants of F are strictly positive.*

Proof. This follows from Result 1.2 and Theorem 1.2.

□

Theorem 1.4 *If F is a* PSD *matrix, whether it is symmetric or not, its determinant is nonnegative.*

Proof. For $0 \leqq \lambda \leqq 1$, define $F(\lambda)$, $f(\lambda)$ as in the proof of Theorem 1.2. Since I is PD, and F is PSD; $F(\lambda)$ is a PD matrix for $0 \leqq \lambda < 1$. $f(0) = 1$, and $f(1)$ is the determinant of F. If $f(1) < 0$, there exists a $\overline{\lambda}$ satisfying $0 < \overline{\lambda} < 1$, and $f(\overline{\lambda}) = 0$, a contradiction since $F(\overline{\lambda})$ is a PD matrix. Hence $f(1) \nless 0$. So the determinant of F is nonnegative.

□

Theorem 1.5 *If F is a* PSD *matrix, whether it is symmetric or not, all its principal subdeterminants are nonnegative.*

Proof. Follows from Result 1.4 and Theorem 1.4.

□

Theorem 1.6 *Let*

$$H = \begin{pmatrix} d_{11} & \dots & d_{1n} & d_{1,n+1} \\ \vdots & & \vdots & \vdots \\ d_{n1} & \dots & d_{nn} & d_{n,n+1} \\ d_{n+1,1} & \dots & d_{n+1,n} & d_{n+1,n+1} \end{pmatrix}, \quad D = \begin{pmatrix} d_{11} & \dots & d_{1n} \\ \vdots & & \vdots \\ d_{n1} & \dots & d_{nn} \end{pmatrix}$$

be symmetric matrices. H is of order $n+1$ and D is a principal submatrix of H. So $d_{ij} = d_{ji}$ for all i, $j = 1$ to $n+1$. Let $x \in \mathbf{R}^n$, $d = (d_{1,n+1}, \dots, d_{n,n+1})^T$, and $Q(x) = x^T D x + 2d^T x + d_{n+1,n+1}$. Suppose D is a PD *matrix. Let $x^* = -D^{-1}d$. Then x^* is the point which minimizes $Q(x)$ over $x \in \mathbf{R}^n$, and*

$$Q(x^*) = \text{(determinant of } H\text{) / (determinant of } D\text{)}. \tag{1.12}$$

Also for any $x \in \mathbf{R}^n$

$$Q(x) = Q(x^*) + (x - x^*)^T D(x - x^*). \tag{1.13}$$

Proof. Since H is symmetric $\frac{\partial Q(x)}{\partial x} = 2(Dx + d)$. Hence x^* is the only point in $\mathbf{R}^n$ which satisfies $\frac{\partial Q(x)}{\partial x} = 0$. Also $Dx^* = -d$ implies

$$\begin{aligned} Q(x^*) &= {x^*}^T Dx^* + 2d^T x^* + d_{n+1,n+1} \\ &= d^T x^* + d_{n+1,n+1} \, . \end{aligned} \tag{1.14}$$

For $i = 1$ to $n+1$, if $g_{i,n+1} = d_{i,n+1} + \sum_{j=1}^{n} d_{ij} x_j^*$, and if $g = (g_{1,n+1}, \ldots, g_{n,n+1})^T$, then $g = d + Dx^* = 0$. Also $g_{n+1,n+1} = d_{n+1,n+1} + d^T x^* = Q(x^*)$ from (1.14). Now, from the properties of determinants, it is well known that the value of a determinant is unaltered if a constant multiple of one of its columns is added to another. For $j = 1$ to n, multiply the jth column of H by x_j^* and add the result to column $n+1$ of H. This leads to

$$\begin{aligned} \text{Determinant of } H &= \text{determinant of} \begin{pmatrix} d_{11} & \ldots & d_{1n} & g_{1,n+1} \\ \vdots & & \vdots & \vdots \\ d_{n1} & \ldots & d_{nn} & g_{n,n+1} \\ d_{n+1,1} & \ldots & d_{n+1,n} & g_{n+1,n+1} \end{pmatrix} \\ &= \text{determinant of} \begin{pmatrix} d_{11} & \ldots & d_{1n} & 0 \\ \vdots & & \vdots & \vdots \\ d_{n1} & \ldots & d_{nn} & 0 \\ d_{n+1,1} & \ldots & d_{n+1,n} & Q(x^*) \end{pmatrix} \\ &= (Q(x^*)) \text{ (determinant of } D) \end{aligned}$$

which yields (1.12). (1.13) can be verified by straight forward expansion of its right hand side, or it also follows from Taylor expansion of $Q(x)$ around x^*, since $\frac{\partial^2 Q(x)}{\partial x^2} = 2D$ and x^* satisfies $\frac{\partial Q(x)}{\partial x} = 0$. Since D is a PD matrix, we have $(x - x^*)^T D(x - x^*) > 0$, for all $x \in \mathbf{R}^n$, $x \neq x^*$. This and (1.13) together imply that: $Q(x) > Q(x^*)$, for all $x \in \mathbf{R}^n$, $x \neq x^*$. Hence x^* is the point which minimizes $Q(x)$ over $x \in \mathbf{R}^n$.

□

Theorem 1.7 *Let H, D be square, symmetric matrices defined as in Theorem 1.6. H is* PD *iff D is* PD *and the determinant of H is strictly positive.*

Proof. Suppose H is PD. By Theorem 1.2 the determinant of H is strictly positive, and by Result 1.2 its principal submatrix D is also PD.

Suppose that D is PD and the determinant of H is strictly positive. Let $x = (x_1, \ldots, x_n)^T$ and $\xi = (x_1, \ldots, x_n, x_{n+1})^T$. Define d, $Q(x)$ as in Theorem 1.6. If $x_{n+1} = 0$, but $\xi \neq 0$ (i. e., $x \neq 0$), $\xi^T H \xi = x^T D x > 0$, since D is PD. Now suppose $x_{n+1} \neq 0$. Let $\eta = (1/x_{n+1})x$. Then $\xi^T H \xi = x^T D x + 2x_{n+1} d^T x + d_{n+1,n+1} x_{n+1}^{\mathbf{2}} = x_{n+1}^{\mathbf{2}} Q(\eta)$.

So, when $x_{n+1} \neq 0$, $\xi^T H \xi = x^{\mathbf{2}}_{n+1} Q(\eta) \geqq x^{\mathbf{2}}_{n+1}$ (minimum value of $Q(\eta)$ over $\eta \in \mathbf{R}^n$) $= x^{\mathbf{2}}_{n+1}$ ((determinant of H)/determinant of D)) > 0. So under our hypothesis that D is PD and the determinant of H is strictly positive, we have $\xi^T H \xi > 0$ for all $\xi \in \mathbf{R}^{n+1}$, $\xi \neq 0$, that is H is PD.

□

Theorem 1.8 *Let H be the square symmetric matrix defined in Theorem 1.6. H is* PD *iff the determinants of these $n+1$ principal submatrices of H,*

$$(d_{11}), \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}, \quad \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}, \ldots, D, H$$

are strictly positive.

Proof. Proof is by induction on the order of the matrix. Clearly, the statement of the theorem is true if H is of order 1. Now suppose the statement of the theorem is true for all square symmetric matrices of order n. By this and the hypothesis, we know that the matrix D is PD. So D is PD and the determinant of H is strictly positive by the hypothesis. By Theorem 1.7 these facts imply that H is PD too. Hence, by induction, the statement of the theorem is true in general.

□

Theorem 1.9 *A square symmetric matrix is* PD *iff all its principal subdeterminants are strictly positive.*

Proof. Let the matrix be H defined as in Theorem 1.6. If H is PD, all its principal subdeterminants are strictly positive by Theorem 1.3. On the other hand, if all the principal subdeterminants of H are strictly positive, the $n+1$ principal subdeterminants of H discussed in Theorem 1.8 are strictly positive, and by Theorem 1.8 this implies that H is PD.

□

Definition: P-matrix

A square matrix, whether symmetric or not, is said to be a P-matrix iff all its principal subdeterminants are strictly positive.

As examples, the matrices I, $\begin{pmatrix} 2 & 24 \\ 0 & 2 \end{pmatrix}$, $\begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}$ are P-matrices. The matrices $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} -1 & 0 \\ 0 & 10 \end{pmatrix}$, $\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ are not P-matrices.

Theorem 1.10 *A symmetric P-matrix is always* PD*. If a P-matrix is not symmetric, it may not be* PD.

Proof. By Theorem 1.9 B, a symmetric matrix is PD iff it is a P-matrix. Consider the matrix B,

$$B = \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}, \quad B + B^T = \begin{pmatrix} 2 & 6 \\ 6 & 2 \end{pmatrix}.$$

Since all its principal subdeterminants are 1, B is a P-matrix. However, the determinant of $(B + B^T)$ is strictly negative, and hence it is not a PD matrix by Theorem 1.9, and by Result 1.9 this implies that B is not PD. Actually, it can be verified that, $(1, -1)B(1, -1)^T = -4 < 0$.

□

Note 1.1 The interesting thing to note is that if H is a symmetric matrix, and if the $n+1$ principal subdeterminants of H discussed in Theorem 1.8 are strictly positive, by Theorems 1.10 and 1.8 all principal subdeterminants of H are positive. This result may not be true if H is not symmetric.

Exercises

1.1 If H is a square symmetric PSD matrix, and its determinant is strictly positive, then prove that H is a PD matrix. Construct a numerical example to show that this result is not necessarily true if H is not symmetric.

1.2 Is the following statement true? "H is PSD iff its $(n+1)$ principal subdeterminants discussed in Theorem 1.8 are all nonnegative." Why? Illustrate with a numerical example.

By Theorem 1.9 the class of PD matrices is a subset of the class of P-matrices. By Theorem 1.10 when restricted to symmetric matrices, the property of being a PD matrix is the same as the property of being a P-matrix. An asymmetric P-matrix may not be PD, it may be a PSD matrix as the matrix $\widetilde{M}(n)$ below is, or it may not even be a PSD matrix. Let

$$\widetilde{M}(n) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 2 & 1 & 0 & \dots & 0 & 0 \\ 2 & 2 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 2 & 2 & \dots & 1 & 0 \\ 2 & 2 & 2 & \dots & 2 & 1 \end{pmatrix}. \tag{1.15}$$

$\widetilde{M}(n)$ is a lower triangular matrix in which all the diagonal entries are 1, and all entries below the diagonal are 2. All the principal subdeterminants of $\widetilde{M}(n)$ are clearly equal to 1, and hence $\widetilde{M}(n)$ is a P-matrix. However, $\widetilde{M}(n) + (\widetilde{M}(n))^T$ is the matrix in which all the entries are 2, and it can be verified that it is a PSD matrix and not a PD matrix.

Theorem 1.11 *Let F be a square* PSD *matrix of order n, whether it is symmetric or not. If $\overline{x} \in \mathbf{R}^n$ is such that $\overline{x}^T F \overline{x} = 0$, then $(F + F^T)\overline{x} = 0$.*

Proof. Let $D = F + F^T$. D is symmetric and by Result 1.9, D is PSD. For all $x \in \mathbf{R}^n$, $x^T Dx = 2x^T Fx$. So $\overline{x}^T D\overline{x} = 0$ too. We wish to prove that $D\overline{x} = 0$. Let $x \in \mathbf{R}^n$. For all real numbers λ, $(\overline{x} + \lambda x)^T D(\overline{x} + \lambda x) \geqq 0$, that is

$$\lambda^2 x^T Dx + 2\lambda \overline{x}^T Dx \geqq 0 \tag{1.16}$$

since $\overline{x}^T D\overline{x} = 0$. If $x^T Dx = 0$, by taking $\lambda = 1$ and then -1 in (1.16), we conclude that $\overline{x}^T Dx = 0$. If $x^T Dx \neq 0$, since D is PSD, $x^T Dx > 0$. In this case, from (1.16) we conclude that $2\overline{x}^T Dx \geqq -\lambda x^T Dx$ for $\lambda > 0$, and $2\overline{x}^T Dx \leqq -\lambda x^T Dx$ for $\lambda < 0$. Taking λ to be a real number of very small absolute value, from these we conclude that $\overline{x}^T Dx$ must be equal to zero in this case. Thus whether $x^T Dx = 0$, or $x^T Dx > 0$, we have $\overline{x}^T Dx = 0$. Since this holds for all $x \in \mathbf{R}^n$, we must have $\overline{x}^T D = 0$, that is $D\overline{x} = 0$.

□

Algorithm for Testing Positive Definiteness

Let $F = (f_{ij})$ be a given square matrix of order n. Find $D = F + F^T$. F is PD iff D is. To test whether F is PD, we can compute the n principal subdeterminants of D determined by the subsets $\{1\}, \{1, 2\}, \dots, \{1, 2, \dots, n\}$. F is PD iff each of these n determinants are positive, by Theorem 1.8. However, this is not an efficient method unless n is very small, since the computation of these separate determinants is time consuming.

We now describe a method for testing positive definiteness of F which requires at most n Gaussian pivot steps on D along its main diagonal; hence the computational effort required by this method is $O(n^3)$. This method is based on Result 1.8.

(i) If any of the principal diagonal elements in D are nonpositive, D is not PD. Terminate.

(ii) Subtract suitable multiples of row 1 from all the other rows, so that all the entries in column 1 and rows 2 to n of D are transformed into zero. That is, transform D into D_1 as in Result 1.8. If any diagonal element in the transformed matrix, D_1, is nonpositive, D is not PD. Terminate.

(iii) In general, after r steps we will have a matrix D_r of the form:

$$\begin{bmatrix} d_{11} & d_{12} & & & & \dots & d_{1n} \\ 0 & \tilde{d}_{22} & & & & \dots & \tilde{d}_{1n} \\ & 0 & \ddots & & & & \vdots \\ & & & \overline{d}_{rr} & & \dots & \overline{d}_{rn} \\ & & & 0 & \hat{d}_{r+1,r+1} & \dots & \hat{d}_{r+1,n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & & 0 & \hat{d}_{n,r+1} & \dots & \hat{d}_{nn} \end{bmatrix}.$$

Subtract suitable multiples of row $r+1$ in D_r from rows i for $i > r+1$, so that all the entries in column $r+1$ and rows i for $i > r+1$ are transformed into 0.

This transforms D_r into D_{r+1}. If any element in the principle diagonal of D_{r+1} is nonpositive, D is not PD. Terminate. Otherwise continue the algorithm in the same manner for $n-1$ steps, until D_{n-1} is obtained, which is of the form

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ 0 & \tilde{d}_{22} & \dots & \tilde{d}_{2n} \\ & 0 & & \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \bar{d}_{nn} \end{bmatrix}.$$

D_{n-1} is upper triangular. That's why this algorithm is called the **superdiagonalization algorithm**. If no termination has occured earlier and all the diagonal elements of D_{n-1} are positive, D, and hence, F is PD.

Example 1.3

Test whether

$$F = \begin{bmatrix} 3 & 1 & 2 & 2 \\ -1 & 2 & 0 & 2 \\ 0 & 4 & 4 & \frac{5}{3} \\ 0 & -2 & -\frac{13}{3} & 6 \end{bmatrix} \text{ is PD,} \quad D = F + F^T = \begin{bmatrix} 6 & 0 & 2 & 2 \\ 0 & 4 & 4 & 0 \\ 2 & 4 & 8 & -\frac{8}{3} \\ 2 & 0 & -\frac{8}{3} & 12 \end{bmatrix}.$$

All the entries in the principal diagonal of D (i. e., the entries d_{ii} for all i) are strictly positive. So apply the first step in superdiagonalization getting D_1. Since all elements in the principal diagonal of D_1 are strictly positive, continue. The matrices obtained in the order are:

$$D_1 = \begin{bmatrix} 6 & 0 & 2 & 2 \\ 0 & 4 & 4 & 0 \\ 0 & 4 & \frac{22}{3} & -\frac{10}{3} \\ 0 & 0 & -\frac{10}{3} & \frac{34}{3} \end{bmatrix}, \quad D_2 = \begin{bmatrix} 6 & 0 & 2 & 2 \\ 0 & 4 & 4 & 0 \\ 0 & 0 & \frac{10}{3} & -\frac{10}{3} \\ 0 & 0 & -\frac{10}{3} & \frac{34}{3} \end{bmatrix},$$

$$D_3 = \begin{bmatrix} 6 & 0 & 2 & 2 \\ 0 & 4 & 4 & 0 \\ 0 & 0 & \frac{10}{3} & -\frac{10}{3} \\ 0 & 0 & 0 & 8 \end{bmatrix}.$$

The algorithm terminates now. Since all diagonal entries in D_3 are strictly positive, conclude that D and, hence, F is PD.

Example 1.4

Test whether $D = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 2 & 4 & 4 & 5 \\ 0 & 0 & 5 & 3 \end{pmatrix}$ is PD.

D is already symmetric, and all its diagonal elements are positive. The first step of the algorithm requires performing the operation: (row 3) – 2(row 1) on D. This leads to

$$D_1 = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 4 & 0 & 5 \\ 0 & 0 & 5 & 3 \end{pmatrix} .$$

Since the third diagonal element in D_1 is not strictly positive, D is not PD.

Algorithm for Testing Positive Semidefiniteness

Let $F = (f_{ij})$ be the given square matrix. Obtain $D = F + F^T$. If any diagonal element of D is 0, all the entries in the row and column of the zero diagonal entry must be zero. Otherwise D (and hence F) is not PSD and we terminate. Also, if any diagonal entries in D are negative, D cannot be PSD and we terminate. If termination has not occurred, reduce the matrix D by striking off the rows and columns of zero diagonal entries.

Start off by performing the row operations as in (ii) above, that is, transform D into D_1. If any diagonal element in D_1 is negative, D is not PSD. Let E_1 be the submatrix of D_1 obtained by striking off the first row and column of D_1. Also, if a diagonal element in E_1 is zero, all entries in its row and column in E_1 must be zero. Otherwise D is not PSD. Terminate. Continue if termination does not occur.

In general, after r steps we will have a matrix D_r as in (iii) above. Let E_r be the square submatrix of D_r obtained by striking off the first r rows and columns of D_r. If any diagonal element in E_r is negative, D cannot be PSD. If any diagonal element of E_r is zero, all the entries in its row and column in E_r must be zero; otherwise D is not PSD. Terminate. If termination does not occur, continue.

Let d_{ss} be the first nonzero (and, hence, positive) diagonal element in E_r. Subtract suitable multiples of row s in D_r from rows i, $i > s$, so that all the entries in column s and rows i, $i > s$ in D_r, are transformed into 0. This transforms D_r into D_s and we repeat the same operations with D_s. If termination does not occur until D_{n-1} is obtained and, if the diagonal entries in D_{n-1} are nonnegative, D and hence F are PSD.

In the process of obtaining D_{n-1}, if all the diagonal elements in all the matrices obtained during the algorithm are strictly positive, D and hence F is not only PSD but actually PD.

Example 1.5

Is the matrix

$$F = \begin{bmatrix} 0 & -2 & -3 & -4 & 5 \\ 2 & 3 & 3 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 0 & 0 & 8 & 4 \\ -5 & 0 & 0 & 4 & 2 \end{bmatrix} \text{ PSD? } \quad D = F + F^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 6 & 0 & 0 \\ 0 & 6 & 6 & 0 & 0 \\ 0 & 0 & 0 & 16 & 8 \\ 0 & 0 & 0 & 8 & 4 \end{bmatrix} .$$

$D_{.1}$ and $D_{1.}$ are both zero vectors. So we eliminate them, but we will call the remaining matrix by the same name D. All the diagonal entries in D are nonnegative. Thus we apply the first step in superdiagonalization. This leads to

$$D_1 = \begin{pmatrix} 6 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 8 \\ 0 & 0 & 8 & 4 \end{pmatrix} \qquad E_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 16 & 8 \\ 0 & 8 & 4 \end{pmatrix} .$$

The first diagonal entry in E_1 is 0, but the first column and row of E_1 are both zero vectors. Also all the remaining diagonal entries in D_1 are strictly positive. So continue with superdiagonalization. Since the second diagonal element in D_1 is zero, move to the third diagonal element of D_1. This step leads to

$$D_3 = \begin{pmatrix} 6 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 8 \\ 0 & 0 & 0 & 0 \end{pmatrix} .$$

All the diagonal entries in D_3 are nonnegative. D and hence F is PSD but not PD.

Example 1.6

Is the matrix D in Example 1.4 PSD? Referring to Example 1.4 after the first step in superdiagonalization, we have

$$E_1 = \begin{pmatrix} 2 & 4 & 0 \\ 4 & 0 & 5 \\ 0 & 5 & 3 \end{pmatrix} .$$

The second diagonal entry in E_1 is 0, but the second row and column of E_1 are not zero vectors. So D is not PSD.

1.3.2 Relationship of Positive Semidefiniteness to the Convexity of Quadratic Functions

Let $\mathbf{\Gamma}$ be a convex subset of $\mathbf{R}^n$, and let $g(x)$ be a real valued function defined on $\mathbf{\Gamma}$. $g(x)$ is said to be a **convex function** on $\mathbf{\Gamma}$, if

$$g(\alpha x^1 + (1-\alpha)x^2) \leqq \alpha g(x^1) + (1-\alpha)g(x^2) \tag{1.17}$$

for every pair of points x^1, x^2 in $\mathbf{\Gamma}$, and for all $0 \leqq \alpha \leqq 1$. $g(x)$ is said to be a **strictly convex function** on $\mathbf{\Gamma}$ if (1.17) holds as a strict inequality for every pair of distinct points x^1, x^2 in $\mathbf{\Gamma}$ (i. e., $x^1 \neq x^2$) and for all $0 < \alpha < 1$. See Appendix 3.

Let F be a given square matrix of order n and c a row vector in $\mathbf{R}^n$. Let $f(x) = cx + x^T F x$. Here we discuss conditions under which $f(x)$ is convex, or strictly convex. Let $D = (1/2)(F + F^T)$. If F is symmetric then $F = D$, otherwise D is the symmetrized form of F. Clearly $f(x) = cx + x^T D x$. It can be verified that $\frac{\partial f(x)}{\partial x} = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^T = c^T + (F + F^T)x = c^T + 2Dx$, and that $\frac{\partial^2 f(x)}{\partial x^2}$ = the Hessian of $f(x) = F + F^T = 2D$. Let x^1, x^2 be two arbitrary column vectors in $\mathbf{R}^n$ and let $\xi = x^1 - x^2$. Let α be a number between 0 and 1. By expanding both sides it can be verified that $\alpha f(x^1) + (1-\alpha) f(x^2) - f(\alpha x^1 + (1-\alpha)x^2) = \alpha(1-\alpha)\xi^T D \xi$ where $\xi = x^1 - x^2$. So $\alpha f(x^1) + (1-\alpha) f(x^2) - f(\alpha x^1 + (1-\alpha)x^2) \geqq 0$ for all $x^1, x^2 \in \mathbf{R}^n$ and $0 \leqq \alpha \leqq 1$, iff $\xi^T D \xi \geqq 0$ for all $\xi \in \mathbf{R}^n$, that is iff D (or equivalently F) is PSD. Hence $f(x)$ is convex on $\mathbf{R}^n$ iff F (or equivalently D) is PSD.

Also by the above argument we see that $\alpha f(x^1) + (1-\alpha) f(x^2) - f(\alpha x^1 + (1-\alpha)x^2) > 0$ for all $x^1 \neq x^2$ in $\mathbf{R}^n$ and $0 < \alpha < 1$, iff $\xi^T D \xi > 0$ for all $\xi \in \mathbf{R}^n$, $\xi \neq 0$. Hence $f(x)$ is strictly convex on $\mathbf{R}^n$ iff $\xi^T D \xi > 0$ for all $\xi \neq 0$, that is iff D (or equivalently F) is PD. These are the conditions for the convexity or strict convexity of the quadratic function $f(x)$ **over the whole space $\mathbf{R}^n$**. It is possible for $f(x)$ to be convex on a lower dimensional convex subset of $\mathbf{R}^n$ (for example, a subspace of $\mathbf{R}^n$) even though the matrix F is not PSD. For example, the quadratic form $f(x) = (x_1, x_2) \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} (x_1, x_2)^T$ is convex over the subspace $\{(x_1, x_2) : x_1 = 0\}$ but not over the whole of $\mathbf{R}^2$.

Exercise

1.3 Let $\mathbf{K} \subset \mathbf{R}^n$ be a convex set and $Q(x) = cx + \frac{1}{2} x^T D x$. If $Q(x)$ is convex over $\mathbf{K}$ and $\mathbf{K}$ has a nonempty interior, prove that $Q(x)$ is convex over the whole space $\mathbf{R}^n$.

1.3.3 Necessary Optimality Conditions for Quadratic Programming

We will now resume our discussion of the quadratic program (1.11).

Theorem 1.12 *If $\overline{x}$ is an optimum solution of (1.11), $\overline{x}$ is also an optimum solution of the LP*

$$\begin{array}{ll} \text{minimize} & (c + \overline{x}^T D)x \\ \text{subject to} & Ax \geqq b \\ & x \geqq 0\,. \end{array} \tag{1.18}$$

Proof. Notice that the vector of decision variables in (1.18) is x; $\overline{x}$ is a given point and the cost coefficients in the LP (1.18) depend on $\overline{x}$. The constraints in both (1.11) and (1.18) are the same. The set of feasible solutions is a convex polyhedron. Let $\hat{x}$ be any feasible solution. By convexity of the set of feasible solutions $x_\lambda = \lambda\hat{x}+(1-\lambda)\overline{x} = \overline{x} + \lambda(\hat{x} - \overline{x})$ is also a feasible solution for any $0 < \lambda < 1$. Since $\overline{x}$ is an optimum feasible solution of (1.11), $Q(x_\lambda) - Q(\overline{x}) \geqq 0$, that is $\lambda(c + \overline{x}^T D)(\hat{x} - \overline{x}) + (1/2)$ $\lambda^2(\hat{x}-\overline{x})^T D(\hat{x}-\overline{x}) \geqq 0$ for all $0 < \lambda < 1$. Dividing both sides by λ leads to $(c+\overline{x}^T D)$ $(\hat{x}-\overline{x}) \geqq (-\lambda/2)(\hat{x}-\overline{x})^T D(\hat{x}-\overline{x})$ for all $0 < \lambda < 1$. This obviously implies $(c+\overline{x}^T D)$ $(\hat{x}-\overline{x}) \geqq 0$, that is, $(c+\overline{x}^T D)\hat{x} \geqq (c+\overline{x}^T D)\overline{x}$. Since this must hold for an arbitrary feasible solution $\hat{x}$, $\overline{x}$ must be an optimum feasible solution of (1.18).

□

Corollary 1.1 *If $\overline{x}$ is an optimum feasible solution of (1.11), there exist vectors $\overline{y} \in \mathbf{R}^m$ and slack vectors $\overline{u} \in \mathbf{R}^N$, $\overline{v} \in \mathbf{R}^m$ such that $\overline{x}$, $\overline{y}$, $\overline{u}$, $\overline{v}$ together satisfy*

$$\begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix} - \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} c^T \\ -b \end{pmatrix}$$
$$\begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix} \geqq 0 \quad \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} \geqq 0 \quad \textit{and} \quad \begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix}^T \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = 0 \,. \tag{1.19}$$

Proof. ¿From the above theorem $\overline{x}$ must be an optimum solution of the LP (1.18). The corollary follows by using the results of Section 1.2 on this fact.

□

Necessary and Sufficient Optimality Conditions for Convex Quadratic Programs

The quadratic minimization problem (1.11) is said to be a **convex quadratic program** if $Q(x)$ is convex, that is, if D is a PSD matrix (by the results in Section 1.3.2, or Theorem 17 of Appendix 3). If D is not PSD, (1.11) is said to be a **nonconvex quadratic program**. Associate a Lagrange multiplier y_i to the ith constraint "$A_{i.}x \geqq b_i$" $i = 1$ to m; and a Lagrange multiplier u_j to the sign restriction on x_j in (1.11), $j = 1$ to N. Let $y = (y_1, \ldots, y_m)^T$, $u = (u_1, \ldots, u_N)^T$. Then the **Lagrangian** corresponding to the quadratic program (1.11) is $L(x, y, u) = Q(x) - y^T(Ax-b) - u^T x$. The Karush-Kuhn-Tucker necessary optimality conditions for (1.11) are

$$\begin{gathered} \frac{\partial L}{\partial x}(x, y, u) = c^T + Dx - A^T y - u = 0 \\ y \geqq 0, \; u \geqq 0 \\ y^T(Ax - b) = 0, \; u^T x = 0 \\ Ax - b \geqq 0, \; x \geqq 0 \,. \end{gathered} \tag{1.20}$$

Denoting the slack variables $Ax - b$ by v, the conditions (1.20) can be verified to be exactly those in (1.19), written out in the form of an LCP. A feasible solution x

for (1.11), is said to be a **Karush-Kuhn-Tucker point** (or abbreviated as a **KKT point**) if there exist Lagrange multiplier vectors y, u, such that x, y, u together satisfy (1.20) or the equivalent (1.19). So the LCP (1.19) is the problem of finding a KKT point for (1.11). We now have the following results.

Theorem 1.13 *If $\overline{x}$ is an optimum solution for (1.11), $\overline{x}$ must be a KKT point for it, whether $Q(x)$ is convex or not.*

Proof. Follows from Theorem 1.12 and Corollary 1.1.

□

Thus (1.20) or equivalently (1.19) provide the necessary optimality conditions for a feasible solution x of (1.11) to be optimal. Or, in other words, every optimum solution for (1.11) must be a KKT point for it. However, given a KKT point for (1.11) we cannot guarantee that it is optimal to (1.11) in general. In the special case when D is PSD, every KKT point for (1.11) is optimal to (1.11), this is proved in Theorem 1.14 below. Thus for convex quadratic programs, (1.20) or equivalently (1.19) provide necessary and sufficient optimality conditions.

Theorem 1.14 *If D is* PSD *and $\overline{x}$ is a KKT point of (1.11), $\overline{x}$ is an optimum feasible solution of (1.11).*

Proof. ¿From the definition of a KKT point and the results in Section 1.2, if $\overline{x}$ is a KKT point for (1.11), it must be an optimum feasible solution of the LP (1.18). Let x be any feasible solution of (1.11).

$$Q(x) - Q(\overline{x}) = (c + \overline{x}^T D)(x - \overline{x}) + \frac{1}{2}(x - \overline{x})^T D(x - \overline{x}) .$$

The first term on the right-hand side expression is nonnegative since $\overline{x}$ is an optimal feasible solution of (1.18). The second term in that expression is also nonnegative since D is PSD. Hence, $Q(x) - Q(\overline{x}) \geqq 0$ for all feasible solutions, x, of (1.11). This implies that $\overline{x}$ is an optimum feasible solution of (1.11).

□

Clearly (1.19) is an LCP. An optimum solution of (1.11) must be a KKT point for it. Solving (1.19) provides a KKT point for (1.11) and if D is PSD, this KKT point is an optimum solution of (1.11). [If D is not PSD and if a KKT point is obtained when (1.19) is solved, it may not be an optimum solution of (1.11).]

Example 1.7 Minimum Distance Problem.

Let $\mathbf{K}$ denote the shaded convex polyhedral region in Figure 1.7. Let P_0 be the point $(-2, -1)$. Find the point in $\mathbf{K}$ that is closest to P_0 (in terms of the usual Euclidean distance). Such problems appear very often in operations research applications.

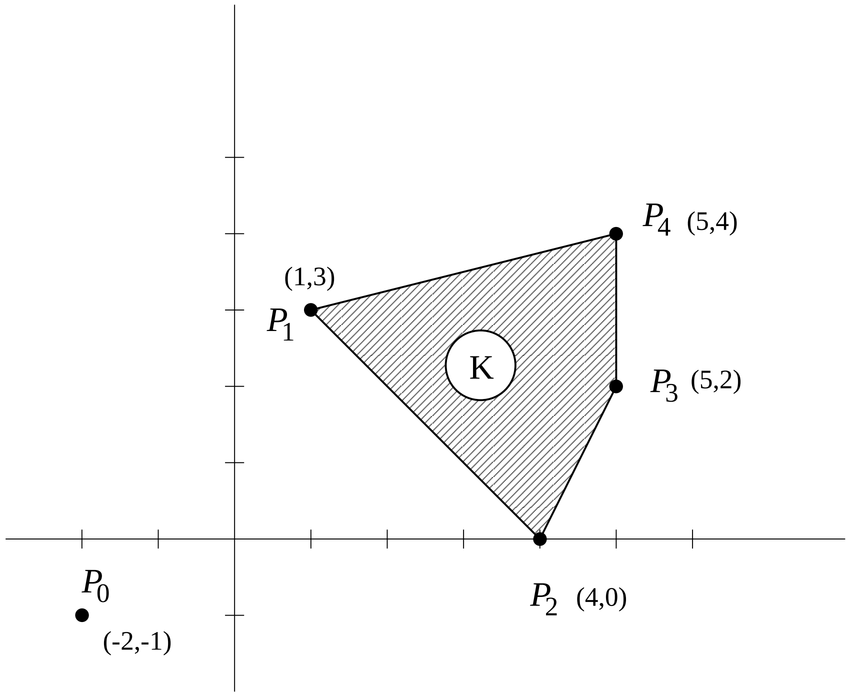


Figure 1.7

Every point in **K** can be expressed as a convex combination of its **extreme points** (or **corner points**) P_1, P_2, P_3, P_4. That is, the coordinates of a general point in **K** are: $(\lambda_1+4\lambda_2+5\lambda_3+5\lambda_4, 3\lambda_1+0\lambda_2+2\lambda_3+4\lambda_4)$ where the λ_i satisfy $\lambda_1+\lambda_2+\lambda_3+\lambda_4=1$ and $\lambda_i \geqq 0$ for all i. Hence, the problem of finding the point in **K** closest to P_0 is equivalent to solving:

$$\begin{array}{ll} \text{Minimize} & (\lambda_1+4\lambda_2+5\lambda_3+5\lambda_4-(-2))^{\mathbf{2}} + (3\lambda_1+2\lambda_3+4\lambda_4-(-1))^{\mathbf{2}} \\ \text{Subject to} & \lambda_1+\lambda_2+\lambda_3+\lambda_4 = 1 \\ & \lambda_i \geqq 0 \quad \text{for all } i\ . \end{array}$$

λ_4 can be eliminated from this problem by substituting the expression $\lambda_4 = 1-\lambda_1-\lambda_2-\lambda_3$ for it. Doing this and simplifying, leads to the quadratic program

$$\begin{array}{ll} \text{Minimize} & (-66,-54,-20)\lambda + \left(\frac{1}{2}\right)\lambda^T \begin{pmatrix} 34 & 16 & 4 \\ 16 & 34 & 16 \\ 4 & 16 & 8 \end{pmatrix} \lambda \\ \text{Subject to} & -\lambda_1-\lambda_2-\lambda_3 \geqq -1 \\ & \lambda \geqq \ \ 0 \end{array}$$

where $\lambda = (\lambda_1,\lambda_2,\lambda_3)^T$. Solving this quadratic program is equivalent to solving the LCP

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \end{pmatrix} - \begin{pmatrix} 34 & 16 & 4 & 1 \\ 16 & 34 & 16 & 1 \\ 4 & 16 & 8 & 1 \\ -1 & -1 & -1 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ y_1 \end{pmatrix} = \begin{pmatrix} -66 \\ -54 \\ -20 \\ 1 \end{pmatrix} .$$

$$\text{All variables } u_1, u_2, u_3, v_1, \lambda_1, \lambda_2, \lambda_3, y_1 \geqq 0$$

$$\text{and } u_1\lambda_1 = u_2\lambda_2 = u_3\lambda_3 = v_1 y_1 = 0\,.$$

Let $(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{v}_1, \tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3, \tilde{y}_1)$ be a solution to this LCP. Let $\tilde{\lambda}_4 = 1 - \tilde{\lambda}_1 - \tilde{\lambda}_2 - \tilde{\lambda}_3$. Then $\tilde{x} = (\tilde{\lambda}_1 + 4\tilde{\lambda}_2 + 5\tilde{\lambda}_3 + 5\tilde{\lambda}_4, 3\tilde{\lambda}_1 + 2\tilde{\lambda}_3 + 4\tilde{\lambda}_4)$ is the point in K that is closest to P_0.

1.3.4 Convex Quadratic Programs and LCPs Associated with PSD Matrices

Consider the LCP (q, M), which is (1.6) – (1.8), in which the matrix M is PSD. Consider also the quadratic program

$$\begin{array}{ll} \text{Minimize} & z^T(Mz + q) \\ \text{Subject to} & Mz + q \geqq 0 \\ & z \geqq 0\,. \end{array}$$

This is a convex quadratic programming problem since M is PSD. If the optimum objective value in this quadratic program is > 0, clearly the LCP (q, M) has no solution. If the optimum objective value in this quadratic program is zero, and $\overline{z}$ is any optimum solution for it, then $(\overline{w} = M\overline{z} + q, \overline{z})$ is a solution of the LCP. Conversely if $(\tilde{w}, \tilde{z})$ is any solution of the LCP (q, M), the optimum objective value in the above quadratic program must be zero, and $\tilde{z}$ is an optimum solution for it. Thus every LCP associated with a PSD matrix can be posed as a convex quadratic program.

Now, consider a convex quadratic program in which $Q(x) = cx + \frac{1}{2}x^T Dx$ (where D is a symmetric PSD matrix) has to be minimized subject to linear constraints. Replace each equality constraint by a pair of opposing inequality constraints (for example, $Ax = b$ is replaced by $Ax \leqq b$ and $Ax \geqq b$). Now the problem is one of minimizing $Q(x)$ subject to a system of linear inequality constraints. This can be transformed into an LCP as discussed in Section 1.3.3. The matrix M in the corresponding LCP will be PSD by Result 1.10, since D is PSD. Thus every convex quadratic programming problem can be posed as an LCP associated with a PSD matrix.

1.3.5 Applications of Quadratic Programming

The Portfolio Problem

A big investment firm has a total of \$ a to invest. It has a list of n stocks in which this money can be invested. The problem is to determine how much of the available money should be invested in each stock. The solution of this problem is called a **portfolio**. In this problem, it is well known that "one should never put all of their eggs in one basket". So after a thorough study, the manager of the company has determined a lower bound \$ l_j and an upper bound \$ k_j for the amount to be invested in stock j, $j = 1$ to n. The yield from each stock varies randomly from year to year. By the analysis of past data, μ_j, the expected (or average) yield per dollar invested in stock j per year has been estimated. The yields from various stocks are not mutually independent, and the analysis of past data has also provided an estimate of the variance-covariance matrix, D, for the annual yields from the various stocks per dollar invested. D is a symmetric positive definite matrix of order n. If \$ x_j is the amount invested in stock j, $j = 1$ to n, the portfolio is $x = (x_1, \ldots, x_n)^T$, the expected annual yield from it is $\sum_{j=1}^{n} \mu_j x_j$ and the variance of the yield is $x^T D x$. The variance is a measure of the random fluctuation in the annual yield and hence it should be minimized. The company would, of course, like to see its expected yield maximized. One way of achieving both of these objectives is to specify a target or lower bound, say μ, on the expected yield and to minimize the variance subject to this constraint. This leads to the problem:

$$\begin{array}{ll} \text{Minimize} & x^T D x \\ \text{Subject to} & \sum_{j=1}^{n} \mu_j x_j \geqq \mu \\ & \sum x_j \leqq a \\ & l_j \leqq x_j \leqq k_j \ , \ j = 1 \text{ to } n \end{array}$$

which is a quadratic programming problem.

Constrained Linear Regression

We will illustrate this application with an example of eggs and chickens due to C. Marmolinero [1.22]. The first step in chicken farming is hatching, carried out by specialized hatcheries. When hatched, a **day-old-chicken** is born. It needs no food for the first two days, at the end of which it is called a **growing pullet** and moved out of the hatchery. Pullets have to grow over a period of approximately 19 weeks before they start producing eggs, and this is done by specialized growing units under optimum conditions of diet, heating, lighting etc. After 19 weeks of age, pullets are moved into the laying flock and are then called **hens**. Consider a geographical region, say a State. Data on the number of chickens hatched by hatcheries in the state during each month is available from published state government statistics. But, day-old-chickens may be bought from, or sold to firms outside the state, statistics on which are not available. Define

y_t = number (in millions) of growing pullets in the state, on the first day of month t.

d_t = number (in millions) of day-old-chickens hatched by hatcheries in the state in month t (from government statistics).

Here d_t are not variables, but are the given data. People in the business of producing chicken feed are very much interested in getting estimates of y_t from d_t. This provides useful information to them in their production planning, etc. Not all the day-old-chickens placed by hatcheries in a month may be alive in a future month. Also, after five months of age, they are recorded as hens and do not form part of the population of growing pullets. So the appropriate linear regression model for y_t as a function of the d_t's seems to be $y_t = \beta_0 + \sum_{i=1}^{5} \beta_i d_{t-i}$, where β_0 is the number of pullets in census, which are not registered as being hatched (pullets imported into the State, or chickens exported from the State), and β_i is a survival rate (the proportion of chickens placed in month $t-i$ that are alive in month t, $i = 1$ to 5). We, of course, expect the parameters β_i to satisfy the constraints

$$0 \leqq \beta_5 \leqq \beta_4 \leqq \beta_3 \leqq \beta_2 \leqq \beta_1 \leqq 1\,. \tag{1.21}$$

To get the best estimates for the parameters $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$ from past data, the least squares method could be used. Given data on y_t, d_t over a period of time (say for the last 10 years), define $L_2(\beta) = \sum_t (y_t - \beta_0 - \sum_{i=1}^{5} \beta_i d_{t-i})^{\mathbf{2}}$. Under the least squares method the best values for β are taken to be those that minimize $L_2(\beta)$ subject to the constraints (1.21). This is clearly a quadratic programming problem.

One may be tempted to simplify this problem by ignoring the constraints (1.21) on the parameters β. The unconstrained minimum of $L_2(\beta)$ can be found very easily by solving the system of equations $\frac{\partial L_2(\beta)}{\partial \beta} = 0$.

There are two main difficulties with this approach. The first is that the solution of this system of equations requires the handling of a square matrix (a_{ij}) with $a_{ij} = 1/(i+j-1)$, known as the **Hilbert matrix**, which is difficult to use in actual computation because of ill-conditioning. It magnifies the uncertainty in the data by very large factors. We will illustrate this using the Hilbert matrix of order 2. This matrix is

$$H_2 = \begin{pmatrix} 1 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix}.$$

Consider the following system of linear equations with H_2 as the coefficient matrix.

x_1	x_2	
1	$\frac{1}{2}$	b_1
$\frac{1}{2}$	$\frac{1}{3}$	b_2

It can be verified that the solution of this system of linear equations is $\overline{x} = (4b_1 - 6b_2, -6b_1 + 12b_2)^T$. Suppose we have the exact value for b_1 but only an approximate value for b_2. In the solution $\overline{x}$, errors in b_2 are magnified by 12 times in $\overline{x}_2$, and 6 times in $\overline{x}_1$. This is only in a small system involving the Hilbert matrix of order 2. The error magnification grows very rapidly in systems of linear equations involving Hilbert matrices of higher orders. In real world applications, the coefficients in the system of linear equations (constants corresponding to b_1, b_2 in the above system) are constructed using observed data, which are always likely to have small errors. These errors are magnified in the solution obtained by solving the system of equations, making that solution very unreliable. See reference [1.36]. The second difficulty is that even if we are able to obtain a reasonable accurate solution $\hat{\beta}$ for the system of equations $\frac{\partial L_2(\beta)}{\partial \beta} = 0$, $\hat{\beta}$ may violate the constraints (1.21) that the parameter vector β is required to satisfy. For example, when this approach was applied on our problem with actual data over a 10-year horizon from a State, it led to the estimated parameter vector $\hat{\beta} = (4, .22, 1.24, .70 - .13, .80)^T$. We have $\hat{\beta}_4 < 0$ and $\hat{\beta}_2 > 1$, these values are not admissible for survival rates. So $\beta = \hat{\beta}$ does not make any sense in the problem. For the same problem, when $L_2(\beta)$ was minimized subject to the constraints (1.21), using a quadratic programming algorithm it gave an estimate for the parameter vector which was quite good.

Parameter estimation in linear regression using the least squares method is a very common problem in many statistical applications, and in almost all branches of scientific research. In a large proportion of these applications, the parameter values are known to satisfy one or more constraints (which are usually linear). The parameter estimation problem in constrained linear regression is a quadratic programming problem when the constraints on the parameters are linear.

1.3.6 Application of Quadratic Programming in Algorithms for NLP, Recursive Quadratic Programming Methods for NLP

Recently, algorithms for solving general nonlinear programs, through the solution of a series of quadratic subproblems have been developed [1.41 to 1.54]. These methods are called **recursive quadratic programming methods**, or **sequential quadratic programming methods**, or **successive quadratic programming methods** in the literature. Computational tests have shown that these methods are especially efficient in terms of the number of function and gradient evaluations required. Implementation of these methods requires efficient algorithms for quadratic programming. We provide here a brief description of this approach for nonlinear programming. Consider the nonlinear program:

$$\begin{array}{ll} \text{Minimize} & \theta(x) \\ \text{Subject to} & g_i(x) = 0, \ i = 1 \text{ to } k \\ & g_i(x) \geqq 0, \ i = k+1 \text{ to } m \end{array} \tag{1.22}$$

where $\theta(x)$ and $g_i(x)$ are real valued twice continuously differentiable functions defined over $\mathbf{R}^n$. Let $g(x) = (g_1(x), \dots, g_m(x))^T$. Given the Lagrange multiplier vector $\pi = (\pi_1, \dots, \pi_k, \pi_{k+1}, \dots, \pi_m)$, the Lagrangian corresponding to (1.22) is $L(x, \pi) = \theta(x) - \pi g(x)$. The first order (or Karush-Kuhn-Tucker) necessary optimality conditions for this problem are

$$\begin{array}{rl} \nabla_x L(x, \pi) = \nabla\theta(x) - \pi\nabla g(x) = 0 & \\ \pi_i \geqq 0 & i = k+1 \text{ to } m \\ \pi_i g_i(x) = 0 & i = k+1 \text{ to } m \\ g_i(x) = 0 & i = 1 \text{ to } k \\ g_i(x) \geqq 0 & i = k+1 \text{ to } m\,. \end{array} \tag{1.23}$$

The methods described here for tackling (1.22) try to obtain a solution $\overline{x}$ and a Lagrange multiplier vector $\overline{\pi}$, which together satisfy (1.23), through an iterative process. In each iteration, a quadratic programming problem is solved, the solution of which provides revised estimates of the Lagrange multipliers and also determines a search direction for a **merit function**. The merit function is an absolute value penalty function (L_1-penalty function) that balances the two competing goals of decreasing $\theta(x)$ and reducing the amounts by which the constraints are violated. The merit function is then minimized in the descent direction by using a line minimization procedure. The solution of this line minimization problem produces a revised point x. With the revised x and π, the method goes to the next iteration. The first iteration begins with an initial point x and Lagrange multiplier vector π satisfying $\pi_i \geqq 0$, $i = k+1$ to m.

At the beginning of an iteration, let $\hat{x}$, $\hat{\pi}$ be the current vectors. Define

$$Q(d) = L(\hat{x}, \hat{\pi}) + \big(\nabla_x L(\hat{x}, \hat{\pi})\big) d + \frac{1}{2} d^T \frac{\partial^2 L(\hat{x}, \hat{\pi})}{\partial x^2} d \tag{1.24}$$

where $d = x - \hat{x}$. $Q(d)$ is the Taylor series approximation for $L(x, \hat{\pi})$ around the current point $\hat{x}$ up to the second order. Clearly $\frac{\partial^2 L(\hat{x}, \hat{\pi})}{\partial x^2}$ changes in each iteration. Since this is an $n \times n$ matrix, recomputing it in each iteration can be very expensive computationally. So in computer implementations of this method, $\frac{\partial^2 L(\hat{x}, \hat{\pi})}{\partial x^2}$ is approximated by a matrix B which is revised from iteration to iteration using the BFGS Quasi-Newton update formula that is widely used for unconstrained minimization. In the initial step, approximate $\frac{\partial^2 L}{\partial x^2}$ by $B_0 = I$, the unit matrix of order n. Let x^t, π^t, B_t, denote the initial point, the initial Lagrange multiplier vector, and the approximation for $\frac{\partial^2 L}{\partial x^2}$ in the t-th iteration. Let x^{t+1} be the point and π^{t+1} the Lagrange multiplier vector at the end of this iteration. Define

$$\begin{aligned} \xi^{t+1} &= x^{t+1} - x^t \\ q^{t+1} &= \big(\nabla_x L(x^{t+1}, \pi^{t+1}) - \nabla_x L(x^t, \pi^{t+1})\big)^T \\ p^{t+1} &= r_{t+1} q^{t+1} + (1 - r_{t+1}) B_t \xi^{t+1} \end{aligned}$$

where

$$r_{t+1} \begin{cases} = 1 & \text{if } (\xi^{t+1})^T q^{t+1} \geqq (0.2)(\xi^{t+1})^T B_t \xi^{t+1} \\ = \dfrac{(0.8)((\xi^{t+1})^T B_t \xi^{t+1})}{(\xi^{t+1})^T B_t \xi^{t+1} - (\xi^{t+1})^T q^{t+1}}, & \text{if } (\xi^{t+1})^T q^{t+1} < (0.2)(\xi^{t+1})^T B_t \xi^{t+1}. \end{cases}$$

Then update $\frac{\partial^2 L}{\partial x^2}$ by the formula

$$B_{t+1} = B_t + \frac{p^{t+1}(p^{t+1})^T}{(\xi^{t+1})^T p^{t+1}} - \frac{(B_t \xi^{t+1})(B_t \xi^{t+1})^T}{(\xi^{t+1})^T B_t \xi^{t+1}}. \tag{1.25}$$

This updating formula is a slight modification of the BFGS (Broyden-Fletcher-Goldfarb-Shanno) formula for updating the Hessian (the BFGS updating formula discussed in Section 10.8.6 is for updating the inverse of the Hessian, the one given here is for updating the actual Hessian itself).

If $r_{t+1} = 1$, then $p^{t+1} = q^{t+1}$ and the updating formula reduces to the standard BFGS formula for the approximation of the Hessian. The definition of p^{t+1} using r_{t+1} is introduced to assure that $(\xi^{t+1})^T p^{t+1} > 0$, which guarantees the hereditary positive definiteness of the updates B_t. The quantities 0.2, 0.8 are choosen from numerical experiments, they can be changed. This updating formula provides a symmetric positive definite approximation for $\frac{\partial^2 L}{\partial x^2}$. Also, in actual implementation, the second term in $Q(d)$ in (1.24) is replaced by $(\nabla\theta(\hat{x}))d$.
Therefore, the quadratic program solved in this iteration is: find d that

$$\begin{array}{ll} \text{minimizes} & (\nabla\theta(\hat{x}))d + (1/2)d^T \hat{B} d \\ \text{subject to} & g_i(\hat{x}) + (\nabla g_i(\hat{x}))d \begin{cases} = 0, & i = 1 \text{ to } k \\ \geqq 0, & i = k+1 \text{ to } m \end{cases} \end{array} \tag{1.26}$$

where $\hat{B}$ is the current approximation for $\frac{\partial^2 L}{\partial x^2}$.

Let $\tilde{d}$ denote the optimum solution of the quadratic program (1.26), and let $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_m)$ denote the associated Lagrange multiplier vector corresponding to the constraints in (1.26). If $\tilde{d} = 0$, from the optimality conditions for the quadratic program (1.26), it can be verified that $(\hat{x}, \tilde{\pi})$ together satisfy (1.23) and we terminate. If $\tilde{d} \neq 0$, it will be a descent direction for the merit function at $\hat{x}$. In the quadratic programm (1.26), to make sure that the Taylor series approximations remain reasonable, one could add additional bound conditions of the form $-\delta_j \leqq d_j \leqq \delta_j$, $j = 1$ to n, where δ_j are suitably choosen small positive numbers.

The form of the function that is minimized in the line search in this iteration is the merit function which is a L_1-penalty function

$$S(x) = \theta(x) + \sum_{i=1}^{k} \hat{\mu}_i |g_i(x)| + \sum_{i=k+1}^{m} \hat{\mu}_i |\text{minimum } \{0, g_i(x)\}| \tag{1.27}$$

where the last two terms are weighted sums of the absolute constraint violations. The weights $\hat{\mu}_i$ used in (1.27) satisfy $\mu_i > |\tilde{\pi}_i|$, they are usually obtained from

$$\hat{\mu}_i = \text{ maximum } \{|\tilde{\pi}_i|, (1/2)(\overline{\mu}_i + |\tilde{\pi}_i|)\}, \ i = 1 \text{ to } m,$$

where $\overline{\mu}_i$ are the weights used in the previous iteration. In Theorem 1.15 given below we prove that if $\tilde{d} \neq 0$, it is a descent direction at the current point $\hat{x}$, for the specially choosen merit functions $S(x)$ defined in (1.27) (this means that for $\alpha > 0$ and small $S(\hat{x}+\alpha\tilde{d}) < S(\hat{x})$, i. e., that $S(x)$ strictly decreases as we move from $\hat{x}$ in the direction $\tilde{d}$). The merit function $S(x)$ is minimized on the half-line $\{x : x = \hat{x} + \alpha\tilde{d},\ \alpha \geqq 0\}$. For this we define $f(\alpha) = S(\hat{x} + \alpha\tilde{d})$ and minimize $f(\alpha)$ over $\alpha \geqq 0$ by using some one dimensional line search algorithm (see Chapter 10). If $\tilde{\alpha}$ is the value of α that minimizes $f(\alpha)$ over $\alpha \geqq 0$, let $\tilde{x} = \hat{x} + \tilde{\alpha}\tilde{d}$. The point $\tilde{x}$ is the new point, it is obtained by moving a step length of $\tilde{\alpha}$ from $\hat{x}$ in the direction $\tilde{d}$.

If $\tilde{x}$, $\tilde{\pi}$ satisfy (1.23) to a reasonalbe degree of approximation, the method terminates, otherwise it moves to the next iteration with them.

The Descent Property

Theorem 1.15 *Suppose $\hat{B}$ is symmetric and* PD. *Let $\tilde{d}$, $\tilde{\pi}$ be the optimum solution and the associated Lagrange multiplier vector for the quadratic program (1.26). If $\tilde{d} \neq 0$, it is a descent direction for the merit function $S(x)$ at $\hat{x}$.*

Proof. By the first order necessary optimality conditions for the quadratic program (1.26) we have

$$\begin{aligned} \nabla\theta(\hat{x}) + (\hat{B}\tilde{d})^T - \tilde{\pi}\nabla g(\hat{x}) &= 0 \\ \tilde{\pi}_i(g_i(\hat{x}) + (\nabla g_i(\hat{x}))\tilde{d}) &= 0\,,\ i = 1 \text{ to } m\,. \end{aligned} \tag{1.28}$$

So, for α positive and sufficiently small, since all the functions are continuously differentiable, we have

$$\begin{aligned} f(\alpha) = \quad & S(\hat{x} + \alpha\tilde{d}) = \theta(\hat{x}) + \alpha(\nabla\theta(\hat{x}))\tilde{d} + \\ & \sum_{i=1}^{k} \hat{\mu}_i |g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d}| \\ & - \sum_{i=k+1}^{m} \hat{\mu}_i(\min\{0, g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d}\}) + o(\alpha) \end{aligned} \tag{1.29}$$

where $o(\alpha)$ is a function of α satisfying the property that the limit $(o(\alpha)/\alpha)$ as $\alpha \to 0$ is 0 (the reason for the minus sign in the last line of (1.29) is the following. Since $\min\{0, g_i(x)\} \leqq 0$, $|\min\{0, g_i(x)\}| = -\min\{0, g_i(x)\}$).

Let $\mathbf{J} = \{i : k+1 \leqq i \leqq m, g_i(\hat{x}) < 0\}$, the index set of inequality constraints in the original problem (1.22) violated by the current point $\hat{x}$. For $k+1 \leqq i \leqq m$, $i \notin \mathbf{J}$, if $g_i(\hat{x}) = 0$, then $(\nabla g_i(\hat{x}))\tilde{d} \geqq 0$, from the constraints in (1.26). So, when α is positive but sufficiently small, for $k+1 \leqq i \leqq m$, $i \notin \mathbf{J}$, $\min\{0, g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d}\} = 0$. Therefore,

$$\sum_{i=k+1}^{m} \hat{\mu}_i(\min\{0, g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d}\}) = \sum_{i\in\mathbf{J}} \hat{\mu}_i(g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d})\,. \tag{1.30}$$

Also, for $1 \leqq i \leqq k$, $(\nabla g_i(\hat{x}))\tilde{d} = -g_i(\hat{x})$ by the constraints in (1.26). Therefore

$$\sum_{i=1}^{k} \hat{\mu}_i |g_i(\hat{x}) + \alpha(\nabla g_i(\hat{x}))\tilde{d}| = (1-\alpha)\sum_{i=1}^{k} \hat{\mu}_i |g_i(\hat{x})|\,. \tag{1.31}$$

¿From (1.28) we have $(\nabla\theta(\hat{x}))\tilde{d} = -\tilde{d}^T\hat{B}\tilde{d} + (\tilde{\pi}\nabla g(\hat{x}))\tilde{d} = -\tilde{d}^T\hat{B}\tilde{d} + \sum_{i=1}^{m} \tilde{\pi}_i(\nabla g_i(\hat{x}))\tilde{d} = -\tilde{d}^T\hat{B}\tilde{d} - \sum_{i=1}^{m}\tilde{\pi}_i g_i(\hat{x})$. Using this and (1.30), (1.31) in (1.29), we get

$$\begin{aligned} f(\alpha) =& \theta(\hat{x}) + \sum_{i=1}^{k} \hat{\mu}_i |g_i(\hat{x})| - \sum_{i\in \mathbf{J}} \hat{\mu}_i g_i(\hat{x}) \\ &+ \alpha[-\tilde{d}^T\hat{B}\tilde{d} - \sum_{i=1}^{k}\hat{\mu}_i|g_i(\hat{x})| - \sum_{i=1}^{m}\tilde{\pi}_i g_i(\hat{x}) - \sum_{i\in\mathbf{J}}\hat{\mu}_i(\nabla g_i(\hat{x}))\tilde{d}] + o(\alpha) \\ =& f(0) + \alpha[-\tilde{d}^T\hat{B}\tilde{d} - \sum_{i=1}^{k}(\hat{\mu}_i|g_i(\hat{x})| + \tilde{\pi}_i g_i(\hat{x})) \\ &- \sum_{i\in\overline{\mathbf{J}}}\tilde{\pi}_i g_i(\hat{x}) - \sum_{i\in\mathbf{J}}(\hat{\mu}_i(\nabla g_i(\hat{x}))\tilde{d} + \tilde{\pi}_i g_i(\hat{x}))] + o(\alpha)\,, \end{aligned} \tag{1.32}$$

where $\overline{\mathbf{J}} = \{k+1,\dots,m\} \setminus \mathbf{J}$. Now $\tilde{d}^T\hat{B}\tilde{d} > 0$ since $\hat{B}$ is PD and $\tilde{d} \neq 0$. Also, $\sum_{i=1}^{k}(\hat{\mu}_i|g_i(\hat{x})| + \tilde{\pi}_i g_i(\hat{x})) \geqq 0$, since $\hat{\mu}_i \geqq |\tilde{\pi}_i|$ for all $i = 1$ to k. Again $\sum_{i\in\overline{\mathbf{J}}}\tilde{\pi}_i g_i(\hat{x}) \geqq 0$ since $\tilde{\pi}_i \geqq 0$ and $g_i(\hat{x}) \geqq 0$ for all $i \in \overline{\mathbf{J}} = \{k+1,\dots,m\}\setminus\mathbf{J}$. Further, for $i \in \mathbf{J}$, $g_i(\hat{x}) < 0$, the constraints in the quadratic program imply $(\nabla g_i(\hat{x}))\tilde{d} \geqq -g_i(\hat{x}) > 0$; therefore, $\sum_{i\in\mathbf{J}}(\hat{\mu}_i(\nabla g_i(\hat{x}))\tilde{d} + \tilde{\pi}_i g_i(\hat{x})) \geqq \sum_{i\in\mathbf{J}}|g_i(\hat{x})|(\hat{\mu}_i - \tilde{\pi}_i) \geqq 0$. All this implies that the coefficient of α on the right hand side of (1.32) is strictly negative, that is, $f(\alpha) - f(0) < 0$ when α is sufficiently small and positive.

□

It is possible that even though the original problem is feasible and has a KKT point, the quadratic program (1.26) may be infeasible in some steps. See Example 1.8. In such steps, it is possible to define an alternate quadratic program of higher dimension which is always feasible, whose solution again provides a descent direction for the merit function $S(x)$. One such modification is given by the following quadratic programming problem

$$\begin{aligned} \text{minimize} \quad & (\nabla\theta(\hat{x}))d + (1/2)d^T\hat{B}d + \rho\Big(\sum_{i=1}^{m} u_i + \sum_{i=1}^{k} v_i\Big) \\ \text{subject to} \quad & g_i(\hat{x}) + (\nabla g_i(\hat{x}))d + u_i - v_i = 0\,,\ i = 1 \text{ to } k \\ & g_i(\hat{x}) + (\nabla g_i(\hat{x}))d + u_i \geqq 0\,,\ i = k+1 \text{ to } m \\ & u_i, v_i \geqq 0,\ \text{for all } i \end{aligned} \tag{1.33}$$

where ρ is a positive penalty parameter.

The quadratic program (1.33) is always feasible, since, $d = 0$ leads to a feasible solution to it. Let $\tilde{d}$, $\tilde{\pi}$ be an optimum solution and the associated Lagrange multiplier vector for (1.33). If $\tilde{d} \neq 0$, it can be shown that it provides a descent direction for the merit function $S(x)$ at the current point $\hat{x}$ using arguments similar to those in the proof of Theorem 1.15, and the method proceeds as usual. If (1.26) is infeasible and $\tilde{d} = 0$ is an optimum solution of (1.33), we cannot conclude that $\hat{x}$ is a KKT point for the original problem (1.22), and the method breaks down; however, the possibility of this occurrence can be discounted in practice.

Example 1.8

Consider the following nonlinear program from the paper of K. Tone [1.53].

$$\begin{array}{lrl} \text{Minimize} & \theta(x) = & x_1^3 + x_2^2 \\ \text{Subject to} & g_1(x) = & x_1^2 + x_2^2 - 10 = 0 \\ & g_2(x) = & x_1 \qquad - 1 \geqq 0 \\ & g_3(x) = & \qquad x_2 - 1 \geqq 0 \,. \end{array} \tag{1.34}$$

The set of feasible solutions for this problem is the thick chord of the circle in $\mathbf{R}^2$ in Figure 1.8. It can be verified that $x = (1, 3)^T$ is an optimum solution of this problem.

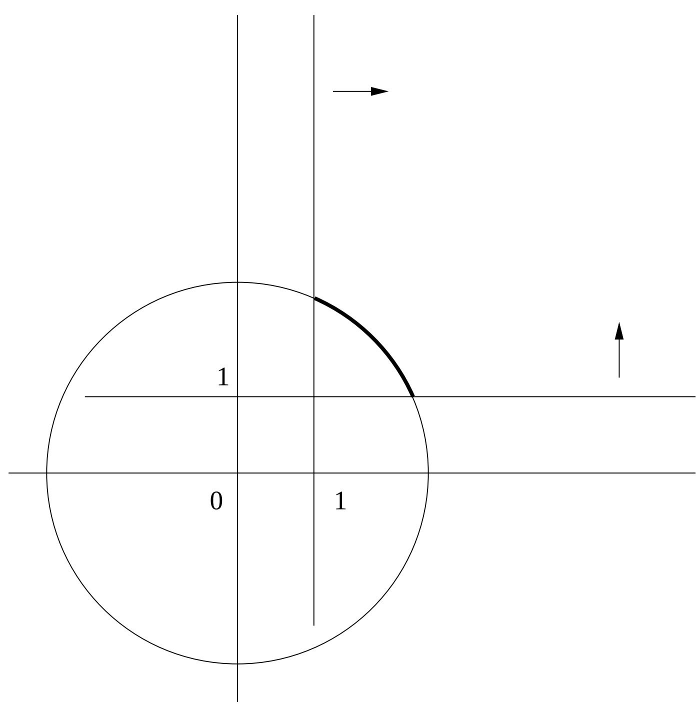


Figure 1.8

We have:

$$\begin{aligned}\nabla\theta(x) &= (3x_1^{\mathbf{2}}, 2x_2)\\ \nabla g_1(x) &= (2x_1, 2x_2)\\ \nabla g_2(x) &= (1, 0)\\ \nabla g_3(x) &= (0, 1)\,.\end{aligned}$$

We try to solve this problem by the recursive quadratic programming method using $x^\circ = (-10, -10)^T$ as the initial point. The constraints for the initial quadratic programming subproblem are

$$\begin{aligned}g_1(x^\circ) + (\nabla g_1(x^\circ))d &= \quad 190 - 20d_1 - 20d_2 = 0\\ g_2(x^\circ) + (\nabla g_2(x^\circ))d &= -\ 11 + \ d_1 \qquad\quad \geqq 0\\ g_3(x^\circ) + (\nabla g_3(x^\circ))d &= -\ 11 \qquad\ + \ d_2 \geqq 0\,.\end{aligned}$$

Even though the original NLP is feasible and has an optimum solution, it can be verified that this quadratic subproblem is infeasible. So, we use the quadratic programming subproblem as in (1.33). Taking the initial approximation to the Hessian of the Lagrangian to be $B_\circ = I_2$, this leads to the following quadratic programming problem.

$$\begin{array}{lllll}\text{minimize} & 300d_1 - 20d_2 + (1/2)(d_1^{\mathbf{2}} + d_2^{\mathbf{2}}) & & & \\ & \quad + \rho(u_1 + u_2 + u_3 + v_1) & & & \\ \text{subject to} & 20d_1 + 20d_2 & +u_1 - v_1 & = & 190\\ & d_1 & +u_2 & \geqq & 11\\ & \qquad d_2 & +u_3 & \geqq & 11\\ & & u_1, v_1, u_2, u_3 & \geqq & 0\,.\end{array} \tag{1.35}$$

Taking the penalty parameter $\rho = 1000$, this quadratic program has $\tilde{d} = (-1.5, 11)^T$ as the optimum solution with $\tilde{\pi} = (-35 \cdot 75, 1000, 692 \cdot 5)$ as the associated Lagrange multiplier vector corresponding to the constraints.

If we take penalty parameter vector $\mu = (1100, 1100, 1100)$ for constructing the merit function, we get the merit function

$$S(x) = x_1^{\mathbf{3}} + x_1^{\mathbf{2}} + 1100|x_1^{\mathbf{2}} + x_2^{\mathbf{2}} - 10| + 1100|\min\{0, x_1 - 1\}| + 1100|\min\{0, x_2 - 1\}|\,.$$

We minimize $S(x)$ on the half-line $\{x^\circ + \lambda\tilde{d} = (-10 - 1 \cdot 5\lambda, -10 + 11\lambda)^T, \lambda \geqq 0\}$. This problem can be solved using some of the line minimization algorithms discussed in Chapter 10. If the output of this problem is x^1, we update the Hessian approximation B, and with x^1, $\tilde{\pi}$ move over to the next quadratic programming subproblem and continue in the same way.

Under the assumptions:

(i) the quadratic program has an optimum solution in each step,

(ii) if $(\overline{x}, \overline{\pi})$ satisfies the KKT optimality conditions (1.23), then letting $\mathbf{J}(\overline{x}) = \{i : 1 \leqq i \leqq m, g_i(\overline{x}) = 0\}$, we have $\{\nabla g_i(\overline{x}) : i \in \mathbf{J}(\overline{x})\}$ is linearly independent; $\overline{\pi}_i > 0$

for all $i \in \mathbf{J}(\overline{x}) \cap \{k+1, \ldots, m\}$; and for any $y \neq 0$, $y \in \{y : (\nabla g_i(\overline{x}))\, y = 0,\ i \in \mathbf{J}(\overline{x})\}$, $y^T\big(\frac{\partial^2 L(\overline{x},\overline{\pi})}{\partial x^2}\big)y > 0$,

(iii) the initial point x^0 is sufficiently close to a KKT point for (1.22);

it has been proved (see references [1.44, 1.45]) that the sequence (x^r, π^r) generated by the algorithm converges superlinearly to $(\overline{x}, \overline{\pi})$ which together satisfy (1.23).

These recursive quadratic programming methods have given outstanding numerical performance and thereby attracted a lot of attention. However, as pointed out above, one difficulty with this approach is that the quadratic programming problem (1.26) may be infeasible in some steps, even if the original nonlinear program has an optimum solution, in addition the modified quadratic program (1.33) may have the optimum solution $\tilde{d} = 0$, in which case the method breaks down. Another difficulty is that constraint gradients need to be computed for each constraint in each step, even for constraints which are inactive. Yet another difficulty is the function $f(\alpha)$ minimized in the line search routine in each step, which is a non-differentiable L_1-penalty function. To avoid these and other difficulties, the following modified sequential quadratic programming method has been proposed for solving (1.22) by K. Schittkowski [1.50, 1.51].

Choose the initial point x^0, multiplier vector π^0, $B_0 = I$ or some PD symmetric approximation for $\frac{\partial^2 L(x^0,\pi^0)}{\partial x^2}$, $\rho_0 \in R^1$, $\gamma^0 \in \mathbf{R}^m$ ($\rho_0 > 0$, $\gamma^0 > 0$) and constants $\varepsilon > 0$, $\overline{\rho} > 1$, $0 < \overline{\delta} < 1$. The choice of $\varepsilon = 10^{-\mathbf{7}}$, $\overline{\delta} = 0.9$, $\overline{\rho} = 100$, and suitable positive values for ρ_0, γ^0 is reported to work well by K. Schittkowski [1.51]. Evaluate $\theta(x^0)$, $g_i(x^0)$, $\nabla g_i(x^0)$, $i = 1$ to m and go to stage 1.

General Stage $r{+}1$: Let x^r, π^r denote the current solution and Lagrange multiplier vector. Define

$$\mathbf{J}_1 = \{1, \ldots, k\} \cup \{i : k+1 \leqq i \leqq m,\ \text{and either } g_i(x^r) \leqq \varepsilon \text{ or } \pi_i^r > 0\}$$
$$\mathbf{J}_2 = \{1, \ldots, m\} \setminus \mathbf{J}_1\,.$$

The constraints in (1.22) corresponding to $i \in \mathbf{J}_1$ are treated as the active set of constraints at this stage, constraints in (1.22) corresponding to $i \in \mathbf{J}_2$ are the current inactive constraints.

Let B_r be the present matrix which is a PD symmetric approximation for $\frac{\partial^2 L(x^r,\pi^r)}{\partial x^2}$, this matrix is updated from step to step using the BFGS quasi-Newton update formula discussed earlier. The quadratic programming subproblem to be solved at this stage contains an additional variable, x_{n+1}, to make sure it is feasible. It is the following

$$\begin{aligned}
\text{minimize} \quad & P(d) = \frac{1}{2} d^T B_r d + (\nabla\theta(x^r))d + \frac{1}{2}\big(\rho_r x_{n+1}^{\mathbf{2}}\big) \\
\text{subject to} \quad & (\nabla g_i(x^r))d + (1 - x_{n+1}) g_i(x^r) \begin{cases} = 0, & i = 1 \text{ to } k \\ \geqq 0, & i \in \mathbf{J}_1 \cap \{k+1, \ldots, m\} \end{cases} \\
& (\nabla g_i(x^{s_i}))d + g_i(x^r) \geqq 0,\ i \in \mathbf{J}_2 \\
& 0 \leqq x_{n+1} \leqq 1
\end{aligned} \tag{1.36}$$

where, for each $i \in \mathbf{J}_2$, x^{s_i} denotes the most recent point in the sequence of points obtained under the method, at which $\nabla g_i(x)$ was evaluated; and ρ_r is a positive penalty parameter which is updated in each step using the formula

$$\rho_r = \text{ maximum } \left\{ \rho_0 \,,\ \frac{\rho^*((d^{r-1})^T A_{r-1} u^{r-1})^{\mathbf{2}}}{\left(1 - x_{n+1}^{r-1}\right)^{\mathbf{2}} (d^{r-1})^T B_{r-1} d^{r-1}} \right\} \tag{1.37}$$

where x_{n+1}^{r-1} u^{r-1}, d^{r-1} are the value of x_{n+1} in the optimum solution, the optimum Lagrange multiplier vector, and the optimum d-vector, associated with the quadratic programming problem in the previous stage; $\rho^* > 1$ is a constant; and A_{r-1} is the $n \times m$ matrix, whose jth column is the gradient vector of $g_i(x)$ computed at the most recent point, written as a column vector.

By definition of the set $\mathbf{J}_2$, the vector $(d = 0, x_{n+1} = 1)$ is feasible to this quadratic program, and hence, when B_r is PD, this quadratic program (1.34) has a finite unique optimum solution. One could also add additional bound constraints on the variables of the form $\delta_j \leqq d_j \leqq \delta_j$, $j = 1$ to n, where δ_j are suitable chosen positive numbers, to the quadratic programming subproblem (1.34), as discussed earlier.

Let (d^r, x_{n+1}^r), u^r, be the optimum solution and the optimum Lagrange multiplier vector, for the quadratic program (1.36). The solution of the quadratic programming subproblem (1.36) gives us the search direction d^r, for conducting a line search for a merit function or line search function corresponding to the original nonlinear program (1.22). If $x_{n+1}^r > \overline{\delta}$, change ρ_r into $\overline{\rho}\rho_r$ in (1.36) and solve (1.36) after this change. If this fails to lead to a solution with the value of x_{n+1} within the upper bound, define

$$\begin{aligned} d^r &= -B_r^{-1}\big(\nabla_x(\phi_{\gamma^r}(x^r, \pi^r))\big)^T \\ u^r &= \pi^r - \nabla_\pi(\phi_{\gamma^r}(x^r, \pi^r)) \end{aligned} \tag{1.38}$$

where $\phi_{\gamma^r}(x^r, \pi^r)$ is the line search function or the merit function defined later on in (1.39).

The new point in this stage is of the form

$$\begin{aligned} x^{r+1} &= x^r + \alpha_r d^r \\ \pi^{r+1} &= \pi^r + \alpha_r(u^r - \pi^r) \end{aligned}$$

where α_r is a step length obtained by solving the line search problem

$$\text{minimize } \ h(\alpha) = \phi_{\gamma^{r+1}}(x^r + \alpha d^r, \pi^r + \alpha(u^r - \pi^r))$$

over $\alpha \in \mathbf{R}^1$, where

$$\phi_\gamma(x, \pi) = \theta(x) - \sum_{i \in \Gamma} (\pi_i g_i(x) - \frac{1}{2}\gamma_i (g_i(x))^{\mathbf{2}}) - \frac{1}{2} \sum_{i \in \Delta} \pi_i^{\mathbf{2}} / \gamma_i \tag{1.39}$$

where $\mathbf{\Gamma} = \{1, \ldots, k\} \cup \{i : k < i \leqq m,\ g_i(x) \leqq \pi_i/\gamma_i\}$, $\mathbf{\Delta} = \{1, \ldots, m\} \setminus \mathbf{\Gamma}$, and the penalty parameters γ_i are updated using the formula

$$\gamma_i^{r+1} = \text{ maximum } \left\{ \sigma_i^r \gamma_i^r \,,\ \frac{2m(u_i^r - \pi_i^r)^{\mathbf{2}}}{(1 - x_{n+1}^r)(d^r)^T B_r d^r} \right\} \,, \quad i = 1 \text{ to } m\,. \tag{1.40}$$

The sequence $\{\sigma_i^r : r = 0, 1, \ldots\}$ is a bounded sequence with $\sigma_i^r \leqq 1$ for all r, and it allows the possibility of decreasing the penalty parameters γ_i. A possible choice for updating these parameters σ^r from stage to stage is by the formula

$$\sigma_i^r = \text{ minimum } \Big\{1, \frac{r}{\sqrt{\gamma_i^r}}\Big\}, \; r = 1, 2, \ldots, \; i = 1 \text{ to } m\,.$$

The function $\phi_\gamma(x, \pi)$ is a differentiable augmented Lagrangian function. If (d^r, u^r) are obtained from the solution of the quadratic program (1.36), let γ^{r+1} be obtained using (1.40). On the other hand, if (d^r, u^r) are obtained from (1.38), let $\gamma^{r+1} = \gamma^r$.

If $\frac{dh(0)}{d\alpha} \geqq 0$, replace ρ_r by $\overline{\rho}\rho_r$, and go back to solving the modified quadratic subproblem (1.36). Otherwise, perform a line search to minimize $h(\alpha)$ with respect to α, over $\alpha \geqq 0$, and let α_r be the optimum value of α for this line minimization problem. Define

$$\begin{aligned} x^{r+1} &= x^r + \alpha_r d^r \\ \pi^{r+1} &= \pi^r + \alpha_r (u^r - \pi^r) \end{aligned}$$

update the matrix B_r by the BFGS updating formula (1.25) and go to the next stage with these new quantities.

The algorithm can be terminated in the rth stage, if the following conditions are satisfied

$$\begin{aligned} (d^r)^T B_r d^r &\leqq \varepsilon^2 \\ \sum_{i=1}^{m} |u_i^r g_i(x^r)| &\leqq \varepsilon \\ \|\nabla_x L(x^r, u^r))\|^2 &\leqq \varepsilon \\ \sum_{i=1}^{k} |g_i(x^r)| + \sum_{i=k+1}^{m} |\text{ minimum } (0, g_i(x^r))| &\leqq \sqrt{\varepsilon}\,. \end{aligned}$$

For a global convergence analysis of this algorithm under suitable constraint qualification assumptions, see [1.51].

Algorithms for Quadratic Programming Problems

In this book we will discuss algorithms for quadratic programming problems which are based on its transformation to an LCP as discussed above. Since the quadratic program is a special case of a nonlinear program, it can also be solved by the reduced gradient methods, linearly constrained nonlinear programming algorithms, and various other methods for solving nonlinear programs. For a survey of all these nonlinear programming algorithms, see Chapter 10.

1.4 TWO PERSON GAMES

Consider a game where in each play of the game, player I picks one out of a possible set of his m choices and independently player II picks one out of a possible set of his

N choices. In a play, if player I has picked his choice, i, and player II has picked his choice j, then player I loses an amount a'_{ij} dollars and player II loses an amount b'_{ij} dollars, where $A' = (a'_{ij})$ and $B' = (b'_{ij})$ are given **loss matrices**.

If $a'_{ij} + b'_{ij} = 0$ for all i and j, the game is known as a **zero sum game**; in this case it is possible to develop the concept of an **optimum strategy** for playing the game using Von Neumann's Minimax theorem. Games that are not zero sum games are called **nonzero sum games** or **bimatrix games**. In bimatrix games it is difficult to define an optimum strategy. However, in this case, an **equilibrium pair of strategies** can be defined (see next paragraph) and the problem of computing an equilibrium pair of strategies can be transformed into an LCP.

Suppose player I picks his choice i with a probability of x_i. The column vector $x = (x_i) \in \mathbf{R}^m$ completely defines player I's strategy. Similarly let the probability vector $y = (y_j) \in \mathbf{R}^N$ be player II's strategy. If player I adopts strategy x and player II adopts strategy y, the expected loss of player I is obviously $x^T A' y$ and that of player II is $x^T B' y$.

The strategy pair $(\overline{x}, \overline{y})$ is said to be an **equilibrium pair** if no player benefits by unilaterally changing his own strategy while the other player keeps his strategy in the pair $(\overline{x}, \overline{y})$ unchanged, that is, if

$$\overline{x}^T A' \overline{y} \leqq x^T A' \overline{y} \quad \text{for all probability vectors } x \in \mathbf{R}^m$$

and

$$\overline{x}^T B' \overline{y} \leqq \overline{x}^T B' y \quad \text{for all probability vectors } y \in \mathbf{R}^N .$$

Let α, β be arbitrary positive numbers such that $a_{ij} = a'_{ij} + \alpha > 0$ and $b_{ij} = b'_{ij} + \beta > 0$ for all i, j. Let $A = (a_{ij})$, $B = (b_{ij})$. Since $x^T A' y = x^T A y - \alpha$ and $x^T B' y = x^T B y - \beta$ for all probability vectors $x \in \mathbf{R}^m$ and $y \in \mathbf{R}^N$, if $(\overline{x}, \overline{y})$ is an equilibrium pair of strategies for the game with loss matrices A', B', then $(\overline{x}, \overline{y})$ is an equilibrium pair of strategies for the game with loss matrices A, B, and vice versa. So without any loss of generality, consider the game in which the loss matrices are A, B.

Since x is a probability vector, the condition $\overline{x}^T A \overline{y} \leqq x^T A \overline{y}$ for all probability vectors $x \in \mathbf{R}^m$ is equivalent to the system of constraints

$$\overline{x}^T A \overline{y} \leqq A_{i.} \overline{y} \quad \text{for all } i = 1 \text{ to } m .$$

Let e_r denote the column vector in R^r in which all the elements are equal to 1. In matrix notation the above system of constraints can be written as $(\overline{x}^T A \overline{y}) e_m \leqq A \overline{y}$. In a similar way the condition $\overline{x}^T B \overline{y} \leqq \overline{x}^T B y$ for all probability vectors $y \in \mathbf{R}^N$ is equivalent to $(\overline{x}^T B \overline{y}) e_N \leqq B^T \overline{x}$. Hence the strategy pair $(\overline{x}, \overline{y})$ is an equilibrium pair of strategies for the game with loss matrices A, B iff

$$\begin{aligned} A \overline{y} &\geqq (\overline{x}^T A \overline{y}) e_m \\ B^T \overline{x} &\geqq (\overline{x}^T B \overline{y}) e_N . \end{aligned} \tag{1.41}$$

Since A, B are strictly positive matrices, $\overline{x}^T A\overline{y}$ and $\overline{x}^T B\overline{y}$ are strictly positive numbers. Let $\overline{\xi} = \overline{x}/(\overline{x}^T B\overline{y})$ and $\overline{\eta} = \overline{y}/(\overline{x}^T A\overline{y})$. Introducing slack variables corresponding to the inequality constraints, (1.41) is equivalent to

$$\begin{aligned} &\begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix} - \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \overline{\xi} \\ \overline{\eta} \end{pmatrix} = \begin{pmatrix} -e_m \\ -e_N \end{pmatrix} \\ &\begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix} \geqq 0, \quad \begin{pmatrix} \overline{\xi} \\ \overline{\eta} \end{pmatrix} \geqq 0, \quad \begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix}^T \begin{pmatrix} \overline{\xi} \\ \overline{\eta} \end{pmatrix} = 0\,. \end{aligned} \tag{1.42}$$

Conversely, it can easily be shown that if $(\overline{u}, \overline{v}, \overline{\xi}, \overline{\eta})$ is a solution of the LCP (1.42) then an equilibrium pair of strategies for the original game is $(\overline{x}, \overline{y})$ where $\overline{x} = \overline{\xi}/(\sum \overline{\xi}_i)$ and $\overline{y} = \overline{\eta}/(\sum \overline{\eta}_j)$. Thus an equilibrium pair of strategies can be computed by solving the LCP (1.42).

Example 1.9

Consider the game in which the loss matrices are

$$A' = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \qquad B' = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}.$$

Player I's strategy is a probability vector $x = (x_1, x_2)^T$ and player II's strategy is a probability vector $y = (y_1, y_2, y_3)^T$. Add 1 to all the elements in A' and 2 to all the elements in B', to make all the elements in the loss matrices strictly positive. This leads to

$$A = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \end{pmatrix}.$$

The LCP corresponding to this game problem is

$$\begin{bmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 \\ 1 & 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \tag{1.43}$$

$$u, v, \xi, \eta \geqq 0 \text{ and } u_1\xi_1 = u_2\xi_2 = v_1\eta_1 = v_2\eta_2 = v_3\eta_3 = 0\,.$$

Example 1.10

The Prisoner's Dilemma:

Here is an illustration of a bimatrix game problem from [1.31]. Two well known criminals were caught. During plea bargaining their Judge urged them both to confess and plead guilty. He explained that if one of them confesses and the other does not, the one who confesses will be acquitted and the other one given a sentence of 10 years in prison. If both of them confess, each will get a 5 year prison sentence. Both of them know very well that the prosecution's case against them is not strong, and the established evidence against them rather weak. However, the Judge said that if both of them decide not to confess, he will book both of them on some traffic violations for a year's prison term each. For each prisoner, let 1 refer to his choice of confessing and 2 to the choice of pleading not guilty. Measuring the loss in years in prison, their loss matrices are:

		A		B	
Player II's Choice →		1	2	1	2
Player I's Choice	1	5	0	5	10
	2	10	1	0	1

In this game it can be verified that the probability vectors $(\overline{x} = (1,0)^T,\ \overline{y} = (1,0)^T)$ provide the unique equilibrium pair for this game, resulting in a loss of a five year prison term for each player. But if both player's collude and agree to use the probability vectors $(\hat{x} = (0,1)^T,\ \hat{y} = (0,1)^T)$, the result, loss of a year's prison term for each player, is much better for both. The trouble with the strategy $(\hat{x}, \hat{y})$ is that each can gain by double-crossing the other.

Example 1.11

The Battle of the Sexes:
Here is another illustration of a bimatrix game from [1.31]. A newly married couple have to decide how they will spend Friday evening. The husband (player II) proposes to go to a boxing match and the wife (player I) proposes to go to a musical concert. The man rates the pleasure (or gain, or negative loss) he derives by going to the concert and the boxing match to be 1 and 4 units respectively on a scale from 0 to 5; and the corresponding figure for the woman are 4 and 1 units respectively. For each player let 1, 2 refer to the choices of insisting on going to concert, boxing match respectively. If their choices disagree, there is a fight, and neither gains any pleasure from going out that evening. Treating loss as negative pleasure, here are the loss matrices.

		A		B	
Player II's Choice →		1	2	1	2
Player I's Choice	1	−4	0	−1	0
	2	0	−1	0	−4

For this game, it can be verified that the probability vectors $(\overline{x} = (1,0)^T,\ \overline{y} = (1,0)^T)$. $(\hat{x} = (0,1)^T,\ \hat{y} = (0,1)^T)$ are both equilibrium pairs. The losses from the two equilibrium pairs $(\overline{x}, \overline{y})$, $(\hat{x}, \hat{y})$ are distinct, $(\overline{x}, \overline{y})$ will be preferred by player I, whereas II will prefer $(\hat{x}, \hat{y})$. Because of this, these equilibrium pairs are unstable. Even if player I knows that II will use the strategy $\hat{y}$, she may insist on using strategy $\overline{x}$ rather than $\hat{x}$, hoping that this will induce II to switch to $\overline{y}$. So, in this game, it is difficult to foresee what will happen. The probability vectors $(\tilde{x} = (4/5, 1/5),\ \tilde{y} = (1/5, 4/5)^T)$ is another equilibrium pair. In this problem, knowledge of these equilibrium pairs seems to have contributed very little towards the development of any "optimum" strategy.

Even though the theory of equilibrium strategies is mathematically elegant, and algorithms for computing them (through the LCP formulation) are practically efficient, they have not found many real world applications because of the problems with them illustrated in the above examples.

1.5 OTHER APPLICATIONS

Besides these applications, LCP has important applications in the nonlinear analysis of certain elastic-plastic structures such as reinforced concrete beams, in the free boundary problems for journal bearings, in the study of finance models, and in several other areas. See references [1.1 to 1.5, 1.8, 1.12, 1.13, 1.19, 1.21, 1.29, 1.32, 1.35].

1.6 THE NONLINEAR COMPLEMENTARITY PROBLEM

For each $j = 1$ to n, let $f_j(z)$ be a real valued function defined on $\mathbf{R}^n$. Let $f(z) = (f_1(z), \ldots, f_n(z))^T$. The problem of finding $z \in \mathbf{R}^n$ satisfying

$$\begin{aligned} z \geqq 0, &\quad f(z) \geqq 0 \\ z_j f_j(z) = 0, &\quad \text{for each } j = 1 \text{ to } n \end{aligned} \tag{1.44}$$

is known as a nonlinear complementarity problem (abbreviated as NLCP). If we define $f_j(z) = M_{j.}z + q_j$ for $j = 1$ to n, it can be verified that (1.44) becomes the LCP (1.1). Thus the LCP is a special case of the NLCP. Often, it is possible to transform the necessary optimality conditions for a nonlinear program into that of an NLCP and thereby solve the nonlinear program using algorithms for NLCP. The NLCP can be transformed into a fixed point computing problem, as discussed in Section 2.7.7, and solved by the piecewise linear simplicial methods presented in Section 2.7. Other than this, we will not discuss any detailed results on NLCP, but the references [1.14 to 1.16, 1.24, 1.25, 1.39] can be consulted by the interested reader.

1.7 Exercises

1.4 Consider the two person game with loss matrices A, B. Suppose $A + B = 0$. Then the game is said to be a **zero sum game** (see references [1.28, 1.31]). In this case prove that every equilibrium pair of strategies for this game is an optimal pair of strategies in the minimax sense (that is, it minimizes the maximum loss that each player may incur. See references [1.28, 1.31]). Show that the same results continue to hold as long as $a_{ij} + b_{ij}$ is a constant independent of i and j.

1.5 Consider the bimatrix game problem with given loss matrices A, B. Let $x = (x_1, \ldots, x_m)^T$ and $y = (y_1, \ldots, y_n)^T$ be the probability vectors of the two players. Let $X = (x_1, \ldots, x_m, x_{m+1})^T$ and $Y = (y_1, \ldots, y_n, y_{n+1})^T$. Let e_r be the column vector in $\mathbf{R}^r$ all of whose entries are 1. Let $\mathbf{S} = \{X : B^T x - e_n^T x_{m+1} \geqq 0, e_m^T x = 1, x \geqq 0\}$ and $\mathbf{T} = \{Y : Ay - e_m^T y_{n+1} \geqq 0, e_n^T y = 1, y \geqq 0\}$. Let $Q(X, Y) = x^T (A+B) y - x_{m+1} - y_{n+1}$. If $(\overline{x}, \overline{y})$ is an equilibrium pair of strategies for the game and $\overline{x}_{m+1} = \overline{x}^T B \overline{y}$, $\overline{y}_{n+1} = \overline{x}^T A \overline{y}$, prove that $(\overline{X}, \overline{Y})$ minimizes $Q(X, Y)$ over $\mathbf{S} \times \mathbf{T} = \{(X, Y) : X \in \mathbf{S},\ Y \in \mathbf{T}\}$. (O. L. Mangasarian)

1.6 Consider the quadratic program:

$$\begin{array}{ll} \text{Minimize} & Q(x) = cx + \frac{1}{2} x^T D x \\ \text{Subject to} & Ax \geqq b \\ & x \geqq 0 \end{array}$$

where D is a symmetric matrix. $\mathbf{K}$ is the set of feasible solutions for this problem. $\overline{x}$ is an interior point of $\mathbf{K}$ (i. e., $A\overline{x} > b$ and $\overline{x} > 0$).

(a) What are the necessary conditions for $\overline{x}$ to be an optimum solution of the problem?

(b) Using the above conditions, prove that if D is not PSD, $\overline{x}$ could not be an optimum solution of the problem.

1.7 For the following quadratic program write down the corresponding LCP.

$$\begin{array}{lll} \text{Minimize} & -6x_1 - 4x_2 - 2x_3 + 3x_1^2 + 2x_2^2 + \frac{1}{3} x_3^2 & \\ \text{Subject to} & x_1 + 2x_2 + x_3 \leqq 4 & \\ & x_j \geqq 0 & \text{for all} j\,. \end{array}$$

If it is known that this LCP has a solution in which all the variables x_1, x_2, x_3 are positive, find it.

1.8 Write down the LCP corresponding to

$$\begin{array}{ll} \text{Minimize} & cx + \frac{1}{2}x^T Dx \\ \text{Subject to} & x \geqq 0\,. \end{array}$$

1.9 Let

$$M = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}, \quad q = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Show that the LCP (q, M) has four distinct solutions. For $n = 3$, construct a square matrix M of order 3 and a $q \in \mathbf{R}^3$ such that (q, M) has eight distinct solutions.

Hint. Try $-M = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 4 \end{pmatrix} \quad q = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$; or try $M = -I, q > 0$.

1.10 Let

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad q = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}.$$

Find out a solution of the LCP (q, M) by inspection. However, prove that there exists no complementary feasible basis for this problem.
(L. Watson)

1.11 Test whether the following matrices are PD, PSD, or not PSD by using the algorithms described in Section 1.3.1

$$\begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 1 & 2 & 1 \end{pmatrix}, \begin{pmatrix} 4 & 3 & -7 \\ 0 & 0 & -2 \\ 0 & 0 & 6 \end{pmatrix}, \begin{pmatrix} 4 & 100 & 2 \\ 0 & 2 & 10 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 5 & -2 & -2 \\ 0 & 5 & -2 \\ 0 & 0 & 5 \end{pmatrix}.$$

1.12 Let $Q(x) = (1/2)x^T Dx - cx$. If D is PD, prove that $Q(x)$ is bounded below.

1.13 Let $\mathbf{K}$ be a nonempty closed convex polytope in $\mathbf{R}^n$. Let $f(x)$ be a real valued function defined on $\mathbf{R}^n$. If $f(x)$ is a concave function, prove that there exists an extreme point of $\mathbf{K}$ which minimizes $f(x)$ on $\mathbf{K}$.

1.14 Let D be an arbitrary square matrix of order n. Prove that, for every positive and sufficiently large λ, the function $Q_\lambda(x) = x^T(D - \lambda I)x + cx$ is a concave function on $\mathbf{R}^n$.

1.15 Consider the following quadratic assignment problem.

$$\begin{aligned}
\text{minimize} \quad & z(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{n}\sum_{q=1}^{n} c_{ijpq} x_{ij} x_{pq} \\
\text{subject to} \quad & \sum_{j=1}^{n} x_{ij} = 1, \text{ for all } i = 1 \text{ to } n \\
& \sum_{i=1}^{n} x_{ij} = 1, \text{ for all } j = 1 \text{ to } n \\
& x_{ij} \geqq 0, \text{ for all } i, j = 1 \text{ to } n
\end{aligned} \tag{1.45}$$

and

$$x_{ij} \text{ integral for } i, j = 1 \text{ to } n\,. \tag{1.46}$$

Show that this discrete problem (1.45), (1.46) can be posed as another problem of the same form as (1.45) without the integrality constraints (1.46).

1.16 Consider an optimization problem of the following form

$$\begin{aligned}
\text{minimize} \quad & \frac{(x^T D x)^{1/2}}{dx + \beta} \\
\text{subject to} \quad & Ax \geqq b \\
& x \geqq 0
\end{aligned}$$

where D is a given PSD matrix and it is known that $dx + \beta > 0$ on the set of feasible solutions of this problem. Using the techniques of fractional programming (see Section 3.20 in [2.26]), show how this problem can be solved by solving a single convex quadratic programming problem. Using this, develop an approach for solving this problem efficiently by algorithms for solving LCPs
(J. S. Pang, [1.33]).

1.17 Let D be a given square matrix of order n. Develop an efficient algorithm which either confirms that D is PSD or produces a vector $y \in \mathbf{R}^n$ satisfying $y^T D y < 0$.

1.18 Consider the following quadratic programming problem

$$\begin{aligned}
\text{minimize} \quad & Q(x) = cx + \frac{1}{2} x^T D x \\
\text{subject to} \quad & a \leqq Ax \leqq b \\
& l \leqq x \leqq u
\end{aligned}$$

where A, D, c, a, b, l, u are given matrices of orders $m \times n$, $n \times n$, $1 \times n$, $m \times 1$, $m \times 1$, $n \times 1$, $n \times 1$ respectively, and D is symmetric. Express the necessary optimality conditions for this problem in the form of an LCP. (R. W. H. Sargent, [1.37])

1.19 Suppose D is a symmetric matrix of order n. Show that the KKT necessary optimality conditions for the quadratic program

$$\begin{array}{ll} \text{minimize} & cx + (1/2)x^T Dx \\ \text{subject to} & 0 \leqq x \leqq b \end{array}$$

where $b > 0$ is a given vector, are of the form: find, $x, y \geqq 0$ in $\mathbf{R}^n$ satisfying $c^T + Dx + y \geqq 0$, $b - x \geqq 0$, $x^T(c^T + Dx + y) = y^T(b - x) = 0$. Express these conditions in the form of an LCP. Also prove that this is equivalent to finding an $x \in \mathbf{R}^n$ satisfying $0 \leqq x \leqq b$ and $(u - x)^T(Dx + c^T) \geqq 0$ for all $0 \leqq u \leqq b$. Prove that this LCP always has a solution and that the solution is unique if D is a P-matrix.
(B. H. Ahn [9.4], S. Karamardian [1.15])

1.20 Weighted Min-Max Location Problem: Given m points $a^i = (a^i_1, \ldots, a^i_n)^T \in \mathbf{R}^n$, $i = 1$ to m, and positive weights δ_i, $i = 1$ to m associated with these points, define the function $\theta(x) =$ maximum $\{\delta_i \sqrt{(x - a^i)^T(x - a^i)} : i = 1 \text{ to } m\}$ over $x \in \mathbf{R}^n$. The weighted min-max location problem is to find an $x \in \mathbf{R}^n$ that minimizes $\theta(x)$. Show that this problem is equivalent to the problem

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \lambda - \delta_i^2(\|a^i\|^2 + \sum_{j=1}^{n} x_j^2 - 2\sum_{j=1}^{n} a^i_j x_j) \geqq 0, \ i = 1 \text{ to } m \end{array} \tag{1.47}$$

where λ is treated as another variable in (1.47). Consider the following quadratic program

$$\begin{array}{ll} \text{minimize} & Q(X) = \sum_{j=1}^{n} x_j^2 - x_{n+1} \\ \text{subject to} & x_{n+1} - 2\sum_{j=1}^{n} a^i_j x_j \leqq \|a^i\|^2 + \dfrac{\lambda}{\delta_i^2}\,, \quad i = 1 \text{ to } m \end{array} \tag{1.48}$$

where x_{n+1} is an additional variable in (1.48), $X = (x_1, \ldots, x_n, x_{n+1})$. Prove that if $(\overline{x}, \overline{\lambda})$ is feasible to (1.47), $(\overline{x}, \overline{\lambda}, \overline{x}_{n+1})$ where $\overline{x}_{n+1} = \sum_{j=1}^{n} \overline{x}_j^2$, is feasible to (1.48) with $Q(\overline{X}) = 0$. Conversely if $(\hat{x}, \hat{\lambda})$ is feasible to (1.48) with $Q(\hat{X}) \leqq 0$, then show that $(\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n), \hat{\lambda})$ is feasible to (1.47). Also, for each $\lambda > 0$, prove that the optimum solution of (1.48) is unique. Treating λ as a parameter, denote the optimum solution of (1.48) as a function of λ by $X(\lambda)$. Let $\tilde{\lambda}$ be the smallest value of λ for which $Q(X(\lambda)) \leqq 0$. Prove that $x(\tilde{\lambda})$ is the optimum solution of the min-max location problem. Use these results to develop an algorithm for the min-max location problem based on solving a parametric right hand side LCP.
(R. Chandrasekaran and M. J. A. P. Pacca, [1.2])

1.21 Let F be a square matrix of order n. In general there may be no relation between determinant $((F + F^T)/2)$ and determinant (F). Establish conditions under which determinant $((F + F^T)/2) \leqq$ determinant (F).

1.22 Let $\mathbf{K} \subset \mathbf{R}^n$ convex and $Q(x) = cx + \frac{1}{2}x^T Dx$. If $Q(x)$ is convex over $\mathbf{K}$ and $\mathbf{K}$ has nonempty interior, prove that $Q(x)$ is convex over the whole space $\mathbf{R}^n$.

1.23 Concave Regression Problem: Here, given a real valued function $\theta(t)$ defined on an interval, it is desired to find a convex (or concave, depending on the application) function that approximates it as closely as possible. Specifically, suppose we are given $\theta_i = \theta(\alpha_i)$, $i = 1$ to n, where $\alpha_1 < \alpha_2 < \ldots < \alpha_n$. So we are given the values of $\theta(t)$ at the points $t = \alpha_1, \ldots, \alpha_n$. It is required to find real values $f_1, \ldots, f_n$ so that $f_i = f(\alpha_i)$, $i = 1$ to n where f is a convex function defined on the real line, that minimizes the measure of deviation $\sum_{i=1}^n \gamma_i(\theta_i - f_i)^2$ where γ_i, $i = 1$ to n are given positive weights. Formulate this problem as an LCP.

1.24 $\mathbf{K}_1$ and $\mathbf{K}_2$ are two convex polyhedra in $\mathbf{R}^n$, each of them provided as the set of feasible solutions of a given system of linear inequalities. Develop an algorithm for the problem

$$\begin{aligned} &\text{minimize } \|x - y\| \\ &x \in \mathbf{K}_1, y \in \mathbf{K}_2 \,. \end{aligned}$$

1.25 Sylvester's Problem: We are given a set of n points in $\mathbf{R}^m$, $\{A_{.1}, \ldots, A_{.n}\}$, where $A_{.j} = (a_{1j}, \ldots, a_{mj})^T$, $j = 1$ to n. It is required to find the smallest diameter sphere in $\mathbf{R}^m$ containing all the points in the set $\{A_{.1}, \ldots, A_{.n}\}$. Transform this into a quadratic program and discuss an algorithm for solving it. Apply your algorithm to find the smallest diameter circle containing all the points in $\{(1, 1), (-3, 2), (1, -5), (-2, 4)\}$ in $\mathbf{R}^2$.
(References [1.5, 1.29])

1.26 Let $\mathbf{K}$ be any convex polyhedral subset of R^n (you can assume that $\mathbf{K}$ is the set of feasible solutions of $Ax \geqq b$ where A, b are given). Let x^0, x^1 be given points in $\mathbf{R}^n$. Let $\tilde{x}$, $\hat{x}$ be respectively the nearest points in $\mathbf{K}$ (in terms of the usual Euclidean distance) to x^0, x^1 respectively. Prove that $\|\tilde{x} - \hat{x}\| \leqq \|x^0 - x^1\|$.

1.27 Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ be a given row vector of $\mathbf{R}^n$ and let $x^0 \in \mathbf{R}^n$ be another given column vector. It is required to find the nearest point in $\mathbf{K} = \{x : \alpha x \leqq 0, x \geqq 0\}$ to x^0, in terms of the usual Euclidean distance. For this, do the following. Let λ be a real valued parameter. Let λ_0 be the smallest nonnegative value of λ for which the piecewise linear, monotonically decreasing function $\alpha(x^0 - \lambda\alpha^T)^+$ assumes a nonpositive value. Let $\overline{x} = (x^0 - \lambda_0\alpha^T)^+$. (For any vector $y = (y_j) \in \mathbf{R}^n$, $y^+ = (y_j^+)$ where $y_j^+ = \text{Maximum } \{0, y_j\}$ for each j.) Prove that $\overline{x}$ is the nearest point in $\mathbf{K}$ to x^0.

Extend this method into one for finding the nearest point in $\mathbf{\Gamma} = \{x : x \geqq 0, \alpha x \leqq \delta\}$ to x^0, where δ is a given number, assuming that $\mathbf{\Gamma} \neq \emptyset$.
(W. Oettli [1.30])

1.28 Let M be a square matrix of order n and $q \in \mathbf{R}^n$. Let $z \in \mathbf{R}^n$ be a vector of variables. Define: $f_i(z) =$ minimum $\{z_i, M_{i.}z + q_i\}$, that is

$$\begin{aligned} f_i(z) &= I_{i.}z && \text{if} \quad (M_{i.} - I_{i.})z + q_i \geqq 0 \\ &= M_{i.}z + q_i && \text{if} \quad (M_{i.} - I_{i.})z + q_i \leqq 0 \end{aligned}$$

for each $i = 1$ to n.

(a) Show that $f_i(z)$ is a piecewise linear concave function defined on $\mathbf{R}^n$

(b) Consider the system of equations

$$f_i(z) = 0 \quad i = 1 \text{ to } n\,.$$

Let $\overline{z}$ be a solution of this system. Let $\overline{w} = M\overline{z} + q$. Prove that $(\overline{w}, \overline{z})$ is a complementary feasible solution of the LCP (q, M).

(c) Using (b) show that every LCP is equivalent to solving a system of piecewise linear equations.
(R. Saigal)

1.29 For $j = 1$ to n define $x_j^+ =$ Maximum $\{0, x_j\}$, $x_j^- = -$ Minimum $\{0, x_j\}$. Let $x = (x_j) \in \mathbf{R}^n$, $x^+ = (x_j^+)$, $x^- = (x_j^-)$. Given the square matrix M of order n, define the piecewise linear function

$$T_M(x) = x^+ - Mx^-\,.$$

Show that $T_M(x)$ is linear in each orthant of $\mathbf{R}^n$. Prove that $(w = x^+, z = x^-)$ solves the LCP (q, M) iff $q = T_M(x)$.
(R. E. Stone [3.71])

1.30 Let D be a given square matrix of order n, and $f(x) = x^T Dx$. Prove that there exists a nonsingular linear transformation: $y = Ax$ (where A is a square nonsingular matrix of order n) such that

$$f(x) = y_1^2 + \ldots + y_p^2 - y_{p+1}^2 - \ldots - y_r^2$$

where $0 \leqq p \leqq r \leqq n$. Discuss an efficient method for finding such a matrix A, given D.
Find such a transformation for the quadratic form $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 - 2x_1x_3 - 2x_2x_3$ (this dates back to Lagrange in 1759, see D. E. Knuth [10.20]).

1.31 Sylvester's Law of Inertia (dates from 1852): Let D be a given square matrix of order n, and $f(x) = x^T Dx$. If there exist nonsingular linear transformations: $y = Ax$, $z = Bx$ (A, B are both square nonsingular matrices of order n) such that

$$f(x) = y_1^2 + \ldots + y_p^2 - y_{p+1}^2 - \ldots - y_r^2 = z_1^2 + \ldots + z_q^2 - z_{q+1}^2 - \ldots - z_s^2$$

then prove that $p = q$ and $r = s$.

This shows that the numbers p and r associated with a quadratic form, defined in Exercise 1.30 are unique
(see D. E. Knuth [10.20]).

1.32 Using the notation of Exercise 1.30 prove that $r = n$ iff the matrix $(D + D^T)/2$ has no zero eigenvalues and that p is the number of positive eigenvalues of
$(D + D^T)/2$.

Let D_0, D_1 be two given square matrices of order n, and let $D_\alpha = (1-\alpha)D_0 + \alpha D_1$. Let $r(D_\alpha)$, $p(D_\alpha)$ be the numbers r, p, associated with the quadratic form $f_\alpha = x^T D_\alpha x$ as defined in Exercise 1.30. If $r(D_\alpha) = n$ for all $0 \leqq \alpha \leqq 1$, prove that $p(D_0) = p(D_1)$.
(See D. E. Knuth [10.20].)

1.33 To Determine Optimum Mix of Ingredients for Moulding Sand in a Foundry: In a heavy casting steel foundry, moulding sand is prepared by mixing sand, resin (Phenol formaledhyde) and catalyst (Para toluene sulfonic acid). In the mixture the resin undergoes a condensation polymerization reaction resulting in a phenol formaldehyde polymer that bonds and gives strength. The bench life of the mixed sand is defined to be the length of the time interval between mixing and the starting point of setting of the sand mix. In order to give the workers adequate time to use the sand and for proper mould strength, the bench life should be at least 10 minutes. Another important characteristic of the mixed sand is the dry compression strength which should be maximized. An important variable which influences these characteristics is the resin percentage in the mix, extensive studies have shown that the optimum level for this variable is 2 % of the weight of sand in the mix, so the company has fixed this variable at this optimal level. The other process variables which influence the output characteristics are:

$x_1 =$ temperature of sand at mixing time
$x_2 =$ % of catalyst, as a percent of resin added
$x_3 =$ dilution of catalyst added at mixing.

The variable x_3 can be varied by adding water to the catalyst before it is mixed. An experiment conducted yielded the following data.

Dry Compression Strength

	$x_3 = 0$				10			
x_1	$x_2 = 25$	30	35	40	25	30	35	40
20^c	31.4	32.4	33.7	37.3	32.7	33.7	36.3	34.0
30^c	33.4	34.1	34.9	32.6	30.1	31.1	35.0	35.2
40^c	33.8	31.4	38.0	32.4	31.6	32.3	34.7	34.8

Bench Life

	$x_3 = 0$				10			
x_1	$x_2 = 25$	30	35	40	25	30	35	40
20^c	13.3	11.5	10.8	10.3	15.8	14.0	12.8	11.8
30^c	10.3	9.0	8.0	6.8	12.3	11.0	10.3	9.3
40^c	7.0	6.3	5.0	4.3	11.8	10.5	7.3	5.8

Bench life can be approximated very closely by an affine function in the variables x_1, x_2, x_3; and dry compression strength can be approximated by a quadratic function in the same variables. Find the functional forms for these characteristics that provide the best approximation. Using them, formulate the problem of finding the optimal values of the variables in the region $0 \leqq x_3 \leqq 10$, $25 \leqq x_2 \leqq 40$, $20 \leqq x_1 \leqq 40$, so as to maximize the dry compression strength subject to the additional constraint that the bench life should be at least ten, as a mathematical programming problem. Find an optimum solution to this mathematical program. (Hint: For curve fitting use either the least squares method discussed in Section 1.3.5, or the minimum absolute deviation methods based on linear programming discussed in [2.26, Section 1.2.5].)
(Bharat Heavy Electricals Ltd., Hardwar, India).

1.34 Synchronous Motor Design Problem: There are 11 important design variables (these are variables like the gauge of the copper wring used, etc. etc.) denoted by x_1 to x_{11} and let $x = (x_1, \dots, x_{11})^T$. These variables effect the raw material cost for this motor, denoted by $f_0(x)$; the efficiency of the motor (= (output energy)/(input energy) measured as a percentage) denoted by $f_1(x)$; and the power factor (this measures leakage, it is a loss measured as a percentage) denoted by $f_2(x)$. Subroutines are available for computing each of the functions $f_0(x)$, $f_1(x)$, $f_2(x)$ for given x. The problem is to find optimal values for the variables which minimizes $f_0(x)$ subject to $f_1(x) \geqq 86.8$ and $f_2(x) \leqq 90$ and $l \leqq x \leqq u$, where l, u are specified lower and upper bound vectors for the variables. Discuss a method for solving this problem.

1.35 Quadratic Programming Model to Determine State Taxes: It is required to determine optimum levels for various state government taxes that minimizes instability while meeting constraints on growth rates over time. Seven different taxes are considered, sales, motor fuel, alcoholic beverages, tobacco, motor vehicle, personal income, and corporate taxes. State government finance is based on the assumption of predictable and steady growth of each tax over time. Instability in tax revenue is measured by the degree to which the actual revenue differs from predicted revenue.

Using past data, a regression equation can be determined to measure the growth in tax revenue over time. Let s be the tax rate for a particular tax and S_t the expected tax revenue from this tax in year t. Then the regression equation used is

$$\log_e S_t = a + bt + cs$$

where a, b, c are parameters to be determined using past data to give the closest fit. Data for the past 10 years from a state is used for this parameter estimation. Clearly, the parameter c can only be estimated, if the tax rate s for that tax has changed during this period, this has happened only for the motor fuel and the tobacco taxes. The best fit parameter values for the various taxes are given below (for all but the motor fuel and tobacco taxes, the tax rate has remained the same over the 10 years period for which the tax data is available, and hence the parameter a given below for these taxes, is actually the value of $a + cs$, as it was not possible to estimate a and c individually from the data).

Table 1: Regression coefficient values

j	Tax j	a	b	c
1	Sales	12.61	.108	
2	Motor fuel	10.16	.020	.276
3	Alcoholic beverages	10.97	.044	
4	Tobacco	9.79	.027	.102
5	Motor vehicle	10.37	.036	
6	Personal income	11.89	.160	
7	Corporate	211.09	.112	

The annual growth rate is simply the regression coefficient b multiplied by 100 to convert it to percent.

For 1984, the tax revenue from each tax as a function of the tax rate can be determined by estimating the tax base. This data, available with the state, is given below.

j	Tax j	Tax base (millions of dollars)
1	Sales	34,329
2	Motor fuel	3,269
3	Alcoholic beverages	811
4	Tobacco	702
5	Motor vehicle	2,935
6	Personal income	30,809
7	Corporate	4,200

If s_j is the tax rate for tax j in 1984 as a fraction, x_j = tax revenue to be collected in 1984 in millions of dollars for the jth tax is expected to be: (tax base for tax j) s_j.

Choosing the decision variables to be x_j for $j = 1$ to 7, let $x = (x_1, \dots, x_7)^T$. The total tax revenue is $\sum_{j=1}^{7} x_j$. Then the variability or instability in this revenue is measured by the quadratic function $Q(x) = x^T V x$ where V, the variance-covariance matrix estimated from past data is

$$\begin{pmatrix} .00070 & -.00007 & .00108 & -.00002 & .00050 & .00114 & .00105 \\ & .00115 & .00054 & -.00002 & .00058 & -.00055 & .00139 \\ & & .00279 & .00016 & .00142 & .00112 & .00183 \\ & & & .00010 & .00009 & -.00007 & -.00003 \\ & & & & .00156 & .00047 & .00177 \\ & & & & & .00274 & .00177 \\ & & & & & & .00652 \end{pmatrix} .$$

Since V is symmetric, only the upper half of V is recorded above.

The problem is to determine the vector x that minimizes $Q(x)$, subject to several constraints. One of the constraints is that the total expected tax revenue for 1984 should be $T = 3300$ in millions of dollars. The second constraint is that a specified growth rate of λ in the total tax revenue should be maintained. It can be assumed that this overall growth rate is the function $\sum_{i=1}^{7} \frac{x_j b_j}{T}$ which is a weighted average of the growth rates of the various taxes. We would like to solve the problem treating λ as a nonnegative parameter. Of particular interest are values $\lambda = 9\ \%$ and $13\ \%$.

The other constraints are lower and upper bounds on tax revenues x_j, these are of the form $0 \leqq x_j \leqq u_j$ for each j; where u_j is twice the 1983 revenue from tax j. The vector $u = (u_j)$ is (2216, 490, 195, 168, 95, 2074, 504) in millions of dollars.

Formulate this problem as an LCP and solve it using the complementary pivot algorithm discussed in Chapter 2. Using the tax base information given above, determine the optimal tax rates for 1984 for each tax.
(F. C. White [1.40], my thanks to H. Bunch for bringing this paper to my attention.)

1.36 Consider the equality constrained nonlinear program

$$\begin{array}{ll} \text{minimize} & \theta(x) \\ \text{subject to} & h_i(x) = 0, \quad i = 1 \text{ to } m\,. \end{array}$$

The quadratic merit function for this problem is $S(x) = \theta(x) + (\rho/2)\sum_{i=1}^{m}(h_i(x))^{2}$ where ρ is a positive penalty parameter. Let $\overline{x} \in \mathbf{R}^n$ be an initial point and $\overline{\mu} = (\overline{\mu}_1, \ldots, \overline{\mu}_m) \in \mathbf{R}^m$ be a given Lagrange multiplier vector. Consider the equality constrained quadratic program in variables $d = (d_1, \ldots, d_n)^T$

$$\begin{array}{ll} \text{minimize} & \nabla\theta(\overline{x})d + \frac{1}{2}d^T Bd \\ \text{subject to} & (h(\overline{x}))^T + (\nabla h(\overline{x}))d = \overline{\mu}/\rho \end{array}$$

where B is a symmetric PD matrix of order n. If $\overline{d} \neq 0$ is an optimum solution of this quadratic program, and $\overline{\pi} = (\overline{\pi}_1, \ldots, \overline{\pi}_m)$ the associated Lagrange multiplier vector, prove that $\overline{d}$ is a descent direction for $S(x)$ at $\overline{x}$.

1.37 Let $A = (a_{ij})$ be a given square matrix of order n. Consider the usual assignment problem

$$\begin{array}{lll} \text{minimize} & z(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_{ij} & \\ \text{subject to} & \sum_{i=1}^{n} x_{ij} = 1, & j = 1 \text{ to } n \\ \text{subject to} & \sum_{j=1}^{n} x_{ij} = 1, & i = 1 \text{ to } n \\ \text{subject to} & x_{ij} \geqq 0, & i, j = 1 \text{ to } n\,. \end{array}$$

i) Prove that if A is PD and symmetric, $\overline{x} = I_n$ = unit matrix of order n, is an optimum solution for this problem. Is the symmetry of A important for this result to be valid?

ii) Using the above, prove that if A is PD and symmetric, there exists a vector $u = (u_1, \ldots, u_n)$ satisfying

$$u_i - u_j \geqq a_{ij} - a_{jj}, \quad i, j = 1 \text{ to } n\,.$$

1.38 Consider the problem of an investor having one dollar to invest in assets $i = 1, \ldots, n$. If x_i is invested in asset i, then $\xi_i x_i$ is returned at the end of the investment period, where $(\xi_1, \ldots, \xi_n)$ are random variables independent of the choice of $x_i s$, with the row-vector of means $\mu = (\mu_1, \ldots, \mu_n)$ $(\mu > 0)$ and a positive definite symmetric variance-covariance matrix D. In portfolio theory, under certain assumptions, it is shown that optimal investment proportions, $x = (x_1, \ldots, x_n)^T$, may be obtained by maximizing the fractional objective function

$$g(x) = \frac{\mu x}{(x^T Dx)^{1/2}}\,.$$

i) A real valued function $f(x)$ defined on a convex set $\mathbf{K} \subset \mathbf{R}^n$ is said to be pseudo-concave on $\mathbf{K}$ if it is differentiable on $\mathbf{K}$ and for every x^1, $x^2 \in \mathbf{K}$, $\nabla f(x^2)(x^1 - x^2) \leqq 0$ implies $f(x^1) \leqq f(x^2)$.

Prove that $g(x)$ is pseudo-concave in $\{x : x > 0\}$, even though it is not in general concave on this set.

For the problem of maximizing a pseudo-concave function on a convex set, prove that every local maximum is a global maximum.

Consider the problem

$$\begin{array}{ll} \text{maximize} & g(x) \\ \text{subject to} & \sum_{j=1}^{n} x_j = 1 \\ & x_j \geqq 0, \text{ for all } j\,. \end{array}$$

Show that this problem has a unique optimum solution. Also, show that an optimum solution of this problem can be obtained from the solution of the LCP $(-\mu, D)$.
(W. T. Ziemba, C. Parkan and R. Brooks-Hill [3.80])

1.39 In Section 1.3.5, the computational problems associated with the Hilbert matrix were mentioned briefly. Consider the following linear program

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \leqq b \end{array}$$

where

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n} \end{pmatrix}$$

$$b = (b_i : i = 1 \text{ to } n)^T = \Big(\sum_{j=1}^{n} \frac{1}{i+j}\Big)$$

$$c = (c_j : j = 1 \text{ to } n) = \Big(\frac{2}{j+1} + \sum_{i=2}^{n} \frac{1}{j+i}\Big)$$

Clearly, this problem has the unique optimum solution $\overline{x} = (1, 1, \ldots, 1)^T$ and the dual problem has the unique optimum solution $\overline{\pi} = (2, 1, 1, \ldots, 1)$. The coefficient matrix A is related to the Hilbert matrix of order n. Verify that when this problem is solved by pivotal algorithms such as the simplex algorithm, or by the complementary pivot algorithm through an LCP formulation, using finite precision arithmetic, the results obtained are very bad, if n exceeds 10, say.
(E. Bernarczuk, "On the results of solving some linear programming problems using program packages of IBM and Robotron computers")

1.40 Consider the LCP (q, M). Define

$$f(z) = \sum_{i=1}^{n}[\text{minimum } \{0, M_{i.}z + q_i - z_i\} + z_i].$$

Show that the LCP (q, M) is equivalent to the following concave minimization problem

$$\begin{array}{ll} \text{minimize} & f(z) \\ \text{subject to} & Mz+q \geqq 0 \\ & z \geqq 0. \end{array}$$

(O. L. Mangasarian [8.15])

1.41 Let n be a positive integer. Consider a square matrix $x = (x_{ij})$ of order n. Order the entries x_{ij} in the matrix in the form of a vector in $\mathbf{R}^{n^2}$, in some order. Let $\mathbf{K} \subset \mathbf{R}^{n^2}$ denote the set of all such vectors corresponding to PSD matrices x. Prove that $\mathbf{K}$ is a convex cone, but not polyhedral, and has a nonempty interior.

1.42 Consider the LCP (q, M) (1.6) to (1.8), of order n. Now consider the following mixed 0-1 integer programming problem (MIP)

$$\begin{array}{ll} \text{maximize} & y_{n+1} \\ \text{subject to} & 0 \leqq My + qy_{n+1} \leqq e - x \\ & 0 \leqq y \leqq x, 0 \leqq y_{n+1} \leqq 1 \\ & x_i = 0 \text{ or } 1 \text{ for all } i = 1 \text{ to } n \end{array} \tag{1.49}$$

where $y = (y_1, \dots, y_n)^T$, $x = (x_1, \dots, x_n)^T$ and e is the vector of all 1s in $\mathbf{R}^n$. Suppose the optimum objective value in the MIP (1.49) is y^*_{n+1}.

If $y^*_{n+1} = 0$, prove that the LCP (q, M) has no solution.

If $y^*_{n+1} > 0$ and (y^*, x^*, y^*_{n+1}) is any optimum solution of the MIP (1.49), prove that (w^*, z^*) is a solution of the LCP (q, M), where

$$z^* = (1/y^*_{n+1})y^*$$
$$w^* = Mz^* + q$$

(J. Ben Rosen, "Solution of general LCP by 0-1 Mixed integer programming", Computer Science Tech. Report 86-23, University of Minnesota, Minneapolis, May, 1986).

1.8 References

1.1 R. Chandrasekaran, "The Weighted Euclidean 1-Center Problem", *O. R. Letters*, 1 (1982) 111–112.

1.2 R. Chandrasekaran and M. J. A. P. Pacca, "Weighted Min-Max, Max-Min Location Problems: Finite and Polynomially Bounded Algorithms", *Opsearch*, 17 (1980) 172–180.

1.3 R. W. Cottle and G. B. Dantzig, "Complementary Pivot Theory of Mathematical Programming", *Linear Algebra and its Applications*, 1 (1968) 103–125.

1.4 O. DeDonato and G. Maier, "Mathematical Programming Methods for the Inelastic Analysis of Reinforced Concrete Frames Allowing for Limited Rotation Capacity", *International Journal for Numerical Methods in Engineering*, 4 (1972) 307–329.

1.5 J. Elzinga and D. W. Hearn, "The Minimum Covering Sphere Problem", *Management Science*, 19 (1972) 96–104.

1.6 M. L. Fisher and F. J. Gould, "A Simplicial Algorithm for the Nonlinear Complementarity Problem", *Mathematical Programming*, 6 (1974) 281-300.

1.7 O. Fujiwara, S. P. Han and O. L. Mangasarian, "Local Duality of Nonlinear Programs", *SIAM Journal on Control and Optimization*, 22 (1984) 162—169.

1.8 R. F. Giese and P. C. Jones, "An Economic Model of Short-Rotation Forestry", *Mathematical Programming*, 28 (1984) 206–217.

1.9 S. P. Han and O. L. Mangasarian, "Characterization of Positive Definite and Semidefinite Matrices via Quadratic Programming Duality", *SIAM Journal of Algebraic and Discrete Methods*, 5 (1984) 26–32.

1.10 S. P. Han and O. L. Mangasarian, "Conjugate Cone Characterization of Positive and Semidefinite Matrices", *Linear Algebra and Its Applications*, (to appear).

1.11 M. R. Hestenes, *Optimization Theory, The Finite Dimensional Case,* Wiley-Interscience, 1975.

1.12 I. Kaneko, "A Mathematical Programming Method for the Inelastic Analysis of Reinforced Concrete Frames", *International Journal for Numerical Methods in Engineering*, 11 (1977) 1137–1154.

1.13 I. Kaneko, "Piecewise Linear Elastic-Plastic Analysis", *International Journal for Numerical Methods in Engineering*, 14 (1979) 757–767.

1.14 S. Karamardian, "The Nonlinear Complementarity Problem With Applications, Parts I and II", *Journal of Optimization Theory and Applications*, 4 (1969) 87–98, 167–181.

1.15 S. Karamardian, "Generalized Complementarity Problem", *Journal of Optimization Theory and Applications*, 8 (1971) 161–168.

1.16 S. Karamardian, "The Complementarity Problem", *Mathematical Programming*, 2 (1972) 107–129.

1.17 G. J. Kolhler, "A Complementarity Approach for Solving Leontief Substitution Systems and (Generalized) Markov Decision Processes", *RAIRO Rochereche Operationnelle/Operations Research*, 13 (1979) 75–80.

1.18 C. E. Lemke and J. T. Howson, "Equilibrium Points of Bimatrix Games", *SIAM Journal of Applied Mathematics*, 12 (1964) 413–423.

1.19 G. Maier, "A Quadratic Programming Approach for Certain Classes of Nonlinear Structural Problem", *Meccanica*, 3 (1968) 121–130.

1.20 A. Majthay, "Optimality Conditions for Quadratic Programming", *Mathematical Programming*, 1 (1971) 359–365.

1.21 H. M. Markowitz, "Portfolio Selection", *The Journal of Finance*, 12 (1952) 77–91.

1.22 C. MarMolinero, "Negative Chickens: An Exercise in Model Building", *Mathematical Spectrum*, 11 (1978-79) 13–18.

1.23 C. McLinden, "An Analogue of Moreau's Approximation Theorem with Application to the Nonlinear Complementarity Problem", *Pacific Journal of Mathematics*, 88 (1980) 101—161.

1.24 N. Megiddo and M. Kojima, "On One Existence and Uniqueness of Solutions in Nonlinear Complementarity Theory", *Mathematical Programming*, 12 (1977) 110–130.

1.25 J. J. Moré, "Classes of Functions and Feasibility Conditions in Nonlinear Complementarity Problems", *Mathematical Programming*, 6 (1974) 327–338.

1.26 K. G. Murty, "On the Number of Solutions of the Complementarity Problem and Spanning Properties of Complementary Cones", *Linear Algebra and Its Applications*, 5 (1972) 65–108.

1.27 K. G. Murty, Article on "Complementarity Problems", in: J. Belzer, A. G. Holzman and A. Kent (editors), *Encyclopedia of Computer Science and Technology*, Marcel Dekker, 1976.

1.28 K. G. Murty, *Linear and Combinatorial Programming,* Krieger Publishing Company, Malabar, Florida, USA, 1976.

1.29 K. P. K. Nair and R. Chandrasekaran, "Optimal Location of a Single Service Center of Certain Types", *Naval Research Logistics Quarterly*, 18 (1971) 503–510.

1.30 W. Oettli, "An Iterative Method Having Linear Rate of Convergence, for Solving a Pair of Dual Linear Programs", *Mathematical Programming*, 3 (1972) 302–311.

1.31 G. Owen, *Game Theory,* 2nd Edition, Academic Press, New York, 1982.

1.32 J. S. Pang, I. Kaneko and W. P. Hallman, "On the Solution of Some (Parametric) Linear Complementarity Problems With Applications to Portfolio Selection, Structural Engineering and Acturial Graduation", *Mathematical Programming*, 16 (1979) 325–347.

1.33 J. S. Pang, "A Parametric Linear Complementarity Technique for Optimal Portfolio Selection With a Risk-Free Asset", *Operations Research*, 28 (1980) 927–941.

1.34 J. S. Pang, "Linear Complementarity Problems Solvable by an Efficient Polynomially Bounded Pivoting Algorithm", Technical Report, MRC, University of Wisconsin, Madison, Wisconsin, 1983.

1.35 A. A. Raimondi and J. Boyd, "A Solution for the Finite Journal Bearing and Its Application to Analysis and Design", III, *Transactions of the American Society of Lubrication Engineers*, 1 (1958) 194–209.

1.36 J. R. Rice, *Matrix Computations and Mathematical Software,* McGraw-Hill, 1981.

1.37 R. W. H. Sargent, "An Efficient Implementation of the Lemke Algorithm and Its Extension to Deal with Upper and Lower Bounds", *Mathematical Programming Study*, 7 (1978) 36–54.

1.38 H. Tuy, "Minimization of a Concave Function Over an Unbounded Polyhedral Convex Set. Application to the LCP", Institute of Mathematics, Hanoi (1981).

1.39 J. W. Tolle and M. L. Fisher, "The Nonlinear Complementarity Problem: Existence and Determination of Solutions", *SIAM Journal on Control and Optimization*, 15 (1977) 612–624.

1.40 F. C. White, "Trade-Off in Growth and Stability in State Taxes", *National Tax Journal*, 36 (1983) 103–114.

References on Recursive (or Sequential) Quadratic Programming Methods for Nonlinear Programming

1.41 D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods,* Academic Press, New York, 1982.

1.42 M. C. Biggs, "On the Convergence of Some Constrained Minimization Algorithms Based on Recursive Quadratic Programming", *Journal of the Institute of Mathematics and Its Applications*, 21 (1978) 67–82.

1.43 U. M. Garcia-Palomares and O. L. Mangasarian, "Superlinearly Convergent Quasi-Newton Algorithms for Nonlinearly Constrained Optimization Problems", Computer Science Technical Report No. 195, University of Wisconsin, Madison, Wisconsin, 1974.

1.44 S. P. Han, "Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems", *Mathematical Programming*, 11 (1976) 263–282.

1.45 S. P. Han, "A Globally Convergent Method for Nonlinear Programming", *Journal of Optimization Theory and Its Applications*, 22 (1977) 297–309.

1.46 M. J. D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", in: G. A. Watson (editor), *Numerical Analysis, Lecture Notes in Mathematics*, Vol. 630, Springer Verlag, New York, 1978.

1.47 M. J. D. Powell, "The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations", in: O. L. Mangasarian, R. R. Meyer and S. M. Robinson (editors), *Nonlinear Programming 3*, Academic Press, New York, 1978.

1.48 B. N. Pshenichny and Y. M. Danilin, *Numerical Methods in Extremal Problems*, (translated from Russian by V. Zhitomirsky), MIR Publishers, Moscow, 1978.

1.49 S. M. Robinson, "A Quadratically Convergent Algorithm for General Nonlinear Programming Problems", *Mathematical Programming*, 3 (1972) 145–156.

1.50 K. Schittkowski, "The Nonlinear Programming Method of Wilson, Han and Powell with an Augmented Lagrangian Type Line Search Function, Part I: Convergence Analysis and Part 2: An Efficient Implementation with Linear Least Squares Subproblems", *Numerische Mathematik*, 38 (1981) 83–114 and 115–127.

1.51 K. Schittkowski, "On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function", Technical Report SOL 82-4, Department of Operations Research, Stanford University, Stanford, California, 1982.

1.52 R. A. Tapia, "Quasi-Newton Methods for Equality Constrained Optimization: Equivalents of Existing Methods and New Implementation", in: O. Mangasarian, R. Meyer and S. M. Robinson (editors), *Symposium on Nonlinear Programming III*, Academic Press, New York, 1978, pp. 125–164.

1.53 K. Tone, "Revisions of Constraint Approximations in the Successive QP method for Nonlinear Programming Problems", *Mathematical Programming*, 26 (1983) 144–152.

1.54 R. B. Wilson, "A Simplicial Algorithm for Concave Programming", Ph. D. Dissertation, Graduate School of Business Administration, Harvard University, Boston, Massachusetts, 1963.