Chapter 2

THE COMPLEMENTARY PIVOT ALGORITHM AND ITS EXTENSION TO FIXED POINT COMPUTING

LCPs of order 2 can be solved by drawing all the complementary cones in the q_1, q_2-plane as discussed in Chapter 1.

Example 2.1

Let $q = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$, $M = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$ and consider the LCP (q, M). The class of complementary cones corresponding to this problem is shown in Figure 1.5.

w_1	w_2	z_1	z_2	q
1	0	2	-1	4
0	1	-1	2	-1

$$w_1, w_2, z_1, z_2 \geqq 0, \quad w_1 z_1 = w_2 z_2 = 0$$

q lies in two complementary cones Pos $(-M_{.1}, I_{.2})$ and Pos $(-M_{.1}, -M_{.2})$. This implies that the sets of usable variables (z_1, w_2) and (z_1, z_2) lead to solutions of the LCP.

Putting $w_1 = z_2 = 0$ and solving the remaining system for the values of the usable variables (z_1, w_2) lead to the solution $(z_1, w_2) = (2, 1)$. Here $(w_1, w_2, z_1, z_2) = (0, 1, 2, 0)$ is a solution of this LCP. Similarly putting $w_1 = w_2 = 0$ and solving it for the value of the usable variables (z_1, z_2) leads to the second solution $(w_1, w_2, z_1, z_2) = (0, 0, \frac{7}{3}, \frac{2}{3})$ of this LCP.

Example 2.2

Let $q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ and $M = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$ and consider the LCP (q, M). The class of complementary cones corresponding to this problem is in Figure 1.5. Verify that q is not contained in any complementary cone. Hence this LCP has no solution.

This graphic method can be conveniently used only for LCPs of order 2. In LCPs of higher order, in contrast to the graphic method where all the complementary cones were generated, we seek only one complementary cone in which q lies. In this chapter we discuss the **complementary pivot algorithm** (which is also called the **complementary pivot method**) for solving the LCP. In the LCP (1.1) if $q \geqq 0$, $(w, z) = (q, 0)$ is a solution and we are done. So we assume $q \not\geqq 0$. First we will briefly review some concepts from linear programming. See [2.26] for complete details.

2.1 BASES AND BASIC FEASIBLE SOLUTIONS

Consider the following system of linear equality constraints in nonnegative variables

$$\begin{aligned} Ax &= b \\ x &\geqq 0 \end{aligned} \tag{2.1}$$

where A is a given matrix of order $m \times n$. Without any loss of generality we assume that the rank of A is m (otherwise either (2.1) is inconsistent, or redundant equality constraints in (2.1) can be eliminated one by one until the remaining system satisfies this property. See [2.26]). In this system, the variable x_j is associated with the column $A_{.j}$, $j = 1$ to n. A **basis** B for (2.1) is a square matrix consisting of m columns of A which is nonsingular; and the column vector of variables x_B associated with the columns in B, arranged in the same order, is the **basic vector** corresponding to it. Let D be the matrix consisting of the $n - m$ columns of A not in B, and let x_D be the vector of variables associated with these columns. When considering the basis B for (2.1), columns in B, D, are called the **basic, nonbasic columns** respectively; and the variables in x_B, x_D are called the **basic, nonbasic variables** respectively. Rearranging the variables, (2.1) can be written in partitioned form as

$$\begin{aligned} Bx_B \ + \ Dx_D \ &= \ b \\ x_B \geqq 0, \qquad x_D &\geqq 0 \ . \end{aligned}$$

The **basic solution** of (2.1) corresponding to the basis B is obtained by setting $x_D = 0$ and then solving the remaining system for the values of the basic variables. Clearly it is $(x_B = B^{-1}b,\ x_D = 0)$. This solution is feasible to (2.1) iff $B^{-1}b \geqq 0$, and in this case B is said to be a **feasible basis** for (2.1) and the solution $(x_B = B^{-1}b,\ x_D = 0)$

is called the **basic feasible solution** (abbreviated as BFS) of (2.1) corresponding to it. A basis B which is not feasible (i. e., if at least one component of $B^{-1}b$ is strictly negative) is said to be an **infeasible basis** for (2.1). Thus each feasible basis B for (2.1) determines a unique BFS for it.

When referring to systems of type (2.1), the word **solution** refers to a vector x satisfying the equality constraints '$Ax = b$', that may or may not satisfy the nonnegativity restrictions '$x \geqq 0$'. A solution x of (2.1) is a **feasible solution** if it satisfies $x \geqq 0$.

Definition: Degeneracy, Nondegeneracy of Basic Solutions for (2.1); of (2.1) itself: and of the b-Vector in (2.1) The basic solution associated with a given basis B for (2.1), whether it is feasible or not, is said to be **degenerate** if at least one component in the vector $B^{-1}b$ is zero, **nondegenerate** otherwise.

A system of constraints of the form (2.1) is said to be nondegenerate if it has no degenerate basic solutions (i. e., iff in every solution of (2.1), at least m variables are nonzero, when the rank of A is m), degenerate otherwise. When A has full row rank, the system (2.1) is therefore degenerate iff the column vector b can be expressed as a linear combination of r columns of A, where $r < m$, nondegenerate otherwise. Thus whether the system of constraints (2.1) is degenerate or nondegenerate depends on the position of the right hand side constants vector b in $\mathbf{R}^m$ in relation to the columns of A; and if the system is degenerate, it can be made into a nondegenerate system by just perturbing the b-vector alone.

The right hand side constants vector b in the system of constraints (2.1) is said to be degenerate or nondegenerate in (2.1) depending on whether (2.1) is degenerate or nondegenerate. See Chapter 10 in [2.26].

The definitions given here are standard definitions of degeneracy, nondegeneracy that apply to either a system of constraints of the form (2.1) or the right hand constants vector b in such a system, or a particular basic solution of such a system. This should not be confused with the concepts of (principal) degeneracy or (principal) nondegeneracy of square matrices defined later on in Section 2.3, or the degeneracy of complementary cones defined in Chapter 1.

As an example, consider the system of constraints given in Example 2.4 in Section 2.2.2. The BFS of this system associated with the basic vector (x_1, x_2, x_3, x_4) is $\bar{x} = (3, 0, 6, 5, 0, 0, 0, 0)^T$ and it is degenerate since the basic variable x_2 is zero in this solution. The BFS of this system associated with the basic vector (x_8, x_2, x_3, x_4) can be verified to be $x = (0, 1, 2, 3, 0, 0, 0, 1)^T$ which is a nondegenerate BFS. Since the system has a degenerate basic solution, this system itself is degenerate, also the b-vector is degenerate in this system.

Definition: Lexico Positive A vector $a = (a_1, \dots, a_r) \in \mathbf{R}^r$, is said to be **lexico positive**, denoted by $a \succ 0$, if $a \neq 0$ and the first nonzero component in a is strictly positive. A vector a is lexico negative, denoted by $a \prec 0$, if $-a \succ 0$. Given two vectors $x, y \in \mathbf{R}^r$, $x \succ y$ iff $x - y \succ 0$; $x \prec y$ iff $x - y \prec 0$. Given a set of vectors

$\{ a^1, \dots, a^k \} \subset \mathbf{R}^r$, a lexico minimum in this set is a vector a^j satisfying the property that $a^i \succeq a^j$ for each $i = 1$ to k. To find the lexico minimum in a given set of vectors from $\mathbf{R}^r$, compare the first component in each vector and discard all vectors not corresponding to the minimum first component, from the set. Compare the second component in each remaining vector and again discard all vectors not corresponding to the minimum in this position. Repeat in the same manner with the third component, and so on. At any stage if there is a single vector left, it is the lexico minimum. This procedure terminates after at most r steps. At the end, if two or more vectors are left, they are all equal to each other, and each of them is a lexico minimum in the set.

Example 2.3

The vector $(0, 0, 0.001, -1000)$ is lexico positive. The vector $(0, -1, 20000, 5000)$ is lexico negative. In the set of vectors $\{ (-2, 0, -1, 0), (-2, 0, -1, 1), (-2, 1, -20, -30), (0, -10, -40, -50) \}$, the vector $(-2, 0, -1, 0)$ is the lexico minimum.

Perturbation of the Right Hand Side Constants Vector in (2.1) to make it Nondegenerate.

If (2.1) is degenerate, it is possible to perturb the right hand side constants vector b slightly, to make it nondegenerate. For example, let ε be a parameter, positive and sufficiently small. Let $b(\varepsilon) = b + (\varepsilon, \varepsilon^{\mathbf{2}}, \dots, \varepsilon^{\mathbf{m}})^T$. It can be shown that if b in (2.1) is replaced by $b(\varepsilon)$, it becomes nondegenerate, for all ε positive and **sufficiently small** (this really means that there exists a positive number $\varepsilon_1 > 0$ such that whenever $0 < \varepsilon < \varepsilon_1$, the stated property holds). This leads to the perturbed problem

$$\begin{aligned} Ax &= b(\varepsilon) \\ x &\geqq 0 \end{aligned} \tag{2.2}$$

which is nondegenerate for all ε positive and sufficiently small. See Chapter 10 in [2.26] for a proof of this fact. A basis B and the associated basic vector x_B for (2.1) are said to be **lexico feasible** if they are feasible to (2.2) whenever ε is positive and sufficiently small, which can be verified to hold iff each row vector of the $m \times (m+1)$ matrix $(B^{-1}b \;\vdots\; B^{-1})$ is lexico positive. Thus lexico feasibility of a given basis for (2.1) can be determined by just checking the lexico positivity of each row of $(B^{-1}b \;\vdots\; B^{-1})$ without giving a specific value to ε. For example, if $b \geqq 0$, and A has the unit matrix of order m as a submatrix, that unit matrix forms a lexico feasible basis for (2.1).

Canonical Tableaus

Given a basis B, the canonical tableau of (2.1) with respect to it is obtained by multiplying the system of equality constraints in (2.1) on the left by B^{-1}. It is

Tableau 2.1 : Canonical Tableau of (2.1) with Respect to the Basis B

basic variables	x	
x_B	$B^{-1}A$	$B^{-1}b$

Let D be the matrix consisting of the $n - m$ columns of A not in B, and let x_D be the vector of variables associated with these columns. When the basic and nonbasic columns are rearranged in proper order, the canonical Tableau 2.1 becomes

Tableau 2.2

basic variables	x_B x_D	
x_B	I $B^{-1}D$	$B^{-1}b = \bar{b}$

$\bar{b}$ is known as the **updated right hand side constants vector** in the canonical tableau. The column of x_j in the canonical tableau, $B^{-1}A_{.j} = \bar{A}_{.j}$ is called the **update column** of x_j in the canonical tableau. The **inverse tableau** corresponding to the basis B is

Tableau 2.3 : Inverse Tableau

basic variables	Inverse	basic values
x_B	B^{-1}	$B^{-1}b$

It just provides the basis inverse and the updated right-hand-side constants column. From the information available in the inverse tableau, the update column corresponding to any nonbasic variable in the canonical tableau can be computed using the formulas given above.

2.2 THE COMPLEMENTARY PIVOT ALGORITHM

We will now discuss a pivotal algorithm for the LCP introduced by C. E. Lemke, known as the **Complementary Pivot Algorithm** (because it chooses the entering variable by a **complementary pivot rule**, the entering variable in a step is always the complement of the dropping variable in the previous step), and also referred to as **Lemke's Algorithm** in the literature.

2.2.1 The Original Tableau

An artificial variable z_0 associated with the column vector $-e_n$ (e_n is the column vector of all 1's in $\mathbf{R}^n$) is introduced into (1.6) to get a feasible basis for starting the algorithm. In detached coefficient tableau form, (1.6) then becomes

w	z	z_0	
I	$-M$	$-e_n$	q

$$w \geqq 0, \quad z \geqq 0, \quad z_0 \geqq 0 \tag{2.3}$$

2.2.2 Pivot Steps

The complementary pivot algorithm moves among feasible basic vectors for (2.3). The primary computational step used in this algorithm is the pivot step (or the **Gauss-Jordan pivot step**, or the Gauss-Jordan elimination pivot step), which is also the main step in the simplex algorithm for linear programs. In each stage of the algorithm, the basis is changed by bringing into the basic vector exactly one nonbasic variable known as the **entering variable**. Its updated column vector is the **pivot column** for this basis change. The **dropping variable** has to be determined according to the **minimum ratio test** to guarantee that the new basis obtained after the pivot step will also be a feasible basis.

For example, assume that the present feasible basic vector is $(y_1, \ldots, y_n)$ with y_r as the r^{th} basic variable, and let the entering variable be x_s. (The variables in (2.3) are $w_1, \ldots, w_n$; $z_1, \ldots, z_n$, z_0. Exactly n of these variables are present basic variables. For convenience in reference, we assume that these basic variables are called $y_1, \ldots, y_n$). After we rearrange the variables in (2.3), if necessary, the canonical form of (2.3), with respect to the present basis is of the form :

Basic variable	$y_1, \ldots, y_n$	x_s	Other variables	Right-hand constant vector
y_1	$1 \ldots 0$	$\bar{a}_{1s}$	$\cdots$	$\bar{q}_1$
$\vdots$	$\vdots \quad \vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	$0 \ldots 1$	$\bar{a}_{ns}$	$\cdots$	$\bar{q}_n$

Keeping all the nonbasic variables other than x_s, equal to zero, and giving the value λ to the entering variable, x_s, leads to the new solutions :

$$\begin{aligned} & x_s = \lambda \\ & y_i = \bar{q}_i - \lambda \bar{a}_{is}, \quad i = 1, \ldots, n \\ & \text{All other variables} = 0 \end{aligned} \tag{2.4}$$

There are two possibilities here.

1. The pivot column may be nonpositive, that is, $\bar{a}_{is} \leqq 0$ for all $1 \leqq i \leqq n$. In this case, the solution in (2.4) remains nonnegative for all $\lambda \geqq 0$. As λ varies from 0 to ∞, this solution traces an **extreme half-line** (or an **unbounded edge**) of the set of feasible solutions of (2.3). In this case the minimum ratio, θ, in this pivot step is $+\infty$. See Example 2.4.
2. There is at least one positive entry in the pivot column. In this case, if the solution in (2.4) should remain nonnegative, the maximum value that λ can take is $\theta = \frac{\bar{q}_r}{\bar{a}_{rs}} = \text{minimum}\{\frac{\bar{q}_i}{\bar{a}_{is}} : i \text{ such that } \bar{a}_{is} > 0\}$. This θ is known as the **minimum ratio** in this pivot step. For any i that attains the minimum here, the present i^{th} basic variable y_i is eligible to be the dropping variable from the basic vector in this pivot step. The dropping basic variable can be chosen arbitrarily among those eligible, suppose it is y_r. y_r drops from the basic vector and x_s becomes the r^{th} basic variable in its place. The r^{th} row is the pivot row for this pivot step. The pivot step leads to the canonical tableau with respect to the new basis.

If the pivot column $(\bar{a}_{1s}, \dots, \bar{a}_{ms})^T$ is placed by the side of the present inverse tableau and a pivot step performed with the element $\bar{a}_{rs}$ in it in the pivot row as the pivot element, the inverse tableau of the present basis gets transformed into the inverse tableau for the new basis.

The purpose of choosing the pivot row, or the dropping variable, by the minimum ratio test, is to guarantee that the basic vector obtained after this pivot step remains feasible.

In this case (when there is at least one positive entry in the pivot column) the pivot step is said to be a nondegenerate pivot step if the minimum ratio computed above is > 0, degenerate pivot step if it is 0. See Examples 2.5, 2.6.

Let B be the basis for (2.3) corresponding to the basic vector $(y_1, \dots, y_n)$. As discussed above, the basic vector $(y_1, \dots, y_n)$ is lexico feasible for (2.3) if each row vector of $(\bar{q} \vdots B^{-1})$ is lexico positive. If the initial basic vector $(y_1, \dots, y_n)$ is lexico feasible, lexico feasibility can be maintained by choosing the pivot row according to the **lexico minimum ratio test**. Here the pivot row is chosen as the r^{th} row where r is the i that attains the lexico minimum in $\{\frac{(\bar{q}_i, \beta_{i1}, \dots, \beta_{in})}{\bar{a}_{is}} : i \text{ such that } \bar{a}_{is} > 0\}$, where $\beta = (\beta_{ij}) = B^{-1}$. The lexico minimum ratio test identifies the pivot row (and hence the dropping basic variable) unambiguously, and guarantees that lexico feasibility is maintained after this pivot step. In the simplex algorithm for linear programming, the lexico minimum ratio test is used to guarantee that cycling will not occur under degeneracy (see Chapter 10 of [2.26]). The lexico minimum ratio test is one of the rules that can be used to resolve degeneracy in the simplex algorithm, and thus guarantee that it terminates in a finite number of pivot steps.

Example 2.4 Extreme Half-line

Consider the following canonical tableau with respect to the basic vector (x_1, x_2, x_3, x_4).

basic variables	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b
x_1	1	0	0	0	1	−1	2	3	3
x_2	0	1	0	0	1	−2	1	−1	0
x_3	0	0	1	0	−1	0	5	4	6
x_4	0	0	0	1	−1	−3	8	2	5

$x_j \geqq 0$ for all j.

Suppose x_6 is the entering variable. The present BFS is $\bar{x} = (3, 0, 6, 5, 0, 0, 0, 0)^T$. The pivot column $(-1, -2, 0, -3)^T$ has no positive entry. Make the entering variable equal to λ, retain all other nonbasic variables equal to 0, this leads to the solution $x(\lambda) = (3+\lambda, 2\lambda, 6, 5+3\lambda, 0, \lambda, 0, 0)^T = \bar{x} + \lambda x^h$, where $x^h = (1, 2, 0, 3, 0, 1, 0, 0)^T$. x^h, the coefficient vector of λ in $x(\lambda)$, is obtained by making the entering variable equal to 1, all other nonbasic variables equal to zero, and each basic variable equal to the negative of the entry in the pivot column in its basic row. Since the pivot column is nonpositive here, $x^h \geqq 0$. it can be verified that x^h satisfies the homogeneous system obtained by replacing the right hand side constants vector by 0. Hence x^h is known as a **homogeneous solution** corresponding to the original system. Since $x^h \geqq 0$ here, $x(\lambda)$ remains $\geqq 0$ for all $\lambda \geqq 0$. The half-line $\{\, \bar{x} + \lambda x^h : \lambda \geqq 0 \,\}$ is known as an **extreme half-line** of the set of feasible solutions of the original system.

A half-line is said to be a **feasible half-line** to a system of linear constraints, if every point on the half-line is feasible to the system.

Example 2.5 Nondegenerate Pivot Step

See Tableau 2.4 in Example 2.8 of Section 2.2.6 a few pages ahead. This is the canonical tableau with respect to the basic vector (w_1, w_2, z_0, w_4) and z_3 is the entering variable. The minimum ratio occurs uniquely in row 4, which is the pivot row in this step, and w_4 is the dropping variable. Performing the pivot step leads to the canonical tableau with respect to the new basic vector (w_1, w_2, z_0, z_3) in Tableau 2.5. This is a nondegenerate pivot step since the minimum ratio in it was $(\frac{4}{2}) > 0$. As a result of this pivot step the BFS has changed from $(w_1, w_2, w_3, w_4; z_1, z_2, z_3, z_4; z_0) = (12, 14, 0, 4; 0, 0, 0, 0; 9)$ to $(6, 8, 0, 0; 0, 0, 2, 0; 5)$.

Example 2.6 Degenerate Pivot Step

Consider the following canonical tableau :

Basic variable	x_1	x_2	x_3	x_4	x_5	x_6	$\bar{b}$	Ratio
x_1	1	0	0	1	2	−3	3	$\frac{3}{1}$
x_2	0	1	0	[1]	−2	1	0	$\frac{0}{1}$ *Min.*
x_3	0	0	1	−1	1	2	0	

$$x_j \geqq 0 \text{ for all } j.$$

Here the BFS is $\bar{x} = (3, 0, 0, 0, 0, 0)^T$. It is degenerate. If x_4 is chosen as the entering variable, it can be verified that the minimum ratio of 0 occurs in row 2. Hence row 2 is the pivot row for this step, and x_2 is the dropping variable. Performing the pivot step leads to the canonical tableau with respect to the new basic vector (x_1, x_4, x_3).

basic variable	x_1	x_2	x_3	x_4	x_5	x_6	
x_1	1	−1	0	0	4	−4	3
x_4	0	1	0	1	−2	1	0
x_3	0	1	1	0	−1	3	0

Eventhough the basic vector has changed, the BFS has remained unchanged through this pivot step. A pivot step like this is called a **degenerate pivot step**.

A **pivot step** is **degenerate**, if the minimum ratio θ in it is 0, **nondegenerate** if the minimum ratio is positive and finite. In every pivot step the basic vector changes by one variable. In a degenerate pivot step there is no change in the correponding BFS (the entering variable replaces a zero valued basic variable in the solution). In a nondegenerate pivot step the BFS changes.

Example 2.7 Ties for Minimum Ratio lead to Degenerate Solution

Consider the following canonical tableau.

basic variable	x_1	x_2	x_3	x_4	x_5	x_6	$\bar{b}$	Ratio
x_1	1	0	0	1	−2	1	3	$\frac{3}{1}$
x_2	0	1	0	2	1	1	6	$\frac{6}{2}$
x_3	0	0	1	2	1	−2	16	$\frac{16}{2}$

The present BFS is $\bar{x} = (3, 6, 16, 0, 0, 0)^T$. Suppose x_4 is chosen as the entering variable. There is a tie for the minimum ratio. Both x_1, x_2 are eligible to be dropping variables. Irrespective of which of them is chosen as the dropping variable, it can be verified that the other remains a basic variable with a value of 0 in the next BFS. So the BFS obtained after this pivot step is degenerate.

In the same way it can be verified that the BFS obtained after a pivot step is always degenerate, if there is a tie for the minimum ratio in that step. Thus, if we know that the right hand side constants vector q is nondegenerate in (2.3), in every pivot step performed on (2.3), the minimum ratio test identifies the dropping variable uniquely and unambiguously.

2.2.3 Initialization

The artificial variable z_0 has been introduced into (2.3) for the sole purpose of obtaining a feasible basis to start the algorithm.

Identify row t such that q_t = minimum $\{ q_i : 1 \leqq i \leqq n \}$. Break ties for t in this equation arbitrarily. Since we assumed $q \not\geqq 0$, $q_t < 0$. When a pivot is made in (2.3) with the column vector of z_0 as the pivot column and the t^{th} row as the pivot row, the right-hand side constants vector becomes a nonnegative vector. The result is the canonical tableau with respect to the basic vector $(w_1, \dots, w_{t-1}, z_0, w_{t+1}, \dots, w_n)$. This is the initial basic vector for starting the algorithm.

2.2.4 Almost Complementary Feasible Basic Vectors

The initial basic vector satisfies the following properties :

(i) There is at most one basic variable from each complementary pair of variables (w_j, z_j).

(ii) It constains exactly one basic variable from each of $(n - 1)$ complementary pairs of variables, and both the variables in the remaining complementary pair are nonbasic.

(iii) z_0 is a basic variable in it.

A feasible basic vector for (2.3) in which there is exactly one basic variable from each complementary pair (w_j, z_j) is known as a **complementary feasible basic vector**. A feasible basic vector for (2.3) satisfying properties (i), (ii), and (iii) above is known as an **almost complementary feasible basic vector**. Given an almost complementary feasible basic vector for (2.3), the complementary pair both of whose variables are nonbasic, is known as the **left-out complementary pair of variables** in it. All the basic vectors obtained in the algorithm with the possible exception of the final basic vector are almost complementary feasible basic vectors. If at some stage of the algorithm, a complementary feasible basic vector is obtained, it is a final basic vector and the algorithm terminates.

Adjacent Almost Complementary Feasible Basic Vectors

Let $(y_1, \ldots, y_{j-1}, z_0, y_{j+1}, \ldots, y_n)$ be an almost complementary feasible basic vector for (2.3), where $y_i \in \{w_i, z_i\}$ for each $i \neq j$. Both the variables in the complementary pair (w_j, z_j) are not in this basic vector. Adjacent almost complementary feasible basic vectors can only be obtained by picking as the entering variable either w_j or z_j. Thus from each almost complementary feasible basic vector there are exactly two possible ways of generating adjcent almost complementary feasible basic vectors.

In the initial almost complementary feasible basic vector, both w_t and z_t are nonbasic variables. In the canonical tableau with respect to the initial basis, the updated column vector of w_t can be verified to be $-e_n$, which is negative. Hence, if w_t is picked as the entering variable into the initial basic vector, an extreme half-line is generated. Hence, the initial almost complementary BFS is at the end of an **almost complementary ray**.

So there is a unique way of obtaining an adjacent almost complementary feasible basic vector from the initial basic vector, and that is to pick z_t as the entering variable.

2.2.5 Complementary Pivot Rule

In the subsequent stages of the algorithm there is a unique way to continue the algorithm, which is to pick as the entering variable, the complement of the variable that just dropped from the basic vector. This is known as the **complementary pivot rule**.

The main property of the path generated by the algorithm is the following. Each BFS obtained in the algorithm has two almost complementary edges containing it. We arrive at this solution along one of these edges. And we leave it by the other edge. So the algorithm continues in a unique manner. It is also clear that a basic vector that was obtained in some stage of the algorithm can never reappear.

The path taken by the complementary pivot algorithm is illustrated in Figure 2.1. The initial BFS is that corresponding to the basic vector $(w_1, \ldots, w_{t-1}, z_0, w_{t+1}, \ldots, w_n)$ for (2.3). In Figure 2.1, each BFS obtained during the algorithm is indicated by a point, with the basic vector corresponding to it entered by its side; and consecutive BFSs are joined by an edge. If w_t is choosen as the entering variable into the initial basic vector we get an extreme half-line (discussed above) and the initial BFS is at end of this extreme half-line. When z_t is choosen as the entering variable into the initial basic vector, suppose w_i is the dropping variable. Then its complement z_i will be the entering variable into the next basic vector (this is the complementary pivot rule).

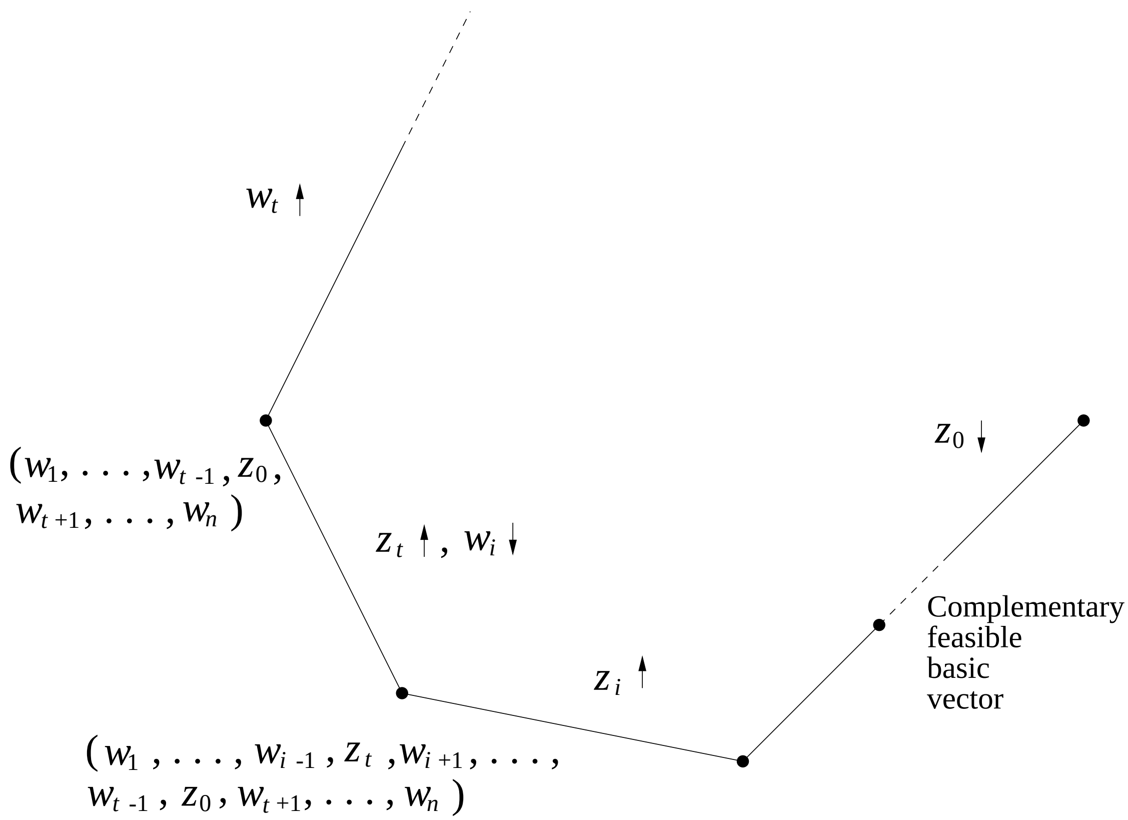

Figure 2.1 Path taken by the complementary pivot method. The ↑ indicates entering variable, ↓ indicates dropping variable. The basic vector corresponding to each point (BFS) is entered by its side. Finally if z_0 drops from the basic vector, we get a complementary feasible basic vector.

The path continues in this unique manner. It can never return to a basic vector visited earlier, since each BFS obtained in the algorithm has exactly two edges of the path incident at it, through one of which we arrive at that BFS and through the other we leave (if the path returns to a basic vector visited earlier, the BFS corresponding to it has three edges in the path incident at it, a contradiction). So the path must terminate after a finite number of steps either by going off along another extreme half-line at the end (**ray termination**, this happens when in some step, the pivot column, the updated column of the entering variable, has no positive entries in it), or by reaching a complementary feasible basic vector of the LCP (which happens when z_0 becomes the dropping variable). If ray termination occurs the extreme half-line obtained at the end, cannot be the same as the initial extreme half-line at the beginning of the path (this follows from the properties of the path discussed above, namely, that it never returns to a basic vector visited earlier).

2.2.6 Termination

There are exactly two possible ways in which the algorithm can terminate.

1. At some stage of the algorithm, z_0 may drop out of the basic vector, or become equal to zero in the BFS of (2.3). If $(\bar{w}, \bar{z}, \bar{z}_0 = 0)$ is the BFS of (2.3) at that stage, then $(\bar{w}, \bar{z})$ is a solution of the LCP (1.6) to (1.8).
2. At some stage of the algorithm, z_0 may be strictly positive in the BFS of (2.3), and the pivot column in that stage may turn out to be nonpositive, and in this case the algorithm terminates with another almost complementary extreme half-line, referred to in some publications as the **secondary ray** (distinct from the initial almost complementary extreme half-line or **initial ray** at the beginning of the algorithm). This is called **ray termination**.

When ray termination occurs, the algorithm is unable to solve the LCP. It is possible that the LCP (1.6) to (1.8) may not have a solution, but if it does have a solution, the algorithm is unable to find it. If ray termination occurs the algorithm is also unable to determine whether a solution to the LCP exists in the general case. However, when M satisfies some conditions, it can be proved that ray termination in the algorithm will only occur, when the LCP has no solution. See Section 2.3.

Problems Posed by Degeneracy of (2.3).

Definition: Nondegeneracy, or Degeneracy of q in the LCP (q, M) As defined earlier, the LCP (q, M) is the problem of finding w, z satisfying

w	z	
I	$-M$	q

$$w, z \geqq 0, \; w^T z = 0$$

This LCP is said to be nondegenerate (in this case q is said to be nondegenrate in the LCP (q, M)) if in every solution (w, z) of the system of linear equations "$w - Mz = q$", at least n variables are non-zero. This condition holds iff q cannot be expressed as a linear combination of $(n-1)$ or less column vectors of $(I \;\vdots\; -M)$.

The LCP (q, M) is said to be degenerate (in this case q is said to be degenerate in the LCP (q, M)) if q can be expressed as a linear combination of a set consisting of $(n-1)$ or less column vectors of $(I \;\vdots\; -M)$.

Definition: Nondegeneracy, Degeneracy of q in the Complementary Pivot Algorithm The system of contraints on which pivot operations are performed in the complementary pivot algorithm is (2.3). This system is said to be degenerate (and q is said to be degenerate in it) if q can be expressed as a linear combination of a set of $(n-1)$ or less column vectors of $(I \;\vdots\; -M \;\vdots\; -e)$; nondegenerate otherwise. If

(2.3) is nondegenerate, in every BFS of (2.3) obtained during the complementary pivot algorithm, all basic variables are strictly positive, and the minimum ratio test identifies the dropping basic variable in each pivot step uniquely and unambiguously.

The argument that each almost complementary feasible basis has at most two adjacent almost complementary feasible bases is used in developing the algorithm. This guarantees that the path taken by the algorithm continues unambiguously in a unique manner till termination occurs in one of the two possibilities. This property that each almost complementary feasible basis has at most two adjacent almost complementary feasible bases holds when (2.3) is nondegenerate. If (2.3) is degenerate, the dropping variable during some pivots may not be uniquely determined. In such a pivot step, by picking different dropping variables, different adjacent almost complementary feasible bases may be generated. If this happens, the almost complementary feasible basis in this step may have more than two adjacent almost complementary feasible bases. The algorithm can still be continued unambiguously according to the complementary pivot rule, but the path taken by the algorithm may depend on the dropping variables selected during the pivots in which these variables are not uniquely identified by the minimum ratio test. All the arguments mentioned in earlier sections are still valid, but in this case termination may not occur in a finite number of steps if the algorithm keeps cycling along a finite sequence of degenerate pivot steps. This can be avoided by using the concept of **lexico feasibility** of the solution. In this case the algorithm deals with **almost complementary lexico feasible bases** throughout. In each pivot step the lexico minimum ratio test determines the dropping variable unambiguously and, hence, each almost complementary lexico feasible basis can have at most two adjacent almost complementary lexico feasible bases. With this, the path taken by the algorithm is again unique and unambiguous, no cycling can occur and termination occurs after a finite number of pivot steps. See Section 2.2.8.

Interpretation of the Path Taken by the Complementary Pivot Algorithm

B. C. Eaves has given a simple haunted house interpretation of the path taken by the complementary pivot algorithm. A man who is afraid of ghosts has entered a haunted house from the outside through a door in one of its rooms. The house has the following properties :

- (i) It has a finite number of rooms.
- (ii) Each door is on a boundary wall between two rooms or on a boundary wall of a room on the outside.
- (iii) Each room may have a ghost in it or may not. However, every room which has a ghost has exactly two doors.

All the doors in the house are open initially. The man's walk proceeds according to the following property.

- (iv) When the man walks through a door, it is instantly sealed permanently and he can never walk back through it.

The man finds a ghost in the room he has entered initially, by properties (iii) and (iv) this room has exactly one open door when the man is inside it. In great fear he runs out of the room through that door. If the next room that he has entered has a ghost again, it also satisfies the property that it has exactly one open door when the man is inside it, and he runs out through that as fast as he can. In his walk, every room with a ghost satisfies the same property. He enters that room through one of its doors and leaves through the other. A **sanctuary** is defined to be either a room that has no ghost, or the outside of the house. The man keeps running until he finds a sanctuary. Property (i) guarantees that the man finds a sanctuary after running through at most a finite number of rooms. The sanctuary that he finds may be either a room without a ghost or the outside of the house.

We leave it to the reader to construct parallels between the ghost story and the complementary pivot algorithm and to find the walk of the man through the haunted house in Figure 2.2. The man walks into the house initially from the outside through the door marked with an arrow.

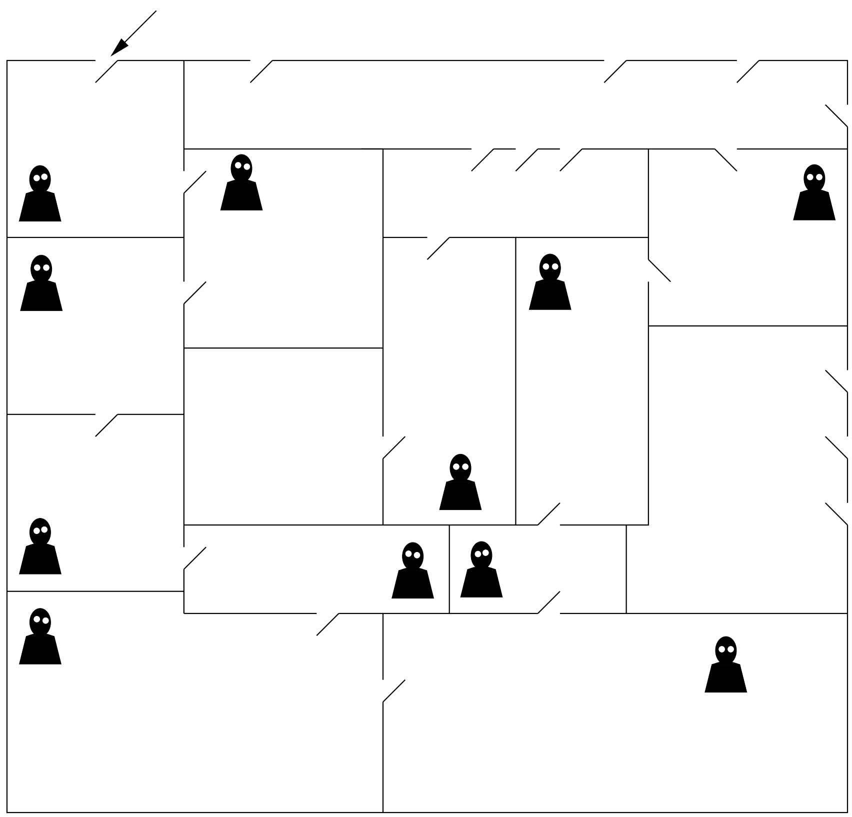

Figure 2.2 Haunted house

Geometric Interpretation of a Pivot Step in the Complementary Pivot Method

In a pivot step of the complementary pivot method, the current point moves between two facets of a complementary cone in the direction of $-e$. This geometric interpretation of a pivot step in the complementary pivot method as a walk between two facets of a complementary cone is given in Section 6.2.

Example 2.8

Consider the following LCP. (This is not an LCP corresponding to an LP.)

w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	q
1	0	0	0	−1	1	1	1	3
0	1	0	0	1	−1	1	1	5
0	0	1	0	−1	−1	−2	0	−9
0	0	0	1	−1	−1	0	−2	−5

$$w_i \geqq 0, \quad z_i \geqq 0, \quad w_i z_i = 0 \quad \text{for all } i$$

When we introduce the artificial variable z_0 the tableau becomes :

w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q
1	0	0	0	−1	1	1	1	−1	3
0	1	0	0	1	−1	1	1	−1	5
0	0	1	0	−1	−1	−2	0	$\boxed{-1}$	−9
0	0	0	1	−1	−1	0	−2	−1	−5

The most negative q_i is q_3. Therefore pivot in the column vector of z_0 with the third row as the pivot row. The pivot element is inside a box.

Tableau 2.4

Basic variables	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q	Ratios
w_1	1	0	−1	0	0	2	3	1	0	12	$\frac{12}{3}$
w_2	0	1	−1	0	2	0	3	1	0	14	$\frac{14}{3}$
z_0	0	0	−1	0	1	1	2	0	1	9	$\frac{9}{2}$
w_4	0	0	−1	1	0	0	$\boxed{2}$	−2	0	4	$\frac{4}{2}$ *Min.*

By the complementary pivot rule we have to pick z_3 as the entering variable. The column vector of z_3 is the pivot column, w_4 drops from the basic vector.

Tableau 2.5

Basic variables	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q	Ratios
w_1	1	0	$\frac{1}{2}$	$-\frac{3}{2}$	0	2	0	$\boxed{4}$	0	6	$\frac{6}{4}$ *Min.*
w_2	0	1	$\frac{1}{2}$	$-\frac{3}{2}$	2	0	0	4	0	8	$\frac{8}{4}$
z_0	0	0	0	-1	1	1	0	2	1	5	$\frac{5}{2}$
z_3	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	1	-1	0	2	

Since w_4 has dropped from the basic vector, its complement, z_4 is the entering variable for the next step. w_1 drops from the basic vector.

Basic variables	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q	Ratios
z_4	$\frac{1}{4}$	0	$\frac{1}{8}$	$-\frac{3}{8}$	0	$\frac{1}{2}$	0	1	0	$\frac{6}{4}$	
w_2	-1	1	0	0	$\boxed{2}$	-2	0	0	0	2	$\frac{2}{2}$ *Min.*
z_0	$-\frac{1}{2}$	0	$-\frac{1}{4}$	$-\frac{1}{4}$	1	0	0	0	1	2	$\frac{2}{1}$
z_3	$\frac{1}{4}$	0	$-\frac{3}{8}$	$\frac{1}{8}$	0	$\frac{1}{2}$	1	0	0	$\frac{14}{4}$	

Since w_1 has dropped from the basic vector, its complement, z_1 is the new entering variable. Now w_2 drops from the basic vector.

Basic variables	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q	Ratios
z_4	$\frac{1}{4}$	0	$\frac{1}{8}$	$-\frac{3}{8}$	0	$\frac{1}{2}$	0	1	0	$\frac{6}{4}$	3
z_1	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	1	-1	0	0	0	1	
z_0	0	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0	$\boxed{1}$	0	0	1	1	1 *Min.*
z_3	$\frac{1}{4}$	0	$-\frac{3}{8}$	$\frac{1}{8}$	0	$\frac{1}{2}$	1	0	0	$\frac{14}{4}$	7

Since w_2 has dropped from the basic vector, its complement, z_2 is the entering variable. Now z_0 drops from the basic vector.

Basic variables	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	q
z_4	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	0	1	$-\frac{1}{2}$	1
z_1	$-\frac{1}{2}$	0	$-\frac{1}{4}$	$-\frac{1}{4}$	1	0	0	0	1	2
z_2	0	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0	1	0	0	1	1
z_3	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	0	0	1	0	$-\frac{1}{2}$	3

Since the present basis is a complementary feasible basis, the algorithm terminates. The correponding solution of the LCP is $w = 0$, $(z_1, z_2, z_3, z_4) = (2, 1, 3, 1)$.

Example 2.9

---

w_1	w_2	w_3	z_1	z_2	z_3	q
1	0	0	1	0	3	−3
0	1	0	−1	2	5	−2
0	0	1	2	1	2	−1

$$w_i \geqq 0, \quad z_i \geqq 0, \quad w_i z_i = 0 \qquad \text{for all } i$$

The tableau with the artificial variable z_0 is :

w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
1	0	0	1	0	3	$\boxed{-1}$	−3
0	1	0	−1	2	5	−1	−2
0	0	1	2	1	2	−1	−1

The initial canonical tableau is :

Basic variables	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q	Ratios
z_0	−1	0	0	−1	0	−3	1	3	
w_2	−1	1	0	−2	2	2	0	1	
w_3	−1	0	1	$\boxed{1}$	1	−1	0	2	$\frac{2}{1}$

The next tableau is :

Basic variables	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
z_0	-2	0	1	0	1	-4	1	5
w_2	-3	1	2	0	4	0	0	5
z_1	-1	0	1	1	1	-1	0	2

The entering variable here is z_3. The pivot column is nonpositive. Hence, the algorithm stops here with ray termination. The algorithm has been unable to solve this LCP.

2.2.7 IMPLEMENTATION OF THE COMPLEMENTARY PIVOT METHOD USING THE INVERSE OF THE BASIS

Let (2.3) be the original tableau for the LCP being solved by the complementary pivot method. Let t be determined as in Section 2.2.3. After performing the pivot with row t as the pivot row and the column vector of z_0 as the pivot column, we get the **initial tableau** for this algorithm. Let P_0 be the pivot matrix of order n obtained by replacing the t^{th} column in I (the unit matrix of order n) by $-e_n$ (the column vector in $\mathbf{R}^n$ all of whose entires are -1). Let $M' = P_0 M$, $q' = P_0 q$. Then the initial tableau in this algorithm is

Tableau 2.6 : Initial Tableau

w	z	z_0	
P_0	$-M'$	$I_{.t}$	q'

The initial basic vector is $(w_1, \dots, w_{t-1}, z_0, w_{t+1}, \dots, w_n)$ and the basis corresponding to it in Tableau 2.6 is I. By choice of t, $q' \geqq 0$. So each row of $(q' \,\vdots\, I)$ is lexico-positive, and hence the initial basic vector in this algorithm is lexico-feasible for the problem in Tableau 2.6.

At some stage of the algorithm, let B be the basis from Tableau 2.6, corresponding to the present basic vector. Let $\beta = (\beta_{ij}) = B^{-1}$ and $\bar{q} = B^{-1} q'$. Then the inverse tableau at this stage is

Basic vector	Inverse	
	$\beta = B^{-1}$	$\bar{q}$

If the entering variable in this step, determined by the complementary pivot rule, is $y_s \in \{w_s, z_s\}$, then the pivot column, the updated column of y_s, is $\beta P_0 I_{.s}$ if $y_s = w_s$, or $\beta P_0(-M_{.s})$ if $y_s = z_s$. Suppose this pivot column is $(\bar{a}_{1s}, \ldots, \bar{a}_{ns})^T$. If $(\bar{a}_{1s}, \ldots, \bar{a}_{ns})^T \leq 0$, we have ray termination and the method has been unable to solve this LCP. If $(\bar{a}_{1s}, \ldots, \bar{a}_{ns})^T \not\leq 0$, the minimum ratio in this step is $\theta = \text{minimum}\left\{ \frac{\bar{q}_i}{\bar{a}_{is}} : i \text{ such that } \bar{a}_{is} > 0 \right\}$. If the i that attains this minimum is unique, it determines the pivot row uniquely. The present basic variable in the pivot row is the dropping variable. If the minimum ratio does not identify the dropping variable uniquely, check whether z_0 is eligible to drop, and if so choose it as the dropping variable. If z_0 is not eligible to drop, one of those eligible to drop can be choosen as the dropping variable arbitrarily, but this can lead to cycling under degeneracy. To avoid cycling, we can use the lexico-minimum ratio rule, which chooses the dropping basic variable so that the pivot row is the row corresponding to the lexico-minimum among $\left\{ \frac{(\bar{q}_i; \beta_{i1}, \ldots, \beta_{in})}{\bar{a}_{is}} : i \text{ such that } \bar{a}_{is} > 0 \right\}$. This lexico minimum ratio rule determines the dropping variable uniquely and unambiguously. If the lexico-minimum ratio rule is used in all steps beginning with the initial step, the dropping variable is identified uniquely in every step, each of the updated vectors $(\bar{q}_i; \beta_{i1}, \ldots, \beta_{in})$, $i = 1$ to n, remain lexico-positive throught, and cycling cannot occur by the properties of the almost complementary path generated by this method, discussed above (see Section 2.2.8). Once the dropping variable is identified, performing the pivot leads to the next basis inverse, and the entering variable in the next step is the complement of the dropping variable, and the method is continued in the same way.

Clearly it is not necessary to maintain the basis inverse explicitly. The complementary pivot algorithm can also be implemented with the basis inverse maintained in product form (PFI) or in elimination form (EFI) just as the simplex algorithm for linear programming (see Chapters 5, 7 of [2.26]).

Example 2.10

Consider the LCP (q, M) where

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix} \qquad q = \begin{pmatrix} -8 \\ -12 \\ -14 \end{pmatrix}$$

To solve this LCP by the complementary pivot algorithm, we introduce the artificial variable z_0 and construct the original tableau as in (2.3). When z_0 replaces w_3 in the basic vector (w_1, w_2, w_3), we get a feasible basic vector for the original tableau. So the initial tableau for this problem is :

Initial Basic Vector	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
w_1	1	0	−1	1	2	1	0	6
w_2	0	1	−1	0	1	1	0	2
z_0	0	0	−1	2	2	1	1	14

The various basis inverses obtained when this LCP is solved by the complementary pivot algorithm are given below.

Basic Vector	Inverse			$\bar{q}$	Pivot Column	Ratios
					z_3	
w_1	1	0	0	6	1	6
w_2	0	1	0	2	$\boxed{1}$	2 *Min.*
z_0	0	0	1	14	1	14
					z_2	
w_1	1	−1	0	4	1	4
z_3	0	1	0	2	$\boxed{1}$	2 *Min.*
z_0	0	−1	1	12	1	12
					w_3	
w_1	1	−2	0	2	$\boxed{1}$	2 *Min.*
z_2	0	1	0	2	−1	
z_0	0	−2	1	10	1	10
					z_1	
w_3	1	−2	0	2	$\boxed{1}$	2 *Min.*
z_2	1	−1	0	4	1	4
z_0	−1	0	1	8	1	8

Basic Vector	Inverse			$\bar{q}$	Pivot Column	Ratios
					z_3	
z_1	1	−2	0	2	−1	
z_2	0	1	0	2	$\boxed{1}$	2 *Min.*
z_0	−2	2	1	6	1	6
					w_2	
z_1	1	−1	0	4	−1	
z_3	0	1	0	2	$\boxed{1}$	2 *Min.*
z_0	−2	1	1	4	1	4
					w_3	
z_1	1	0	0	6	−1	
w_2	0	1	0	2	−1	
z_0	−2	0	1	2	$\boxed{1}$	2 *Min.*
z_1	−1	0	1	8		
w_2	−2	1	1	4		
w_3	−2	0	1	2		

So the solution of this LCP is $(w_1, w_2, w_3; z_1, z_2, z_3) = (0, 4, 2; 8, 0, 0)$.

2.2.8 Cycling Under Degeneracy in the Complementary Pivot Method

Whenever there is a tie for the pivot row in any step of the complementary pivot method, suppose we adopt the rule that the pivot row will be chosen to be the topmost among those eligible for it in that step. Under this rule it is possible that cycling occurs under degeneracy. Here we provide an example of cycling under this rule, constructed by M. M. Kostreva [2.20]. Let

$$M = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix} \qquad q = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

and solve the LCP (q, M) by the complementary pivot method using the above pivot row choice rule in each pivot step. It can be verified that we get the following almost complementary feasible basic vectors: initial basic vector (z_0, w_2, w_3) followed by (z_0, z_1, w_3), (z_0, z_2, w_3), (z_0, z_2, w_1), (z_0, z_3, w_1), (z_0, z_3, w_2), (z_0, z_1, w_2), (z_0, z_1, w_3), in this order. After the initial basic vector (z_0, w_2, w_3) is obtained, all pivots made are degenerate pivot steps, and at the end the method has returned to the basic vector (z_0, z_1, w_3) and so the method has cycled on this problem. The matrix M is a P-matrix, it will be proved later on the LCP (q, M) has a unique solution, and that the complementary pivot method always terminates in a finite number of pivot steps with that solution, if it is carried out in such a way that cycling does not occur under degeneracy. Actually, for the LCP (q, M) considered here, it can be verified that (z_1, z_2, z_3) is the complementary feasible basic vector.

As discussed above, after obtaining the initial basic vector, if the complementary pivot method is carried out using the lexico-minimum ratio rule for choosing the pivot row in each pivot step, cycling cannot occur, and the method must terminate either by obtaining a complementary feasible vector, or in ray termination, after a finite number of pivot steps, because of the following arguments. If q is nondegenerate in (2.3), the dropping basic variable is identified uniquely by the usual minimum ratio test, in every step of the complementary pivot algorithm applied on it. Using the properties of the path traced by this algorithm we verify that in this case, the algorithm must terminate after a finite number of pivot steps either with a complementary feasible basic vector or in ray termination. Suppose q is degenerate in (2.3). Perturb (2.3) by replacing q by $q(\varepsilon) = q + (\varepsilon, \varepsilon^{\mathbf{2}}, \dots, \varepsilon^{\mathbf{n}})^T$, as in (2.2). When ε is positive but sufficiently small, the perturbed problem is nondegenerate. So when the perturbed problem is solved by the complementary pivot algorithm treating $\varepsilon > 0$ to be sufficiently small, it must terminate in a finite number of pivot steps. If a complementary feasible basic vector is obtained at the end for the perturbed problem, that basic vector is also a complementary basic vector for the original LCP (unperturbed original problem, with $\varepsilon = 0$). If ray termination occurs at the end on the perturbed problem, the final almost complementary feasible basic vector is also feasible to the original LCP and satisfies the condition for ray termination in it. The sequence of basic vectors obtained when the complementary pivot algorithm is applied on the original problem (2.3) using the lexico-minimum ratio rule for chosing the dropping variable in every pivot step, is exactly the same as the sequence of basic vectors obtained when the complementary pivot algorithm is applied on the perturbed problem got by replacing q in (2.3) by $q(\varepsilon)$ with $\varepsilon > 0$ and sufficiently small. These facts show that the complementary pivot algorithm must terminate in a finite number of pivot steps (i. e., can not cycle) when operated with the lexico minimum ratio test for chosing the dropping variable in every pivot step.

2.3 CONDITIONS UNDER WHICH THE COMPLEMENTARY PIVOT ALGORITHM WORKS

We define several classes of matrices that are useful in the study of the LCP. Let $M = (m_{ij})$ be a square matrix of order n. It is said to be a

Copositive matrix if $y^T M y \geqq 0$ for all $y \geqq 0$.

Strict copositive matrix if $y^T M y > 0$ for all $y \geq 0$.

Copositive plus matrix if it is a copositive matrix and whenever $y \geqq 0$, and satisfies $y^T M y = 0$, we have $y^T(M + M^T) = 0$.

P**-matrix** if all its principal subdeterminantes are positive.

Q**-matrix** if the LCP (q, M) has a solution for every $q \in \mathbf{R}^n$.

Negative definite matrix if $y^T M y < 0$ for all $y \neq 0$.

Negative semidefinite matrix if $y^T M y \leqq 0$ for all $y \in \mathbf{R}^n$.

Z**-matrix** if $m_{ij} \leqq 0$ for all $i \neq j$

Principally nondegenerate matrix if all its principal subdeterminants are non-zero.

Principally degenerate matrix if at least one of its principal subdeterminants is zero.

L_1**-matrix** if for every $y \geq 0$, $y \in \mathbf{R}^n$, there is an i such that $y_i > 0$ and $M_{i.}y \geqq 0$. If M is an L_1-matrix, an i like it is called a **defining index** for M and y. These matrices are also called **semimonotone matrices**.

L_2**-matrix** if for every $y \geq 0$, $y \in \mathbf{R}^n$, such that $My \geqq 0$ and $y^T M y = 0$, there are diagonal matrices, $\Lambda \geqq 0$, $\Omega \geqq 0$ such that $\Omega y \neq 0$ and $(\Lambda M + M^T \Omega) y = 0$. An equivalent definition is that for each $z \geq 0$, satisfying $w = Mz \geqq 0$ and $w^T z = 0$; there exists a $\hat{z} \geq 0$ satisfying $\hat{w} = -(\hat{z}^T M)^T$, $w \geqq \hat{w} \geqq 0$, $z \geqq \hat{z} \geqq 0$.

L**-matrix** if it is both an L_1-matrix and an L_2-matrix.

$L_\star$**-matrix** if for every $y \geq 0$, $y \in \mathbf{R}^n$, there is an i such that $y_i > 0$ and $M_{i.}y > 0$. If M is an $L_\star$-matrix, an i like it is called a **defining index** for M and y.

P_0**-matrix** if all its principal subdeterminants are $\geqq 0$.

Row adequate matrix if it is a P_0-matrix and whenever the principal subdeterminant corresponding to some subset $\mathbf{J} \subset \{1, \ldots, n\}$ is zero, then the set of row vectors of M corresponding to $\mathbf{J}$, $\{M_{i.} : i \in \mathbf{J}\}$ is linearly dependent.

Column adequate matrix if it is a P_0-matrix and whenever the principal subdeterminant corresponding to some subset $\mathbf{J} \subset \{1, \ldots, n\}$ is zero, then the set of column vectors of M corresponding to $\mathbf{J}$, $\{ M_{.j} : j \in \mathbf{J} \}$ is linearly dependent.

Adequate matrix if it is both row and column adequate.

In this book the only type of degeneracy, nondegeneracy of square matrices that we discuss is principal degeneracy or principal nondegeneracy defined above. So, for notational convenience we omit the term "principally" and refer to these matrices as being degenerate or nondegenerate matrices. Examples of degenerate matrices are $\begin{pmatrix} 0 & 4 \\ 3 & -10 \end{pmatrix}$, $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Examples of nondegenerate matrices are $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$, $\begin{pmatrix} 3 & 1 \\ 0 & -2 \end{pmatrix}$. The notation C_0-**matrix** is used to denote copositive matrices, and the notation C_+-**matrix** is used to denote copositive plus matrices.

Theorem 1.11 implies that every PSD matrix is also a copositive plus matrix. Also, the square matrix M is negative definite or negative semi-definite, iff $-M$ is PD or PSD respectively.

2.3.1 Results on LCPs Associated with Copositive Plus Matrices

Theorem 2.1 *If M is a copositive plus matrix and the system of constraints (1.6) and (1.7) of Section 1.1.3 has a feasible solution, then the LCP (1.6) — (1.8) has a solution and the complementary pivot algorithm will terminate with the complementary feasible basis. Conversely, when M is a copositive plus matrix, if the complementary pivot algorithm applied on (1.6) — (1.8) terminates in ray termination, the system of constraints (1.6), (1.7) must be infeasible.*

Proof. Assume that either (2.3) is nondegenerate, or that the lexico-minimum ratio rule is used throughout the algorithm to determine the dropping basic variable in each step of the algorithm. This implies that each almost complementary feasible (or lexico feasible) basis obtained during the algorithm has exactly two adjacent almost complementary feasible (or lexico feasible) bases, excepting the initial and terminal bases, which have exactly one such adjacent basis only. The complementary pivot algorithm operates on the system (2.3).

The initial basic vector is $(w_1, \dots, w_{t-1}, z_0, w_{t+1}, \dots, w_n)$ (as in Section 2.2.3). The corresponding BFS is $z = 0$, $w_t = 0$, $z_0 = -q_t$, and $w_i = q_i - q_t$ for all $i \neq t$. If w_t is taken as the entering variable into this basic vector, it generates the half-line (called the **initial extreme half-line**)

$$\begin{aligned} w_i &= q_i - q_t + \lambda \qquad \text{for all } i \neq t \\ w_t &= \lambda \\ z &= 0 \\ z_0 &= -q_t + \lambda \end{aligned}$$

where $\lambda \geqq 0$. (This can be seen by obtaining the canonical tableau corresponding to the initial basic vector.) This initial extreme half-line contains the initial BFS of (2.3) as its end point. Among the basic vectors obtained during the algorithm, the only one that can be adjacent to the initial basic vector is the one obtained by introducing z_t into it. Once the algorithm moves to this adjacent basic vector, the initial basic vector will never again appear during the algorithm. Hence, if the algorithm terminates with ray termination, the extreme half-line obtained at termination cannot be the initial extreme half-line.

At every point on the initial extreme half-line all the variables w, z_0 are strictly positive. It is clear that the only edge of (2.3) that contains a point in which all the variables w, z_0 are strictly positive is the initial extreme half-line.

Suppose the algorithm terminates in ray termination without producing a solution of the LCP. Let B_k be the terminal basis. When the complementary pivot algorithm is continued from this basis B_k, the updated column vector of the entering variable must be nonpositive resulting in the generation of an extreme half-line. Let the terminal extreme half-line be

$$\{ (w, z, z_0) = (w^k + \lambda w^h, z^k + \lambda z^h, z_0^k + \lambda z_0^h) : \quad \lambda \geqq 0 \} \tag{2.5}$$

where (w^k, z^k, z_0^k) is the BFS of (2.3) with respect to the terminal basis B_k, and (w^h, z^h, z_0^h) is a homogeneous solution corresponding to (2.3) that is,

$$\begin{aligned} &w^h - Mz^h - e_n z_0^h = 0 \\ &w^h \geqq 0, \quad z^h \geqq 0, \quad z_0^h \geqq 0 \end{aligned} \tag{2.6}$$

$(w^h, z^h, z_0^h) \neq 0$. If $z^h = 0$, (2.6) and the fact that $(w^h, z^h, z_0^h) \neq 0$ together imply that $w^h \neq 0$ and hence $z_0^h > 0$, and consequently $w^h > 0$. Hence, if $z^h = 0$, points on this terminal extreme half-line have all the variables w, z_0 strictly positive, which by earlier arguments would imply that the terminal extreme half-line is the initial extreme half-line, a contradiction. So $z^h \neq 0$.

Since every solution obtained under the algorithm satisfies the complementarity constraint, $w^T z = 0$, we must have $(w^k + \lambda w^h)^T (z^k + \lambda z^h) = 0$ for all $\lambda \geqq 0$. This implies that $(w^k)^T z^k = (w^k)^T z^h = (w^h)^T z^k = (w^h)^T z^h = 0$. From (2.6) $(w^h)^T = (Mz^h + e_n z_0^h)^T$. Hence from $(w^h)^T z^h = 0$, we can conclude that $(z^h)^T M^T z^h = (z^h)^T M z^h = -e_n^T z^h z_0^h \leqq 0$. Since $z^h \geqq 0$, and M is copositive plus by the hypothesis, $(z^h)^T M z^h$ cannot be < 0, and hence, by the above, we conclude that $(z^h)^T M z^h = 0$. This implies that $(z^h)^T (M + M^T) = 0$, by the copositive plus property of M. So $(z^h)^T M = -(z^h)^T M^T$. Also since $-e_n^T z^h z_0^h = (z^h)^T M z^h = 0$, z_0^h must be zero (since $z^h \geq 0$). Since (w^k, z^k, z_0^k) is the BFS of (2.3) with respect to the feasible basis B_k, $w^k = Mz^k + q + e_n z_0^k$. Now

$$\begin{aligned}
0 = (w^k)^T z^h &= (Mz^k + q + e_n z_0^k)^T z^h \\
&= (z^k)^T M^T z^h + q^T z^h + z_0^k e_n^T z^h \\
&= (z^h)^T M z^k + (z^h)^T q + z_0^k e_n^T z^h \\
&= -(z^h)^T M^T z^k + (z^h)^T q + z_0^k e_n^T z^h \\
&= -(z^k)^T M z^h + (z^h)^T q + z_0^k e_n^T z^h \\
&= -(z^k)^T w^h + (z^h)^T q + z_0^k e_n^T z^h \\
&= (z^h)^T q + z_0^k e_n^T z^h
\end{aligned}$$

So $(z^h)^T q = -z_0^k e_n^T z^h$. Since $z^h \geq 0$ and $z_0^k > 0$ [otherwise (w^k, z^k) would be a solution of the LCP], $z_0^k e_n^T z^h > 0$. Hence, $(z^h)^T q < 0$. Hence, if $\pi = (z^h)^T$ we have, $\pi q < 0$, $\pi \geqq 0$, $\pi(-M) = -(z^h)^T M = (z^h)^T M^T = (w^h)^T \geqq 0$, that is,

$$\begin{aligned}
\pi q &< 0 \\
\pi(I \;\vdots\; -M) &\geqq 0
\end{aligned}$$

By Farakas lemma (Theorem 3 of Appendix 1), this implies that the system :

$$(I \;\vdots\; -M)\begin{pmatrix} w \\ \cdots \\ z \end{pmatrix} = q \;, \qquad \begin{pmatrix} w \\ \cdots \\ z \end{pmatrix} \geqq 0$$

has no feasible solution. Hence, if the complementary pivot algorithm terminates in ray termination, the system (1.6) and (1.7) has no feasible solutions in this case and thus there cannot be any solution to the LCP.

This also implies that whenever (1.6) and (1.7) have a feasible solution, the LCP (1.6) to (1.8) has a solution in this case and the complementary pivot algorithm finds it.

□

The following results can be derived as corollaries.

Result 2.1 *In the LCPs corresponding to LPs and convex quadratic programs, the matrix M is* PSD *and hence copositive plus. Hence, if the complementary pivot algorithm applied to the LCP corresponding to an LP or a convex quadratic program terminates in ray termination, that LP or convex quadratic program must either be infeasible, or if it is feasible, the objective function must be unbounded below on the set of feasible solutions of that problem.*

Hence the complementary pivot algorithm works when used to solve LPs or convex quadratic programs.

Result 2.2 *If M is strict copositive the complementary pivot algorithm applied on (1.6) to (1.8) terminates with a solution of the LCP.*

Proof. If the complementary pivot algorithm terminates in ray termination, as seen in the proof of the above theorem there exists a $z^h \geq 0$ such that $(z^h)^T M z^h = 0$, contradicting the hypothesis that M is strict copositive.

□

Thus all strict copositive matrices are Q-matrices. Also, if $M = (m_{ij}) \geqq 0$ and $m_{ii} > 0$ for all i, M is strict copositive and hence a Q-matrix.

Exercise

2.1 Suppose $M \geqq 0$ and $m_{11} = 0$. Prove that if $q = (-1, 1, \dots, 1)^T$, the LCP (1.6) to (1.8) cannot have a solution. Thus prove that a square nonegative matrix is a Q-matrix iff all its diagonal entries are strictly positive.

Later on we prove that if M is a P-matrix, the complementary pivot algorithm terminates with a complementary feasible solution when applied on the LCP (q, M). When the complementary pivot algorithm is applied on a LCP in which the matrix M is not a copositive plus matrix or a P-matrix, it is still possible that the algorithm terminates with a complementary feasible basis for the problem. However, in this general case it is also possible that the algorithm stops with ray termination even if a solution to the LCP exists.

To Process an LCP (q, M)

An algorithm for solving LCPs is said to **process** a particular LCP (q, M) for given q and M, if the algorithm is guaranteed to either determine that the LCP (q, M) has no solution, or find a solution for it, after a finite amount of computational effort.

Suppose M is a copositive plus matrix, and consider the LCP (q, M), for given q. When the complementary pivot algorithm is applied on this LCP (q, M), either it finds a solution; or ends up in ray termination which implies that this LCP has no solution by the above theorem. Hence, the complementary pivot algorithm processes the LCP (q, M) whenever M is a copositive plus matrix.

2.3.2 Results on LCPs Associated with L- and $L_\star$-Matrices

Here we show that the complementary pivot algorithm will process the LCP (q, M) whenever M is an L- or $L_\star$-matrix. The results in this section are from B. C. Eaves [2.8, 2.9], they extend the results proved in Section 2.3.1 considerably. Later on, in Section 2.9.2 we derive some results on the general nonconvex programming problem using those proved in this section.

Lemma 2.1 *If M is an L_1-matrix, the LCP (q, M) has a unique solution for all $q > 0$, and conversely.*

Proof. When $q > 0$, one solution of the LCP (q, M) is $(w = q, z = 0)$. So if $(\bar{w}, \bar{z})$ is an alternate solution, we must have $\bar{z} \geq 0$. But $\bar{w} - M\bar{z} = q$. Let M be an L_1-matrix and let i be the defining index for M and $\bar{z}$. We have

$$\bar{w}_i = (M\bar{z})_i + q_i > 0$$

So $\bar{w}_i \bar{z}_i > 0$, contradiction to complementarity.

Now suppose M is not an L_1-matrix. So, there must exist a $\bar{y} = (\bar{y}_i) \geq 0$ such that for all i such that $\bar{y}_i > 0$, $M_{i.}\bar{y} < 0$. Let $\mathbf{J} = \{ i : \bar{y}_i > 0 \}$. Select a positive number α such that $\alpha > \{ |M_{i.}\bar{y}| : i \notin \mathbf{J} \}$. Define the vector $q = (q_j) \in \mathbf{R}^n$ by

$$q_j = \begin{cases} -M_{j.}\bar{y}, & \text{for all } j \in \mathbf{J} \\ \alpha, & \text{for all } j \notin \mathbf{J}. \end{cases}$$

Then $q > 0$ and the LCP (q, M) has two distinct solutions namely $(w, z) = (q, 0)$ and $\big(\bar{w} = (\bar{w}_j), \bar{z} = \bar{y}\big)$, where

$$\bar{w}_j = \begin{cases} 0, & \text{for all } j \in \mathbf{J} \\ \alpha + M_{j.}\bar{y}, & \text{for all } j \notin \mathbf{J}. \end{cases}$$

This establishes the converse.

□

Lemma 2.2 *If M is an $L_\star$-matrix, the LCP (q, M) has a unique solution for every $q \geqq 0$, and conversely.*

Proof. Similar to Lemma 2.1.

□

Lemma 2.3 *If M is an L_2-matrix and the complementary pivot method applied on the LCP (q, M) terminates with the secondary ray $\{ (\bar{w}^k, z^k, z_0^k) + \lambda(w^h, z^h, z_0^h) : \lambda \geqq 0 \}$ as in (2.5), where (w^k, z^k, z_0^k) is the terminal BFS of (2.3) and (w^h, z^h, z_0^h) is a homogeneous solution corresponding to (2.3) satisfying (2.6); and $z_0^k > 0$ and $z_0^h = 0$; then the LCP (q, M) is infeasible, that is, the system "$w - Mz = q$, $w \geqq 0$, $z \geqq 0$" has no feasible solution.*

Proof. As in the proof of Theorem 2.1 we assume that either (2.3) is nondegenerate or that the lexico minimum ratio rule is used throughout the algorithm to determine the dropping basic variable in each step of the algorithm. Using the hypothesis that $z_0^h = 0$ in (2.6), we have

$$w^h - Mz^h = 0$$
$$(z^h)^T w^h = 0$$

Since $(w^h, z^h, z_0^h) \neq 0$, this implies that $z^h \geq 0$. Therefore $0 = (z^h)^T w^h = (z^h)^T M z^h = 0$, and $z^h \geq 0$. So, using the hypothesis that M is an L_2-matrix, we have diagonal matrices $\Omega \geqq 0$, $\Lambda \geqq 0$ such that $\Omega z^h \neq 0$ and $(\Lambda M + M^T \Omega) z^h = 0$. Since $\Lambda M z^h \geqq 0$ (since $M z^h = w^h \geqq 0$ and $\Lambda \geqq 0$ is a diagonal matrix) this implies that $M^T \Omega z^h = (z^h)^T \Omega M \leqq 0$. Now $0 = (z^k)^T w^h = (z^k)^T \Lambda w^h$ (since Λ is a diagonal matrix with nonnegative entries and $w^h \geqq 0$, $z^k \geqq 0$) $= (z^k)^T \Lambda M z^h = (z^k)^T(-M^T \Omega z^h)$. So $(z^h)^T \Omega M z^k = 0$. Now

$$(z^h)^T \Omega (w^k - M z^k - e z_0^k) = (z^h)^T \Omega q$$

Since Ω is a nonnegative diagonal matrix and $(z^h)^T w^k = 0$ and $(z^h)^T \geqq 0$, $w^k \geqq 0$, we have $(z^h)^T \Omega w^k = 0$. Also $(z^h)^T \Omega M z^k = (z^k)^T M^T \Omega z^h = -(z^k)^T \Lambda M z^h = -(z^k)^T \Lambda w^h = 0$ (since $z^k \geqq 0$, $w^h \geqq 0$, Λ is a diagonal matrix which is $\geqq 0$, $(z^k)^T w^h = 0$ implies $(z^k)^T \Lambda w^h = 0$). Using these in the above equation, we get

$$-(z^h)^T \Omega e z_0^k = (z^h)^T \Omega q$$

since $(z^h)^T \geq 0$, $\Omega \geqq 0$, $\Omega z^h \neq 0$, we have $\Omega z^h = (z^h)^T \Omega \geq 0$, this implies that $(z^h)^T \Omega e > 0$. Also, by hypothesis $z_0^k > 0$. So from the above equation $(z^h)^T \Omega q < 0$. So if $\pi = (z^h)^T \Omega$, we have

$$\begin{aligned} \pi &\geq 0 \\ -\pi M = -(z^h)^T \Omega M &= -M^T \Omega z^h = \Lambda M z^h = \Lambda w^h \geqq 0 \\ \pi q &< 0 \end{aligned}$$

which implies that $q \notin \text{Pos}(I, -M)$ by Farakas$'$ theorem (Theorem 3 of Appendix 1). So the system

$$\begin{aligned} w - Mz &= q \\ w, z &\geqq 0 \end{aligned}$$

is itself infeasible.

□

Theorem 2.2 *The complementary pivot algorithm processes the LCP (q, M) if M is an L-matrix.*

Proof. When we apply the complementary pivot algorithm on the LCP (q, M), suppose the secondary ray $\{ (w^k + \lambda w^h, z^k + \lambda z^h, z_0^k + \lambda z_0^h) : \lambda \geqq 0 \}$ is generated. So we have

$$(w^k + \lambda w^h) - M(z^k + \lambda z^h) = q + e(z_0^k + \lambda z_0^h) \ .$$

If $z_0^h > 0$, and in the above equation if $\bar{\lambda}$ is a large positive value such that $q + e(z_0^k + \lambda z_0^h) > 0$, then $(w^k + \bar{\lambda} w^h, z^k + \bar{\lambda} z^h)$ is a complementary solution for the LCP $\big(q + e(z_0^k + \bar{\lambda} z^h), M\big)$ which by Lemma 2.1 implies that $z^k + \bar{\lambda} z^h = 0$, which means that $z^k = z^h = 0$, a contradiction to the fact that this is the secondary ray. So z_0^h cannot be > 0, that is $z_0^h = 0$, and in this case (q, M) has no solution by Lemma 2.3. So the complementary pivot algorithm processes the LCP (q, M).

□

Theorem 2.3 *If M is an $L_\star$-matrix, when the complementary pivot algorithm is applied on the LCP (q, M), it terminates with a complementary feasible solution.*

Proof. In this case we show that there can be no secondary ray. Suppose $\{ (w^k+\lambda w^h, z^k+\lambda z^h, z_0^k+\lambda z_0^h) : \lambda \geqq 0 \}$ is a secondary ray. As in the proof of Theorem 2.1, $z^h \geq 0$ (otherwise this ray will be the same as the initial ray, a contradiction). Let i be the defining index of M, z^h. So we have $z_i^h > 0$ which implies $w_i^h = 0$ by complementarity and

$$0 < (Mz^h)_i = -(ez_0^h)_i \leqq 0$$

a contradiction. So a secondary ray cannot exist in this case, and the complementary pivot method must terminate with a complementary feasible solution.

□

Theorem 2.2 and 2.3 make it possible for us to conclude that the complementary pivot algorithm processes that LCP (q, M) for a much larger class of matrices M than the copositive plus class proved in Theorem 2.1. We will now prove several results establishing that a variety of matrices are in fact L- or $L_\star$-matrices. By virtue of Theorem 2.2 and 2.3, this establishes that the complementary pivot method processes the LCP (q, M) whenever M is a matrix of one of these types.

All copositive plus matrices are L-matrices. This follows because when M is copositive plus, $y \geqq 0$ implies $y^T My \geqq 0$, and if y is such that $y \geqq 0$, $y^T My = 0$ then $(M + M^T)y = 0$, hence M satisfies the definition of being an L-matrix by taking the diagonal matrices Λ and Ω to be both I. A strictly copositive matrix is clearly an $L_\star$-matrix. From the definitions, it can be verified that PMP^T (obtained by principal rearrangement of M), $\Lambda M\Omega$ (obtained by positive row and column scaling of M) are L-matrices if M is, whenever P is a permutation matrix and Λ, Ω are diagonal matrices with positive diagonal elements. Copositive plus matrices M satisfy the property that PMP^T is also copositive plus whenever P is a permutation matrix, but if M is copositive plus, $\Lambda M\Omega$ may not be copositive when Λ, Ω are diagonal matrices with positive diagonal entries. Also if M, N are L-matrices, so is $\begin{pmatrix} M & 0 \\ 0 & N \end{pmatrix}$. Again, from Theorem 3.11 of Section 3.3, it follows that all P-matrices are $L_\star$ matrices.

Lemma 2.4 *M is row adequate iff for any y, $(y^T M_{.i})y_i \leqq 0$ for $i = 1$ to n implies that $y^T M = 0$.*

Proof. Suppose M is row adequate, and there exists a $y \geqq 0$ such that $(y^T M_{.i})y_i \leqq 0$ for $i = 1$ to n. By a standard reduction technique used in linear programming (see Section 3.4.2 in [2.26]) we can get a solution x of

$$x^T M = y^T M$$
$$x \geqq 0$$

such that $\{ M_{i.} : x_i > 0 \} \subset \{ M_{i.} : y_i > 0 \}$ and $\{ M_{i.} : x_i > 0 \}$ is linearly independent. So we also have $(x^T M_{.i})x_i \leqq 0$ for all $i = 1$ to n. Let $\mathbf{J} = \{ i : x_i > 0 \}$. Since M is a P_0-matrix, so is its principal submatrix $M_{\mathbf{JJ}} = (m_{ij} : i \in \mathbf{J}, j \in \mathbf{J})$. By linear independence of the set of row vectors $\{ M_{i.} : i \in \mathbf{J} \}$, since M is row adequate, we know that the determinant of $M_{\mathbf{TT}} \neq 0$ for all $\mathbf{T} \subset \mathbf{J}$, and therefore that $M_{\mathbf{JJ}}$ is a P-matrix. The facts $\mathbf{J} = \{ i : x_i > 0 \}$, $x_i = 0$ if $i \notin \mathbf{J}$, and $(x^T M_{.i})x_i \leqq 0$ for all $i = 1$ to n, together imply that $M_{\mathbf{JJ}} x_{\mathbf{J}} \leqq 0$ where $x_{\mathbf{J}} = (x_j : j \in \mathbf{J})$, which implies by Theorem 3.11 of Section 3.3 that $x_{\mathbf{J}} = 0$ since $M_{\mathbf{JJ}}$ is a P-matrix, a contradiction. So $\mathbf{J}$ must be empty and $x = 0$, and hence $y^T M = 0$. Now if $y \in \mathbf{R}^n$, y not necessarily $\geqq 0$, satisfies $(y^T M_{.i})y_i \leqq 0$ for all $i = 1$ to n, let $\lambda_i = 1$ if $y_i \geqq 0$, or -1 if $y_i < 0$; and let Δ be the diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$. Then $y^T \Delta \geqq 0$ and $\big(y^T \Delta (\Delta M)_{.i}\big)\lambda_i^{\mathbf{2}} y_i = \big((y^T \Delta)(\Delta M \Delta)_{.i}\big)(\lambda_i y_i) \leqq 0$ for all i. But $\Delta M \Delta$ is row adequate since M is, and by the above we therefore have $y^T \Delta (\Delta M \Delta) = 0$ or $y^T M = 0$.

Conversely, if M is a square matrix such that for any y, $(y^T M_{.i})y_i \leqq 0$ for all $i = 1$ to n implies that $y^T M = 0$, it follows that M is a P_0-matrix by the result in Exercise 3.5 and that M is row adequate.

□

Lemma 2.5 *Let M be a P_0-matrix. If*

$$\begin{aligned} My &= 0 \\ y &> 0 \end{aligned}$$

has a solution y, then the system

$$\begin{aligned} x^T M &= 0 \\ x &\geq 0 \end{aligned}$$

has a solution.

Proof. Let y satisfy $My = 0$, $y > 0$. By the result in Exercise 3.6 we know that since M is a P_0-matrix, there is a x satisfying $x^T M \geqq 0$, $x \geq 0$. If $x^T M \neq 0$, then $(x^T M)y > 0$ but $x^T (My) = 0$, a contradiction. So this x must satisfy $x^T M = 0$.

□

Theorem 2.4 *If M is row adequate, then M is an L-matrix.*

Proof. By the result in Exercise 3.5 M is a P_0-matrix iff for all $y \neq 0$, there exists an i such that $y_i \neq 0$ and $y_i(M_{i.}y) \geqq 0$. This implies that all P_0-matrices are L_1-matrices.

Suppose y satisfies $y \geq 0$, $My \geqq 0$, $y^T My = 0$. Let $\mathbf{J} = \{ i : y_i > 0 \}$. These facts imply $M_{\mathbf{JJ}} y_{\mathbf{J}} = 0$ where $M_{\mathbf{JJ}}$ is the principal submatrix $(m_{ij} : i \in \mathbf{J}, j \in \mathbf{J})$, and $y_{\mathbf{J}} = (y_j : j \in \mathbf{J}) > 0$. By Lemma 2.5, there exists an $x_{\mathbf{J}} = (x_j : j \in \mathbf{J})$ satisfying $x_{\mathbf{J}} \geq 0$, $x_{\mathbf{J}}^T M_{\mathbf{JJ}} = 0$. From Lemma 2.4, these facts imply that $x_{\mathbf{J}}^T M_{\mathbf{J}.} = 0$, where $M_{\mathbf{J}.}$ is the matrix with rows $M_{i.}$ for $i \in \mathbf{J}$. Select the diagonal matrix Ω so that $x_{\mathbf{J}} = (\Omega y)_{\mathbf{J}}$ (possible because $y_{\mathbf{J}} > 0$) and $0 = (\Omega y)_{\bar{\mathbf{J}}}$ where $\bar{\mathbf{J}} = \{ 1, \ldots, n \} \setminus \mathbf{J}$. Then $y^T \Omega M = 0$ and $(\Lambda M + M^T \Omega)y = 0$ with $\Lambda = 0$. So M is an L_2-matrix too. Thus M is an L-matrix.

□

Lemma 2.6 *If R, S are L-matrices and $P > 0$, $N < 0$ are matrices of appropriate orders, then*

$$A = \begin{pmatrix} R & P \\ N & S \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} S & N \\ P & R \end{pmatrix}$$

are also L-matrices.

Proof. Consider the product $A\xi$ where

$$\xi = \begin{pmatrix} x \\ y \end{pmatrix} \geq 0$$

Case 1 : Let $x \geq 0$, $y \geq 0$. Select a defining index for R and x, suppose it is i. Then

$$x_i(Rx + Py)_i > 0, \text{ since } P > 0 \text{ and } y \geq 0 \ .$$

This verifies that in this case the same i will serve as a defining index for A to satisfy the condition for being an L_1-matrix with this vector ξ. Also verify that in this case, A satisfies the condition for being an L_2-matrix, with this vector ξ, trivially.

Case 2 : Let $x \geq 0$, $y = 0$. The select i as in case 1 and it will serve as a defining index for A to satisfy the conditions for being an L_1-matrix, with this vector ξ. Also verify that in this case A satisfies the condition for being an L_2-matrix, with this vector ξ, trivially, since $A\xi \geqq 0$ would imply in this case $x = 0$, a contradiction.

Case 3 : Let $x = 0$, $y \geq 0$. Select a defining index for S and y, suppose it is i. Verify that the same i will serve as a defining index for A to satisfy the condition for being an L_1-matrix. If y is such that $A\xi \geqq 0$ and $\xi^T A\xi = 0$, then $Sy \geqq 0$, $y^T Sy = 0$. Since S is an L_2-matrix, there must exist diagonal matrices Λ_2, $\Omega_2 \geqq 0$ such that $\Omega_2 y \neq 0$ and $(\Lambda_2 S + S^T\Omega_2)y = 0$. Now, it can be verified easily that there is an appropriate choice of diagonal matrices Λ_1, Ω_1 such that (since $x = 0$ in this case)

$$\begin{aligned} &\begin{pmatrix} \Lambda_1 P + N^T\Omega_2 \\ \Lambda_2 S + S^T\Omega_2 \end{pmatrix} y = \\ &= \left(\begin{pmatrix} \Lambda_1 & \\ & \Lambda_2 \end{pmatrix} \begin{pmatrix} R & P \\ N & S \end{pmatrix} + \begin{pmatrix} R^T & N^T \\ P^T & S^T \end{pmatrix} \begin{pmatrix} \Omega_1 & \\ & \Omega_2 \end{pmatrix} \right) \begin{pmatrix} x \\ y \end{pmatrix} \\ &= 0 \end{aligned}$$

So A satisfies the condition for being an L_2-matrix, with this vector ξ.

These facts establish that A is an L-matrix. The proof that B is an L-matrix is similar.

□

Lemma 2.7 *If R, S are $L_\star$-matrices and $P \geqq 0$, Q arbitrary, are matrices of appropriate orders, then*

$$A = \begin{pmatrix} R & P \\ Q & S \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} S & Q \\ P & R \end{pmatrix}$$

are also $L_\star$-matrices.

Proof. Let

$$\xi = \begin{pmatrix} x \\ y \end{pmatrix}$$

Consider the product $A\xi$. If $x \geq 0$, select i to be a defining index for R and x. Since $P \geqq 0$, the same i serves as a defining index for A and ξ in the condition for A to be an $L_\star$-matrix with this ξ. If $x = 0$, then select i to be a defining index for S and y, the same i serves as a defining index for A and ξ in the condition for A to be an $L_\star$-matrix, with this ξ. So A is an $L_\star$-matrix. The proof that B is an $L_\star$-matrix is similar.

□

Lemma 2.8 *If $P > 0$ is of order $n \times m$ and $N < 0$ is of order $m \times n$, then $\begin{pmatrix} 0 & P \\ N & 0 \end{pmatrix}$ is an L-matrix.*

Proof. Since 0 is an L-matrix, this results follows from Lemma 2.6.

□

In Exercise 2.24 we ask the reader to prove that one formulation of the bimatrix game problem as an LCP can be solved directly by the complementary pivot algorithm, to yield a solution, using this lemma.

Lemma 2.9 *Let T $(n \times n)$, R $(n \times m)$, ρ $(n \times 1)$, S $(m \times n)$, σ $(1 \times n)$ be given matrices with $\rho > 0$, $\sigma < 0$; where $n \geqq 0$, $m \geqq 0$. If for each $x = (x_1, \ldots, x_m)^T$, δ real satisfying $(x_1, \ldots, x_m, \delta) \geq 0$, $Rx + \rho\delta \geqq 0$; there exist diagonal matrices $\Lambda \geqq 0$, $\Gamma \geqq 0$ of orders $n \times n$ and $(m+1) \times (m+1)$ respectively such that*

$$\Gamma \begin{pmatrix} x \\ \delta \end{pmatrix} \neq 0 \quad \text{and} \quad \big(\Lambda(R, \rho) + (S^T, \sigma^T)\Gamma\big) \begin{pmatrix} x \\ \delta \end{pmatrix} = 0$$

then the following matrix M is an L_2-matrix

$$M = \begin{pmatrix} T & R & \rho \\ S & 0 & 0 \\ \sigma & 0 & 0 \end{pmatrix}$$

Proof. Follows from the definition of L_2-matrices.

□

Notice that in Lemma 2.9, m could be zero, this will correspond to R, S being vacuous.

Theorem 2.5 *Let T $(n \times n)$, R $(n \times m)$, ρ $(n \times 1)$, S $(m \times n)$, σ $(1 \times n)$ be given matrices satisfying $\rho > 0$, $\sigma < 0$. Let $N = n + m + 1$. Let $\mathbf{J}_1 = \{1, \ldots, n\}$, $\mathbf{J}_2 = \{n+1, \ldots, n+m\}$, $\mathbf{J}_3 = \{n+m+1\}$. For vectors $w, z, q \in \mathbf{R}^N$, let $w_{\mathbf{J}_t}$ etc. be defined to be the vectors $w_{\mathbf{J}_t} = (w_j : j \in \mathbf{J}_t)$, etc. Assume that $q \in \mathbf{R}^N$ is a given column vector satisfying $q_{\mathbf{J}_3} = (q_{n+m+1}) > 0$. Let*

$$M = \begin{pmatrix} T & R & \rho \\ S & 0 & 0 \\ \sigma & 0 & 0 \end{pmatrix}$$

If M is an L_2-matrix, when the complementary pivot method is applied on the LCP (q, M) with the original column vector of the artificial variable z_0 taken to be $(-1, \dots, -1, 0)^T \in \mathbf{R}^N$, either we get a complementary feasible solution of the problem, or the system

$$-\begin{pmatrix} S \\ \sigma \end{pmatrix} x \leqq \begin{pmatrix} q_{\mathbf{J}_2} \\ q_{\mathbf{J}_3} \end{pmatrix}$$

$$x \geqq 0$$

must be infeasible.

Proof. Suppose the complementary pivot algorithm is applied on the LCP (q, M) with the original column vector of the artificial variable z_0 taken to be $(-1, \dots, -1, 0)^T \in \mathbf{R}^N$, and it terminates with the secondary ray $\{ (w^k + \lambda w^h, z^k + \lambda z^h, z_0^k + \lambda z_0^h) : \quad \lambda \geqq 0 \}$. Then

$$\begin{pmatrix} w^h_{\mathbf{J}_1} \\ w^h_{\mathbf{J}_2} \\ w^h_{\mathbf{J}_3} \end{pmatrix} - \begin{pmatrix} T & R & \rho \\ S & 0 & 0 \\ \sigma & 0 & 0 \end{pmatrix} \begin{pmatrix} z^h_{\mathbf{J}_1} \\ z^h_{\mathbf{J}_2} \\ z^h_{\mathbf{J}_3} \end{pmatrix} - \begin{pmatrix} e_n \\ e_m \\ 0 \end{pmatrix} z_0^h = 0$$

So $w^h_{\mathbf{J}_3} = \sigma z^h_{\mathbf{J}_1}$ and since $z^h_{\mathbf{J}_1} \geqq 0$, $w^h_{\mathbf{J}_3} \geqq 0$ and $\sigma < 0$, we have $z^h_{\mathbf{J}_1} = 0$, $w^h_{\mathbf{J}_3} = 0$.

If $z_0^h > 0$, then $w^h_{\mathbf{J}_2} = S z^h_{\mathbf{J}_1} + e_m z_0^h = e_m z_0^h > 0$, which by complementarity implies that $z^h_{\mathbf{J}_2} = z^k_{\mathbf{J}_2} = 0$. So $w^h_{\mathbf{J}_1} = R z^h_{\mathbf{J}_2} + \rho z^h_{\mathbf{J}_3} + e_n z_0^h = \rho z^h_{\mathbf{J}_3} + e_n z_0^h > 0$ (since $\rho > 0$). By complementarity $z^k_{\mathbf{J}_1} = 0$, and so $w^k_{\mathbf{J}_3} = \sigma z^k_{\mathbf{J}_1} + q_{\mathbf{J}_3} = q_{\mathbf{J}_3} > 0$. So by complementarity, $z^k_{\mathbf{J}_3} = z^h_{\mathbf{J}_3} = 0$. Thus $z^h = z^k = 0$, contradiction to the fact that this is a secondary ray. Therefore z_0^h must be zero. Since M is an L_2-matrix, by Lemma 2.3, the existence of this secondary ray with $z_0^h = 0$ implies that

$$w - Mz = q$$

$$w, z \geqq 0$$

has no feasible solution, which, by Faraka′s theorem (Theorem 3 of Appendix 1) implies that there exists a row vector $\alpha \in \mathbf{R}^N$ such that

$$\alpha M \leqq 0$$

$$\alpha q < 0$$

$$\alpha \geq 0$$

$\alpha M \leqq 0$ includes the constraints $\alpha_{\mathbf{J}_1} \leqq 0$ and since $\alpha_{\mathbf{J}_1} \geqq 0$, $\rho > 0$, this implies that $\alpha_{\mathbf{J}_1} = 0$. So the above system of constraints becomes

$$(\alpha_{\mathbf{J}_2}, \alpha_{\mathbf{J}_3}) \begin{pmatrix} S \\ \sigma \end{pmatrix} \leqq 0$$

$$(\alpha_{\mathbf{J}_2}, \alpha_{\mathbf{J}_3}) \begin{pmatrix} q_{\mathbf{J}_2} \\ q_{\mathbf{J}_3} \end{pmatrix} < 0$$

$$(\alpha_{\mathbf{J}_2}, \alpha_{\mathbf{J}_3}) \geqq 0$$

By Faraka′s theorem (Theorem 3 of Appendix 1) this implies that the system

$$-\begin{pmatrix} S \\ \sigma \end{pmatrix} x \leqq \begin{pmatrix} q_{\mathbf{J}_2} \\ q_{\mathbf{J}_3} \end{pmatrix}$$

$$x \geqq 0$$

is infeasible.

□

In Section 2.9.2, Lemma 2.9 and Theorem 2.5 are applied to show that KKT points for general quadratic programs can be computed, when they exist, using the complementary pivot algorithm.

2.3.3 A Variant of the Complementary Pivot Algorithm

In the version of complementary pivot algorithm discussed so far, we have choosen the original column vector associated with the artificial variable z_0 to be $-e_n$. Given a column vector $d \in \mathbf{R}^n$ satisfying $d > 0$, clearly we can choose the original column vector associated with z_0 to be $-d$ instead of $-e_n$ in the complementary pivot algorithm. If this is done, the original tableau turns out to be :

w	z	z_0	
I	$-M$	$-d$	q

$w \geqq 0, \quad z \geqq 0, \quad z_0 \geqq 0$ (2.7)

If $q \geqq 0$, $(w = q, z = 0)$ is a solution of the LCP (q, M) and we are done. So assume $q \ngeqq 0$. Determine t to satisfy $\left(\frac{q_t}{d_t}\right) =$ minimum $\left\{ \left(\frac{q_i}{d_i}\right) : i = 1 \text{ to } n \right\}$. Ties for t can be broken arbitrarily. It can be verified that if a pivot step is performed in (2.7), with the column vector of z_0 as the pivot column, and the t^{th} row as the pivot row; the right hand side constants vector becomes nonnegative after this pivot step. So $(w_1, \dots, w_{t-1}, z_0, w_{t+1}, \dots, w_n)$ is a feasible basic vector for (2.7). It is an almost complementary feasible basic vector as defined earlier. Choose z_t as the entering variable into this initial almost complementary feasible basic vector $(w_1, \dots, w_{t-1}, z_0, w_{t+1}, \dots, w_n)$, and continue by choosing entering variables using the complementary pivot rule as before.

We will now illustrate this variant of the complementary pivot algorithm using a numerical example by M. M. Kostreva [4.11].

Example 2.11

Consider the LCP (q, M), where

$$M = \begin{pmatrix} -1.5 & 2 \\ -4 & 4 \end{pmatrix} \qquad q = \begin{pmatrix} -5 \\ 17 \end{pmatrix}$$

Let $d = (5, 16)^T$. We will apply the complementary pivot algorithm on this LCP, using $-d$ as the original column of the artificial variable z_0.

Basic variables	w_1	w_2	z_1	z_2	z_0	q	
	1	0	1.5	-2	$\boxed{-5}$	-5	$t = 1$
	0	1	4	-4	-16	17	
z_0	$-\frac{1}{5}$	0	$-\frac{3}{10}$	$\frac{2}{5}$	1	1	
w_2	$-\frac{16}{5}$	1	$-\frac{8}{10}$	$\frac{12}{5}$	0	33	

The entering variable is z_1. The updated column vector of z_1 in the canonical tableau with respect to the basic vector (z_0, w_2) is nonpositive. So the algorithm ends up in ray termination.

Example 2.12

Consider the LCP (q, M) discussed in Example 2.11. Let $d = e_2 = (1, 1)^T$. We will apply the complementary pivot algorithm on this LCP with $-e_2$ as the original column of the artificial variable z_0.

Basic variables	w_1	w_2	z_1	z_2	z_0		
	1	0	1.5	-2	$\boxed{-1}$	-5	$t = 1$
	0	1	4	-4	-1	17	
z_0	-1	0	$-\frac{3}{2}$	2	1	5	
w_2	-1	1	$\boxed{\frac{5}{2}}$	-2	0	22	
z_0	$-\frac{8}{5}$	$\frac{3}{5}$	0	$\boxed{\frac{4}{5}}$	1	$\frac{91}{5}$	
z_1	$-\frac{2}{5}$	$\frac{2}{5}$	1	$-\frac{4}{5}$	0	$\frac{44}{5}$	
z_2	-2	$\frac{3}{4}$	0	1	$\frac{5}{4}$	$\frac{91}{4}$	
z_1	-2	1	1	0	1	27	

Now we have terminated with a complementary feasible basic vector, and the corresponding solution of the LCP is $w = 0$, $z = (z_1, z_2) = (27, \frac{91}{4})$.

These examples taken from M. M. Kostreva [4.11] illustrate the fact that, given a general LCP (q, M), the complementary pivot algorithm applied on it with a given positive vector d may end up in ray termination; and yet when it is run with a different positive d vector it may terminate with a solution of the LCP. The question of how to find a good d vector seems to be a hard problem, for which no answer is known. There are LCPs which are known to have solutions, and yet when the complementary pivot algorithm is applied on them with any positive d vector, it always ends up in ray termination. See Exercise 2.11.

If M is a copositive plus matrix, and if the complementary pivot algorithm with any positive d vector ends up in ray termination when applied on the LCP (q, M), then it can be proved that the LCP (q, M) has no solution (in fact it can be proved that "$w - Mz = q$" does not even have a nonnegative solution), using arguments exactly similar to those in the proof of Theorem 2.1. Thus any LCP (q, M) where M is a copositive plus matrix, will be processed by the complementary pivot algorithm with any positive d vector.

Exercise

2.2 Prove that when M is an L-matrix or an $L_\star$-matrix, the variant of the complementary pivot algorithm discussed in this section, with any vector $d > 0$ of appropriate dimension, will process the LCP (q, M). (Proofs are similar to those in Section 2.3.2.)

2.3.4 Lexicographic Lemke Algorithm

This variant of the complementary pivot algorithm is known as the **Lexicographic Lemke Algorithm** if the original column vector of the artificial variable z_0 is taken to be $-d = -(\delta^{\mathbf{n}}, \delta^{\mathbf{n-1}}, \ldots, \delta)^T$ where δ is a sufficiently small positive number. It is not necessary to give δ a specific numerical value, but the algorithm can be executed leaving δ as a small positive parameter and remembering that $\delta^{\mathbf{r+1}} < \delta^{\mathbf{r}}$ for any nonnegative r, and that δ is smaller than any positive constant not involving δ. In this case, if D is any square matrix of order n, $Dd = D(\delta^{\mathbf{n}}, \delta^{\mathbf{n-1}}, \ldots, \delta)^T = \delta^{\mathbf{n}} D_{.1} + \delta^{\mathbf{n-1}} D_{.2} + \ldots + \delta D_{.n}$. Using this, it is possible to execute this algorithm without giving the small positive parameter δ any specific value, but using the equivalent lexicographic rules, hence the name.

2.3.5 Another Sufficient Condition for the Complementary Pivot Method to Process the LCP (q, M)

We will now discuss some results due to J. M. Evers [2.11] on another set of sufficient conditions under which the complementary pivot algorithm can be guaranteed to process the LCP (q, M). First, we discuss some lemmas. These lemmas are used later on in Theorem 2.6 to derive some conditions under which the complementary pivot algorithm can be guaranteed to solve the LCP (q, M) when M is a matrix of the form $E + N$ where E is a symmetric PSD matrix, and N is copositive.

Lemma 2.10 *Let $M = E + N$ where E is a symmetric PSD matrix and N is copositive. If the system*

$$\begin{aligned}(E+N)z &\geqq 0\\ cz &> 0\\ z^T(E+N)z &= 0\\ z &\geqq 0\end{aligned} \tag{2.8}$$

has a solution z, then the system

$$\begin{aligned}Ex - N^T y &\geqq c^T\\ y &\geqq 0\end{aligned} \tag{2.9}$$

has no solution (x, y).

Proof. Let $\bar{z}$ be a feasible solution for (2.8). Since E is PSD and N is copositive, $\bar{z}^T(E+N)\bar{z} = 0$ implies that $\bar{z}^T E\bar{z} = \bar{z}^T N\bar{z} = 0$. Since E is symmetric, by Theorem 1.11, $\bar{z}^T E\bar{z} = 0$ implies that $E\bar{z} = 0$. So by (2.8), $N\bar{z} \geqq 0$. Let $(\bar{x}, \bar{y})$ be feasible to (2.9). So $0 \leqq \bar{y}^T N\bar{z} = -\bar{x}^T E\bar{z} + \bar{y}^T N\bar{z}$ (since $E\bar{z} = 0$) $= \bar{z}^T(-E\bar{x} + N^T\bar{y}) \leqq -c\bar{z} < 0$, a contradiction.

□

Lemma 2.11 *If the variant of the complementary pivot algorithm starting with an arbitrary positive vector d for the column of the artificial variable z_0 in the original tableau ends up in ray termination when applied on the LCP (q, M) in which M is copositive, there exists a $\bar{z}$ satisfying*

$$\begin{aligned}M\bar{z} &\geqq 0\\ q^T\bar{z} &< 0\\ \bar{z}^T M\bar{z} &= 0\\ \bar{z} &\geqq 0\end{aligned} \tag{2.10}$$

Proof. Let the terminal extreme half-line obtained in the algorithm be $\{\,(w, z, z_0) = (w^k + \lambda w^h, z^k + \lambda z^h, z_0^k + \lambda z_0^h) : \quad \lambda \geqq 0\,\}$ where (w^k, z^k, z_0^k) is the BFS of (2.7) and (w^h, z^h, z_0^h) is a homogeneous solution corresponding to (2.7), that is

$$\begin{aligned}w^h - Mz^h - dz_0^h &= 0\\ w^h, z^h, z_0^h &\geqq 0\\ (w^h, z^h, z_0^h) &\neq 0\end{aligned} \tag{2.11}$$

and every point on the terminal extreme half-line satisfies the complementarity constraint, that is

$$(w^k + \lambda w^h)^T(z^k + \lambda z^h) = 0 \quad \text{for all } \lambda \geqq 0\ . \tag{2.12}$$

Clearly $z^h \neq 0$ (otherwise the terminal extreme half-line is the intial one, a contradiction), so $z^h \geq 0$. By complementarity, we have $(w^h)^T z^h = 0$, from (2.11) this implies that $(z^h)^T M z^h = -d^T z^h z_0^h \leqq 0$, (since $d > 0$, $z^h \geq 0$ implies that $d^T z^h > 0$) which implies by the copositivity of M, that $(z^h)^T M z^h = 0$ and $z_0^h = 0$. Using this in (2.11) we conclude that

$$M z^h = w^h \geqq 0 \ . \tag{2.13}$$

Since (w^k, z^k, z_0^k) is a BFS of (2.7) we have $w^k = Mz^k + dz_0^k + q$. Using this and (2.13) in (2.12) we get, for all $\lambda \geqq 0$, $(z^k+\lambda z^h)^T dz_0^k + (z^k+\lambda z^h)^T q = -(z^k+\lambda z^h)^T M(z^k+\lambda z^h) \leqq 0$ (since M is copositive and $z^k + \lambda z^h \geqq 0$). Make $\lambda > 0$, divide this inequality by λ and take the limit as λ tends to $+\infty$. This leads to

$$(z^h)^T dz_0^k + (z^h)^T q \leqq 0 \ . \tag{2.14}$$

But $z_0^k > 0$ (otherwise (w^k, z^k) will be a solution to the LCP (q, M), contradicting the hypothesis that the algorithm terminated with ray termination without leading to a solution of the LCP), $d > 0$, $z^h \geq 0$. Using these facts in (2.14) we conclude that $q^T z^h < 0$. All these facts imply that $z^h = \bar{z}$ satisfies (2.10).

□

Theorem 2.6 *Let $M = E + N$ where E is a symmetric PSD matrix and N is copositive. If the system (2.9) with $c^T = -q$ has a solution (x, y) there exists no secondary ray, and the complementary pivot algorithm terminates with a solution of the LCP (q, M).*

Proof. Follows from Lemma 2.10 and 2.11.

□

Corollary 2.1 *Putting $E = 0$ in Theorem 2.6, we conclude that if N is copositive, for every $u \geqq 0$, $v \geqq 0$ in $\mathbf{R}^n$, there exists $w, z \in \mathbf{R}^n$ satisfying*

$$\begin{aligned} Nz - w &= -N^T u - v \\ z, w &\geqq 0, \quad z^T w = 0 \ . \end{aligned}$$

2.3.6 Unboundedness of the Objective Function

Consider a mathematical program in which an objective function $f(x)$ is required to be minimized subject to constraints on the decision variables $x = (x_1, \ldots, x_n)^T$. This problem is said to be **unbounded below** if the set of feasible solutions of the problem is nonempty and $f(x)$ is not bounded below on it, that is, iff there exists an infinite sequence of feasible solutions $\{x^1, \ldots, x^r, \ldots\}$ such that $f(x^r)$ diverges to $-\infty$ as r goes to $+\infty$.

It is well known that if a linear program is unbounded below, there exists a feasible half-line (in fact an extreme half-line of the set of feasible solutions, see [2.26]) along which the objective function diverges to $-\infty$. This half-line is of the form

$\{ x^0 + \lambda x^1 : \lambda \geqq 0 \}$ satisfying the property that $x^0 + \lambda x^1$ is a feasible solution for all $\lambda \geqq 0$, and the objective value at $x^0 + \lambda x^1$ diverges to $-\infty$ as λ goes to $+\infty$. This property may not hold in general convex programming problems, that is, problems in which a convex function is required to be minimized over a closed convex set. Consider the following example due to R. Smith and K. G. Murty.

$$\begin{aligned} \text{Minimize} \quad & -x_1 \\ \text{Subject to} \quad & x_2 - x_1^2 \geqq 0 \ . \\ & x_1, x_2 \geqq 0 \end{aligned} \tag{2.15}$$

The set of feasible solutions of this problem is drawn in Figure 2.3.

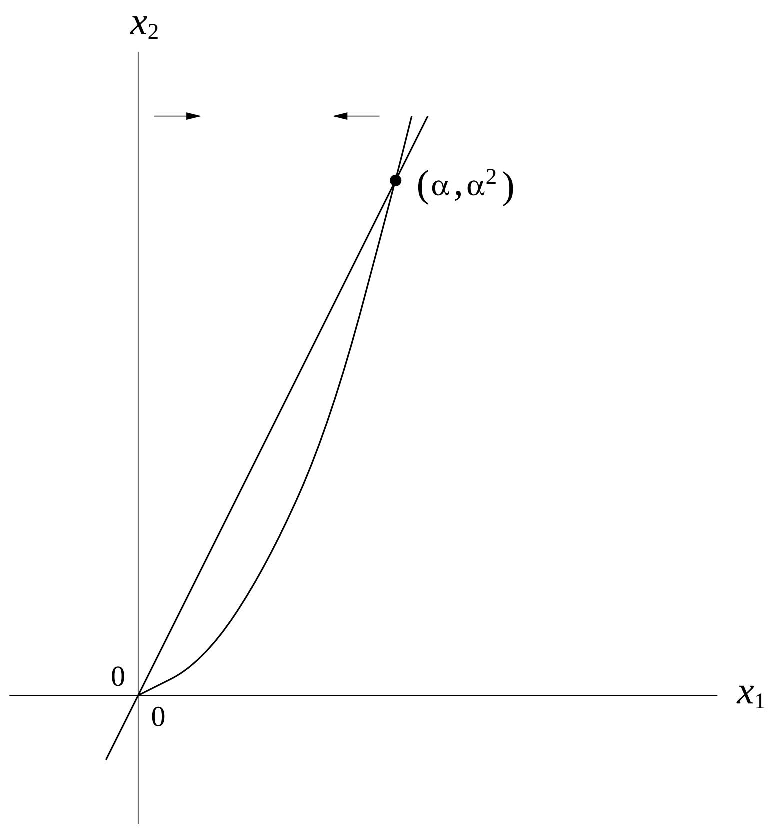

Figure 2.3 The feasible region for (2.15) is the area between the x_2 axis and the parabola. For every $\alpha > 0$, the straight line $x_2 - \alpha x_1 = 0$ intersects the parabola at exactly two points.

The equation $x_2 - x_1^2 = 0$ represents a parabola in the x_1, x_2-Cartesian plane. For every $\alpha > 0$, the straight line $x_2 - \alpha x_1 = 0$ intersects this parabola at the two points $(0, 0)$ and (α, α^2). These facts clearly imply that even though $-x_1$ is unbounded below in (2.15), there exists no half-line in the feasible region along which $-x_1$ diverges to $-\infty$.

However, for convex quadratic programs (i.e., problems of the form (1.11) in which the matrix D is PSD) we have the following theorem.

Theorem 2.7 *Consider the quadratic program (1.11) in which D is* PSD *and symmetric. Suppose (1.11) is feasible and that $Q(x)$ is unbounded below in it. Then there exists a feasible half-line for (1.11) along which $Q(x)$ diverges to $-\infty$. Such a half-line can be constructed from the data in the terminal tableau obtained when the complementary pivot algorithm is applied to solve the corresponding LCP (1.19).*

Proof. For any positive integer r, let e_r denote the column vector in $\mathbf{R}^r$, all of whose entries are 1. By Theorem 2.1, when the complementary pivot algorithm is applied to solve (1.19) it must end in ray termination. When this happens, by the results established in the proof of Theorem 2.1, we get vectors (u^k, v^k, x^k, y^k) and (u^h, v^h, x^h, y^h) satisfying

$$\begin{pmatrix} u^k \\ v^k \end{pmatrix} - \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \begin{pmatrix} e_n \\ e_m \end{pmatrix} z_0^k = \begin{pmatrix} c^T \\ -b \end{pmatrix} \tag{2.16}$$

$$u^k, v^k, x^k, y^k \geqq 0, \quad (u^k)^T x^k = (v^k)^T y^k = 0, \quad z_0^k > 0 \ .$$

$$\begin{pmatrix} u^h \\ v^h \end{pmatrix} - \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^h \\ y^h \end{pmatrix} = 0 \tag{2.17}$$

$$u^h, v^h, x^h, y^h \geqq 0, \quad (u^h)^T x^h = (v^h)^T y^h = 0, \quad (x^h, y^h) \geq 0 \ .$$

$$(u^k)^T x^h = (v^k)^T y^h = (u^h)^T x^k = (v^h)^T y^k = 0 \ . \tag{2.18}$$

$$\left((x^h)^T, (y^h)^T\right) \begin{pmatrix} c^T \\ -b \end{pmatrix} < 0 \ . \tag{2.19}$$

So we have $v^h = Ax^h$ and $0 = (y^h)^T v^h = (y^h)^T Ax^h$. We also have $u^h - Dx^h + A^T y^h = 0$, and hence $0 = (x^h)^T u^h = (x^h)^T Dx^h - (x^h)^T A^T y^h = (x^h)^T Dx^h$. Since D is PSD and symmetric by Theorem 1.11, this implies that $Dx^h = 0$. So $A^T y^h = -u^h \leqq 0$, that is $(y^h)^T A \leqq 0$. From (2.16), $-b = v^k - Ax^k - e_m z_0^k$, $z_0^k > 0$. So $(-b^T y^h) = (v^k)^T y^h - (x^k)^T A^T y^h - z_0^k e_m^T y^h = -(x^k)^T A^T y^h - z_0^k e_m^T y^h = -(x^k)^T(-u^h) - z_0^k e_m^T y^h = -z_0^k (e_m^T y^h) \leqq 0$ since $z_0^k > 0$ and $y^h \geq 0$. So $b^T y^h = z_0^k (e_m^T y^h) \geqq 0$.

If $b^T y^h > 0$, (1.11) must be infeasible. To see this, suppose $\hat{x}$ is a feasible solution of (1.11). Then $A\hat{x} \geqq b$, $\hat{x} \geqq 0$. So $(y^h)^T A\hat{x} \geqq (y^h)^T b$. But it has been established earlier that $(y^h)^T A = -(u^h)^T \leqq 0$. Using this in the above, we have, $(y^h)^T b \leqq (y^h)^T A\hat{x} = -(u^h)^T \hat{x} \leqq 0$ (since both u^h and $\hat{x}$ are $\geqq 0$), and this contradicts the fact that $(y^h)^T b > 0$.

So, under the hypothesis that (1.11) is feasible, we must have $b^T y^h = 0$. In this case, from (2.19) we have $cx^h < 0$. From earlier facts we also have $Ax^h = v^h \geqq 0$, $x^h \geqq 0$ and $Dx^h = 0$. Let $\tilde{x}$ be any feasible solution to (1.11). These facts together imply that $\tilde{x} + \lambda x^h$ is also feasible to (1.11) for any $\lambda \geqq 0$ and $Q(\tilde{x} + \lambda x^h) = Q(\tilde{x}) + \lambda(cx^h)$ (this equation follows from the fact that $Dx^h = 0$) diverges to $-\infty$ as

λ tends to $+\infty$. Thus in this case, $\{\,\tilde{x} + \lambda x^h : \quad \lambda \geqq 0\,\}$ is a feasible half-line along which $Q(x)$ diverges to $-\infty$.

□

Since D is assumed to be PSD, we have $x^T D x \geqq 0$ for all $x \in \mathbf{R}^n$. So, in this case, if $Q(x)$ is unbounded below in (1.11), the linear function cx must be unbounded below on the set of feasible solutions of (1.11), and this is exactly what happens on the half-line constructed above.

If ray termination occurs in the complementary pivot algorithm applied on (1.19) when D is PSD, we get the vectors satisfying (2.16), (2.17), (2.18) and (2.19) from the terminal tableau. If $b^T y^h > 0$, we have shown above that (1.11) must be infeasible. On the other hand, if $b^T y^h = 0$, $Q(x)$ is unbounded below in (1.11) if (1.11) is feasible. At this stage, whether (1.11) is feasible or not can be determined by using Phase I of the Simplex Method or some other algorithm to find a feasible solution of the system $Ax \geqq b$, $x \geqq 0$.

With a slight modification in the formulation of a convex quadratic program as an LCP, we can make sure that at termination of the complementary pivot algorithm applied to this LCP, if ray termination has occurred, then either a proof of infeasibility or a feasible extreme half-line along which the objective function is unbounded, are readily available, without having to do any additional work. See Section 2.9.2 for this version.

2.3.7 Some Results on Complementary BFSs

Theorem 2.8 *If the LCP (q, M) has a complementary feasible solution, then it has a complementary feasible solution which is a BFS of*

$$\begin{aligned} w - Mz &= q \\ w \geqq 0, \quad z &\geqq 0 \,. \end{aligned} \tag{2.20}$$

Proof. Let $(\bar{w}, \bar{z})$ be a complementary feasible solution for the LCP (q, M). So for each $j = 1$ to n, we have $\bar{w}_j \bar{z}_j = 0$. If $(\bar{w}, \bar{z})$ is a BFS of (2.20), we are done. Otherwise, using the algorithm discussed in Section 3.5.4 of [2.26], starting with $(\bar{w}, \bar{z})$, we can obtain a BFS $(\hat{w}, \hat{z})$ of (2.20) satisfying the property that the set of variables which have positive values in $(\hat{w}, \hat{z})$, is a subset of the set of variables which have positive values in $(\bar{w}, \bar{z})$. So $\hat{w}_j \hat{z}_j = 0$, for $j = 1$ to n. Hence $(\hat{w}, \hat{z})$ is a complementary feasible solution of the LCP (q, M) and it is also a BFS of (2.20).

□

Note 2.1 The above theorem does not guarantee that whenever the LCP (q, M) has a complementary feasible solution, there exists a complementary feasible basis for (2.20). See Exercise 1.10.

Theorem 2.9 *Suppose M is nondegenerate. If $(\bar{w}, \bar{z})$ is a complementary feasible solution for the LCP (q, M), the set of column vectors $\{ I_{.j} : j$ such that $\bar{w}_j > 0 \} \cup \{ -M_{.j} : j$ such that $\bar{z}_j > 0 \}$ is linearly independent. Also, in this case, define a vector of variables $y = (y_1, \ldots, y_n)$ by*

$$y_j = \begin{cases} w_j \ , & \text{if } \bar{w}_j > 0 \\ z_j \ , & \text{if } \bar{z}_j > 0 \\ \text{either } w_j \text{ or } z_j \text{ choosen arbitrarily} \ , & \text{if both } \bar{w}_j \text{ and } \bar{z}_j \text{ are } 0 \ . \end{cases}$$

Then y is a complementary feasible basic vector for (2.20).

Proof. From Corollary 3.1 of Chapter 3, when M is nondegenerate, every complementary vector is basic. Since $(\bar{w}, \bar{z})$ is a complementary feasible solution, this implies that the set $\{ I_{.j} : j$ such that $\bar{w}_j > 0 \} \cup \{ -M_{.j} : j$ such that $\bar{z}_j > 0 \}$ is linearly independent. Also from this result, y is a complementary basic vector, and the BFS of (2.20) with y, as the basic vector is $(\bar{w}, \bar{z})$, and hence y is a complementary feasible basic vector.

□

Theorem 2.10 *If M is* PSD *or copositive plus, and (2.20) is feasible, then there exists a complementary feasible basic vector for (2.20).*

Proof. When the complementary pivot algorithm is applied to solve the LCP (q, M), it terminates with a complementary feasible basic vector when M is copositive plus and (2.20) is feasible, by Theorem 2.1.

□

2.4 A METHOD OF CARRYING OUT THE COMPLEMENTARY PIVOT ALGORITHM WITHOUT INTRODUCING ANY ARTIFICIAL VARIABLES, UNDER CERTAIN CONDITIONS

Consider the LCP (q, M) of order n, suppose the matrix M satisfies the condition :

> there exists a column vector of M in which all the entries are strictly positive. (2.21)

Then a variant of the complementary pivot algorithm which uses no artificial variable at all, can be applied on the LCP (q, M). We discuss it here. The original tableau for this version of the algorithm is :

w	z	
I	$-M$	q

$$w \geqq 0, \quad z \geqq 0 \tag{2.22}$$

As before, we assume that $q \not\geqq 0$. Let s be such that $M_{.s} > 0$. So the column vector associated with z_s is strictly negative in (2.22). Hence the variable z_s can be made to play the same role as that of the artificial variable z_0 in versions of the complementary pivot algorithm discussed earlier, and thus there is no need to introduce the artificial variable. Determine t to satisfy $\left(\frac{q_t}{m_{ts}}\right) =$ minimum $\left\{ \left(\frac{q_i}{m_{is}}\right) : i = 1 \text{ to } n \right\}$. Ties for t can be broken arbitrarily. When a pivot step is carried out in (2.22) with the column of z_s as the pivot column and row t as the pivot row, the right hand side constants vector becomes nonnegative after this pivot step (this follows because $-m_{is} < 0$ for all i and by the choice of t). Hence, $(w_1, \ldots, w_{t-1}, z_s, w_{t+1}, \ldots, w_n)$ is a feasible basic vector for (2.22), and if $s = t$, it is a complementary feasible basic vector and the solution corresponding to it is a solution of the LCP (q, M), terminate. If $s \neq t$, the feasible basic vector $(w_1, \ldots, w_{t-1}, z_s, w_{t+1}, \ldots, w_n)$ for (2.22) satisfies the following properties :

i) It contains exactly one basic variable from the complementary pair (w_i, z_i) for $n - 2$ values of i (namely $i \neq s$, t here).

ii) It contains both the variables from a fixed complementary pair (namely (w_s, z_s) here), as basic variables.

iii) There exists exactly one complementary pair both the variables in which are not contained in this basic vector (namely (w_t, z_t) here).

The complementary pair of variables identified by property (iii), both of which are not contained in the basic vector, is known as the **left out complementary pair of variables** in the present basic vector.

For carrying out this version of the complementary pivot algorithm, any feasible basic vector for (2.22) satisfying (i), (ii), (iii) is known as an **almost complementary feasible basic vector**. All the basic vectors obtained during this version of the algorithm, with the possible exception of the terminal one (which may be a complementary basic vector), will be such almost complementary feasible basic vectors, and the complementary pair in property (ii) both of whose variables are basic, will be the same for all of them.

In the canonical tableau of (2.22) with respect to the initial almost complementary feasible basic vector, the updated column vector of w_t can be verified to be strictly negative (because the pivot column in the original tableau, $-M_{.s}$, is strictly negative). Hence if w_t is selected as the entering variable into the initial basic vector, an almost complementary extreme half-line is generated. Hence the initial almost complementary BFS of (2.22) is at the end of an almost complementary ray.

The algorithm chooses z_t as the entering variable into the initial almost complementary feasible basic vector $(w_1, \ldots, w_{t-1}, z_s, w_{t+1}, \ldots, w_n)$. In all subsequent steps, the entering variable is uniquely determined by the complementary pivot rule, that is, the entering variable in a step is the complement of the dropping variable in the previous step. The algorithm can terminate in two possible ways :

1. At some stage one of the variables form the complementary pair (w_s, z_s) (this is the pair specified in property (ii) of the almost complementary feasible basic vectors obtained during the algorithm) drops out of the basic vector, or becomes equal to zero in the BFS of (2.22). The BFS of (2.22) at that stage is a solution of the LCP (q, M).
2. At some stage of the algorithm both the variables in the complementary pair (w_s, z_s) may be strictly positive in the BFS, and the pivot column in that stage may turn out to be nonpositive, and in this case the algorithm terminates with another almost complementary ray. This is **ray termination**.

When ray termination occurs, the algorithm bas been unable to solve the LCP (q, M).

Example 2.13

Consider the LCP (q, M), where

$$M = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \qquad q = \begin{pmatrix} -4 \\ -5 \\ -1 \end{pmatrix}$$

All the column vectors of M are strictly positive here. We will illustrate the algorithm on this problem using $s = 3$.

Original Tableau

w_1	w_2	w_3	z_1	z_2	z_3	q
1	0	0	-2	-1	-1	-4
0	1	0	-1	-2	$\boxed{-1}$	-5
0	0	1	-1	-1	-2	-1

$-M_{.3} < 0$. The minimum $\{\frac{-4}{1}, \frac{-5}{1}, \frac{-1}{2}\} = -5$, and hence $t = 2$ here. So the pivot row is row 2, and the pivot element for the pivot operation to get the initial almost complementary feasible basic vector is inside a box in the original tableau. Applying the algorithm we get the following canonical tableaus:

Basic variables	w_1	w_2	w_3	z_1	z_2	z_3	q	Ratios
w_1	1	−1	0	−1	$\boxed{1}$	0	1	$\frac{1}{1}$ *Min.*
z_3	0	−1	0	1	2	1	5	$\frac{5}{2}$
w_3	0	−2	1	1	3	0	9	$\frac{9}{3}$
z_2	1	−1	0	−1	1	0	1	
z_3	−2	1	0	$\boxed{3}$	0	1	3	$\frac{3}{3}$ *Min.*
w_3	−3	1	1	4	0	0	6	$\frac{6}{4}$
z_2	$\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	$\frac{1}{3}$	2	
z_1	$-\frac{2}{3}$	$\frac{1}{3}$	0	1	0	$\frac{1}{3}$	1	
w_3	$-\frac{4}{3}$	$-\frac{1}{3}$	1	0	0	$-\frac{4}{3}$	2	

So the solution of this LCP is $w = (w_1, w_2, w_3) = (0, 0, 2)$; $z = (z_1, z_2, z_3) = (1, 2, 0)$.

Exercise

2.3 Show that the version of the complementary pivot algorithm discussed in this section can be used to process all LCPs (q, M) in which M is copositive plus and at least one of its columns is strictly positive. In this case, prove that ray termination cannot occur, and that the algorithm will terminate with a complementary feasible basic vector for the problem.

2.5 TO FIND AN EQUILIBRIUM PAIR OF STRATEGIES FOR A BIMATRIX GAME USING THE COMPLEMENTARY PIVOT ALGORITHM

The LCP corresponding to the problem of finding an equilibrium pair of strategies in a bimatrix game is (1.42), where A, B^T are positive matrices. The original tableau for this problem is :

u	v	ξ	η	
I_m	0	0	$-A$	$-e_m$
0	I_N	$-B^T$	0	$-e_N$

$u \geqq 0, \quad v \geqq 0, \quad \xi \geqq 0, \quad \eta \geqq 0$ (2.23)

where for any r, I_r denotes the identity matrix of order r. The complementary pairs of variables in this problem are (u_i, ξ_i), $i = 1$ to m, and (v_j, η_j), $j = 1$ to N.

We leave it to the reader to verify that when the complementary pivot algorithm discussed in Section 2.2 is applied on this problem, it ends up in ray termination right after obtaining the initial almost complementary feasible basic vector. However, it turns out that the variant of the complementary pivot algorithm discussed in Section 2.4 can be applied to this problem, and when it is applied it works. We discuss the application of this version of the algorithm here.

So, here, an **almost complementary feasible basic vector** for (2.23), is defined to be a feasible basic vector that contains exactly one basic variable from each complementary pair excepting two pairs. Both variables of one of these pairs are basic variables, and both variables in the other pair are nonbasic variables. These are the conditions for almost complementarity (i), (ii), (iii), discussed in Section 2.4.

The column vectors of the variables ξ_i, η_j, in (2.23) are all nonpositive, but none of them is strictly negative. But, because of their special structure, an almost complementary feasible basic vector for (2.23) can be constructed by the following special procedure.

Initially make the variable ξ_1 a basic variable and the variables $\xi_2, \ldots, \xi_m$ nonbasic variables. Make ξ_1 equal to ξ_1^0, the smallest positive number such that $v^0 = -e_N + (B^T)_{.1}\xi_1^0 \geq 0$. At least one of the components in v^0, say, v_r^0 is zero. Make v_r a nonbasic variable too. The complement of v_r is η_r. Make the value of η_r to be the smallest positive value, η_r^0, such that $u^0 = A_{.r}\eta_r^0 - e_m \geq 0$. At least one of the components in u^0, say u_s^0 is 0. If $s = 1$, the basic vector $(u_2, \ldots, u_m, v_1, \ldots, v_{r-1}, v_{r+1}, \ldots, v_N, \xi_1, \eta_r)$ is a complementary feasible basic vector, and the feasible solution corresponding to it is a solution of the LCP (1.42), terminate.

If $s \neq 1$, the basic vector, $(u_1, \ldots, u_{s-1}, u_{s+1}, \ldots, u_m, v_1, \ldots, v_{r-1}, v_{r+1}, \ldots, v_N, \xi_1, \eta_r)$ is a feasible basic vector. Both the variables in the complementary pair (u_1, ξ_1) are basic variables in it. Both variables in the complementary pair (u_s, ξ_s) are nonbasic variables. And this basic vector contains exactly one basic variable from every complementary pair in (2.23), excepting (u_1, ξ_1), (u_s, ξ_s). Hence this initial basic vector is an almost complementary feasible basic vector. All the basic vectors obtained during the algorithm (excepting the terminal complementary feasible basic vector) will be almost complementary feasible basic vectors containing both the variables in the pair (u_1, ξ_1) as basic variables.

When u_s is made as the entering variable into the initial basic vector, an almost complementary extreme half-line is generated. Hence the BFS of (2.23) with respect to the initial basic vector is an almost complementary BFS at the end of an almost complementary extreme half-line.

The algorithm begins by taking ξ_s as the entering variable into the initial basic vector. In all subsequent steps, the entering variable is picked by the complementary pivot rule. The algorithm terminates when one of the variables in the pair (u_1, ξ_1) drops from the basic vector. It can be proved that temination occurs after at most a finite number of pivots. The terminal basis is a complementary feasible basis. In this algorithm if degeneracy is encountered, its should be resolved using the lexico minimum ratio rule (see Section 2.2.8).

Example 2.14

We will solve the LCP (1.43) corresponding to the 2 person game in Example 1.9. In tableau form it is

u_1	u_2	v_1	v_2	v_3	ξ_1	ξ_2	η_1	η_2	η_3	q
1	0	0	0	0	0	0	−2	−2	−1	−1
0	1	0	0	0	0	0	$\boxed{-1}$	−2	−2	−1
0	0	1	0	0	$\boxed{-1}$	−2	0	0	0	−1
0	0	0	1	0	−3	−1	0	0	0	−1
0	0	0	0	1	−2	−3	0	0	0	−1

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

Making $\xi_2 = 0$, the smallest value of ξ_1 that will yield nonnegative values to the v's is 1. When $\xi_2 = 0$, $\xi_1 = 1$ the value of v_1 is 0. Hence, v_1 will be made a nonbasic variable. The complement of v_1 is η_1. So make η_2 and η_3 nonbasic variables. The smallest value of η_1 that will make the u's nonnegative is $\eta_1 = 1$. When $\eta_1 = 1$ with $\eta_2 = \eta_3 = 0$, u_2 becomes equal to 0. So make u_2 a nonbasic variable. The canonical tableau with respect to the initial basic vector is therefore obtained as below by performing pivots in the columns of ξ_1 and η_1, with the elements inside a box as pivot elements.

Basic variables	u_1	u_2	v_1	v_2	v_3	ξ_1	ξ_2	η_1	η_2	η_3	q	Ratios
u_1	1	−2	0	0	0	0	0	0	2	3	1	
η_1	0	−1	0	0	0	0	0	1	2	2	1	
ξ_1	0	0	−1	0	0	1	2	0	0	0	1	$\frac{1}{2}$
v_2	0	0	−3	1	0	0	$\boxed{5}$	0	0	0	2	$\frac{2}{5}$ *Min.*
v_3	0	0	−2	0	1	0	1	0	0	0	1	$\frac{1}{1}$

The algorithm continues by selecting ξ_2, the complement of u_2, as the entering variable. v_2 drops from the basic vector.

Basic variables	u_1	u_2	v_1	v_2	v_3	ξ_1	ξ_2	η_1	η_2	η_3	q
u_1	1	-2	0	0	0	0	0	0	$\boxed{2}$	3	1
η_1	0	-1	0	0	0	0	0	1	2	2	1
ξ_1	0	0	$\frac{1}{5}$	$-\frac{2}{5}$	0	1	0	0	0	0	$\frac{1}{5}$
ξ_2	0	0	$-\frac{3}{5}$	$\frac{1}{5}$	0	0	1	0	0	0	$\frac{2}{5}$
v_3	0	0	$-\frac{7}{5}$	$-\frac{1}{5}$	1	0	0	0	0	0	$\frac{3}{5}$

Since v_2 has dropped from the basic vector, its complement η_2 is the next entering variable. There is a tie in the minimum ratio when η_2 is the entering variable, since it can replace either u_1 or η_1 from the basic vector. Such ties should be resolved by the lexico minimum ratio test, but in this case we will let u_1 drop from the basic vector, since that leads to a complementary feasible basis to the problem.

Basic variables	u_1	u_2	v_1	v_2	v_3	ξ_1	ξ_2	η_1	η_2	η_3	q
η_2	$\frac{1}{2}$	-1	0	0	0	0	0	0	1	$\frac{3}{2}$	$\frac{1}{2}$
η_1	-1	1	0	0	0	0	0	1	0	-1	0
ξ_1	0	0	$\frac{1}{5}$	$-\frac{2}{5}$	0	1	0	0	0	0	$\frac{1}{5}$
ξ_2	0	0	$-\frac{3}{5}$	$\frac{1}{5}$	0	0	1	0	0	0	$\frac{2}{5}$
v_3	0	0	$-\frac{7}{5}$	$-\frac{1}{5}$	1	0	0	0	0	0	$\frac{3}{5}$

The present basic vector is a complementary feasible basic vector. The solution $(u_1, u_2; v_1, v_2, v_3; \xi_1, \xi_2; \eta_1, \eta_2, \eta_3) = (0, 0; 0, 0, \frac{3}{5}; \frac{1}{5}, \frac{2}{5}; 0, \frac{1}{2}, 0)$ is a solution of the LCP. In this solution $\xi_1 + \xi_2 = \frac{3}{5}$ and $\eta_1 + \eta_2 + \eta_3 = \frac{1}{2}$. Hence the probability vector $x = \frac{\xi}{(\sum \xi_i)} = \left(\frac{1}{3}, \frac{2}{3}\right)^T$ and $y = \frac{\eta}{(\sum \eta_j)} = (0, 1, 0)^T$ constitute an equilibrium pair of strategies for this game.

Theorem 2.11 *If the lexicographic minimum ratio rule is used to determine the dropping variable in each pivot step (this is to prevent cycling under degeneracy) of the complementary pivot algorithm discussed above for solving (1.42), it terminates in a finite number of pivot steps with a complementary feasible solution.*

Proof. The original tableau for this problem is (2.23), in which $A > 0$, $B^T > 0$, by the manner in which the problem is formulated. In the algorithm discussed above for this problem, both variables from exactly one complementary pair are nonbasic in every almost complementary feasible basic vector obtained, and this pair is known as the **left out complementary pair of variables**. The left out complementary pair may be different in the various almost complementary feasible basic vectors obtained during the algorithm, but the complementary pair both of whose variables are basic, remains the same in all of them.

Let $(u_1, \ldots, u_{s-1}, u_{s+1}, \ldots, u_m, v_1, \ldots, v_{r-1}, v_{r+1}, \ldots, v_N, \xi_1, \eta_r)$ be the initial almost complementary feasible basic vector obtained in the algorithm, by the special procedure discussed above. Let the initial tableau be the canonical tableau of (2.23) with respect to the initial almost complementary feasible basic vector. In this, the left out complementary pair is (u_s, ξ_s) both of which are nonbasic at present. Let

$$u^1 = (u^1_i), \quad u^1_i = -1 + \big(\frac{a_{ir}}{a_{sr}}\big), \qquad \text{for } i \neq s, \quad u^1_s = 0 \ .$$

$$v^1 = (v^1_j), \quad v^1_j = -1 + \big(\frac{b_{1j}}{b_{1r}}\big), \qquad \text{for } j \neq r, \quad v^1_r = 0 \ .$$

$$\xi^1 = \big(\frac{1}{b_{1r}}, 0, \ldots, 0\big)$$

$$\eta^1 = (\eta^1_j), \quad \eta^1_j = 0, \qquad \text{for } j \neq r, \quad \eta^1_r = \frac{1}{a_{sr}} \ .$$

$$\bar{u}^h = (\bar{u}^h_i), \quad \bar{u}^h_i = \big(\frac{a_{is}}{a_{ir}}\big), \qquad \text{for } i \neq s, \quad \bar{u}^h_s = 1 \ .$$

$$\bar{v}^h = 0, \quad \bar{\xi}^h = 0$$

$$\bar{\eta}^h = (\bar{\eta}^h_j), \quad \bar{\eta}^h_j = 0, \qquad \text{for } j \neq r, \quad \bar{\eta}^h_r = \big(\frac{1}{a_{sr}}\big) \ .$$

The present BFS can be verified to be $(u^1, v^1, \xi^1, \eta^1)$. It can also be verified that $(\bar{u}^h, \bar{v}^h, \bar{\xi}^h, \bar{\eta}^h)$ is a homogeneous solution corresponding to the initial tableau, and that the initial almost complementary extreme half-line generated when u_s is brought into the basic vector in the initial tableau is $\{ (u^1, v^1, \xi^1, \eta^1) + \lambda(\bar{u}^h, \bar{v}^h, \bar{\xi}^h, \bar{\eta}^h) : \lambda \geqq 0 \}$.

The algorithm begins by bringing the nonbasic variable ξ_s into the basic vector in the initial tableau, and continues by using the complementary pivot rule to choose the entering variable and the lexico-minimum ratio rule to choose the dropping variable in each step.

Let B be the basis consisting of the columns of the basic variables in the initial tableau (not the original tableau), in a step of this procedure and let $\beta = (\beta_{ij}) = B^{-1}$. Let $\bar{q}$ be the updated right hand side constants vector in this step. If u_1 or ξ_1, is eligible to be a dropping variable in this step by the usual minimum ratio test, it is choosen as the dropping variable, and the pivot step is carried out, leading to a complementary feasible basic vector for the problem. If both u_1 and ξ_1 are ineligible to be dropping variables in this step, the lexico minimum ratio rule chooses the dropping variable so that the pivot row corresponds to the row which is the lexico minimum $\{ \frac{(\bar{q}_i, \beta_{i.})}{p_{it}} : i \text{ such that } p_{it} > 0 \}$ where $p = (p_{1t}, \ldots, p_{m+N,t})^T$ is the pivot column (updated column of the entering variable) in this step. This lexico minimum ratio rule determines the dropping variable uniquely and unambiguously in each pivot step.

In each almost complementary feasible basic vector, obtained during the algorithm, there is exactly one left out complementary pair of variables; and hence it can have at most two adjacent almost complementary feasible basic vectors, that can be obtained by bringing one variable from the left out complementary pair into it.

The left out complementary pair in the initial almost complementary feasible basic vector is (u_s, ξ_s), and when u_s is brought into the initial almost complementary feasible basic vector, we obtain the initial almost complementary extreme half-line. So the only manner in which the almost complementary path can be continued from the initial almost complementary BFS is by bringing ξ_s into the basic vector. The updated column of ξ_s in the initial tableau can be verified to contain at least one positive entry. Hence when ξ_s is brought into the initial basic vector, we get an adjacent almost complementary feasible basic vector, and the almost complementary path continues uniquely and unambiguously from there. Each almost complementary feasible basic vector has at most two adjacent ones, from one of them we arrive at this basic vector; we move to the other when we leave this basic vector. These facts, and the perturbation interpretation of the lexico minimum ratio rule imply that an almost complementary feasible basic vector obtained in the algorithm can never reappear later on. Since there are at most a finite number of almost complementary feasible basic vectors, the algorithm must terminate in a finite number of pivot steps. If it terminates by obtaining a complementary feasible basic vector, the BFS corresponding to it is a solution of the LCP (1.42) and we are done. The only other possibility in which the algorithm can terminate is if the updated column vector of the entering variable in some step has no positive entries in it, in which case we get a terminal almost complementary extreme half-line (this is the **ray termination** discussed earlier). We will now show that this second possibility (ray termination) cannot occur in this algorithm.

Suppose ray termination occurs in pivot step k. Let the almost complementary BFS in this step be $(u^k, v^k, \xi^k, \eta^k)$ and let the terminal extreme half-line be: $\{\, (u^k, v^k, \xi^k, \eta^k) + \lambda(u^h, v^h, \xi^h, \eta^h) : \lambda \geqq 0 \,\}$. From this and from the almost compelementary property being maintained in the algorithm, we have :

$$\begin{pmatrix} u^k + \lambda u^h \\ v^k + \lambda v^h \end{pmatrix} - \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \xi^k + \lambda \xi^h \\ \eta^k + \lambda \eta^h \end{pmatrix} = \begin{pmatrix} -e_m \\ -e_N \end{pmatrix} \tag{2.24}$$

$$(u_i^k + \lambda u_i^h)(\xi_i^k + \lambda \xi_i^h) = 0 \quad \text{for all } i \neq 1 \tag{2.25}$$

$$(v_j^k + \lambda v_j^h)(\eta_j^k + \lambda \eta_j^h) = 0 \quad \text{for all } j \tag{2.26}$$

$$u^k, v^k, \xi^k, \eta^k, u^h, v^h, \xi^h, \eta^h \geqq 0 \tag{2.27}$$

for all $\lambda \geqq 0$. $(u^h, v^h, \xi^h, \eta^h)$ is a homogeneous solution satisfying the nonnegativity restrictions and

$$\begin{pmatrix} u^h \\ v^h \end{pmatrix} - \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \xi^h \\ \eta^h \end{pmatrix} = 0 \,.$$

That is :

$$\begin{aligned} u^h &= A\eta^h \\ v^h &= B^T \xi^h \,. \end{aligned} \tag{2.28}$$

We also have $(u^h, v^h, \xi^h, \eta^h) \neq 0$, which implies by (2.27) and (2.28) that $(\xi^h, \eta^h) \geq 0$. Now suppose $\xi^h \neq 0$. So $\xi^h \geq 0$. Since $B > 0$, this implies by (2.28) that $v^h = B^T \xi^h > 0$. From (2.26) this implies that $\eta_j^k + \lambda \eta_j^h = 0$ for all j and for all $\lambda \geqq 0$. From (2.24) this implies that $(u^k + \lambda u^h) = -e_m < 0$, a contradiction.

Suppose we have $\xi^h = 0$ but $\eta^h \neq 0$. So $\eta^h \geq 0$ and since $A > 0$, $u^h = A\eta^h > 0$.So from (2.25), we must have $\xi_i^k = 0$ for all $i \neq 1$. Since $\xi^h = 0$, by (2.28) $v^h = 0$. So from (2.24)

$$v^k = -e_N + B^T \xi^k \tag{2.29}$$

and since $\xi_i^k = 0$ for all $i \neq 1$, v^k is obtained by the same procedure as v^1, the value of v in the initial BFS (since ξ_1^k must be the smallest value that makes v^k nonnegative in (2.29) in order to get an extreme point solution). So v^k is the same as v^1 in the initial BFS in (2.23). By our discussion earlier, this implies that $v_j^k > 0$ for all $j \neq r$, and $v_r^k = 0$. By (2.26) this implies that $\eta_j^k + \lambda \eta_j^h = 0$ for all $\lambda \geqq 0$ and $j \neq r$. These facts clearly imply that $(u^k, v^k, \xi^k, \eta^k)$ is the same as the initial BFS obtained for (2.23). This is a contradiction, since a BFS obtained in a step of the algorithm cannot reappear later on, along the almost complementary path.

These facts imply that ray termination cannot occur. So the algorithm must terminate in a finite number of steps by obtaining a complementary feasible basic vector, and the terminal BFS is therefore a solution of the LCP (1.42).

□

Comments 2.1 The complementary pivot algorithm for computing equilibrium strategies in bimatrix games is due to C. E. Lemke and J. T. Howson [1.18]. C. E. Lemke [2.21] extended this into the complementary pivot algorithm for LCPs discussed in Section 2.2. The proof of Theorem 2.1 is from the paper of R. W. Cottle and G. B. Dantzig [1.3] which also discusses various applications of the LCP and some principal pivoting methods for solving it. C. E. Lemke was awarded the ORSA/TIMS John Von Neumann Theory Prize in 1978 for his contributions to this area. The citation of the award says "Nash's equilibrium proofs were nonconstrutive, and for many years it seemed that the nonlinearity of the problem would prevent the actual numerical solution of any but the simplest noncooperative games. The breakthrough came in 1964 with an ingenious algorithm for the bimatrix case devised by Carlton Lemke and L. T. Howson Jr. It provided both a constructive existence proof and a practical means of calculation. The underlying logic, involving motions on the edges of an appropriate polyhedron, was simple and elegant yet conceptually daring in an epoch when such motions were typically contemplated in the context of linear programming. Lemke took the lead in exploiting the many ramifications and applications of this procedure, which range from the very basic linear complementary problem of mathematical programming to the problem of calculating fixed points of continuous, nonlinear mappings arising in various contexts. A new chapter in the theory and practice of mathematical programming was thereby opened which quickly became a very active and well-populated area of research...".

The geometric interpretation of the LCP using complementary cones was initiated in K. G. Murty [3.47, 1.26].

2.6 A VARIABLE DIMENSION ALGORITHM

We consider the LCP (q, M) which is to find $w, z \in \mathbf{R}^n$ satisfying

$$w - Mz = q \tag{2.30}$$

$$w, z \geqq 0 \tag{2.31}$$

$$\text{and} \quad w^T z = 0 \tag{2.32}$$

Definition : Principal Subproblem

Let $\mathbf{J} \subset \{1, \ldots, n\}$. *Denote* $w_{\mathbf{J}} = (w_j : j \in \mathbf{J})$, $z_{\mathbf{J}} = (z_j : j \in \mathbf{J})$, $q_{\mathbf{J}} = (q_j : j \in \mathbf{J})$, *and the principal submatrix of* M *corresponding to* $\mathbf{J}$, $M_{\mathbf{JJ}} = (m_{ij} : i, j \in \mathbf{J})$. *The* **principal subproblem** *of the LCP (2.30)–(2.32) in the variables* $w_{\mathbf{J}}$, $z_{\mathbf{J}}$ *(or the principal subproblem of the LCP (2.30)–(2.32) associated with the subset* $\mathbf{J}$*) is the LCP* $(q_{\mathbf{J}}, M_{\mathbf{JJ}})$ *of order* $|\mathbf{J}|$, *the complementary pairs of variables in it are* $\{w_j, z_j\}$ *for* $j \in \mathbf{J}$ *and it is: find* $w_{\mathbf{J}}$, $z_{\mathbf{J}}$ *satisfying*

$$\begin{aligned} & w_{\mathbf{J}} - M_{\mathbf{JJ}} z_{\mathbf{J}} = q_{\mathbf{J}} \\ & w_{\mathbf{J}}, z_{\mathbf{J}} \geqq 0 \\ & w_{\mathbf{J}}^T z_{\mathbf{J}} = 0 \ . \end{aligned}$$

This principal subproblem is therefore obtained from (2.30)–(2.32) by striking off the columns of all the variables w_j, z_j *for* $j \notin \mathbf{J}$ *and the equation in (2.30) corresponding to* $j \notin \mathbf{J}$.

Let $\mathbf{J} = \{1, \ldots, n\} \setminus \{i\}$, $\omega = (w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)^T$, $\xi = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)^T$. The following results follow by direct verification.

Results 2.2 If $(\hat{w} = (\hat{w}_1, \ldots, \hat{w}_n)^T; \hat{z} = (\hat{z}_1, \ldots, \hat{z}_n)^T)$ is a solution of the LCP (q, M) and $\hat{z}_i = 0$, then $(\hat{\omega} = (\hat{w}_1, \ldots, \hat{w}_{i-1}, \hat{w}_{i+1}, \ldots, \hat{w}_n)^T; \hat{\xi} = (\hat{z}_1, \ldots, \hat{z}_{i-1}, \hat{z}_{i+1}, \ldots, \hat{z}_n)^T)$ is a solution of its principal subproblem in the variables ω, ξ.

Results 2.3 Suppose that $(\tilde{\omega} = (\tilde{w}_1, \ldots, \tilde{w}_{i-1}, \tilde{w}_{i+1}, \ldots, \tilde{w}_n)^T; \ \tilde{\xi} = (\tilde{z}_1, \ldots, \tilde{z}_{i-1}, \tilde{z}_{i+1}, \ldots, \tilde{z}_n)^T)$ is a solution of the principal subproblem of the LCP (q, M) in the variables ω, ξ. Define $\tilde{z}_i = 0$ and let $\tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_{i-1}, \tilde{z}_i, \tilde{z}_{i+1}, \ldots, \tilde{z}_n)^T$. If $q_i + M_{i.}\tilde{z} \geqq 0$, define $\tilde{w}_i = q_i + M_{i.}\tilde{z}$, and let $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_{i-1}, \tilde{w}_i, \tilde{w}_{i+1}, \ldots, \tilde{w}_n)^T$, then $(\tilde{w}, \tilde{z})$ is a solution of the original LCP (q, M).

Example 2.15

Consider the following LCP (q, M)

w_1	w_2	w_3	z_1	z_2	z_3	q
1	0	0	2	0	−3	4
0	1	0	−1	−4	−3	−14
0	0	1	1	2	−2	13

$w_j \geqq 0,\ z_j \geqq 0,\ w_j z_j = 0$ for all $j = 1$ to 3

Let $\omega = (w_1, w_2)^T$, $\xi = (z_1, z_2)^T$. Then the principal subproblem of this LCP in the variable ω, ξ is

w_1	w_2	z_1	z_2	γ
1	0	2	0	4
0	1	−1	−4	−14

$w_j \geqq 0,\ z_j \geqq 0,\ w_j z_j = 0$ for $j = 1, 2$

$(\hat{w} = (0,0,5)^T; \hat{z} = (2,3,0)^T)$ is a solution of the original LCP and $\hat{z}_3$ is equal to zero in this solution. This implies that $(\hat{\omega} = (0,0); \hat{\xi} = (2,3))$ is a solution of this principal subproblem which can easily be verified. Also, $(\tilde{\omega} = (4,0)^T; \tilde{\xi} = (0, \frac{14}{4})^T)$ is another solution of the principal subproblem. Defining $\tilde{z}_3 = 0$, $\tilde{z} = (0, \frac{14}{4}, 0)^T$, we verify that $q_3 + M_{3.}\tilde{z} = 13 + (-1,-2,2)(0, \frac{14}{4}, 0)^T = 6 > 0$. Hence, define $\tilde{w}_3 = 6$, and $\tilde{w} = (4,0,6)^T$. It can be verified that $(\tilde{w} = (4,0,6)^T; \tilde{z} = (0, \frac{14}{4}, 0)^T)$ is another solution of the original LCP.

We now discuss a variable dimension algorithm for the LCP (q, M) due to L. Van der Heyden [2.38]. If $q \geqq 0$, $(w = q, z = 0)$ is a readily available solution. So we assume that $q \not\geqq 0$. The method proceeds by solving a sequence of principal subproblems of (2.30), (2.31), (2.32) always associated with subsets of the form $J = \{1, \ldots, k\}$ (this problem is called the k-problem), for some k satisfying $1 \leqq k \leqq n$. When the method is working on the k-problem, the bottom $n-k$ constraints in (2.30) as well as the columns of variables w_j, z_j for $j > k$ can be ignored, hence the reason for the name. All the intermediate solutions for (2.30) obtained during the method (with the exception of the terminal solution which is a complementary feasible solution satisfying (2.30), (2.31), (2.32)) are of two types called **position 1** and **position 2 solutions** defined below.

Position 1 Solution : This is a solution $(\tilde{w}, \tilde{z})$ for (2.30) satisfying the following properties :

i) there exists an index k such that $\tilde{z}_k = 0$ and $\tilde{w}_k < 0$.

ii) $\tilde{z}_j = 0$ for $j > k$.

iii) if $k > 1$, $\tilde{w}^{(k-1)} = (\tilde{w}_1, \ldots, \tilde{w}_{k-1})$, $\tilde{z}^{(k-1)} = (\tilde{z}_1, \ldots, \tilde{z}_{k-1})$ is a solution for the principal subproblem of (2.30), (2.31), (2.32) determined by the subset $\{1, \ldots, k-1\}$, that is, $\tilde{w}^{(k-1)} \geqq 0$, $\tilde{z}^{(k-1)} \geqq 0$ and $(\tilde{w}^{(k-1)})^T \tilde{z}^{(k-1)} = 0$.

From the definition, a position 1 solution $(\bar{w}, \bar{z})$ always satisfies $\bar{w}^T \bar{z} = 0$, it is complementary (but infeasible) and it will be a complementary basic solution associated with a complementary (but infeasible in the same sense that the solution violates (2.31)) basic vector for (2.30).

Position 2 Solution : This is a solution $(\hat{w}, \hat{z})$ for (2.30) satisfying the following properties :

a) there exists an index k such that $\hat{z}_k > 0$, $\hat{w}_k < 0$.

b) $\hat{z}_j = 0$ for $j > k$.

c) there is a $u < k$ such that both $\hat{z}_u$ and $\hat{w}_u$ are zero.

d) $\hat{w}^{(k-1)} = (\hat{w}_1, \ldots, \hat{w}_{k-1})^T \geqq 0$, $\hat{z}^{(k-1)} = (\hat{z}_1, \ldots, \hat{z}_{k-1})^T \geqq 0$ and $(\hat{w}^{(k-1)})^T \hat{z}^{(k-1)} = 0$.

From the definition, a position 2 solution discussed above is an almost complementary solution (not feasible, since some of the variables are < 0) of the type discussed in Section 2.4, it satisfies $\hat{w}^T \hat{z} = \hat{w}_k \hat{z}_k$. It will be an almost complementary basic solution associated with an almost complementary basic vector for (2.30) which has both w_k, z_k as basic variables, and contains exactly one basic variable from the complementary pair (w_j, z_j) for each $j \neq k$ or u (both variables w_u, z_u are out of this almost complementary basic vector, so the complementary pair (w_u, z_u) is the left out complementary pair in this basic vector). This almost complementary basic vector has w_j as a basic variable for all $j > k$. All intermediate (i. e., except the initial and terminal) solutions obtained by the method when it is working on the k-problem wiil be position 2 solutions of (2.30) as defined above.

Note 2.1 As mentioned above, all the solutions obtained during the algorithm will be basic solutions of (2.30).The definitions given above for positions 1, 2 solutions are under the assumption that q is nondegenerate in the LCP (q, M) (i. e., that every solution to (2.30) has at least n nonzero variables). In the general case when q may be degenerate, the algorithm perturbs q by adding the vector $(\varepsilon, \varepsilon^{\mathbf{2}}, \ldots, \varepsilon^{\mathbf{n}})^T$ to it, where ε is treated as a sufficiently small positive number without giving any specific value to it (see Section 2.1, 2.2.2, 2.2.8), and all the inequalities for the signs of the variables should be understood in the usual lexico sense.

The Algorithm

The algorithm takes a path among basic vectors for (2.30) using pivot steps. All basic vectors obtained will be almost complementary basic vectors as defined in Section 2.4, or complementary basic vectors.

Initial Step: STEP 0 : The initial basic vector is $w = (w_1, \ldots, w_n)$. The initial solution is the Position 1 basic solution of (2.30) corresponding to it, define $k =$ minimum $\{i : q_i < 0\}$. Begin with the k-problem, by making a type 1 pivot step to increase the value of the nonbasic variable z_k from 0, as described below.

STEP 1 : Type 1 Pivot Step, to increase the Value of a Nonbasic Variable from Zero. Let $(y_1, \ldots, y_k, w_{k+1}, \ldots, w_n)$ be the basic vector in some stage of working for the k-problem. If this is the initial basic vector, $(y_1, \ldots, y_k)$ will be a complementary basic vector for the principal subproblem of (2.30), (2.31), (2.32) defined by the subset $\{1, \ldots, k\}$. Except possibly at termination of work on the k-problem, y_k will always be w_k; $y_1, \ldots, y_{k-1}$ will all be w_j or z_j for $j \leqq k - 1$. This type of pivot step occurs when the value of a nonbasic variable, say v, selected by the rules specified in the algorithm, is to be increased from its present value of zero. The variable v will be either w_j or z_j for some $j \leqq k$. Let the canonical tableau for (2.30) with respect to the present basic vector be

Tableau 2.7 Canonical Tableau

$y_1 \ldots y_k \quad w_{k+1} \ldots w_n$	$\ldots v \ldots$	
	a_1	$\bar{q}_1$
I	$\ldots \vdots \ldots$	$\vdots$
	a_n	$\bar{q}_n$

While working on the k-problem, in all the canonical tableaus, we will have $\bar{q}_1, \ldots, \bar{q}_{k-1} \geqq 0$ and $\bar{q}_k < 0$ (and $y_k = w_k$). Let $\beta = B^{-1}$ be the inverse of the present basis. The algorithm always maintains $(\bar{q}_i, \beta_{i.}) \succ 0$ for $i = 1$ to $k - 1$. Let λ denote the nonnegative value given to the nonbasic variable v. The new solution as a function of λ is

$$
\begin{aligned}
&\text{all nonbasic variables other than } v \text{ are } 0 \\
&v = \lambda \\
&y_i = \bar{q}_i - \lambda a_i, \quad i = 1 \text{ to } k \\
&w_j = \bar{q}_j - \lambda a_j, \quad j = k + 1 \text{ to } n
\end{aligned}
\tag{2.33}
$$

We will increase the value of λ from 0 until one of the variables y_i for $i = 1$ to k, changes its value from its present to zero in (2.33), and will change sign if λ increases any further. This will not happen if the updated column of the entering variable v satisfies

$$a_i \leqq 0, \quad i = 1, \ldots, k - 1 \quad \text{and} \quad a_k \geqq 0 \tag{2.34}$$

If condition (2.34) is satisfied, the method is unable to proceed further and termination occurs with the conclusion that the method is unable to process this LCP. If condition (2.34) is not satisfied, define

$$\theta = \text{ Max}\Big\{\frac{\bar{q}_i}{a_i} : \text{ Over } 1 \leqq i \leqq k - 1 \text{ such that } a_i > 0 \ ; \text{ and } \frac{\bar{q}_k}{a_k}, \text{ if } a_k < 0\Big\} \tag{2.35}$$

Let $\mathbf{\Delta}$ be the set of all i between 1 to k which tie for the maximum in (2.35). If $\mathbf{\Delta}$ is a singleton set, let r be the element in it. Otherwise let r be the element which attains the lexicomaximum in lexicomaximum $\{ (\frac{(\bar{q}_i, \beta_{i.})}{a_i}) : i \in \mathbf{\Delta} \}$. If $r = k$, v replaces $y_k (= w_k)$ from the basic vector. After this pivot step we are lead to the basic vector $(y_1, \ldots, y_{k-1}, \; v, w_{k+1}, \ldots, w_n)$ which will be a complementary basic vector for (2.30) (except that the variables $y_1, \ldots, y_{k-1}, v$ may have to be rearranged so that the j^{th} variable here is from the j^{th} complementary pair), and $(y_1, \ldots, y_{k-1}, v)$ is a complementary lexico feasible basic vector for the k-problem (except for the rearrangement of the basic variables as mentioned above). If $(y_1, \ldots, y_{k+1}, v, w_{k+1}, \ldots, w_n)$ is feasible to (2.30) (this happens if the updated right hand side constants vector is $\geqq 0$ after the pivot step of replacing y_k by v), it is a complementary feasible basic vector for (2.30), the method terminates with the basic solution corresponding to it as being a solution for (2.30), (2.31), (2.32). On the other hand, if $(y_1, \ldots, y_{k-1}, v, w_{k+1}, \ldots, w_n)$ is not a feasible basic vector for (2.30), the k-problem has just been solved and the method moves to another principal subproblem with index greater than k (this is called a **forward move**), go to Step 3.

If $r < k$, v replaces y_r from the basic vector, leading to the new basic vector $(y_1, \ldots, y_{r-1}, v, y_{r+1}, \ldots, y_k, w_{k+1}, \ldots, w_n)$. Two things can happen now. If $y_r = z_k$, then this new basic vector is a complementary basic vector for (2.30) (except for rearrangement of the variables as mentioned above), but $(y_1, \ldots, y_{r-1}, v, y_{r+1}, \ldots, y_k)$ is not lexico feasible for the k-problem. In this case the method moves to make a type 2 pivot step (discussed next) leading to a principal subproblem with index less than k (this is called a **regressive move**, moving to a smaller principal subproblem already solved earlier). The next steps of the algorithm will be concerned with finding yet another solution for this smaller principal subproblem. Go to Step 2.

The second possibility is that $y_r \neq z_k$. In this case the basic vector $(y_1, \ldots, y_{r-1}, v, y_{r+1}, \ldots, y_k, w_{k+1}, \ldots, w_n)$ is another almost complementary basic vector, the basic solution of (2.30) associated with which is another position 2 solution. In this case, the method continues the work on the k-problem by making a type 1 pivot step next, to increase the value of the complement of y_r from zero.

STEP 2 : Type 2 Pivot Step to Decrease the Value of a Nonbasic Variable w_g from Zero. This pivot step will be made whenever we obtain a complementary basic vector $(y_1, \ldots, y_k, w_{k+1}, \ldots, w_n)$ after doing some work on the k-problem, with $y_k = w_k$. Let Tableau 2.7 be the canonical tableau with respect to this complementary basic vector. We will have $\bar{q}_i \geqq 0$, $i = 1$ to $k-1$ and $(\bar{q}_k, \beta_{k.}) \prec 0$ at this stage ($\beta_{k.}$ is the k^{th} row of the present basis inverse). Let g be the maximum j such that $y_j = z_j$. Now the algorithm decreases the value of the nonbasic variable w_g from zero. Letting $v = w_g$, and giving this variable a value λ (we want to make $\lambda \leqq 0$), the new solution obtained is of the same form as in (2.33). We will decrease the value of λ from 0 until one of the variables y_i for $i = 1$ to g, changes its value from its present to zero in (2.33), and will change sign if λ decreases any further. This will not happen if the updated column of the entering variable v satisfies

$$a_i \geqq 0, \quad i = 1 \text{ to } g \tag{2.36}$$

in which case termination occurs with the conclusion that the method is unable to process this LCP. If (2.36) is not satisfied, define

$$\theta = \text{ Minimum } \left\{ -\left(\frac{\bar{q}_i}{a_i}\right) : 1 \leqq i \leqq g, \quad i \text{ such that } a_i < 0 \right\} . \tag{2.37}$$

Let $\boldsymbol{\Delta}$ be the set of all i between 1 to g which tie for the minimum in (2.37). If $\boldsymbol{\Delta}$ is a singleton set let r be the element in it. Otherwise let r be the element which attains the lexicominimum in lexico minimum $\left\{ -\frac{(\bar{q}_i, \beta_{i.})}{a_i} : i \in \boldsymbol{\Delta} \right\}$. Replace y_r in the present basic vector by v ($= w_g$ here) and move over to the g-problem after this pivot step, by going to Step 1 to increase the value of the complement of y_r from 0.

STEP 3 : We move to this step when we have solved a k-problem after performing a type 1 pivot step on it in Step 1. Let $(y_1, \dots, y_k, w_{k+1}, \dots, w_n)$ be the complementary basic vector at this stage with $y_j \in \{w_j, z_j\}$ for $j = 1$ to k. Let $\bar{q} = (\bar{q}_1, \dots, \bar{q}_n)^T$ be the current updated right hand side constants vector. Since $(y_1, \dots, y_k)$ is a complementary feasible basic vector for the k-problem, we have $\bar{q}_i \geqq 0$ for $i = 1$ to k. If $\bar{q}_i \geqq 0$ for $i = k+1$ to n also, this basic vector is complementary feasible to the original problem (2.30), (2.31), (2.32), and we would have terminated. So $\bar{q}_i < 0$ for at least one i between $k+1$ to n. Let u be the smallest i for which $\bar{q}_i < 0$, replace k by u and go back to Step 1 to increase the value of z_k from zero.

Numerical Example 2.16

We provide here a numerical example for this algorithm from the paper [2.38] of L. Van der Heyden. Consider the LCP (q, M) where

$$q = \begin{pmatrix} -1 \\ -2 \\ -10 \end{pmatrix} \quad , \qquad M = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}$$

Since $q_1 < 0$, the algorithm begins with $k = 1$, on the 1-problem. Pivot elements are inside a box.

Basic Vector	w_1	w_2	w_3	z_1	z_2	z_3		
w_1	1	0	0	$\boxed{-1}$	−1	−1	−1	$k = 1$. Increase z_1. In this
w_2	0	1	0	−3	−1	−1	−2	type 1 pivot step, w_1
w_3	0	0	1	−2	−2	−1	−10	drops from basic vector.
z_1	−1	0	0	1	1	1	1	$k = 3$.
w_2	−3	1	0	0	2	$\boxed{2}$	1	Increase z_3.
w_3	−2	0	1	0	0	1	−8	w_2 drops.

Basic Vector	w_1	w_2	w_3	z_1	z_2	z_3		
z_1	$\frac{1}{2}$	$-\frac{1}{2}$	0	1	0	0	$\frac{1}{2}$	$k = 3$. Increase z_2
z_3	$-\frac{3}{2}$	$\frac{1}{2}$	0	0	$\boxed{1}$	1	$\frac{1}{2}$	(complement of w_2).
w_3	$-\frac{1}{2}$	$-\frac{1}{2}$	1	0	-1	0	$-\frac{17}{2}$	z_3 drops.
z_1	$\frac{1}{2}$	$\boxed{-\frac{1}{2}}$	0	1	0	0	$\frac{1}{2}$	Need a type 2 pivot step.
z_2	$-\frac{3}{2}$	$\frac{1}{2}$	0	0	1	1	$\frac{1}{2}$	Decrease w_2.
w_3	-2	0	1	0	0	1	-8	z_1 drops.
w_2	$\boxed{-1}$	1	0	-2	0	0	-1	$k = 1$. Increase w_1 (compl.
z_2	-1	0	0	1	1	1	1	of z_1 that just dropped)
w_3	-2	0	1	0	0	1	-8	w_2 drops.
w_1	1	-1	0	2	0	0	1	$k = 3$.
z_2	0	-1	0	3	1	$\boxed{1}$	2	Increase z_3.
w_3	0	-2	1	4	0	1	-6	z_2 drops.
w_1	1	-1	0	2	0	0	1	Increase w_2 (complement
z_3	0	-1	0	3	1	1	2	of z_2 that just dropped).
w_3	0	$\boxed{-1}$	1	1	-1	0	-8	w_3 drops.
w_1	1	0	-1	1	1	1	9	Complementary
z_3	0	0	-1	2	2	1	10	feasible
w_2	0	1	-1	-1	1	0	8	basic vector

Thus $(w_1, w_2, w_3; z_1, z_2, z_3) = (9, 8, 0; 0, 0, 10)$ is a complementary feasible solution of this problem.

Conditions Under Which the Algorithm is Guaranteed to Work

Theorem 2.12 *For every* $\mathbf{J} \subset \{1, \ldots, n\}$, *if the principal submatrix* $M_{\mathbf{JJ}}$ *of* M *associated with* $\mathbf{J}$ *satisfies the property that there exists no positive vector* $z_{\mathbf{J}}$ *such that the last component of* $M_{\mathbf{JJ}} z_{\mathbf{J}}$ *is nonpositive and the other components are zero, the termination criteria (2.34) or (2.36) will never be satisfied and the algorithm terminates* *with a complementary feasible basic vector for the LCP (q, M) after a finite number of steps.*

Proof. When (2.34) or (2.36) is satisfied, we have a solution of the type given in equation (2.33), which we denote by $(w(\lambda), z(\lambda)) = (\bar{w} + \lambda w^h, \bar{z} + \lambda z^h)$ satisfying the property that for $\lambda > 0$, there exists a k such that $w_k(\lambda) < 0$, $z_k(\lambda) > 0$, $z_j(\lambda) = 0$ for $j > k$, and if $k > 1$ the vectors $w^{(k-1)}(\lambda) = (w_1(\lambda), \ldots, w_{k-1}(\lambda))$, $z^{(k-1)}(\lambda) = (z_1(\lambda), \ldots, z_{k-1}(\lambda))$ are nonnegative and complementary. Let $\mathbf{J} = \{1, \ldots, k\}$, $w^h_{\mathbf{J}} = (w^h_1, \ldots, w^h_k)^T$, $z^h_{\mathbf{J}} = (z^h_1, \ldots, z^h_k)^T$. Then

$$\begin{aligned} w^h_{\mathbf{J}} - M_{\mathbf{JJ}} z^h_{\mathbf{J}} &= 0 \\ z^h_{\mathbf{J}} &\geqq 0 \\ w^h_k &\leqq 0 \end{aligned} \tag{2.38}$$

and if $k > 1$, $(w^h_1, \ldots, w^h_{k-1}) \geqq 0$, and $w^h_j z^h_j = 0$ for $j = 1$ to $k - 1$. Let $\mathbf{P} = \{ j : 1 \leqq j \leqq k, \text{ and } z^h_j > 0 \}$. Clearly $\mathbf{P} \neq \emptyset$, otherwise $(w^h_{\mathbf{J}}, z^h_{\mathbf{J}}) = 0$. Letting $z^h_{\mathbf{P}} = (z^h_j : j \in \mathbf{P})$, all the components of $M_{\mathbf{PP}} z^h_{\mathbf{P}}$ are zero except possibly the last one because of (2.38) and the fact that $w^h_j z^h_j = 0$ for $j = 1$ to $k - 1$. Also, the last component of $M_{\mathbf{PP}} z^h_{\mathbf{P}}$ is $\leqq 0$ because of (2.38). And since $z^h_{\mathbf{P}} > 0$, this contradicts the hypothesis in the theorem.

The finiteness of the algorithm follows from the path argument used in Sections 2.2, 2.3, the argument says that the algorithm never returns to a previous position as this situation implies the existence of a position with three adjacent positions, a contradiction. Since there are only a finite number of positions we must terminate with a solution for the original LCP.

□

Corollary 2.2 *If M has the property that for every $\mathbf{J} \subset \{1, \ldots, n\}$, the corresponding submatrix $M_{\mathbf{JJ}}$ of M satisfies the property that the system*

$$\begin{aligned} M_{\mathbf{JJ}} z_{\mathbf{J}} &\leqq 0 \\ z_{\mathbf{J}} &\geqq 0 \end{aligned}$$

has the unique solution $z_{\mathbf{J}} = 0$, then the variable dimension algorithm discussed above will terminate with a solution of the LCP (q, M) for any $q \in \mathbf{R}^n$.

Proof. Follows from Theorem 2.12.

□

R. W. Cottle [3.9] has shown that the class of matrices M satisfying the hypothesis in Theorem 2.12 or Corollary 2.2, is the strictly semi-monotone matrices defined later on in Section 3.4, which is the same as $\bar{Q}$ (completely Q-matrices, that is, matrices all of whose principal submatrices are Q-matrices). This class includes all P-matrices and positive or strictly copositive matrices.

By the results discussed in Chapter 3, the LCP (q, M) has a unique solution when M is a P-matrix. So if M is s P-matrix and the LCP (q, M) is solved by the variable dimension algorithm, type 2 pivot steps will never have to be performed.

M. J. Todd [2.35, 2.36] has shown that when q is nondegenerate in (2.30) and M is a P-matrix, the variable dimension algorithm discussed above corresponds to the lexicographic Lemke algorithm discussed in Section 2.3.4.

Now consider the LCP (q, M) of order n. Let e_n denote the column vector of all 1's in $\mathbf{R}^n$. Introduce the artificial variable z_0 associated with the column vector $-e_n$, as in the complementary pivot algorithm (see equation (2.3)). Introduce an additional artificial variable w_0, which is the complement of z_0, and the artificial constraint "$w_0 - e_n^T z = q_0$", where q_0 is treated as a large positive number, without giving it any specific value. This leads to an LCP of order $n+1$, in which the variables are $(w_0, w_1, \dots, w_n)$, $(z_0, z_1, \dots, z_n)$ and the data is

$$M^\star = \begin{pmatrix} 0 & -e_n^T \\ e_n & M \end{pmatrix} , q^\star = \begin{pmatrix} q_0 \\ q \end{pmatrix} .$$

Since q_0 is considered as a large positive parameter, $w_0 > 0$ and $z_0 = 0$ in any complementary solution of this larger dimensional LCP $(q^\star, M^\star)$, and hence if $((\bar{w}_0, \bar{w}), (\bar{z}_0, \bar{z}))$ is a solution of this LCP, then $(\bar{w}, \bar{z})$ is a solution of the original LCP (q, M).

Essentially by combining the arguments in Theorems 2.1 and 2.12, L. Van der Heyden [2.39] has shown that if M is a copositive plus matrix and the system "$w - Mz = q$, $w \geqq 0$, $z \geqq 0$" has a feasible solution, when the variable dimension algorithm is applied on the LCP $(q^\star, M^\star)$, it will terminate with a complementary feasible solution $((\bar{w}_0, \bar{w}), (\bar{z}_0, \bar{z}))$ in a finite number of steps. This shows that the variable dimension algorithm will process LCP's associated with copositive plus matrices, by introducing an artificial dimension and by applying the variable dimension algorithm to the enlarged LCP.

2.7 EXTENSIONS TO FIXED POINT COMPUTING, PIECEWISE LINEAR AND SIMPLICIAL METHODS

It has also been established that the arguments used in the complementary pivot algorithm can be generalized, and these generalizations have led to algorithms that can compute approximate Brouwer and Kakutani fixed points! Until now, the greatest single contribuition of the complementarity problem is probably the insight that it has provided for the development of fixed point computing algorithms. In mathematics, fixed point theory is very higly developed, but the absence of efficient algorithms for computing these fixed points has so far frustrated all attempts to apply this rich theory to real life problems. With the development of these new algorithms, fixed point theory is finding numerous applications in mathematical programming, in mathematical economics, and in various other areas. We present one of these fixed point computing algorithms, and some of its applications, in this section. We show that the problem of computing a KKT point for an NLP can be posed as a fixed point problem and solved by these methods.

The algorithms that are discussed later in this section trace a path through the simplices of a triangulation in $\mathbf{R}^n$, that is why they are called **simplicial methods**. Since they use piecewise linear approximations of maps, these methods are also called **piecewise linear methods**. Since the path traced by these methods has exactly the same features as that of the complementary pivot algorithm (see Sections 2.2.5, 2.2.6) these methods are also called **complementary pivot methods**.

2.7.1 Some Definitions

Let $g(x)$ be a real valued function defined over a convex subset $\mathbf{\Gamma} \subset \mathbf{R}^n$. We assume that the reader is familiar with the definition of continuity of $g(x)$ at a point $x^0 \in \mathbf{\Gamma}$, and the definition of the vector of partial derivatives of $g(x)$ at x^0, $\nabla g(x^0) = \big(\frac{\partial g(x^0)}{\partial x_1}, \dots, \frac{\partial g(x^0)}{\partial x_n}\big)$, when it exists. The function $g(x)$ is said to be differentiable at x^0 if $\nabla g(x^0)$ exists, and for any $y \in \mathbf{R}^n$, $\frac{1}{\alpha}\left(g(x^0+\alpha y) - g(x^0) - \alpha\big(\nabla g(x^0)\big)y\right)$ tends in the limit to zero as α tends to zero. If $g(x)$ is differentiable at x^0, for any $y \in \mathbf{R}^n$, we can approximate $g(x^0+\alpha y)$ by $g(x^0) + \alpha\big(\nabla g(x^0)\big)y$ for values of α for which $|\alpha|$ is small. This is the **first order Taylor series expansion** for $g(x+\alpha y)$ at $x = x^0$. If $g(x)$ is differentiable at x^0, the partial derivative vector $\nabla g(x^0)$ is known as the **gradient vector** of $g(x)$ at x^0.

When the second order partial derivatives of $g(x)$ exist at x^0, we denote the $n \times n$ matrix of second order partial derivatives $\big(\frac{\partial^2 g(x^0)}{\partial x_i\, \partial x_j}\big)$ by the symbol $H(g(x^0))$. It is called the **Hessian matrix** of $g(x)$ at x^0.

Let $g_1(x), \dots, g_m(x)$ be m real valued convex functions defined on the convex subset $\mathbf{\Gamma} \subset \mathbf{R}^n$. For each $x \in \mathbf{\Gamma}$, define $s(x) = \text{Maximum}\,\{\, g_1(x), \dots, g_m(x)\,\}$. The function $s(x)$ is known as the **pointwise supremum** or **maximum** of $\{g_1(x), \dots, g_m(x)\}$. It is also convex on $\mathbf{\Gamma}$. See Figure 2.4 where we illustrate the pointwise supremum of several affine functions defined on the real line.

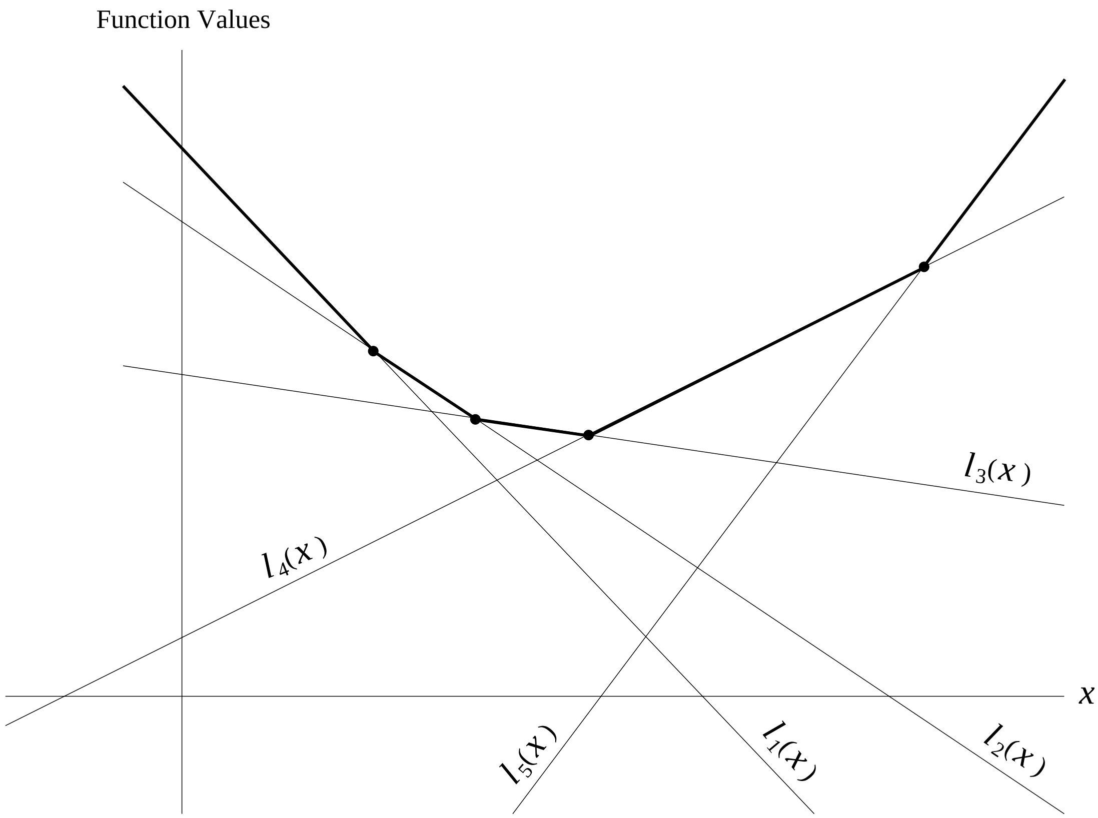

Figure 2.4 $l_1(x)$ to $l_5(x)$ are five affine functions defined on $\mathbf{R}^1$. Function values are plotted on the vertical axis. Their pointwise maximum is the function marked with thick lines here.

Subgradients and Subdifferentials of Convex Functions

Let $g(x)$ be a real valued convex function defined on $\mathbf{R}^n$. Let $x^0 \in \mathbf{R}^n$ be a point where $g(x^0)$ is finite. The vector $d = (d_1, \ldots, d_n)^T$ is said to be a subgradient of $g(x)$ at x^0 if

$$g(x) \geqq g(x^0) + d^T(x - x^0), \quad \text{for all } x \in \mathbf{R}^n \ . \tag{2.39}$$

Notice that the right hand side of (2.39) is $l(x) = (g(x^0) - d^t x^0) + d^T x$, is an affine function in x; and we have $g(x^0) = l(x^0)$. One can verify that $l(x)$ is the first order Taylor expansion for $g(x)$ around x^0, constructed using the vector d in place of the gradient vector of $g(x)$ at x^0. So d is a subgradient of $g(x)$ at x^0, iff this modified Taylor approximation is always an underestimate for $g(x)$ at every x.

Example 2.17

Let $x \in \mathbf{R}^1$, $g(x) = x^{\mathbf{2}}$. $g(x)$ is convex. Consider the point $x^0 = 1$, $d = 2$. It can be verified that the inequality (2.39) holds in this case. So $d = (2)$ is a subgradient for $g(x)$ at $x^0 = 1$ in this case. The affine function $l(x)$ on the right hand side of (2.39) in this case is $1 + 2(x - 1) = 2x - 1$. See Figures 2.5, 2.6 where the inequality (2.39) is illustrated.

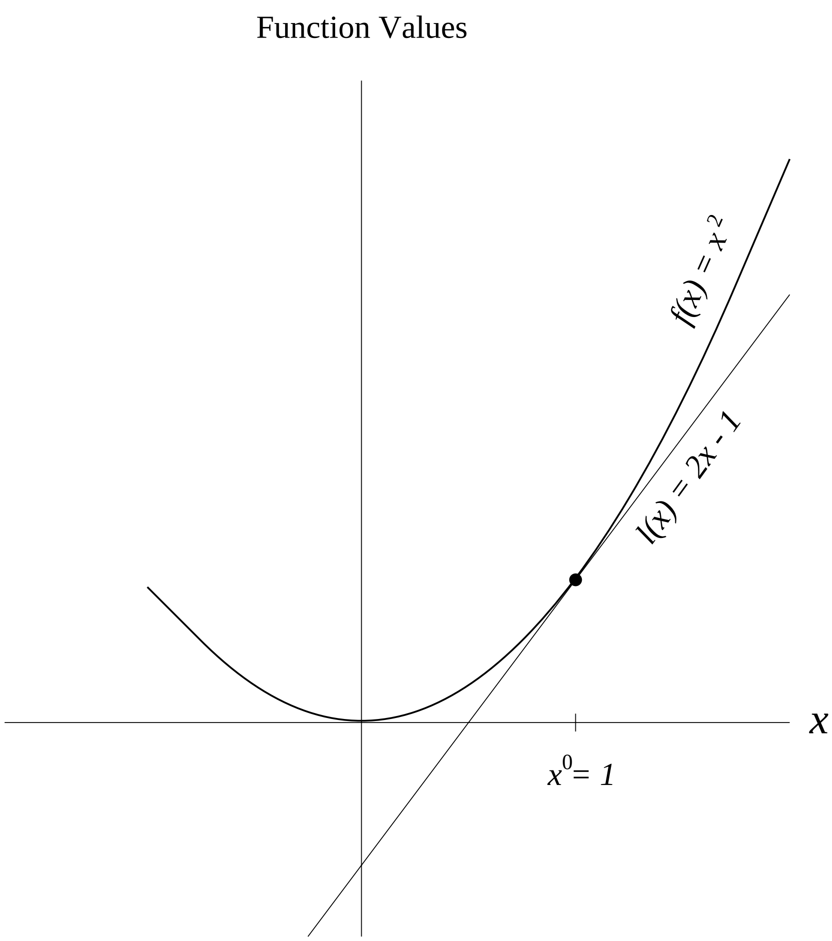

Figure 2.5 A Convex Function, and the Affine Function Below it Constructed Using a Subgradient for it at the Point x^0.

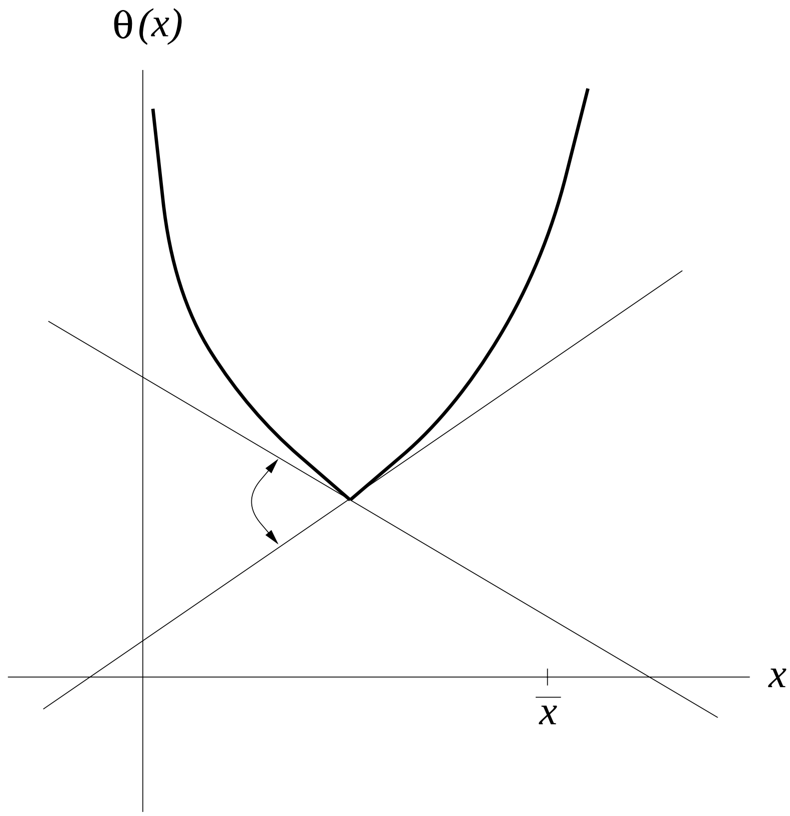

Figure 2.6 The subdifferential to $\theta(x)$ at $\bar{x}$ is the set of slope vectors of all lines in the cone marked by the angle sign.

The set of all subgradients of $g(x)$ at x^0 is denoted by the symbol $\partial g(x^0)$, and called the **subdifferential set** of $g(x)$ at x^0. It can be proved that if $g(x)$ is differentiable at x^0, then its gradient $\nabla g(x^0)$ is the unique subgradient of $g(x)$ at x^0. Conversely if $\partial g(x^0)$ contains a single vector, then $g(x)$ is differentiable at x^0 and $\partial g(x^0) = \{\nabla g(x^0)\}$. See references [2.92–2.94] for these and other related results.

Subgradients of Concave Functions

Let $h(x)$ be a concave function defined on a convex subset $\mathbf{\Gamma} \subset \mathbf{R}^n$. In defining a subgradient vector for $h(x)$ at a point $x^0 \in \mathbf{\Gamma}$, the inequality in (2.39) is just reversed; in other words, d is a subgradient for the concave function $h(x)$ at x^0 if $h(x) \leqq h(x^0) + d^T(x - x^0)$ for all x. With this definition, all the results stated above also hold for concave functions.

Computing a Subgradient

Let $\theta(x)$ be a convex function defined on $\mathbf{R}^n$. Let $\bar{x} \in \mathbf{R}^n$, if $\theta(x)$ is differentiable at $\bar{x}$, then the gradient vector $\nabla\theta(\bar{x})$ is the only subgradient of $\theta(x)$ at $\bar{x}$. If $\theta(x)$ is not differentiable at $\bar{x}$, in general, the computation of a subgradient for $\theta(x)$ at $\bar{x}$ may be hard. However, if $\theta(x)$ is the pointwise supremum of a finite set of differentiable convex functions, say

$$\theta(x) = \text{ Maximum } \{ g_1(x), \dots, g_m(x) \}$$

where each $g_i(x)$ is differentiable and convex, then the subdifferential of $\theta(x)$ is easily obtained. Let

$$\mathbf{J}(\bar{x}) = \{ i : \quad \theta(\bar{x}) = g_i(\bar{x}) \}$$

the the subdifferential of $\theta(x)$ at $\bar{x}$,

$$\partial\theta(\bar{x}) = \text{ convex hull of } \{ \nabla g_i(\bar{x}) : i \in \mathbf{J}(\bar{x}) \} .$$

See references [2.92–2.94].

2.7.2 A Review of Some Fixed Point Theorems

Let $\mathbf{\Gamma} \subset \mathbf{R}^n$ be a compact convex subset with a nonempty interior. Let $f(x) : \mathbf{\Gamma} \to \mathbf{\Gamma}$ be a single valued map, that is, for each $x = (x_1, \dots, x_n)^T \in \mathbf{\Gamma}$, $f(x) = (f_1(x), \dots, f_n(x))^T \in \mathbf{\Gamma}$, which is continuous. We have the following celebrated theorem.

Theorem 2.13 : Brouwer's Fixed Point Theorem *If $f(x) : \mathbf{\Gamma} \to \mathbf{\Gamma}$ is continuous, it has a fixed point, that is, the system*

$$f(x) - x = 0 \tag{2.40}$$

which is a system of n equations in n unknowns, has a solution $x \in \mathbf{\Gamma}$.

See references [2.48, 2.50, 2.68, 2.69, 2.72] for proofs of this theorem. We now provide an illustration of this theorem.

Example 2.18

Consider $n = 1$. Let $\mathbf{\Gamma} = \{ x : \ x \in \mathbf{R}^1, \ 0 \leqq x \leqq 1 \}$ denoted by $[0, 1]$. Consider the continuous function $f(x) : [0, 1] \to [0, 1]$. We can draw a diagram for $f(x)$ on the two dimensional Cartesian plane by plotting x on the horizontal axis, and the values of $f(x)$ along the vertical axis, as in Figure 2.7. Since $f(x)$ is defined on $[0, 1]$ the curve of $f(x)$ begins somewhere on the thick vertical line $x = 0$, and goes all the way to the thick vertical line $x = 1$, in a continuous manner. Since $f(x) \in [0, 1]$, the curve for $f(x)$ lies between the two thin horizontal lines $f(x) = 0$ and $f(x) = 1$. The dashed diagonal line is $f(x) - x = 0$. It is intuitively clear that the curve of $f(x)$ must cross the diagonal of the unit square, giving a fixed point for $f(x)$.

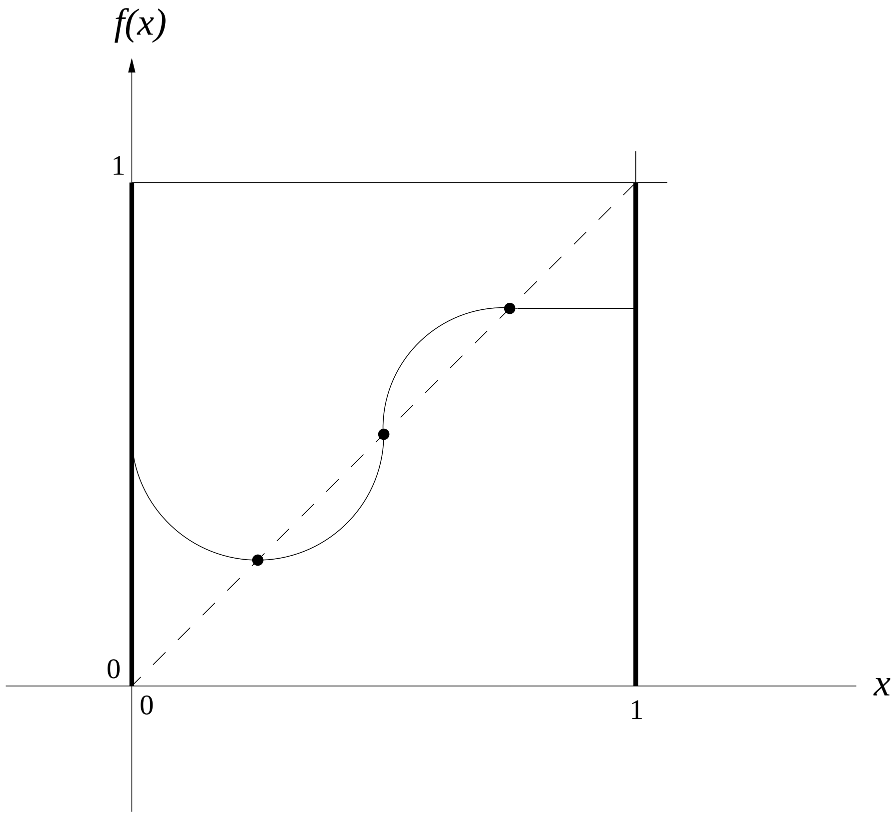

Figure 2.7 The curve of $f(x) : [0, 1] \to [0, 1]$. Points of intersection of the curve with the dashed diagonal line are the Brouwer fixed points of $f(x)$.

Example 2.19

This example illustrates the need for convexity in Theorem 2.13. Let $n = 2$. Let $\mathbf{K}$ denote the dotted ring in Figure 2.8 between two concentric circles. Let $f(x)$ denote the continuous mapping $\mathbf{K} \to \mathbf{K}$ obtained by rotating the ring through a specified angle θ in the anti-clockwise direction. Clearly this $f(x)$ has no fixed points in $\mathbf{K}$.

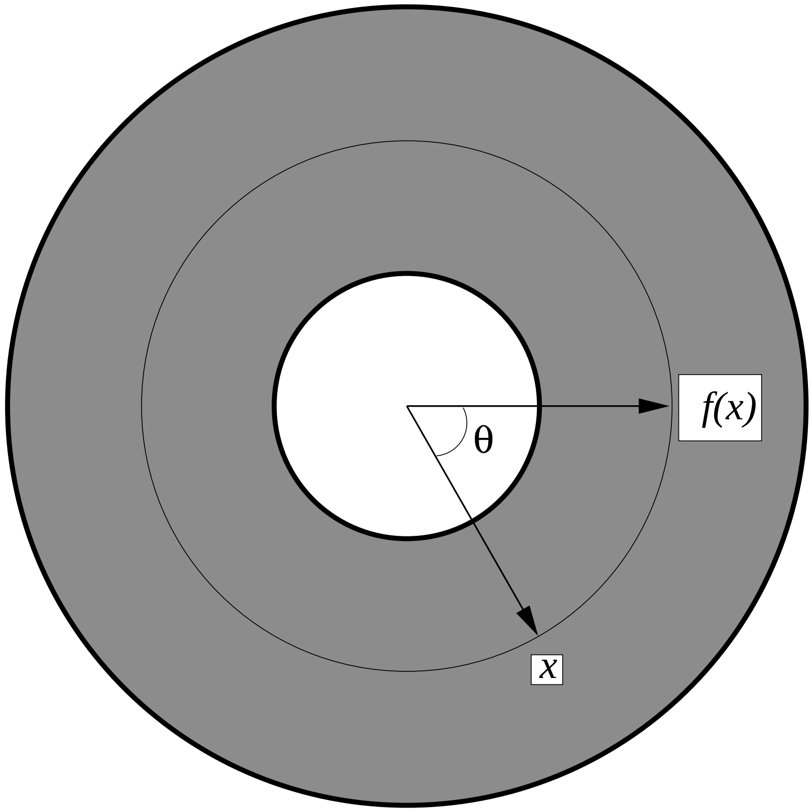

Figure 2.8 The need of convexity for the validity of Brouwer's fixed point theorem.

The need for the boundedness of the set $\mathbf{\Gamma}$ for the validity of Theorem 2.13 follows from the fact that the mapping $f(x) = x + a$ for each $x \in \mathbf{R}^n$, where $a \neq 0$ is a specified point in $\mathbf{R}^n$, has no fixed points. The need for the closedness of the set $\mathbf{\Gamma}$ for the validity of Theorem 2.13 follows from the fact that the mapping $f(x) = \frac{1}{2}(x+1)$ from the set $\{x : 0 \leqq x < 1\}$ into itself has no fixed point in the set.

The system (2.40) is a system of n equality constraints in n unknowns. An effort can be made to solve (2.40) using methods for solving nonlinear equations.

A Monk's Story

The following story of a monk provides a nice intuitive justification for the concept and the existence of a fixed point. A monk is going on a pilgrimage to worship in a temple at the top of a mountain. He begins his journey on Saturday morning at 6:00 AM promptly. The path to the temple is steep and arduous and so narrow that trekkers on it have to go in a single file. Our monk makes slow progress, he takes several breaks on the way to rest, and at last reaches the temple by evening. He spends the night worshipping at the temple. Next morning, he begins his return trip from the temple exactly at 6:00 AM, by the same path. On the return trip, since the path is downhill, he makes fast progress and reaches the point from where he started his hjourney on Saturday morning, well before the evening.

Suppose we call a point (or spot or location) on the path, a fixed point, if the monk was exactly at that spot at precisely the same time of the day on both the forward and return trips.

The existence of a fixed point on the path can be proved using Brouwer's fixed point theorem, but there is a much simpler and intuitive proof for its existence (see A. Koestler, *The Act of Creation*, Hutchinson, 1976, London). Imagine that on Saturday morning exactly at 6:00 AM, a duplicate monk starts from the temple, down the mountain, proceeding at every point of time at exactly the same rate that the original monk would on Sunday. So, at any point of time of the day on Saturday, the duplicate monk will be at the same location on the path as the original monk will be at the time on Sunday. Since the path is so narrow that both cannot pass without being in each other's way, the two monks must meet at some time during the day, and the spot on the path where they meet is a fixed point.

Successive Substitution Method for Computing a Brouwer's Fixed Point

One commonly used method to compute a Brouwer's fixed point of the single valued map $f(x) : \mathbf{\Gamma} \to \mathbf{\Gamma}$ is an iterative method that begins with an arbitrary point $x^0 \in \mathbf{\Gamma}$, and obtains a sequence of points $\{x^r : r = 0, 1, \ldots\}$ in $\mathbf{\Gamma}$ using the iteration

$$x^{r+1} = f(x^r) \ .$$

The sequence so generated, converges to a Brouwer's fixed point of $f(x)$ if $f(x)$ satisfies the **contraction property**, that is, if there exists a constant ν satisfying $0 \leqq \nu < 1$ such that for every $x, y \in \mathbf{\Gamma}$, we have

$$\|f(x) - f(y)\| \leqq \nu \|x - y\| \ . \tag{2.41}$$

If the map $f(x)$ satisfies the contraction proprety, this successsive substituitions method is a very convenient method for computing a Brouwer's fixed point of $f(x)$. Unfortunately, the contraction property is a strong property and does not usually hold in many practical applications.

Newton-Raphson Method for Solving a System of n Equations in n Unknowns

The system (2.40) is a system of n equations in n unknowns, and we can try to solve it using approaches for solving nonlinear equations of this type, like **Newton-Raphson method**, which we now present. The method is also called Newton's method often in the literature, or Newton's method for solving equations. Consider the system

$$g_i(x) = 0 \quad i = 1 \text{ to } n \tag{2.42}$$

where each $g_i(x)$ is a real valued function defined on $\mathbf{R}^n$. Assume that each function $g_i(x)$ is differentiable. Let $\nabla g_i(x)$ be the row vector of partial derivatives and let the **Jacobian** be

$$\nabla g(x) = \begin{pmatrix} \nabla g_1(x) \\ \vdots \\ \nabla g_n(x) \end{pmatrix}$$

in which the i^{th} row vector is the partial derivative vector of $g_i(x)$ written as a row.

To solve (2.42) the Newton-Raphson method begins with an arbitrary point x^0 and generates a sequence of points $\{x^0, x^1, x^2, \ldots\}$. Given x^r in the sequence, the method approximates (2.42) by its first order Taylor approximation around x^r leading to

$$g(x^r) + \nabla g(x^r)(x - x^r) = 0$$

whose solution is $x^r - (\nabla g(x^r))^{-1} g(x^r)$, which is taken as the next point in the sequence. This leads to the iteration

$$x^{r+1} = x^r - (\nabla g(x^r))^{-1} g(x^r) \ .$$

If the Jacobian is nonsingular, the quantity $y = (\nabla g(x^r))^{-1} g(x^r)$ can be computed efficiently by solving the system of linear equations

$$(\nabla g(x^r))y = g(x^r)$$

If the Jacobian $\nabla g(x^r)$ is singular, the inverse $(\nabla g(x^r))^{-1}$ does not exist and the method is unable to proceed further. Several modifications have been proposed to remedy this situation, see references [10.9, 10.13, 10.33]. Many of these modifications are based on the applications of Newton's method for unconstrained minimization or a modified version of it (see Sections 10.8.4, 10.8.5) to the least squares formulation of (2.42) leading to problem of finding the unconstrained minimum of

$$\sum_{i=1}^{n} (g_i(x))^{\mathbf{2}} \ .$$

As an example, consider the system

$$g_1(x) = x_1^{\mathbf{2}} + x_2^{\mathbf{2}} - 1 = 0$$
$$g_2(x) = x_1^{\mathbf{2}} - x_2 = 0$$

The Jacobian matrix is

$$\begin{pmatrix} 2x_1 & 2x_2 \\ 2x_1 & -1 \end{pmatrix}$$

Let $x^0 = (1,0)^T$ be the initial point. So $g(x^0) = (0,1)^T$. The Jacobian matrix at x^0 is $\begin{pmatrix} 2 & 0 \\ 2 & -1 \end{pmatrix}$. This leads to the next point $x^1 = (1,1)^T$. It can be verified that

$x^2 = \left(\frac{5}{6}, \frac{2}{3}\right)^T$, and so on. The actual solution in this example can be seen from Figure 2.9.

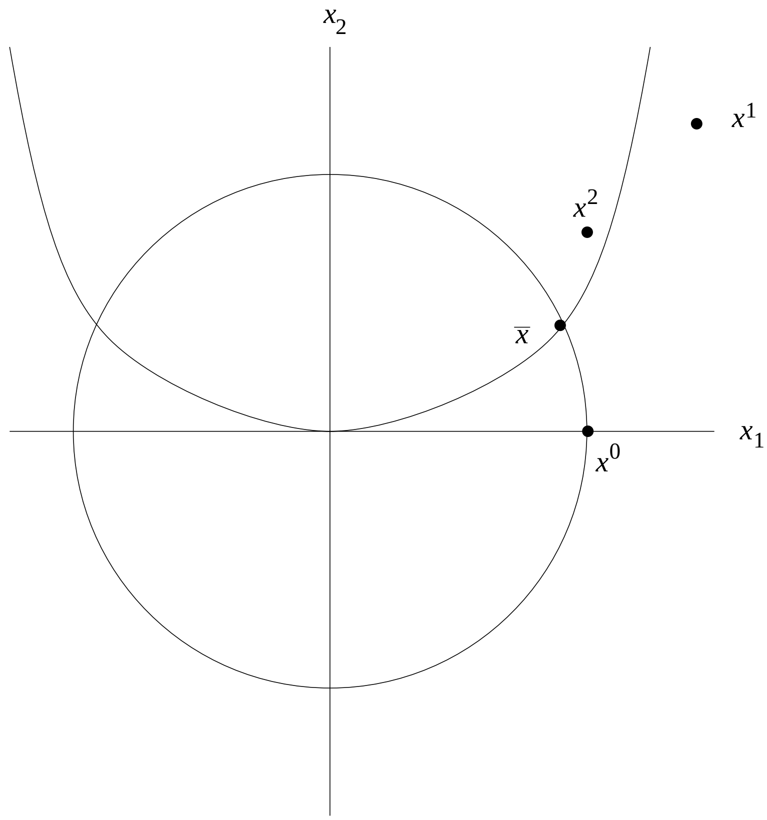

Figure 2.9 The circle here is the set of all points (x_1, x_2) satisfying $x_1^2 + x_2^2 - 1 = 0$. The parabola is the set of all points satisfying $x_1^2 - x_2 = 0$. The two intersect in two points (solutions of the system) one of which is $\bar{x}$. Beginning with x^0, the Newton-Raphson method obtains the sequence x^1, $x^2, \ldots$ converging to $\bar{x}$.

In order to solve (2.40) by Newton-Raphson method or some modified versions of it, the map $f(x)$ must satisfy strong properties like being differentiable etc., which do not hold in may many practical applications. Thus, to use Brouwer's fixed point theorem in practical applications we should devise methods for solving (2.40) without requiring the map $f(x)$ to satisfy any conditions besides continuity. In 1967 H. Scarf in a pioneering paper [2.68] developed a method for finding an approximate solution of (2.40) using a triangulation of the space, that walks through the simplices of the triangulation along a path satisfying properties similar to the one traced by the complementary pivot algorithm for the LCP. This method has the advantage that it works without requiring any conditions on the map $f(x)$ other than those required by Brouwer's theorem for the existence of the fixed point (i. e., continuity). Subsequently vastly improved versions of these methods have been developed by many researches. We will discuss one of these methods in detail.

Approximate Brouwer Fixed Points

Let $f(x) : \mathbf{\Gamma} \to \mathbf{\Gamma}$ be continuous as defined in Theorem 2.13. A true Brouwer fixed point of $f(x)$ is a solution of (2.40). However, in general, we may not be able to compute an exact solution of (2.40) using finite precision arithmetic. In practice, we attempt to compute an approximate Brouwer fixed point. There are two types of approximate Brouwer fixed points, we define them below.

Type 1: A point $\bar{x} \in \mathbf{\Gamma}$ is said to be an approximate Brouwer fixed point of $f(x)$ of Type 1 if

$$||\bar{x} - f(\bar{x})|| < \varepsilon$$

for some user selected tolerance ε (a small positive quantity).

Type 2: A point $x^* \in \mathbf{\Gamma}$ is said to be an approximate Brouwer fixed point of Type 2 if there exists an exact solution y of (2.40) such that

$$||x^* - y|| < \varepsilon \ .$$

In general, a Type 1 approximate Brouwer fixed point $\bar{x}$ may not be a Type 2 approximate Brouwer fixed point, that is, $\bar{x}$ may be far away from any exact solution of (2.40). If some strong conditions hold (such as: $f(x)$ is continuously differentiable in the interior of $\mathbf{\Gamma}$ and all the derivatives are Lipschitz continuous, or $f(x)$ is twice continuously differentiable in the interior of $\mathbf{\Gamma}$) a Type 1 approximate Brouwer fixed point can be shown to be also a Type 2 approximate Brouwer fixed point with a modified tolerance. At any rate, the algorithms discussed in the following sections are only able to compute approximate Brouwer fixed points of Type 1.

Kakutani Fixed Points

In many applications, the requirement that $f(x)$ be a point-to-point map is itself too restrictive. In 1941 S. Kakutani generalized Theorem 2.13 to point-to-set maps. As before, let $\mathbf{\Gamma}$ be a compact convex subset of $\mathbf{R}^n$. Let $\mathbf{F}(x)$ be a point-to-set map on $\mathbf{\Gamma}$, that is, for each $x \in \mathbf{\Gamma}$, $\mathbf{F}(x)$ is itself a specified subset of $\mathbf{\Gamma}$.

Example 2.20

Let $n = 1$. Let $\mathbf{\Gamma} = \{ x \in \mathbf{R}^1 : 0 \leqq x \leqq 1 \}$. For each $x \in \mathbf{\Gamma}$, suppose $\mathbf{F}(x) = \{ y : x \leqq y \leqq 1 \} = [x, 1]$. See Figure 2.10.

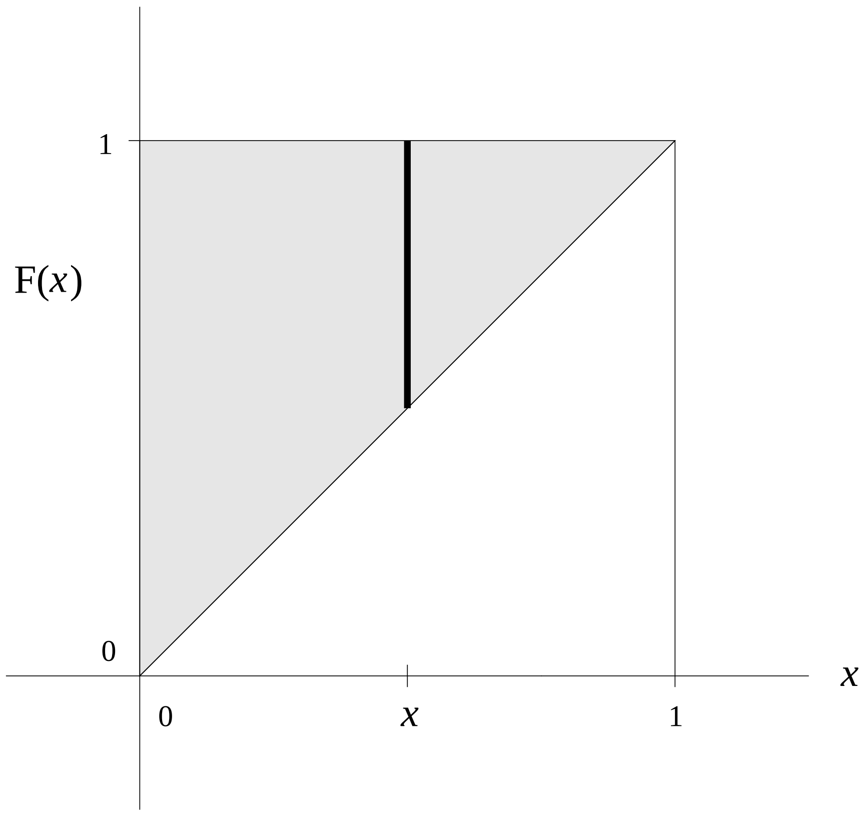

Figure 2.10 For each $x \in \mathbf{\Gamma}$, $\mathbf{F}(x)$ is the closed interval $[x, 1]$.

We consider only maps in which $\mathbf{F}(x)$ is a compact convex subset of $\mathbf{\Gamma}$ for each $x \in \mathbf{\Gamma}$. The point-to-set map $\mathbf{F}(x)$ is said to be an USC (Upper Semi-Continuous) map if it satisfies the following properties. Let $\{ x^k : k = 1, 2, \dots \}$ be any sequence of points in $\mathbf{\Gamma}$ converging to a point $x^\star \in \mathbf{\Gamma}$. For each k, suppose y^k is an arbitrary point selected from $\mathbf{F}(x^k)$, $k = 1, 2, \dots$. Suppose that the sequence $\{ y^k : k = 1, 2, \dots \}$ converges to the point $y^\star$. The requirement for the upper semi-continuity of the point-to-set map $\mathbf{F}(x)$ is that these conditions imply that $y^\star \in \mathbf{F}(x^\star)$.

It can be verified that the point-to-set map $\mathbf{F}(x)$ given in Figure 2.8 satisfies this USC property.

Theorem 2.14 Kakutani's Fixed Point Theorem

If $\mathbf{F}(x)$ is a USC point-to-set map defined on the compact convex subset $\mathbf{\Gamma} \subset \mathbf{R}^n$, there exists a point $x \in \mathbf{\Gamma}$ satisfying

$$x \in \mathbf{F}(x) \ . \tag{2.43}$$

Any point satisfying (2.43) is known as a **Kakutani's fixed point** of the point-to-set map $\mathbf{F}(x)$. To prove his theorem, Kakutani used the fundamental notion of a **piecewise linear approximation** to the map $\mathbf{F}(x)$. The same picewise linear approximation scheme is used in the method discussed later on for computing fixed points. See reference [2.50] for the proof of Kakutani's theorem.

For each $x \in \mathbf{\Gamma}$, if $\mathbf{F}(x)$ is a singleton set (i. e., a set containing only a single element) $\{f(x)\} \subset \mathbf{\Gamma}$, it can be verified that this $\mathbf{F}(x)$ is USC iff $f(x)$ is continuous. Thus the USC property of point-to-set maps is a generalization of the continuity property of point-to-point maps. Also, every Brouwer fixed point of the point-to-point map $f(x)$ can be viewed as a Kakutani fixed point of $\mathbf{F}(x) = \{f(x)\}$.

Approximate Kakutani Fixed Points

Given the USC point-to-set map $\mathbf{F}(x)$ as defined in Theorem 2.14, a Kakutani fixed point is a point $x \in \mathbf{\Gamma}$ satisfying (2.43). As under the Brouwer fixed point case, using finite precision arithmetic, we may not be able to find $x \in \mathbf{\Gamma}$ satisfying (2.43) exactly. We therefore attempt to compute an approximate Kakutani fixed point. Again, there are two types of approximate Kakutani fixed points, we define them below

Type 1: A point $\bar{x} \in \mathbf{\Gamma}$ is said to be an approximate Kakutani fixed point of $\mathbf{F}(x)$ of Type 1 if there exists a $z \in \mathbf{F}(\bar{x})$ satisfying

$$||\bar{x} - z|| < \varepsilon$$

for some user selected tolerance ε (a small positive quantity).

Type 2: A point $x^* \in \mathbf{\Gamma}$ is said to be an approximative Kakutani fixed point of $\mathbf{F}(x)$ of Type 2 if there exists a y satisfying (2.43) and

$$||x^* - y|| < \varepsilon \ .$$

The algorithms discussed in the following sections are only able to compute Type 1 approximate Kakutani fixed points.

Use in Practical Applications

In pratical applications we have to deal with either point-to-point or point-to-set maps defined over the whole space $\mathbf{R}^n$, not necessarily on only a compact convex subset of $\mathbf{R}^n$. Also, it is very hard, if not computationally impossible, to check whether properties like USC etc. hold for our maps. For such maps, the existence of a fixed point is not guaranteed. Because of this, the algorithms that we discuss for computing fixed points may not always work on these problems. Also, it is impossible for us to continue the computation indefinitely, we have to terminate after a finite number of steps. In practice, from the path traced by the algorithm, it will be clear whether it seems to be converging, or running away. If it seems to be converging, from the point obtained at termination, an approximate solution of the problem can be obtained. If the algorithm seems to be running away, either we can conclude that the algorithm has failed to solve the problem, or an effort can be made to run the algorithm again with different initial conditions. Before discussing the algorithm, we will now discuss some standard applications of fixed point computing.

2.7.3 Applications in Unconstrained Optimization

Let $\theta(x)$ be a real valued function defined on $\mathbf{R}^n$ and suppose it is required to solve the problem

$$\begin{array}{ll} \text{minimize} & \theta(x) \\ \text{over} & x \in \mathbf{R}^n \end{array} \tag{2.44}$$

If $\theta(x)$ is differentiable, a necessary condition for a point $x \in \mathbf{R}^n$ to be a local minimum for (2.44) is

$$\nabla\theta(x) = 0 \tag{2.45}$$

which is a system of n equations in n unknowns. Define $f(x) = x - (\nabla\theta(x))^T$. Then every Brouwer fixed point of $f(x)$ is a solution of (2.45) and vice versa. Hence every fixed point of $f(x)$ satisfies the first order necessary optimality conditions for (2.44). If $\theta(x)$ is convex, every solution of (2.45) is a global minimum for (2.44) and vice versa, and hence in this case (2.44) can be solved by computing a fixed point for $f(x)$ defined above. However, if $\theta(x)$ is not convex, there is no guarantee that a solution of (2.45), (i.e., a fixed point of $f(x) = x - (\nabla\theta(x))^T$) is even a local minimum for (2.44) (it could in fact be a local maximum). So, after obtaining an approximate fixed point, $\bar{x}$, of $f(x)$, one has to verify whether it is a local minimum or not. If $\theta(x)$ is twice continuously differentiable, a sufficient condition for a solution of (2.45) to be a local minimum for (2.44) is that the Hessian matrix $H(\theta(\bar{x}))$ be positive definite.

If $\theta(x)$ is not differentiable at some points, but is convex, then the subdifferential set $\partial\theta(x)$ exists for all x. In this case define $\mathbf{F}(x) = \{\, x - y : y \in \partial\theta(x) \,\}$. Then every Kakutani fixed point of $\mathbf{F}(x)$ is a global minimum for (2.44) and vice versa.

One strange feature of the fixed point formulation for solving (2.45) is worth mentioning. Define $\mathbf{G}(x) = \{\, x + y : y \in \partial\theta(x) \,\}$. Clearly, every Kakutani fixed point of $\mathbf{G}(x)$ also satisfies the necessary optimality conditions for (2.44). Mathematically, the problems of finding a Kakutani fixed point of $\mathbf{F}(x)$ or $\mathbf{G}(x)$ are equivalent, but the behavior of the fixed point computing algorithm discussed in Section 2.7.8 on the two problems could be very different. This is discussed later on under the subsection entitled, "Sufficient Conditions for Finite Termination" in Section 2.7.8. In practical applications, one might try computing the Kakutani fixed point of $\mathbf{F}(x)$ using the algorithm discussed in Section 2.7.8, and if its performance is not satisfactory switch over and use the same algorithm on $\mathbf{G}(x)$ instead.

2.7.4 Application to Solve a System of Nonlinear Inequalities

Consider the system

$$g_i(x) \leqq 0, \quad \text{for } i = 1 \text{ to } m \tag{2.46}$$

where each $g_i(x)$ is a real valued convex function defined on $\mathbf{R}^n$. Define the pointwise supremum function $s(x) = \text{Maximum}\,\{\, g_1(x), \dots, g_m(x) \,\}$. As discussed earlier, $s(x)$ is itself convex, and $\partial s(x) \supset \bigcup_{i \in \mathbf{J}(x)} \partial g_i(x)$, where $\mathbf{J}(x) = \{\, i : g_i(x) = s(x) \,\}$. If each $g_i(x)$ is differentiable, then $\partial s(x) =$ convex hull of $\{\, \nabla g_i(x) : i \in \mathbf{J}(x) \,\}$. If (2.46) has a feasible solution $\bar{x}$, then $s(\bar{x}) \leqq 0$, and conversely every point x satisfying $s(x) \leqq 0$ is feasible to (2.46). So the problem of finding a feasible solution of (2.46) can be tackled by finding the unconstrained minimum of $s(x)$, which is the same as the problem of finding a Kakutani fixed point of $\mathbf{F}(x) = \{\, x - y : y \in \partial s(x) \,\}$ as discussed in Section 2.7.3. If $\bar{x}$ is a Kakutani fixed point of this map and $\bar{s}(x) > 0$, (2.46) is infeasible. On the other hand if $s(\bar{x}) \leqq 0$, $\bar{x}$ is a feasible solution of (2.46).

2.7.5 Application to Solve a System of Nonlinear Equations

Consider the system of equations

$$h_i(x) = 0, \quad i = 1 \text{ to } r \tag{2.47}$$

where each $h_i(x)$ is a real valued function defined on $\mathbf{R}^n$. Let $h(x) = (h_1(x), \dots, h_r(x))^T$. If $r > n$, (2.47) is said to be an **overdetermined system**. In this case there may be no solution to (2.47), but we may be interested in finding a point $x \in \mathbf{R}^n$ that satisfies (2.47) as closely as possible. The **least squares approach** for finding this is to look for the unconstrained minimum of $\sum_{i=1}^{r} (h_i(x))^{\mathbf{2}}$, which can be posed as a fixed point problem as in Section 2.7.3.

If $r < n$, (2.47) is known as an **underdetermined system**, and it may have many solutions. It may be possible to develop additional $n - r$ equality constraints which when combined with (2.47) becomes a system of n equations in n unknowns. Or the least squares method discussed above can be used here also.

Assume that $r = n$. In this case define $f(x) = x - h(x)$. Then every Brouwer fixed point of $f(x)$ solves (2.47) and vice versa. As mentioned in Section 2.7.3, it may be worthwhile to also consider the equivalent problem of computing the fixed point of $d(x) = x + h(x)$ in this case.

2.7.6 Application to Solve the Nonlinear Programming Problem

Consider the nonlinear program

$$\begin{array}{ll} \text{Minimize} & \theta(x) \\ \text{subject to} & g_i(x) \leqq 0, \quad i = 1 \text{ to } m \end{array} \tag{2.48}$$

where $\theta(x)$, $g_i(x)$ are real valued functions defined over $\mathbf{R}^n$. We will assume that each of these functions is convex, and continuously differentiable. We make an additional assumption that if (2.48) is feasible (i. e., the set $\{ x : g_i(x) \leqq 0, i = 1 \text{ to } m \} \neq \emptyset$), then there exists an $x \in \mathbf{R}^n$ satisfying $g_i(x) < 0$, for each $i = 1$ to m. This assumption is known as a **constraint qualification**. As before, let $s(x)$ be the pointwise supremum function, maximum $\{ g_1(x), \ldots, g_m(x) \}$. Then (2.48) is equivalent to

$$\begin{array}{ll} \text{Minimize} & \theta(x) \\ & s(x) \leqq 0 \end{array} \tag{2.49}$$

By our assumption, and the results discussed earlier, $s(x)$ is also convex and $\partial s(x) =$ convex hull of $\{ \nabla g_i(x) : i \in \mathbf{J}(x) \}$, where $\mathbf{J}(x) = \{ i : s(x) = g_i(x) \}$. Consider the following point-to-set mapping defined on $\mathbf{R}^n$.

$$\mathbf{F}(x) = \begin{cases} \{ x - (\nabla\theta(x))^T \}, & \text{if } s(x) < 0 \,, \\ \{ x - y : \ y \in \text{ convex hull of } \{\nabla\theta(x), \partial s(x)\} \}, & \text{if } s(x) = 0 \,, \\ \{ x - y : \ y \in \partial s(x) \}, & \text{if } s(x) > 0 \,. \end{cases} \tag{2.50}$$

Under our assumptions of convexity and differentiability, it can be verified that $\mathbf{F}(x)$ defined in (2.50) is USC. Let $\bar{x}$ be a Kakutani fixed point of $\mathbf{F}(x)$. If $s(\bar{x}) < 0$, then $0 = \nabla\theta(\bar{x})$, and thus $\bar{x}$ is a global minimum for $\theta(x)$ over $\mathbf{R}^n$ and is also feasible to (2.48), and therefore solves (2.48). If $s(\bar{x}) > 0$, then $0 \in \partial s(\bar{x})$, thus 0 is a global minimum of $s(x)$, and since $s(\bar{x}) > 0$, (2.48) has no feasible solution. If $s(\bar{x}) = 0$, then $0 \in$ convex hull of $\{\nabla\theta(\bar{x}), \partial s(\bar{x}) \} =$ convex hull of $\{\nabla\theta(\bar{x}), \nabla g_i(\bar{x})$ for $i \in \mathbf{J}(\bar{x})\}$, so there exists nonnegative numbers λ_0, λ_i for $i \in \mathbf{J}(\bar{x})$ satisfying

$$\begin{aligned} \lambda_0 \nabla\theta(\bar{x}) + \sum_{i \in \mathbf{J}(\bar{x})} \lambda_i \nabla g_i(\bar{x}) &= 0 \\ \lambda_0 + \sum_{i \in \mathbf{J}(\bar{x})} \lambda_i &= 1 \\ \lambda_0, \ \lambda_i \geqq 0 \quad \text{for all } i &\in \mathbf{J}(\bar{x}) \end{aligned} \tag{2.51}$$

If $\lambda_0 = 0$, (2.51) implies that $0 \in \partial s(\bar{x})$ and so $s(\bar{x})$ is a global minimizer of $s(\bar{x})$, $\bar{x}$ is feasible to (2.48) since $s(\bar{x}) = 0$, and these facts lead to the conclusion that $\{ x : g_i(x) \leqq 0, \text{ for } i = 1 \text{ to } m \} \neq \emptyset$ and yet there exists no x satisfying $g_i(x) < 0$ for all $i = 1$ to m, violating our constraint qualification assumption. So $\lambda_0 > 0$ in (2.51).

So if we define $\bar{\pi}_i = \frac{\lambda_i}{\lambda_0}$ if $i \in \mathbf{J}(\bar{x})$, $= 0$ otherwise, then from (2.51) we conclude that $\bar{x}$, $\bar{\pi}$ together satisfy the Karush-Kuhn-Tucker necessary conditions for optimality for (2.48), and our convexity assumption imply that $\bar{x}$ is the global minimum for (2.48).

Thus solving (2.48) is reduced to the problem of finding a Kakutani fixed point of the mapping $\mathbf{F}(x)$ defined in (2.50).

Example 2.21

Consider the problem :

$$\begin{array}{ll} \text{minimize} & \theta(x) = x_1^2 + x_2^2 - 2x_1 - 3x_2 \\ \text{subject to} & g_1(x) = x_1 + x_2 \leqq 1 \end{array} \tag{2.52}$$

Clearly $\nabla\theta(x) = (2x_1 - 2, 2x_2 - 3)$, $\nabla g_1(x) = (1, 1)$. The mapping $\mathbf{F}(x)$ for this problem is

$$\mathbf{F}(x) = \begin{cases} \{ -x_1 + 2, -x_2 + 3)^T \}, & \text{if } x_1 + x_2 < 1 \ , \\ \text{Convex hull of } \{(-x_1 + 2, -x_2 + 3)^T, (x_1 - 1, x_2 - 1)^T\}, & \text{if } x_1 + x_2 = 1 \ , \\ \{ (x_1 - 1, x_2 - 1)^T \}, & \text{if } x_1 + x_2 > 1 \ . \end{cases}$$

It can be verified that $\bar{x} = (\frac{1}{4}, \frac{3}{4})^T$ is a Kakutani fixed point of this mapping $\mathbf{F}(x)$, and that $\bar{x}$ is the global optimum solution of the nonlinear program (2.52).

If $\theta(x)$, $g_i(x)$ are all continuously differentiable, but not necessarily convex, we can still define the point-to-set mapping $\mathbf{F}(x)$ as in (2.50) treating $\partial s(x) =$ convex hull of $\{ \nabla g_i(x) : i \in \mathbf{J}(x) \}$. In this general case, any Kakutani fixed point $\bar{x}$ of $\mathbf{F}(x)$ satisfies the first order necessary optimality conditions for (2.48), but these conditions are not sufficient to guarantee that $\bar{x}$ is a global or even a local minimum for (2.48), see Section 10.2 for definitions of a global minimum, local minimum. One can then try to check whether $\bar{x}$ satisfies some sufficient condition for being a local minimum for (2.48) (for example, if all the functions are twice continuously differentiable, a sufficient condition for $\bar{x}$ to be a local minimum for (2.48) is that the Hessian matrix of the Lagrangian with respect to x is positive definite at $\bar{x}$. See references [10.2, 10.3, 10.13, 10.17, A8, A12]). If these sufficient optimality conditions are not satisfied, it may be very hard to verify whether $\bar{x}$ is even a local minimum for (2.48). As an example, consider the problem: minimize $x^T D x$, subject to $x \geqq 0$. The point $0 \in \mathbf{R}^n$ is a global minimum for this problem if D is PSD. If D is not PSD, 0 is a local minimum for this problem iff D is a copositive matrix. Unfortunately, there are as yet no efficient methods known for checking whether a matrix which is not PSD, is copositive. See Section 2.9.3.

Thus, in the general nonconvex case, the fixed point approach for (2.48) finds a point satisfying the first order necessary optimality conditions for (2.48), by computing a Kakutani fixed point of $\mathbf{F}(x)$ defined in (2.50). In this general case, many of the other solution techniques of nonlinear programming for solving (2.48) (see Chapter 10) are usually based on **descent methods**. These techniques generate a sequence of points

$\{ x^r : r = 0, 1, \dots \}$. Given x^r, they generate a $y^r \neq 0$ such that the direction $x^r + \lambda y^r$, $\lambda \geqq 0$, is a **descent direction**, that is, it is either guaranteed to decrease the objective value or a measure of the infeasibility of the current solution to the problem or some criterion function which is a combination of both. The next point in the sequence x^{r+1} is usually taken to be the point which minimizes the criterion function on the half line $\{ x^r + \lambda y^r : \lambda \geqq 0 \}$ obtained by using some one dimensional (λ is the only variable to be determined in this problem) line minimization algorithm. And the whole process is then repeated with the new point. On general problems, these methods suffer from the same difficulties, they cannot theoretically guarantee that the point obtained at termination is even a local minimum. However, these descent methods do seem to have an edge over the fixed point method presented above in the general case. In the absence of convexity, one has more confidence that a solution obtained through a descent process is likely to be a local minimum, than a solution obtained through fixed point computation which is based purely on first order necessary conditions for optimality.

The approach for solving the nonlinear program (2.48) using the fixed point transformation has been used quite extensively, and seems to perform satisfactorily. See references [2.40, 2.58, 2.59].

Many practical nonlinear programming models tend to be nonconvex. The fixed point approach outlined above, provides additional arsenal in the armory for tackling such general problems.

Now consider the general nonlinear programming problem in which there are both equality and inequality constraints.

$$\begin{array}{lll} \text{minimize} & \theta(x) & \\ \text{subject to} & g_i(x) \leqq 0, & i = 1 \text{ to } m \\ & h_t(x) = 0, & t = 1 \text{ to } p \end{array} \tag{2.53}$$

The usual approach for handling (2.53) is the **penality function method** which includes a term with a large positive coefficient corresponding to a measure of violation of the equality constraints in the objective function. One such formulation leads to the problem

$$\begin{array}{ll} \text{minimize} & \theta(x) + \alpha \sum_{t=1}^{p} (h_t(x))^{\mathbf{2}} \\ \text{subject to} & g_i(x) \leqq 0, \quad i = 1 \text{ to } m \end{array} \tag{2.54}$$

In (2.54), α, a large positive number, is the **penalty parameter**. If (2.53) has a feasible solution, every optimum solution of (2.54) would tend to satisfy $h_t(x) = 0$, $t = 1$ to p as α becomes very large, and thus would also be optimal to (2.53). When α is fixed to be a large positive number, (2.54) is in the same form as (2.48), and can be tackled through a fixed point formulation as discussed above.

Advantages and Disadvantages of this Approach

In the NLP (2.48) there may be several constraints (i, e., m may be large) and the problem difficulty can be expected to increase with the number of constraints. The fixed point approach for solving (2.48), first transforms (2.48) into the equivalent (2.49), which is an NLP in which there is only a single constraint. The fact that (2.49) is a single constraint problem is definitely advantageous.

The original problem (2.48) is a smooth problem since the objective and constraint functions are all assumed to be continuously differentiable. Eventhough $g_i(x)$ are continuously differentiable for all i, there may be points x where $s(x)$ is not differentiable. However, $s(x)$ is differentiable almost everywhere and so (2.49) is a nonsmooth NLP. That this approach transforms a nice smooth NLP into a nonsmooth NLP is a disadvantage. But, because of the special nature of the function $s(x)$, for any x, we are able to compute a point in the subdifferential set $\partial s(x)$ efficiently, as discussed above. For computing a fixed point of the map $\mathbf{F}(x)$ defined in (2.50), the algorithms discussed in the following sections need as inputs only subroutines to compute $\nabla\theta(x)$, or a point from $\partial s(x)$ for any given x, which are easy to provide. Thus, eventhough (2.49) is a nonsmooth NLP, the fixed point approach is able to handle it efficiently. Practical computational experience with this approach is quite encouraging.

The fixed point approach solves NLPs using only the first order necessary conditions for optimality. The objective value is never computed at any point. This is a disadvantage in this approach. In nonconvex NLPs, a solution to the first order necessary conditions for optimality, may not even be a local minimum. Since the objective value is not used or even computed in this approach, we lack the circumstantial evidence, or the neighborhood information about the behaviour of objective values, to conclude that the final solution obtained is at least likely to be a local minimum.

2.7.7 Application to Solve the Nonlinear Complementarity Problem

As discussed in Section 1.6, the nonlinear complementary problem (NLCP) is the following. Given $g(x) = (g_1(x), \dots, g_n(x))^T : \mathbf{R}^n_+ \to \mathbf{R}^n$, where $\mathbf{R}^n_+$ is the nonnegative orthant of $\mathbf{R}^n$, find $x \geqq 0$ satisfying $g(x) \geqq 0$, $x^T g(x) = 0$.

Define $\psi(x) =$ Maximum $\{-x_1, \dots, -x_n\}$. So $\partial\psi(x) =$ convex hull of $\{-I_{.j} : j$ such that $-x_j \geqq -x_i$ for all $i = 1$ to n in $x\}$. Define the point-to-set map on $\mathbf{R}^n$,

$$\mathbf{F}(x) = \begin{cases} \{x - y : y \in \partial\psi(x)\}, & \text{if } \psi(x) > 0 \,, \\ \{x - y : y \in \text{ convex hull of } \{g(x), \partial\psi(x)\}\}, & \text{if } \psi(x) = 0 \,, \\ \{x - g(x)\}, & \text{if } \psi(x) < 0 \,. \end{cases} \tag{2.55}$$

It can be verified that every Kakutani fixed point of $\mathbf{F}(x)$ defined here is a solution of the NLCP and vice versa. Thus the NLCP can be solved by computing a Kakutani fixed point of $\mathbf{F}(x)$.

2.7.8 Merrill's Algorithm for Computing a Kakutani Fixed Point

Let $\mathbf{F}(x)$ be a point-to-set map defined on $\mathbf{R}^n$. We describe in this section, Merrill's method for computing a Kakutani fixed point of $\mathbf{F}(x)$.

Data Requirements of the Algorithm

If the algorithm requires the storage of the complete set $\mathbf{F}(x)$ for any x, it will not be practically useful. Fortunately, this algorithm does not require the whole set $\mathbf{F}(x)$ for even one point $x \in \mathbf{R}^n$. It only needs a computational procedure (or a subroutine), which, for any given $x \in \mathbf{R}^n$, outputs one point from the set $\mathbf{F}(x)$. The algorithm will call this subroutine a finite number of times. Thus the data requirements of the algorithm are quite modest, considering the complexity of the problem being attempted, and it can be implemented for the computer very efficiently. Also, the primary computational step in the algorithm is the pivot step, which is the same as that in the simplex method for linear programs.

n-Dimensional Simplex

The points $v_1, \ldots, v_r$ in $\mathbf{R}^n$ are the vertices of an $(r-1)$ dimensional simplex if the set of column vectors $\left\{ \begin{pmatrix} 1 \\ v_1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ v_r \end{pmatrix} \right\}$ in $\mathbf{R}^{n+1}$ form a linearly independent set. The simplex itself is the convex hull of its vertices and will be denoted by the symbol $\langle v_1, \ldots, v_r \rangle$. Given the simplex with vertices $v_1, \ldots, v_r$, the convex hull of any subset of its vertices is a **face** of the simplex. An n-dimensional simplex has $(n+1)$ vertices. See Figure 2.11. Clearly a 1-dimensional simplex is a line segment of positive length joining two distinct points, a 2-dimensional simplex is the triangle enclosed by three points which are not collinear, etc.

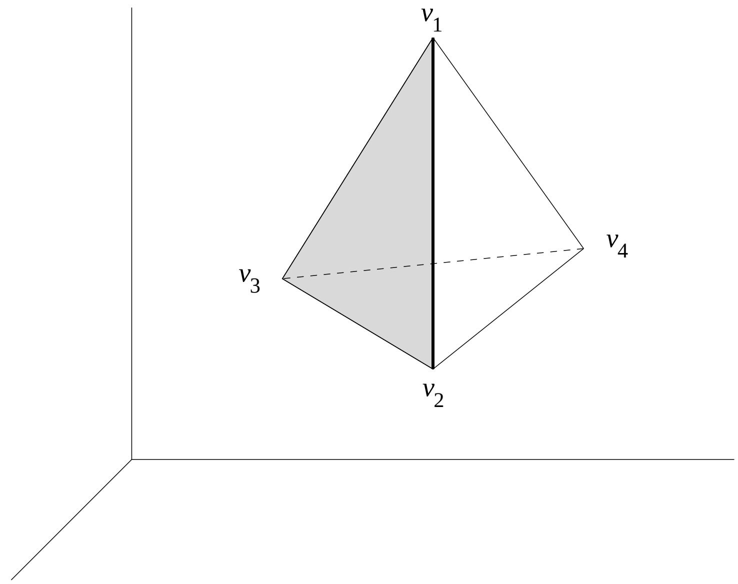

Figure 2.11 The tetrahedron which is the convex hull of vertices $\{v_1, v_2, v_3, v_4\}$ is a 3-dimensional simplex. Its vertices v_1, v_2, v_3, v_4 are its 0-dimensional faces. Its 6 edges, of which the thick line segment joining v_1 and v_2 is one, are its 1-dimensional faces. The dashed 2-dimensional simplex which is the convex hull of $\{v_1, v_2, v_3\}$ is one of the four 2-dimensional faces of the tetrahedron.

Triangulations

Let $\mathbf{K}$ be either $\mathbf{R}^n$ or a convex polyhedral subset of $\mathbf{R}^n$ of dimension n. A **triangulation** of $\mathbf{K}$ is a partition of $\mathbf{K}$ into simplexes satisfying the following properties

i) the simplexes cover $\mathbf{K}$,

ii) if two simplexes meet, their intersection is a common face,

iii) each point $x \in \mathbf{K}$ has a neighborhood meeting only a finite number of the simplexes,

iv) each $(n-1)$ dimensional simplex in the triangulation is the face of either two n-dimensional simplexes (in which case, the $(n-1)$ dimensional simplex is said to be an interior face in the triangulation) or exactly one n-dimensional simplex (in this case the $(n-1)$ dimensional simplex is said to be a boundary face in the triangulation),

v) for every point $x \in \mathbf{K}$ there exists a unique least dimension simplex, say σ, in the triangulation, containing x. If dimension of σ is $< n$, σ may be a face of several simplexes in the triangulation of dimension $>$ dimension of σ, and x is of course contained on the boundary of each of them. There exists a unique expression for x as a convex combination of vertices of σ, and this is the same expression for x as the convex combination of the vertices of any simplex in the triangulation containing x.

Example 2.22

In Figure 2.12 we give a triangulation of the unit cube in $\mathbf{R}^2$. The two 2-dimensional simplexes in this triangulation are the convex hulls of $\{v_0, v_1, v_2, \}$, $\{v_0, v_3, v_2\}$. The thick line segments in Figure 2.12 are the 1-dimensional simplexes in this triangulation which are the faces of exactly one two dimensional simplex. These 1-dimensional simplexes are the boundary faces in this triangulation. The thin diagonal line segment joining vertices v_0 and v_2 is the face of exactly two 2-dimensional simplexes, and hence is an interior face in this triangulation.

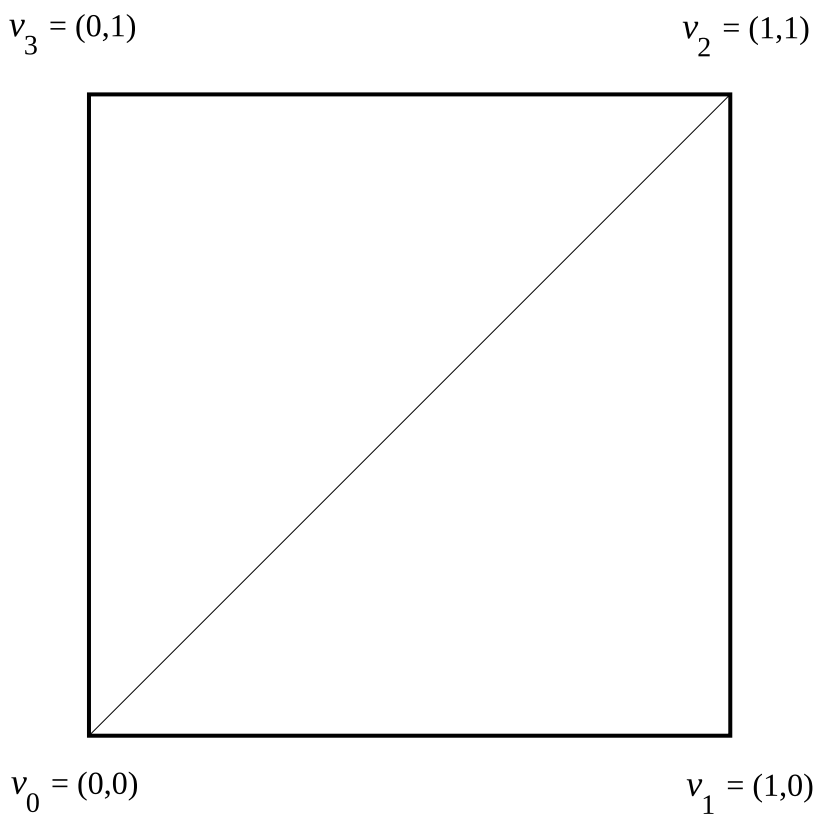

Figure 2.12 Triangulation K_1 of the unit square in $\mathbf{R}^2$.

Example 2.23

Consider the partition of the unit square in $\mathbf{R}^2$ into simplexes in Figure 2.13. It is not a triangulation since the two simplexes $\langle v_1, v_2, v_3\rangle$ and $\langle v_3, v_4, v_5\rangle$ intersect in $\langle v_3, v_5\rangle$ which is a face of $\langle v_3, v_4, v_5\rangle$ but not a face of $\langle v_1, v_2, v_3\rangle$ (it is a proper subset of the face $\langle v_2, v_3\rangle$ of $\langle v_1, v_2, v_3\rangle$). So the partition of the unit square in $\mathbf{R}^2$ in Figure 2.13 into simplexes violates property (ii) given above, and is therefore not a triangulation.

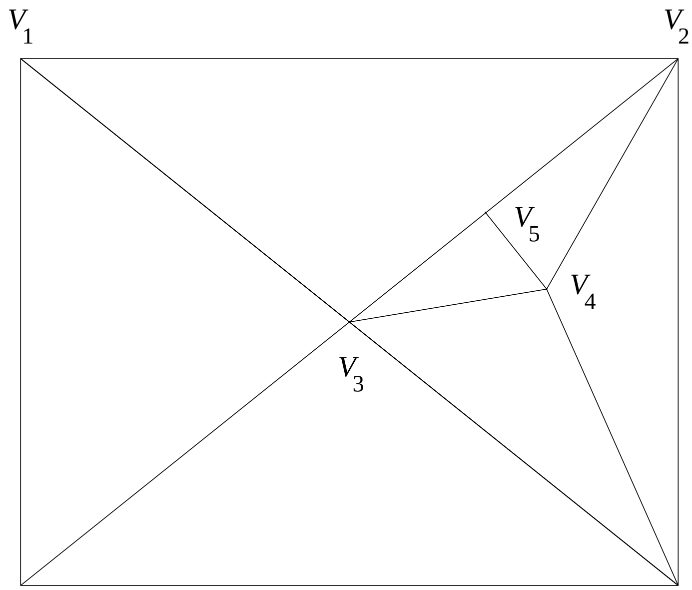

Figure 2.13 A partition of the unit cube in $\mathbf{R}^2$ into simplexes which is not a triangulation.

The triangulation for the unit cube in $\mathbf{R}^2$ given in Example 2.22 can be generalized to a triangulation of the unit cube in $\mathbf{R}^n$ which we call triangulation K_1, discussed by Freudenthal in 1942. The vertices of the simplexes in this triangulation are the same as the vertices of the unit cube. There are $n!$ n-dimensional simplexes in this triangulation. Let $v_0 = 0 \in \mathbf{R}^n$. Let $p = (p_1, \dots, p_n)$ be any permutation of $\{1, \dots, n\}$. Each of the $n!$ permutations p leads to an n-dimensional simplex in this triangulation. The n-dimensional simplex associated with the permutation p, denoted by (v_0, p), is $\langle v_0, v_1, \dots, v_n \rangle$ where

$$v_i = v_{i-1} + I_{\cdot p_i}, \quad i = 1 \text{ to } n \ . \tag{2.56}$$

In (2.56), I is the unit matrix of order n. For example, for $n = 2$, $p = (1, 2)$, we get the simplex $\langle v_0 = (0, 0)^T, v_1 = (1, 0)^T, v_2 = (1, 1)^T \rangle$. See Figure 2.12. See reference [2.72] for a proof that this does provide a triangulation of the unit cube of $\mathbf{R}^n$.

In this representation (v_0, p) for the simplex discussed above, v_0 is known as the **initial** or the 0^{th} **vertex** of this simplex. The other vertices of this simplex are obtained recursively as in (2.56). The vertex v_i is called the i^{th} **vertex** of this simplex for $i = 1$ to n.

This triangulation can be extended to provide a triangulation for the whole space $\mathbf{R}^n$ itself, which we call triangulation K_1 (it has been called by other symbols like K, I, etc., in other references) by first partitioning $\mathbf{R}^n$ into unit cubes using the integer points in $\mathbf{R}^n$, and then triangulating each unit cube as above. The vertices in this triangulation are all the points with integer coordinates in $\mathbf{R}^n$. Let $\bar{v}$ be any such vertex, and let $p = (p_1, \dots, p_n)$ be any permutation of $\{1, \dots, n\}$. Define $v_0 = \bar{v}$, and obtain v_i for $i = 1$ to n as in (2.56). Let $(\bar{v}, p)$ denote the simplex $\langle v_0, v_1, \dots,$

$v_n\rangle$. The set of all such simplexes as $\bar{v}$ ranges over all points with integer coordinates in $\mathbf{R}^n$, and p ranges over all the permutations of $\{1, \ldots, n\}$ is the collection of all the n-dimensional simplexes in this triangulation K_1. Again see reference [2.72] for a proof that this is indeed a triangulation of $\mathbf{R}^n$. See Figure 2.14.

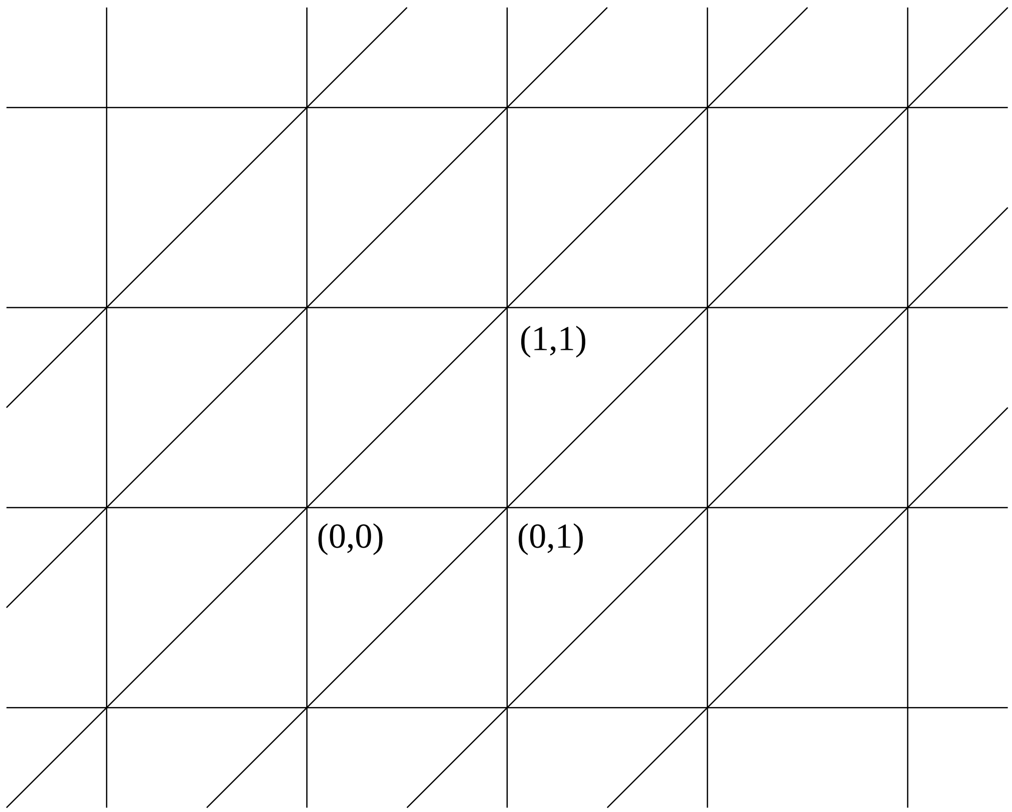

Figure 2.14 A partition of the unit cube in $\mathbf{R}^2$ into simplexes which is not a triangulation.

The **mesh** of a triangulation is defined to be the maximum Euclidean distance between any two points in a simplex in the triangulation. Clearly the mesh of triangulation K_1 of $\mathbf{R}^n$ is $\sqrt{n}$.

We can get versions of triangulation K_1 with smaller mesh by scaling the variables appropriately. Also the origin can be translated to any specified point. Let $x^0 \in \mathbf{R}^n$ be any specified point and δ a positive number. Let $\mathbf{J} = \{ x : x = (x_j) \in \mathbf{R}^n, x_j - x_j^0$ is an integer multiple of δ for all $j = 1$ to $n \}$. For any $v_0 \in \mathbf{J}$, and $p = (p_1, \ldots, p_n)$, a permutation of $\{1, \ldots, n\}$, define

$$v_i = v_{i-1} + \delta I_{\cdot p_i}, \quad i = 1 \text{ to } n \ . \tag{2.57}$$

Let (v_0, p) denote the simplex $\langle v_0, v_1, \ldots, v_n\rangle$. The set $\mathbf{J}$ are the vertices, and the set of all simplexes (v_0, p) as v_0 ranges over $\mathbf{J}$ and p ranges over all the permutations of $\{1, 2, \ldots, n\}$ are the n-dimensional simplexes, in the triangulation of $\mathbf{R}^n$. We denote this triangulation by the symbol $\delta K_1(x^0)$. Its mesh is $\delta\sqrt{n}$.

How is the Triangulation used by the Algorithm ?

The algorithm traces a path. Each step in the path walks from one $(n-1)$-dimensional face of an n-dimensional simplex in the triangulation, to another $(n-1)$-dimensional face of the same simplex, and continues this way. See Figure 2.15. The path traced is unambiguous once it is started, and is similar to the one in the ghost story mentioned earlier, or the path traced by the complementary pivot method for the LCP. Computationally, the algorithm associates a column vector in $\mathbf{R}^n$ to each vertex in the triangulation. At each stage, the columns associated with the vertices of the current $(n-1)$-dimensional simplex form a basis, and the inverse of this basis is maintained. A step in the algorithm corresponds to the pivot step of entering the column associated with a new entering vertex into the basis. The path never returns to a simplex it has visited earlier.

To execute the path, one may consider it convenient to store all the simplexes in the triangulation explicitly. If this is necessary, the algorithm will not be practically useful. For practical efficiency the algorithm stores the simplexes using the mathematical formulae given above, which are easily programmed for the computer. The current simplex is always maintained by storing its 0^{th} vertex and the permutation corresponding to it. To proceed along the path efficiently, the algorithm provides very simple rules for termination once a desirable $(n-1)$-dimensional simplex in the triangulation is reached (this is clearly spelled out later on). If the termination condition is not satisfied, a mathematical formula provides the entering vertex. A minimum ratio procedure is then carried out to determine the dropping vertex, and another mathematical formula then provides the 0^{th} vertex and the permutation corresponding to the new simplex. All these procedures make it very convenient to implement this algorithm for the computer.

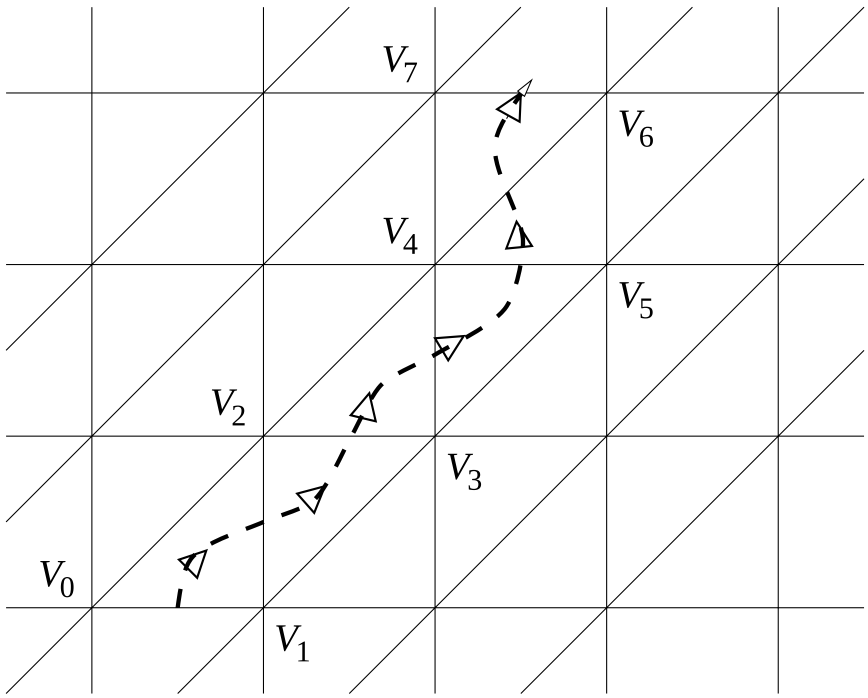

Figure 2.15 A path traced by the algorithm through the simplexes in the triangulation.

Special Triangulations of $\mathbf{R}^n \times [0,1]$

For computing a Kakutani fixed point of $\mathbf{F}(x)$ defined on $\mathbf{R}^n$, Merrill's algorithm uses a triangulation of $\mathbf{R}^n \times [0,1]$, which is a restriction of triangulation K_1 for $\mathbf{R}^{n+1}$ to this region, known as the special triangulation $\widetilde{K}_1$.

We will use the symbol $X = \begin{pmatrix} x \\ x_{n+1} \end{pmatrix}$ with $x \in \mathbf{R}^n$, to denote points in $\mathbf{R}^n \times [0,1]$. The set of vertices $\mathbf{J}$ in the special triangulation K_1 of $\mathbf{R}^n \times [0,1]$ are all the points $X = \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \in \mathbf{R}^{n+1}$ with x an integer vector in $\mathbf{R}^n$ and $x_{n+1} = 0$ or 1. The set of these vertices of the form $\begin{pmatrix} v \\ 1 \end{pmatrix}$ is denoted by $\mathbf{J}_1$, and the set of vertices of the form $\begin{pmatrix} v \\ 0 \end{pmatrix}$ is denoted by $\mathbf{J}_0$. $\mathbf{J} = \mathbf{J}_0 \cup \mathbf{J}_1$. The boundary of $\mathbf{R}^n \times [0,1]$ corresponding to $x_{n+1} = 1$ is known as the **top layer**, and the boundary corresponding to $x_{n+1} = 0$ is called the **bottom layer**. So $\mathbf{J}_1$, $\mathbf{J}_0$ are respectively the points with integer coordinates in the top and bottom layers. The $(n+1)$-dimensional simplexes in the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^n \times [0,1]$ are those of the form (V_0, P) where $P = (p_1, \ldots, p_{n+1})$ is a permutation of $\{1, \ldots, n+1\}$ and $V_0 \in \mathbf{J}_0$, and $(V_0, P) = \langle V_0, V_1, \ldots, V_{n+1} \rangle$ where

$$V_i = V_{i-1} + I_{.p_i}, \quad i = 1 \text{ to } n+1 \ . \tag{2.58}$$

In (2.58), I is the unit matrix of order $n+1$. It can be verified that the set of all simplexes of the form (V_0, P) as V_0 ranges over $\mathbf{J}_0$, and P ranges over all permutations of $\{1, 2, \ldots, n+1\}$ forms a triangulation of $\mathbf{R}^n \times [0, 1]$. V_0 is the 0^{th} **vertex** and for $i = 1$ to $n+1$, the vertex V_i determined as in (2.58) in the i^{th} **vertex** of the $(n+1)$-dimensional simplex denoted by (V_0, P). The following properties should be noted.

Property 1 : In the representation (V, P) for an $(n+1)$-dimensional simplex in the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^n \times [0, 1]$, the 0^{th} vertex V is always an integer point in the bottom layer, that is, belongs to $\mathbf{J}_0$.

Property 2 : In the representation (V, P) for an $(n+1)$-dimensional simplex in the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^n \times [0, 1]$, there exists a positive integer r such that for all $i \leqq r-1$, the i^{th} vertex of (V, P) belongs to the bottom layer; and for all $i \geqq r$, the i^{th} vertex of (V, P) belongs to the top layer. The i here is the index satisfying the property that if the permutation $P = (p_1, \ldots, p_{n+1})$, then $p_i = n+1$. This property follows from the fact that the vertices of the simplex (V, P) are obtained by letting $V_0 = V$, and using (2.58) recursively.

Two $(n+1)$-dimensional simplexes in the special triangulation $\widetilde{K}_1$ are said to be adjacent, if they have a common n-dimensional simplex as a face (i. e., if $(n+1)$ of their vertices are the same). Merrill's algorithm generates a sequence of $(n+1)$-dimensional simplexes $\sigma_1, \sigma_2, \sigma_3, \ldots$ of $\widetilde{K}_1$ in which every pair of consecutive simplexes are adjacent. So, given σ_j, σ_{j+1} is obtained by dropping a selected vertex V^- of σ_j and adding a new vertex V^+ in its place. The rules for obtaining σ_{j+1} given σ_j and V^- are called the **entering vertex choice rules** of the algorithm. These rules are very simple, they permit the generation of vertices as they are needed. We provide these rules here.

Let $\sigma_j = (V, P)$, where $P = (p_1, \ldots, p_{n+1})$ is a permutation of $\{1, \ldots, n+1\}$. The vertices of σ_j are $V_0 = V, V_1, \ldots, V_{n+1}$, as determined by (2.58). Let V^- be the dropping vertex. So V^- is V_i for some $i = 0$ to $n+1$. There are several cases possible which we consider separately.

Case 1 : $\{V_0, V_1, \ldots, V_{n+1}\} \setminus \{V^-\} \subset \mathbf{J}_1$. By property 2, this can only happen if $V^- = V_0$ and $V_1 \in \mathbf{J}_1$, that is, $p_1 = n+1$. The face of σ_j obtained by dropping the vertex V^-, is the n-dimensional simplex $\langle V_1, \ldots, V_{n+1} \rangle$ in the top layer, and hence is a boundary face. $\langle V_1, \ldots, V_{n+1} \rangle$ is the face of exactly one $(n+1)$ dimensional simplex in the triangulation $\widetilde{K}_1$, σ_j, and the algorithm terminates when this happens.

Case 2 : $\{V_0, V_1, \ldots, V_{n+1}\} \setminus \{V^-\} \subset \mathbf{J}_0$. By property 2, this implies that $V^- = V_{n+1}$ and $V_n \in \mathbf{J}_0$, that is $p_{n+1} = n+1$. We will show that this case cannot occur in the algorithm. So whenever $V^- = V_{n+1}$, we will have $p_{n+1} \neq n+1$ in the algorithm.

Case 3 : $\{V_0, V_1, \ldots, V_{n+1}\} \setminus \{V^-\}$ contains vertices on both the top and bottom layers. So the convex hull of $\{V_0, V_1, \ldots, V_{n+1}\} \setminus \{V^-\}$ is an n-dimensional simplex in the triangulation $\widetilde{K}_1$ which is an interior face, and hence is a face of exactly two $(n+1)$ dimensional simplexes in the triangulation, one is the present σ_j. The other σ_{j+1} is $(\widehat{V}, \widehat{P})$ as given below (V^+ given below is the new vertex in σ_{j+1} not in σ_j, it is the entering vertex that replaces the dropping vertex V^-).

V^-	V^+ (entering vertex)	$\widehat{V}$	$\widehat{P}$
$V^- = V_0$ $p_1 \neq n+1$ (see Case 1)†	$V_{n+1} + I_{\cdot p_1}$	$V_0 + I_{\cdot p_1}$	$(p_2, \ldots, p_{n+1}, p_1)$
$V^- = V_i$ $0 < i < n+1$	$V_{i-1} + I_{\cdot p_{i+1}}$	V_0	$(p_1, \ldots, p_{i-1}, p_{i+1},$ $p_i, p_{i+2}, \ldots, p_{n+1})$
$V^- = V_{n+1}$ $p_{n+1} \neq n+1$ (see Case 2)*	$V_0 - I_{\cdot p_{n+1}}$	$V_0 - I_{\cdot p_{n+1}}$	$(p_{n+1}, p_1, \ldots, p_n)$

It can be verified that if $(\widehat{V}, \widehat{P})$ is defined as above, then $\widehat{V} \in \mathbf{J}_0$ (since $V \in \mathbf{J}_0$ where V is the 0^{th} vertex of σ_j) and so $(\widehat{V}, \widehat{P})$ is an $(n+1)$-dimensional simplex in the special triangulation, and that $(\widehat{V}, \widehat{P})$ and (V, P) share $(n+1)$ common vertices, so they are adjacent $(n+1)$ dimensional simplexes in this triangulation. See Figure 2.16 for an illlustration of the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^1 \times [0,1]$.

The restriction of the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^n \times [0,1]$ to either the top layer (given by $x_{n+1} = 1$) or the bottom layer (given by $x_{n+1} = 0$) in the same as the triangulation K_1 of $\mathbf{R}^n$. The mesh of the special triangulation $\widetilde{K}_1$ of $\mathbf{R}^n \times [0,1]$ is defined to be the mesh of the triangulation of $\mathbf{R}^n$ on either the top and bottom layer, and hence it is $\sqrt{n}$.

We can get special triangulation of $\mathbf{R}^n \times [0,1]$ of smaller mesh by scaling the variables in $\mathbf{R}^n$ appropriately. Also, the origin in the $\mathbf{R}^n$ part can be translated to any specified point in $\mathbf{R}^n$. Let $x^0 \in \mathbf{R}^n$ be a specified point and δ a positive number. Let $\mathbf{J}(x^0, \delta) = \left\{ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} : x = (x_j) \in \mathbf{R}^n, x_j - x_j^0 \text{ is an integer multiple of } \delta \text{ for each } j = 1 \text{ to } n,\ x_{n+1} = 0 \text{ or } 1 \right\}$. Then the points is $\mathbf{J}(x^0, \delta)$ are the vertices of the special triangulation of $\mathbf{R}^n \times [0,1]$ denoted by $\delta\widetilde{K}_1(x^0)$. $\mathbf{J}_0(x^0, \delta) = \left\{ X = \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} : X \in \mathbf{J}(x^0, \delta), x_{n+1} = 0 \right\}$, $\mathbf{J}_1(x^0, \delta) = \left\{ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} : x \in \mathbf{J}(x^0, \delta),\ x_{n+1} = 1 \right\}$. For any $V \in \mathbf{J}_0(x^0, \delta)$, and $P = (p_1, \ldots, p_{n+1})$ a permutation of $\{1, \ldots, n+1\}$ define

$$\begin{aligned} V_0 &= V \\ V_i &= V_{i-1} + \delta I_{\cdot p_i}, \quad i = 1 \text{ to } n+1 \end{aligned} \tag{2.59}$$

and let $(V, P) = \langle V_0, V_1, \ldots, V_{n+1} \rangle$. The set of all $(n+1)$ dimensional simplexes (V, P) given by (2.59) with $V \in \mathbf{J}_0(x^0, \delta)$ and P ranging over all the permutation of

† In this case, if $p_1 = n+1$, as discussed in Case 1 above, the algorithm terminates. So the algorithm continues only if $p_1 \neq n+1$ when this case occurs.

* In this case, we cannot have $p_{n+1} = n+1$, as discussed in Case 2 above. So, whenever this case occurs in the algorithm, we will have $p_{n+1} \neq n+1$.

$\{1,\ldots,n+1\}$ are the $(n+1)$-dimensional simplexes in the special triangulation $\delta\widetilde{K}_1(x^0)$ of $\mathbf{R}^n \times [0,1]$. Its mesh in $\delta\sqrt{n}$. In this triangulation, the vertex $\begin{pmatrix} x^0 \\ 0 \end{pmatrix}$ plays the same role as the origin $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ in the triangulation $\widetilde{K}_1$.

The Piecewise Linear Approximation and a Linear Approximate fixed Point of $F(x)$

Consider the special triangulation $\widetilde{K}_1$ of $\mathbf{R}\times[0,1]$ defined above, and let $\mathbf{J}_0$, $\mathbf{J}_1$ be the vertices in this triangulation on the bottom and top layers respectively. On the top layer, we define a a piecewise linear map $f(X)$ known as a piecewise linear approximation of $\mathbf{F}(x)$ relative to the present triangulation. For each $V = \begin{pmatrix} v \\ 1 \end{pmatrix} \in \mathbf{J}_1$ define $f(V) = \begin{pmatrix} f(v) \\ 1 \end{pmatrix}$, where $f(v) \in \mathbf{F}(v)$. The point $f(v)$ can be selected from the set $\mathbf{F}(v)$ arbitrarily, in fact it can be determined using the subroutine for finding a point from the set $\mathbf{F}(v)$, which was pointed out as a required input for this algorithm. Any nonvertex point $X = \begin{pmatrix} x \\ 1 \end{pmatrix}$ on the top layer must lie in an n-dimensional simplex in the triangulation on this layer. Suppose the vertices of this simplex are $V_i = \begin{pmatrix} v_i \\ 1 \end{pmatrix}$, $i = 1$ to $n+1$. Then x can be expressed as a convex combinations of $v_1,\ldots,v_{n+1}$ in a unique manner. Suppose this expression is $\alpha_1 v_1 + \ldots + \alpha_{n+1}v_{n+1}$ where $\alpha_1 + \ldots + \alpha_{n+1} = 1$, $\alpha_1,\ldots,\alpha_{n+1} \geqq 0$. Then define $f(x) = \alpha_1 f(v_1) + \ldots + \alpha_{n+1} f(v_{n+1})$. $f(x)$ is the piecewise linear approximation of $\mathbf{F}(x)$ defined on the top layer relative to the present triangulation. For $X = \begin{pmatrix} x \\ 1 \end{pmatrix}$ define $f(X) = \begin{pmatrix} f(x) \\ 1 \end{pmatrix}$. In each n-dimensional simplex in the top layer in this triangulation $f(x)$ is linear. So $f(x)$ is a well defined piecewise linear continuous function defined on the top layer. Remember that the definition of $f(x)$ depends on the choice of $f(v)$ from $\mathbf{F}(v)$ for $V = \begin{pmatrix} v \\ 1 \end{pmatrix} \in \mathbf{J}_1$.

The point $x \in \mathbf{R}^n$ is said to be a linear approximate fixed point of $\mathbf{F}(x)$ relative to the present piecewise linear approximation if

$$x = f(x)\ . \tag{2.60}$$

The n-dimensional simplex $\left\langle V_i = \begin{pmatrix} v_i \\ 1 \end{pmatrix} : i = 1 \text{ to } n+1 \right\rangle$ on the top layer contains a fixed point of the piecewise linear map $f(x)$ iff the system

λ_1	$\ldots$	λ_{n+1}	
1	$\ldots$	1	1
$f(v_1) - v_1$	$\ldots$	$f(v_{n+1}) - v_{n+1}$	0

$\lambda_i \geqq 0, \quad i = 1 \text{ to } n+1$ (2.61) has a feasible solution. Thus the problem of finding a fixed point of the piecewise linear approximation $f(x)$ boils down to the problem of finding an n-dimensional simplex on the top layer whose vertices are such that (2.61) is feasible.

For each vertex $V = \begin{pmatrix} v \\ 1 \end{pmatrix}$ in the top layer associate the column vector $\begin{pmatrix} 1 \\ f(v) - v \end{pmatrix} \in \mathbf{R}^{n+1}$, which we denote by $A_{.V}$ and call the label of the vertex V. The coefficient matrix in (2.61) whose columns are the labels of the vertices of the simplex is called the **label matrix** corresponding to the simplex. Because of the nature of the labels used on the vertices, this is called a **vector labelling method**.

An n-dimensional simplex on the top layer is said to be a **completely labelled simplex** if the system (2.61) corresponding to it has a nonnegative solution, that is, if it contains a fixed point of the current piecewise linear approximation.

Let $V_0 = \begin{pmatrix} v_0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $P = (1, \ldots, n+1)$ and let $(V_0, P) = \langle V_0, V_1, \ldots, V_{n+1} \rangle$, where $V_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix}$, $i = 1$ to n. Then $\langle V_0, V_1, \ldots, V_n \rangle$ is the n-dimensional face of (V_0, P) in the bottom layer. Let $W = \begin{pmatrix} w \\ o \end{pmatrix}$ be an arbitrary point in the interior of this n-dimensional simplex $\langle V_0, \ldots, V_n \rangle$, for example, $w = \frac{(v_0 + \ldots + v_n)}{(n+1)}$. For every vertex $V = \begin{pmatrix} v \\ 0 \end{pmatrix} \in \mathbf{J}_0$ in the bottom layer, define $f(V) = \begin{pmatrix} f(v) \\ 0 \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}$. For any nonvertex X in $\mathbf{R}^n \times [0,1]$, X must lie in some $(n+1)$-dimensional simplex in the present triangulation, say $\langle V_0^1, V_1^1, \ldots, V_{n+1}^1 \rangle$. So there exist unique numbers $\alpha_0, \ldots, \alpha_{n+1} \geqq 0$ such that $\alpha_0 + \alpha_1 + \ldots + \alpha_{n+1} = 1$, $X = \alpha_0 V_0^1 + \alpha_1 V_1^1 + \ldots + \alpha_{n+1} V_{n+1}^1$. Then define $f(X) = \alpha_0 f(V_0^1) + \ldots + \alpha_{n+1} f(V_{n+1}^1)$. The map $f(X)$ is thus a continuous piecewise linear map defined on $\mathbf{R}^n \times [0,1]$. In each $(n+1)$ dimensional simplex in the present triangulation, $f(X)$ is linear. Also, under this map, every point in the bottom layer maps into the point W. Define the label of any vertex $V = \begin{pmatrix} v \\ 0 \end{pmatrix} \in J_0$ to be the column vector $A_{.V} = \begin{pmatrix} 1 \\ w - v \end{pmatrix} \in \mathbf{R}^{n+1}$.

Let $\langle V_0, V_1, \ldots, V_n \rangle$ be the n-dimensional simplex in the bottom layer, from the interior of which we selected the point W. Since W is in the interior of this simplex, B_1, the $(n+1) \times (n+1)$ label matrix corresponding to this simplex is nonsingular. Let $b = (1, 0, 0, \ldots, 0)^T \in \mathbf{R}^{n+1}$. Then the system corresponding to (2.61) for this simplex is

λ	
B_1	b

$\lambda \geqq 0$ (2.62)

This system has the unique positive solution $\lambda = \bar{b} = B_1^{-1} b > 0$, since W is in the interior of this simplex. Incidentally, this $\langle V_0, V_1, \ldots, V_n \rangle$ is the only n-dimensional simplex in the bottom layer whose label matrix leads to a nonnegative solution to the system like (2.61). The reason for it is that since W is in the interior of $\langle V_0, V_1, \ldots,$

$V_n\rangle$, W is not contained in any other simplex in the triangulation in the bottom layer. Also, since $\bar{b} > 0$, the $n \times (n+1)$ matrix $(\bar{b} \,\vdots\, B_1^{-1})$ has all rows lexicopositive. The inverse tableau corresponding to the initial system (2.62) is

basic vector	basis inverse	
λ	B_1^{-1}	$\bar{b}$

(2.63)

The initial simplex $\langle V_0, V_1, \ldots, V_n\rangle$ in the bottom layer is an n-dimensional face of the unique $(n+1)$-dimensional simplex $\langle V_0, V_1, \ldots, V_n, V_{n+1}\rangle$ in the present triangulation $\widetilde{K}_1$. Introduce a new variable, say λ_{n+1}, in (2.62) with its column vector equal to the label of this new vertex V_{n+1}, and bring this variable into the present basic vector. The pivot column for this pivot operation is $B_1^{-1}A_{.V_{n+1}}$. If this pivot column is nonpositive, it would imply that the set of feasible solutions of this augmented system (2.62) with this new variable is unbounded, which is impossible since the first constraint in the system says that the sum of all the variables is 1, and all the variables are nonnegative. So, the pivot column contains at least one positive entry, and it is possible to bring the new variable into the present basic vector. The dropping variable is determined by the usual lexico minimum ratio test of the primal simplex algorithm, this always determines the dropping variable uniquely and unambiguously and maintains the system lexico feasible. If the label of V_i is the dropping column, the next basis is the label matrix of the n-dimensional simplex $\langle V_0, \ldots, V_{i-1}, V_{i+1}, \ldots, V_{n+1}\rangle$. The inverse tableau corresponding to this new basis is obtained by entering the pivot column by the side of the present inverse tableau in (2.63) and performing a pivot step in it, with the row in which the dropping variable λ_i is basic, as the pivot row.

By the properties of the triangulation, the new n-dimensional simplex $\langle V_0, \ldots, V_{i-1}, V_{i+1}, \ldots, V_{n+1}\rangle$ is the face of exactly one or two $(n+1)$ dimensional simplexes in the triangulation. One is the simplex $\langle V_0, \ldots V_{n+1}\rangle$. If there is another, it must be a simplex of the form $\langle Y, V_0, \ldots, V_{i-1}, V_{i+1}, \ldots, V_{n+1}\rangle$. Then bring the column $A_{.Y}$ into the basis next. Continuing in this manner, we generate a unique path of the form S_1^n, S_1^{n+1}, S_2^n, S_2^{n+1}, ... Here S_k^n, S_k^{n+1} represent the k^{th} n-dimensional simplex and $(n+1)$-dimensional simplex respectively in this path. Termination can only occur if at some stage the basis corresponds to an n-dimensional simplex S_r^n all of whose vertices are on the top layer. Each n-dimensional simplex in this path is the face of at most two $(n+1)$-dimensional simplexes, we arrive at this face through one of these $(n+1)$-dimensional simplexes, and leave it through the other. The initial n-dimensional simplex in the bottom layer is a boundary face, and hence is the face of a unique $(n+1)$-dimensional simplex in the triangulation. So the path continues in a unique manner and it cannot return to the initial n-dimensional simplex again. Also, since the initial n-dimensional simplex is the only n-dimensional simplex in the bottom layer for which the system corresponding to (2.61) is feasible, the path will never pass through any other n-dimensional simplex in the bottom layer after the first step. Any n-dimensional simplex obtained on the path whose vertices belong to both the bottom and top layers is an interior face, so it is incident to two $(n+1)$-dimensional simplexes, we arrive at this n-face through one of these $(n+1)$-dimensional simplexes and leave it through the other, and the algorithm continues. The reader can verify that the properties of the path generated are very similar to the almost complementary basic vector path traced by the complementary pivot algorithm for the LCP. Thus we see that the path continues uniquely and unambiguously and it can only terminate when the columns of the current basis are the labels of vertices all of whom belong to the top layer. When it terminates, from the final BFS we get a fixed point of the current piecewise linear approximation.

Example 2.24

Consider $n = 1$. We consider a single-valued map from $\mathbf{R}^1$ to $\mathbf{R}^1$, $\mathbf{F}(x) = \{x^{\mathbf{2}} - 5x + 9\}$, $x \in \mathbf{R}^1$. The special triangulation of $\mathbf{R}^1 \times [0,1]$ is given in Figure 2.16.

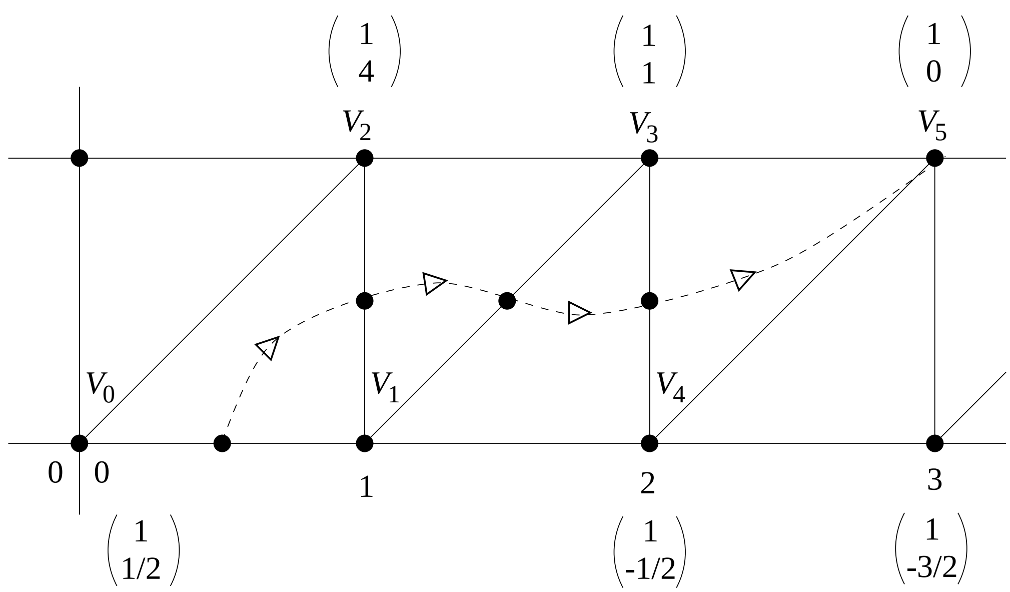

Figure 2.16 The column vector by the side of a vertex is its vector label.

The vertices for the triangulation are all the points with integer coordinates in $\mathbf{R}^1 \times [0,1]$. For each $V = \begin{pmatrix} v \\ 1 \end{pmatrix}$ on the top layer with v integer, we define $f(v) = v^2 - 5v + 9$. We take the initial 1-dimensional simplex on the bottom layer to be $\langle V_0, V_1 \rangle$ and the point W to be the interior point $(w, 0)^T = \left(\frac{1}{2}, 0\right)^T$ in it. For each $V = \begin{pmatrix} v \\ 0 \end{pmatrix}$ in the bottom layer, define $f(V) = W = \left(\frac{1}{2}, 0\right)^T$. The label of the vetex $V = \begin{pmatrix} v \\ x_{n+1} \end{pmatrix}$ is $\begin{pmatrix} 1 \\ f(v) - v \end{pmatrix}$ if $x_{n+1} = 1$, or $\begin{pmatrix} 1 \\ w - v \end{pmatrix}$ if $x_{n+1} = 0$. The labels of some of the vertices are entered in Figure 2.16. The initial system corresponding to (2.62) here is

λ_0	λ_1	
1	1	1
$\frac{1}{2}$	$-\frac{1}{2}$	0

$\lambda_0, \lambda_1 \geqq 0$

The feasible solution of this system and the basis inverse are given below.

Basic variable	Basis Inverse		$\bar{b}$	Pivot Column $\bar{A}_{\cdot V_2}$	Ratios
λ_0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\boxed{\frac{9}{2}}$	$\frac{1}{2}/\frac{9}{2}$ *Min.*
λ_1	$\frac{1}{2}$	-1	$\frac{1}{2}$	$-\frac{7}{2}$	

The initial simplex $\langle V_0, V_1 \rangle$ is the face of the unique 2-dimensional simplex $\langle V_0, V_1, V_2 \rangle$ in the triangulation. So we associate the label of V_2 with a variable λ_2 and bring it into the basic vector. The pivot column is

$$\begin{pmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{9}{2} \\ -\frac{7}{2} \end{pmatrix} = \bar{A}_{\cdot V_2}$$

and this is entered on the inverse tableau. The dropping variable is λ_0 and the pivot element is inside a box. Pivoting leads to the next inverse tableau. For ease in understanding, the vertices are numbered as V_i, $i = 0, 1, \dots$ in Figure 2.16 and we will denote the variable in the system associated with the label of the vertex V_i by λ_i.

Basic variable	Basis Inverse		$\bar{b}$	Pivot Column $\bar{A}_{\cdot V_3}$	Ratios
λ_2	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{1}{9}$	$\boxed{\frac{3}{9}}$	$\frac{1}{3}$ *Min.*
λ_1	$\frac{8}{9}$	$-\frac{2}{9}$	$\frac{8}{9}$	$\frac{6}{9}$	$\frac{8}{6}$

The current 1-simplex $\langle V_2, V_1 \rangle$ is the face of $\langle V_0, V_1, V_2 \rangle$ and $\langle V_3, V_1, V_2 \rangle$. We came to the present basic vector through $\langle V_0, V_1, V_2 \rangle$, so we have to leave $\langle V_2, V_1 \rangle$ through the 2-simplex $\langle V_3, V_1, V_2 \rangle$. Hence the updated column of the label of V_3, $\bar{A}_{\cdot V_3}$, is the entering column. It is already entered on the inverse tableau. The dropping variable is λ_2. Continuing, we get the following

Basic variable	Basis Inverse		$\bar{b}$	pivot Column $\bar{A}_{\cdot V_4}$	Ratios
λ_3	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{2}{3}$	
λ_1	$\frac{2}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$	$\boxed{\frac{5}{3}}$	$\frac{2}{5}$ *Min.*
				$\bar{A}_{\cdot V_5}$	
λ_3	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{3}{5}$	1
λ_4	$\frac{2}{5}$	$-\frac{2}{5}$	$\frac{2}{5}$	$\boxed{\frac{2}{5}}$	1
λ_3	0	1	0		
λ_5	1	-1	1		

In the basic vector λ_3, λ_4, there is a tie for the dropping variable by the usual primal simplex minimum ratio test, and hence the lexico minimum ratio test was used in determining the dropping variable. The algorithm terminates with the basic vector (λ_3, λ_5) since the corresponding vertices V_3, V_5 are both in the top layer. The fixed point of the piecewise linear approximation is $0 \times v_3 + 1 \times v_5 = 0 \times 2 + 1 \times 3 = 3$, from the terminal BFS. It can be verified that $x = 3$ is indeed a fixed point of $\mathbf{F}(x)$, since $\mathbf{F}(3) = \{3\}$.

Sufficient Conditions for Finite Termination with a Linear Approximate Fixed Point

Once the triangulation of $\mathbf{R}^n \times [0, 1]$ and the piecewise linear approximation are given, the path generated by this algorithm either terminates with an n-dimensional simplex on the top layer (leading to a fixed point of the present piecewise linear approximation) after a finite number of pivot steps, or continues indefinitely. Sufficient conditions to guarantee that the path terminates after a finite number of steps are discussed in [2.58], where the following theorem is proved.

Theorem 2.15 *Given $\hat{x} \in \mathbf{R}^n$ and $\alpha > 0$ let $\mathbf{B}(\hat{x}, \alpha) = \{ x : x \in \mathbf{R}^n$ satisfying $\|x - \hat{x}\| \leqq \alpha \}$. Suppose there are fixed positive numbers ν and γ and a point $\bar{x} \in \mathbf{R}^n$ satisfying: for each $x \in \mathbf{B}(\bar{x}, \nu)$, $y \in \mathbf{B}(x, \gamma) \backslash \mathbf{B}(\bar{x}, \nu)$ and $u \in \mathbf{F}(x)$, $(u - x)^T (y - \bar{x}) < 0$. Let x^0 by an arbitrary point in $\mathbf{R}^n$. If the above algorithm is executed using the starting point $x \in \{x^0\} \cup \mathbf{B}(\bar{x}, \nu + \gamma)$ and a special triangulation $\widetilde{K}_1$ with its mesh $\leqq \gamma$, then, the algorithm terminates in a finite number of steps with a linear approximate fixed point of $\mathbf{F}(x)$. Also, every linear approximate fixed point lies in $\mathbf{B}(\bar{x}, \nu + \gamma)$.*

We refer the reader to O. H. Merril's Ph. D. thesis [2.58] for a proof of this theorem. But it is very hard to verify whether these conditions hold in practical applications. In practical applications we apply the algorithm and let the path continue until some prescribed upper bound on computer time is used up. If termination does not occur by then, one usually stops with the conclusion that the method has failed on that problem.

One strange feature of the sufficient conditions to guarantee finite termination of the above algorithm is the following. Let $f(x) = (f_1(x), \ldots, f_n(x))^T$ be a continuously differentiable function from $\mathbf{R}^n$ into $\mathbf{R}^n$, and suppose we are applying the algorithm discussed above, on the fixed point formulation for the problem of solving the system of equations "$f(x) = 0$". Solving the system "$f(x) = 0$" is equivalent to finding the Kakutani fixed point of either $\mathbf{F}_1(x) = \{f(x) + x\}$ or $\mathbf{F}_2(x) = \{-f(x) + x\}$. Mathematically, the problem of finding a fixed point of $\mathbf{F}_1(x)$ or $\mathbf{F}_2(x)$ are equivalent. However, if $\mathbf{F}_1(x)$ satisfies the sufficiency condition for finite termination, $\mathbf{F}_2(x)$ will not. Thus, if the algorithm is applied to find the fixed points of $\mathbf{F}_1(x)$, and $\mathbf{F}_2(x)$; the behavior of the algorithm on the two problems could be very different. On one of them the algorithm may have finite termination, and on the other it may never terminate. This point should be carefully noted in using this algorithm in practical applications.

Algorithm to generate an Approximate Fixed Point of $\mathbf{F}(x)$

Select a sequence of positive numbers $\delta_0 = 1, \delta_1, \delta_2, \ldots$ converging to zero. Let $x^0 = 0$. Set $t = 0$ and go to Step 1.

Step 1 : Define the piecewise linear approximation for $\mathbf{F}(x)$ relative to the special triangulation $\delta_t \widetilde{K}_1(x^t)$ choosing the point W from the interior of the translate of the n-dimensonal face of the initial simplex $\langle 0, I_{.1}, \ldots, I_{.n} \rangle$ on the bottom layer in this triangulation. Find a fixed point of this piecewise linear approximation using this special triangulation by the algorithm discussed above. Suppose the fixed point obtained is x^{t+1}. x^{t+1} is a linear approximate fixed point of $\mathbf{F}(x)$ relative to this special triangulation $\delta_t \widetilde{K}_1(x^t)$. If $x^{t+1} \in \mathbf{F}(x^{t+1})$, terminate, x^{t+1} is a fixed point of $\mathbf{F}(x)$. Otherwise go to Step 2.

Step 2 : Replace t by $t + 1$ and do Step 1.

So this method generates the sequence $\{x^1, x^2, x^3, \ldots\}$ of linear approximate fixed points for $\mathbf{F}(x)$. If at any stage $x^t \in \mathbf{F}(x^t)$, it is a fixed point of $\mathbf{F}(x)$ and we terminate. Otherwise, any limit point of the sequence $\{x^t : t = 1, 2, \ldots\}$ can be shown to be a fixed point of $\mathbf{F}(x)$. In practice, if finite termination does not occur, we continue until δ_t becomes sufficiently small and take the final x^t as an approximate fixed point of $\mathbf{F}(x)$.

To find Fixed Points of USC Maps Defined on a Compact Convex Subset $\mathbf{\Gamma} \subset \mathbf{R}^n$

Without any loss of generality we can assume that $\mathbf{\Gamma}$ has a nonempty interior (if the interior of $\mathbf{\Gamma}$ in $\mathbf{R}^n$ is $\emptyset$, the problem is not altered by replacing $\mathbf{R}^n$ by the affine hull of $\mathbf{\Gamma}$, in which $\mathbf{\Gamma}$ has a nonempty interior). Let $\mathbf{F}(x)$ be the given USC map. So $\mathbf{F}(x)$

is defined for all $x \in \mathbf{\Gamma}$, and for all such x, $\mathbf{F}(x)$ is a compact convex subset of $\mathbf{\Gamma}$. Since this map is only defined on $\mathbf{\Gamma}$, and not on the whole of $\mathbf{R}^n$, the algorithm discussed above does not apply to this problem directly. However, as pointed out by B. C. Eaves [2.44], we can extend the definition of $\mathbf{F}(x)$ to the whole of $\mathbf{R}^n$ as below. Let c be any point from the interior of $\mathbf{\Gamma}$.

$$\mathbf{F}^1(x) = \begin{cases} \{c\}, & \text{if } x \notin \mathbf{\Gamma} \\ \text{convex hull of } \{c, \mathbf{F}(x)\}, & \text{if } x \in \text{boundary of } \mathbf{\Gamma} \\ \mathbf{F}(x), & \text{if } x \in \text{interior of } \mathbf{\Gamma}. \end{cases}$$

It can be verified that $\mathbf{F}^1(x)$ is now a USC map defined on $\mathbf{R}^n$, and that every fixed point of $\mathbf{F}^1(x)$ is in $\mathbf{\Gamma}$ and is also a fixed point of $\mathbf{F}(x)$ and vice versa. Since $\mathbf{F}^1(x)$ is defined over all of $\mathbf{R}^n$, the method discussed above can be applied to find a fixed point of it.

Homotopy Interpretation

In the algorithm discussed above for computing a fixed point of the piecewise linear approximation, there are two layers, the bottom layer and the top layer. We have the same triangulation of $\mathbf{R}^n$ in both the bottom and top layers. The labels for the vertices on the bottom layer are artificial labels corresponding to a very simple map for which we know the fixed point. The labels for the vertices on the top layer are natural labels corresponding to the piecewise linear map whose fixed point we want to find. The algorithm starts at the known fixed point of the artificial map of the bottom layer and walks its way through the triangulation until it reaches a fixed point of the piecewise linear map on the top layer. This makes it possible to interpret the above algorithm as a homotopy algorithm. Other homotopy algorithms for computing fixed points with continuous refinement of the grid size have been developed by B. C. Eaves [2.44] and B. C. Eaves and R. Saigal [2.47] and several others [2.40 to 2.80].

Comments 2.2 H. Scarf [2.68] first pointed out that the basic properties of the path followed by the complementary pivot algorithm in the LCP can be used to compute approximate Brouwer's fixed points using partitions of the space into sets called primitive sets, and T. Hansen and H. Scarf [2.69] extended this into a method for approximating Kakutani fixed points. The earliest algorithms for computing approximate fixed points using triangulations are those by B. C. Eaves [2.44], H. W. Kuhn [2.54]. These early algorithms suffered from computational inefficiency because they start from outside the region of interest. The first method to circumvent this difficulty is due to O. H. Merrill [2.57, 2.58] discussed above. The applications of fixed point methods in nonlinear programming discussed in Sections 2.7.3, 2.7.4, 2.7.5, 2.7.6 and 2.7.7 are due to O. H. Merrill [2.58]. Besides the triangulation K_1 discussed above, Merrill's algorithm can be implemented using other triangulations, see M. J. Todds book [2.72] and the papers [2.40 to 2.80].

2.8 COMPUTATIONAL COMPLEXITY OF THE COMPLEMENTARY PIVOT ALGORITHM

The **computational complexity** of an algorithm measures the growth of the computational effort involved in executing the algorithm as a function of the size of the problem. In the complementary pivot algorithm, we will assess the computational effort by the number of pivot steps carried out before the algorithm terminates. There are three commonly used measures for studying the computational complexity of an algorithm. These are discussed below.

Worst Case Computational Complexity

This measure is a tight mathematical upper bound on the number of pivot steps required before termination, as a function of the size of the problem. In studying the worst case computational complexity we will assume that the data is integer, or more generally, rational, that is, each m_{ij}, q_i in the matrices q, M is a ratio of two integers. In this case by multiplying all the data by a suitable positive integer, we can transform the problem into an LCP in which all the data is integer. Hence without any loss of generality we assume that all the data is integer, and define the size of the problem to be the total number of bits of storage needed to store all the data in the problem in binary form. See Chapter 6 where a mathematical definition of this size is given. The worst case computational complexity of an algorithm provides a guaranteed upper limit on the computational effort needed to solve any instance of the problem by the algorithm, as a function of the size of the instance. The algorithm is said to be **polynomially bounded** if this worst case computational complexity is bounded above by a polynomial of fixed degree in the size of the problem, that is, if there exist constants α, r independent of the size, such that the computational effort needed is always $\leqq \alpha s^{\mathbf{r}}$ when the algorithm is applied on problems of size s. Even though the worst case computational complexity is measured in terms of the number of pivot steps, each pivot step needs $\mathcal{O}(n^{\mathbf{2}})$ basic arithmetical operations (addition, multiplication, division, comparison) on data each of which has at most s digits, where s is the size and n the order of the instance; so if the algorithm is polynomially bounded in terms of the number of pivot steps, it is polynomially bounded in terms of the basic arithmetical operations. In Chapter 6 we conclusively establish that the complementary pivot algorithm is not a polynomially bounded algorithm in this worst case sense. Using our examples discussed in Chapter 6, in [2.74] M. J. Todd constructed examples of square nonsingular systems of linear equations "$Ax - b = 0$", with integer data, for solving which the computational effort required by Merrill's algorithm of Section 2.7.8, grows exponentially with the size of the problem.

An algorithm may have a worst case computational complexity which is an exponentially growing function of the size of the problem, just because it performs very poorly on problem instances with a very rare pathological structure. Such an algorithm might be extremely efficient on instances of the problem not having the rare pathological structure, which may never show up in practical applications. For this reason, the worst case measure is usually very poor in judging the computational efficiency of an algorithm, or its practical utility.

The Probabilistic Average Computational Complexity

Here we assume that the data in the problem is randomly generated according to some assumed probability distribution. The average computational complexity of the algorithm under this model is then defined to be the statistical expectation of the number of steps needed by the algorithm before termination, on problem instances with this data. Since the expectation is a multiple integral, this average analysis requires techniques for bounding the values of multiple integrals. If the probability distributions are continuous distributions, the data generated will in general be real numbers (not rational), and so in this case we define the size of the LCP to be its order n. We assume that each pivot step in the algorithm is carried out on the real data using exact arithmetic, but assess the computational complexity by the average number of pivot steps carried out by the algorithm before termination.

M. J. Todd performed the average analysis in [2.36] under the folowing assumptions on the distribution of the data (q, M).

i) With probability one, every square submatrix of M whose sets of row indices and column indices differ in at most one element, is nonsingular.

ii) q is nondegenerate in the LCP (q, M).

iii) The distributions of (q, M) are sign-invariant; that is, (q, M) and (Sq, SMS) have identical distributions for all sign matrices S (i. e., diagonal matrices with diagonal entries of $+1$ or -1).

Under these assumptions he showed that the expected number of pivot steps taken by the lexicographic Lemke algorithm (see Section 2.3.4) before termination when applied on the LCP (q, M) is at most $\frac{n(n+1)}{4}$.

M. J. Todd [2.36] also analysed the average computational complexity of the lexicographic Lemke algorithm applied on the LCP corresponding to the LP

$$\begin{aligned} \text{minimize} \quad & cx \\ \text{subject to} \quad & Ax \geqq b \\ & x \geqq 0 \end{aligned}$$

under the following assumptions. A is a matrix of order $m \times N$. The probability distribution generating the data (A, b, c) and hence the data (q, M) in the corresponding LCP satisfies the following assumptions :

i) with probability one, the LP and its dual are nondegenerate (every solution of $Ax - u = b$ has at least m nonzero variables, and every solution of $yA + v = c$ has at least N nonzero variables), and every square submatrix of A is nonsingular.

ii) the distributions of (A, b, c) and of (S_1AS_2, S_1b, S_2c) are identical for all sign matrices S_1, S_2 of appropriate dimension). This is the sign invariance requirement.

Under these assumptions he showed that the expected number of pivot steps taken by the lexicographic Lemke algorithm when applied on the LCP corresponding to this LP is at most, minimum $\{\frac{m^2+5m+11}{2}, \frac{2N^2+5N+5}{2}\}$. See also [2.31] for similar results under slightly different probabilistic models.

In a recent paper, [8.20] R. Saigal showed that the expected number of pivot steps taken by the lexicographic Lemke algorithm when applied on the LCP corresponding to the above LP is actually bounded above by m and asymptotically approaches $\frac{m}{2} - 1$, where m is the number of rows in A.

Unfortunately, these nice quadratic or linear bound expected complexity results seem very dependent on the exact manner in which the algorithm is implemented, and on the problabilistic model of the data. For example, it has not been possible so far to obtain comparable results for the complementary pivot algorithm of Section 2.2 which uses the column vector e of all 1's as the original column vector of the artificial variable z_0.

Empirical Average Computation Complexity

This measure of computational complexity is used more in the spirit of simulation. Here, a computational experiment is usually performed by applying the algorithm on a large number of problem instances of various sizes, and summary statistics are then prepared on how the algorithm performed on them. The data is usually generated according to some distribution (typically we may assume that each data element is a uniformly distributed random variable from an interval such as -100 to $+100$, etc.). In the LCP, we may also want to test how the complementary pivot algorithm performs under varying degrees of sparsity of q and M. For this, a certain percentage of randomly chosen entries in q and M can be fixed as zero, and the remaining obtained randomly as described above. It may also be possible to generate M so that it has special properties. As an example, if we want to experiment on LCPs associated with PSD symmetric matrices, we can generate a random square matrix A as above and take M to be A^TA. Such computational experiments can be very useful in practice. The experiments conducted on the complementary pivot algorithm, suggest that the empirical average number of pivot steps before termination grows linearly with n, the order of the LCP.

We know that Merrill's simplicial method for computing the fixed point of a piecewise linear map discussed in Section 2.7.8 may not terminate on some problems. Computational experiments indicate that on problems on which it did terminate, the average number of simplices that the algorithm walked through before termination, is $\mathcal{O}(n^2)$, as a function of the dimension of the problem. See [2.62 to 2.67].

2.9 THE GENERAL QUADRATIC PROGRAMMING PROBLEM

From the results in Section 2.3 we know that the complementary pivot method processes convex quadratic programs with a finite computational effort. Here we discuss the general, possibly nonconvex, quadratic programming problem. This is a problem in which a general quadratic objective function is to be minimized subject to linear constraints.

The Reduction Process

If there is an equality constraint on the variables, using it, obtain an expression for one of the variables as an affine function of the others, and eliminate this variable and this constraint from the optimization portion of the problem. A step like this is called a **reduction step**, it reduces the number of variables in the optimization problem by one, and the number of constraints by one. In the resulting problem, if there is another equality constraint, do a reduction step using it, and continue in the same manner. When this work is completed, only inequality constraints remain, and the system of constraints assumes the form $FX \geqq f$, which includes any sign restrictions and lower or upper bound constraints on the variables. We assume that this system is feasible. An inequality constraint in this system is said to be a **binding inequality constraint** if it holds as an equation at all feasible solutions. A binding inequality constraint can therefore be treated as an equality constraint without affecting the set of feasible solutions. Binding inequality constraints can be identified using a linear programming formulation. Introduce the vector of slack variables v and transform the system of constraints into $FX - v = f$, $v \geqq 0$. The i^{th} constraint in the system, $F_{i.}X \geqq f_i$, is a binding constraint iff the maximum value of v_i subject to $FX - v = f$, $v \geqq 0$, is zero. Using this procedure identify all the binding constraints, change each of them into an equality constraint in the system. Carry out further reduction steps using these equality constraints. At the end, the optimization portion of the problem reduces to one of the following form

$$\begin{array}{ll} \text{Minimize} & \theta(x) = cx + \frac{1}{2}x^T Dx \\ \text{Subject to} & Ax \geqq b \end{array} \tag{2.64}$$

satisfying the property that $Ax > b$ is feasible. Let A be of order $m \times n$. Without any loss of generality we assume that D is symmetric (because $x^T Dx = x^T \frac{D+D^T}{2} x$ and $\frac{D+D^T}{2}$ is a symmetric matrix). Let $\mathbf{K} = \{x : Ax \geqq b\}$. By our assumptions here $\mathbf{K} \neq \emptyset$ and in fact $\mathbf{K}$ has a nonempty interior. Every interior point of $\mathbf{K}$ satisfies $Ax > b$ and vice versa. We also assume that $\mathbf{K}$ is bounded. The solution of the problem when $\mathbf{K}$ is unbounded can be accomplished by imposing additional constraints $-\alpha \leqq x_j \leqq \alpha$ for each j, where α is a large positive valued parameter. The parameter α is not given any specific value, but treated as being larger than any number with which it may be compared. The set of feasible solution of the augmented problem is bounded, and so the augmented problem can be solved by the method discussed below. If the optimum solution of the augmented problem is independent of α when α is positive and large, it is the optimum solution of the original problem (2.64). On the other hand if the optimum solution of the augmented problem depends on α however large α may be, and the optimum objective value diverges to $-\infty$ as α tends to $+\infty$, the objective function is unbounded below in the original problem. In the sequel we assume that $\mathbf{K}$ is bounded. Under these assumptions, (2.64) will have an optimum solution. If D is not PSD, we have the following theorem.

Theorem 2.16 *If D is not PSD, the optimum solution of (2.64) cannot be an interior point of* $\mathbf{K}$.

Proof. Proof is by contradiction. Suppose $\bar{x}$, an interior point of $\mathbf{K}$, is an optimum solution of (2.64). Since $\bar{x}$ is an interior point of $\mathbf{K}$, we have $A\bar{x} > b$, and a necessary condition for it to be optimum for (2.64) (or even for it to be a local minimum for (2.64)) is that the gradient vector of $\theta(x)$ at $\bar{x}$, which is $\nabla\theta(\bar{x}) = c + \bar{x}^T D = 0$. Since D is not PSD, there exists a vector $y \neq 0$ satisfying $y^T Dy < 0$. Using $c + \bar{x}^T D = 0$, it can be verified that $\theta(\bar{x} + \lambda y) = \theta(\bar{x}) + \frac{\lambda^2}{2} y^T Dy$. Since $\bar{x}$ satisfies $A\bar{x} > b$, we can find $\lambda > 0$ and sufficiently small so that $\bar{x} + \lambda y$ is feasible to (2.64), and $\theta(\bar{x} + \lambda y) = \theta(\bar{x}) + \frac{\lambda^2}{2} y^T Dy < \theta(\bar{x})$, contradiction to the hypothesis that $\bar{x}$ is optimal to (2.64). So if D is not PSD, every optimum solution must be a boundary point of $\mathbf{K}$, that is, it must satisfy at least one of the constraints in (2.64) as an equation.

□

The Method

Express the problem in the form (2.64), using the reduction steps discussed above as needed, so that the system $Ax > b$ is feasible. Suppose A is of order $m \times n$. Then we will refer to the problem (2.64) as being of order (m, n), where n is the number of decision variables in the problem, and m the number of inequality constraints on these variables.

Check whether D is PSD. This can be carried out by the efficient algorithm discussed in Section 1.3.1 with a computational effort of $\mathcal{O}(n^3)$. If D is PSD, (2.64) is a convex quadratic program, the optimum solution for it can be computed using the complementary pivot algorithm discussed in earlier sections, with a finite amount of computational effort. If D is not PSD, generate m candidate problems as discussed below. This operation is called the **branching operation**.

For $i = 1$ to m, the i^{th} **candidate problem** is the following :

$$\begin{array}{ll} \text{Minimize} & cx + \frac{1}{2}x^T Dx \\ \text{Subject to} & A_{p.}x \geqq b_p, \quad p = 1 \text{ to } m,\ p \neq i \\ & A_{i.}x = b_i \ . \end{array} \tag{2.65}$$

If D is not PSD, by Theorem 2.16, every optimum solution for (2.64) must be an optimum solution of at least one of the m candidate problems.

Each of the candidate problems is now processed independently. The set of feasible solutions of each candidate problem is a subset (a face) of $\mathbf{K}$, the set of feasible solutions of the original problem (2.64). Using the equality constraint, a reduction step can be carried out in the candidate problem (2.65). In the resulting reduced problem identify any binding inequality constraints by a linear programming formulation discussed earlier. Treat binding constraints as equality constraints and carry out further reduction steps. The final reduced problem is one of the same form (2.64), but of order $\leqq (m-1, n-1)$. Test whether it is a convex quadratic programming problem (this could happen even if the original problem (2.64) is not a convex quadratic program) and if it is so, find the optimum solution for it using the complementary pivot algorithm and store its solution in a **solution list**. If it is not a convex quadratic program carry out the branching operation on it and generate additional candidate problems from it, and process each of them independently in the same way.

The total number of candidate problems to be processed is $\leqq 2^{\mathbf{m}}$. When there are no more candidate problems left to be procesed, find out the best solution (i. e., the one with the smallest objective value) among those in the solution list at that stage. That solution is an optimum solution of the original problem.

This provides a finite method for solving the general quadratic programming problem. It may be of practical use only if m and n are small numbers, or if the candidate problems turn out to be convex quadratic programs fairly early in the branching process. On some problems the method may require a lot of computation. For example, if D in the original problem (2.64) is negative definite, every candidate problem with one or more inequality constraints will be nonconvex, and so the method will only terminate when all the extreme points of $\mathbf{K}$ are enumerated in the solution list. In such cases, this method, eventhough finite, is impractical, and one has to resort to heuristics or some approximate solution methods.

2.9.1 Testing Copositiveness

Let M be a given square matrix of order n. Suppose it is required to check whether M is copositive. From the definition, it is clear that M is copositive iff the optimum objective value in the following quadratic program is zero.

$$\begin{array}{ll} \text{Minimize} & x^T M x \\ \text{Subject to} & x \geqq 0 \\ & e^T x \leqq 1 \ . \end{array} \tag{2.66}$$

where e is the column vector of all 1's in $\mathbf{R}^n$. We can check whether M is PSD with a computational effort of $\mathcal{O}(n^{\mathbf{3}})$ by the efficient pivotal methods discussed in Section 1.3.1. If M is PSD, it is also copositive. If M is not PSD, to check whether it is copositive, we can solve the quadratic program (2.66) by the method discussed above. If the optimum objective value in it is zero, M is copositive, not otherwise. This provides a finite method for testing copositiveness. However, this method is not practilly useful when n is large. Other methods for testing copositiveness are discussed in [3.29, 3.59]. See also Section 2.9.3.

Exercise

2.4 Using the results from Section 8.7, prove that the general quadratic programming problem (2.64) with integer data is an $\mathcal{NP}$-hard problem.

Comments 2.2. Theorem 2.16 is from R. K. Mueller [2.23]. The method for the general quadratic programming problem discussed here is from [2.24] of K. G. Murty.

2.9.2 Computing a KKT point for a General Quadratic Programming Problem

Consider the QP (quadratic program)

$$\begin{array}{lrl} \text{minimize} & Q(x) = & cx + \frac{1}{2}x^T Dx \\ \text{subject to} & Ax \geqq & b \\ & x \geqq & 0 \end{array} \tag{2.67}$$

where D is a symmetric matrix of order n, and A, b, c are given matrices of orders $m \times n$, $m \times 1$, and $1 \times n$ respectively. We let $\mathbf{K}$ denote the set of feasible solutions of this problem. If D is PSD, this is a convex quadratic program, and if $\mathbf{K} \neq \emptyset$, the application of the complementary pivot algorithm discussed in Sections 2.2, 2.3 on the LCP corresponding to this QP will either terminate with the global minimum for this problem, or provide a feasible half-line along which $Q(x)$ diverges to $-\infty$.

Here, we do not assume that D is PSD, so (2.67) is the general QP. In this case there can be local minima which are not global minima (see Section 10.2 for definitions of a global minimum, local minimum), the problem may have KKT points which are not even local minima (for example, for (2.66) verify that $x = 0$ is a KKT point, and that this is not even a local minimum for that problem if D is not copositive). The method discussed at the beginning of Section 2.9 is a total enumeration method (enumerating over all the faces of $\mathbf{K}$) applicable when $\mathbf{K}$ is bounded. In this section we do not make any boundedness assumption on $\mathbf{K}$. We prove that if $Q(x)$ is unbounded below on $\mathbf{K}$, there exists a half-line in $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$. We also prove that if $Q(x)$ is bounded below on $\mathbf{K}$, then (2.67) has a finite global minimum point. This result was first proved by M. Frank and P. Wolfe [10.14] but our proofs are based on the results of B. C. Eaves [2.9]. We also show that the complementary pivot method applied on an LCP associated with (2.67) will terminate with one of three possible ways

(i) establish that $\mathbf{K} = \emptyset$, or
(ii) find a feasible half-line in $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$, or
(iii) find a KKT point for (2.67).

From the results in Chapter 1, we know that $\bar{x} \in \mathbf{K}$ is a KKT point for (2.67) iff there exist vectors $\bar{y}, \bar{v} \in \mathbf{R}^m$ and $\bar{u} \in \mathbf{R}^n$ which together satisfy

$$\begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} - \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} c^T \\ -b \end{pmatrix}$$

$$\begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} \geqq 0, \quad \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \geqq 0, \quad \begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix}^T \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = 0 \tag{2.68}$$

which is an LCP. We will call $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ a KKT solution corresponding to the KKT point $\bar{x}$. For the sake of simplicity, we denote

$$\begin{pmatrix} u \\ v \end{pmatrix} \text{ by } w, \text{ and } \begin{pmatrix} x \\ y \end{pmatrix} \text{ by } z$$

$$\begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \text{ by } M, \text{ and } \begin{pmatrix} c^T \\ -b \end{pmatrix} \text{ by } q$$

$$n + m \text{ by } N .$$

So, if $(\bar{w}, \bar{z})$ is complementary solution of the LCP (2.68), then $(\bar{z}_1, \ldots, \bar{z}_n) = \bar{x}$ is a KKT point for (2.67).

A KKT point $\bar{x}$ for (2.67) is said to be a **reduced** KKT **point** for (2.67) if the set of column vectors $\left\{ M_{.j} = \begin{pmatrix} D_{.j} \\ A_{.j} \end{pmatrix} : j \text{ such that } \bar{x}_j > 0 \right\}$ is linearly independent.

Lemma 2.12 *Let $\bar{x}$ be a KKT point for (2.67). From $\bar{x}$, we can derive either a reduced KKT point $\tilde{x}$ such that $Q(\tilde{x}) \leqq Q(\bar{x})$, or a feasible half-line in $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$.*

Proof. Let $\big(\bar{w} = (\bar{u}, \bar{v}), \bar{z} = (\bar{x}, \bar{y})\big)$ be a KKT solution associated with $\bar{x}$. Let $\mathbf{J}_1 = \{j : \bar{w}_j = 0\}$, $\mathbf{J}_2 = \{j : \bar{z}_j = 0\}$. By complementarity $\mathbf{J}_1 \cup \mathbf{J}_2 = \{1, \ldots, N\}$. From the fact that $(\bar{w}, \bar{z})$ is a KKT solution (i.e., it satisfies (2.68)) it can be verified that $Q(\bar{x}) = \frac{1}{2}(c\bar{x} + \bar{y}^T b) = \frac{1}{2}(c, b^T)\bar{z}$. Consider the following LP

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}(c, b^T)\bar{z} \\ \text{subject to} & w - Mz = q \\ & w_j = 0 \text{ for } j \in \mathbf{J}_1 \\ & z_j = 0 \text{ for } j \in \mathbf{J}_2 \\ & w_j \geqq 0 \text{ for } j \notin \mathbf{J}_1 \\ & z_j \geqq 0 \text{ for } j \notin \mathbf{J}_2 \end{array} \tag{2.69}$$

If (w, z) is any feasible solution to this LP, from the constraints in (2.69) it is clear that the corresponding $x = (z_1, \dots, z_n)$ is in $\mathbf{K}$, and that $w^T z = 0$ (complementarity), by this complementarity we have $Q(x) = \frac{1}{2}(c, b^T)z$.

There are only two possibilities for the LP (2.69). Either the objective function is unbounded below in it, in which case there exists a feasible half-line, say $\{(w^1, z^1) + \lambda(w^h, z^h) : \lambda \geqq 0\}$ along which the objective value diverges to $-\infty$ (this implies that the corresponding half-line $\{x^1 + \lambda x^h : \lambda \geqq 0\}$ is in $\mathbf{K}$ and $Q(x)$ diverges to $-\infty$ on it), or that it has an optimum solution, in which case it has an optimum BFS. If $(\tilde{w}, \tilde{z})$ is an optimum BFS of (2.69), the corresponding $\tilde{x}$ is a reduced KKT point for (2.67) and $Q(\tilde{x}) = \frac{1}{2}(c, b^T)\tilde{z} \leqq \frac{1}{2}(c, b^T)\bar{z} = Q(\bar{x})$.

□

Lemma 2.13 *If the QP has a global optimum solution, it has a global optimum solution $\bar{x}$ satisfying the property that the set of vectors $\left\{ \begin{pmatrix} D_{\cdot j} \\ A_{\cdot j} \end{pmatrix} : j \text{ such that } \bar{x}_j > 0 \right\}$ is linearly independent.*

Proof. Follows from Lemma 2.12.

□

Lemma 2.14 *For given D, A; there exists a finite set of matrices $L_1, \dots, L_l$, each of order $n \times N$, such that for any c, b if x is a reduced KKT point of (2.67), then $x = L_t \begin{pmatrix} c^T \\ -b \end{pmatrix}$ for some t.*

Proof. Let x be a reduced KKT point for (2.67). Let $\big(w = (u, v), z = (x, y)\big)$ be the corresponding KKT solution. Then (w, z) is a BFS of an LP of the form (2.69). Since it is a BFS, there exists a basic vector and associated basis B for (2.69) such that this (w, z) is defined by

$$\begin{aligned} &\text{nonbasic variables} = 0 \\ &\text{basic vector} = B^{-1}q \end{aligned}$$

The matrix L_t can have its j^{th} row to be 0 if x_j is a nonbasic variable, or the r^{th} row of B^{-1}if x_j is the r^{th} basic variable in this basic vector. By complementarity, there are only $2^{\mathbf{N}}$ systems of the form (2.69), and each system has a finite number of basic vectors, so the collection of matrices of the form L_t constructed as above is finite and depends only on D, A. So, for any q, any reduced KKT point must be of the form $L_t q$ for some L_t in this finite collection.

□

Theorem 2.17 *Assume that $\mathbf{K} \neq \emptyset$. Either the QP (2.67) has a global minimum, or there exists a feasible half-line in $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$.*

Proof. Let $\{\alpha_p : p = 1, 2, \dots\}$ be an increasing sequence of positive numbers diverging to $+\infty$, such that $\mathbf{K} \cap \{x : ex \leqq \alpha_1\} \neq \emptyset$. Consider the QP

$$\begin{array}{ll} \text{minimize} & cx + \frac{1}{2}x^T Dx \\ \text{subject to} & Ax \geqq b \\ & x \geqq 0 \\ & ex \leqq \alpha_p \end{array} \tag{2.70}$$

For every p in this sequence, (2.70) has a non-empty bounded solution set, and hence has a global nimimum. By Lemma 2.12, it has a global minimum which is a reduced KKT point for (2.70). Applying Lemma 2.14 to the QP (2.70), we know that there exists a finite collection of matrices $\{L_1, \ldots, L_l\}$ independent of the data in the right hand side constants vector in (2.70), such that every reduced KKT point for (2.70) is of the form

$$L_t \begin{pmatrix} c^T \\ -b \\ \alpha_p \end{pmatrix} = L_t \begin{pmatrix} c^T \\ -b \\ 0 \end{pmatrix} + \alpha_p L_t \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{2.71}$$

for some t. So, for each $p = 1, 2, \ldots$, there exists a t between 1 to l such that the global minimum of (2.70) for that p is of the form given in (2.71). Since there are only a finite number l, of these t's, there must exist a t, say t_1, which gives the global minimum for an infinite number of p's. Let the subsequence corresponding to these p's in increasing order be $P = \{p_1, p_2, \ldots\}$. Let

$$\tilde{x} = L_{t_1} \begin{pmatrix} c^T \\ -b \\ 0 \end{pmatrix}, \quad \bar{y} = L_{t_1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Then the global minimum for (2.70) is $x(p_r) = \tilde{x} + \alpha_{p_r}\bar{y}$ when $p = p_r$, for $r = 1, 2, \ldots$. So, the optimum objective value in this problem is $Q\big(x(p_r)\big) = Q\big(\tilde{x} + \alpha_{p_r}\bar{y}\big)$, and this is of the form $a_0 + a_1\alpha_{p_r} + a_2\alpha_{p_r}^{\mathbf{2}}$. The quantity α_{p_r} is monotonic increasing with r, so the set of feasible solutions of (2.70) for $p = p_r$ becomes larger as r increases, so $Q\big(x(p_r)\big)$ is monotonic decreasing with r. These facts imply that either $a_2 < 0$ or $a_2 = 0$ and $a_1 \leqq 0$. If $a_2 < 0$ or $a_2 = 0$ and $a_1 < 0$, $Q\big(x(p_r)\big)$ diverges to $-\infty$ as r tends to $+\infty$, in this case $\{\tilde{x} + \lambda\bar{y} : \lambda \geqq \alpha_{p_1}\}$ is a half-line in $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$. On the other hand, if $a_2 = a_1 = 0$, $Q(x)$ is bounded below by a_0 on $\mathbf{K}$, and in this case $\tilde{x} + \alpha_{p_r}\bar{y}$ is a global minimum for (2.67) for any r.

□

The Algorithm

To compute a KKT point for (2.67), apply the complementary pivot method on the LCP (γ, F) of order $n + m + 1$, where

$$\gamma = \begin{pmatrix} c^T \\ -b \\ q_{n+m+1} \end{pmatrix}, \qquad F = \begin{pmatrix} D & -A^T & e \\ A & 0 & 0 \\ -e^T & 0 & 0 \end{pmatrix}$$

where q_{n+m+1} is treated as a large positive valued parameter without giving any specific value for it (i.e., q_{n+m+1} is treated as being larger than any numer with which it is compared), with the original column vector of the artificial variable z_0 taken to be $(-1, -1, \dots, -1, 0) \in \mathbf{R}^{n+m+1}$. By Lemma 2.9, it can be verified that the matrix M defined above is an L_2-matrix. If the complementary pivot method terminates in a secondary ray, by Theorem 2.5, we conclude that

$$\begin{aligned} -Ax &\leqq -b \\ ex &\leqq q_{n+m+1} \\ x &\geqq 0 \end{aligned}$$

is infeasible for q_{n+m+1} arbitrarily large, that is

$$\begin{aligned} Ax &\geqq b \\ x &\geqq 0 \end{aligned}$$

is infeasible. So (2.67) is infeasible, if ray termination occurs in the complementary pivot algorithm when applied on the LCP (γ, F).

Suppose the complementary pivot method terminates with a complementary solution $\big(\bar{w} = (\bar{w}_j),\ \bar{z} = (\bar{z}_j)\big)$ where $\bar{w}, \bar{z} \in \mathbf{R}^{n+m+1}$. If $\bar{w}_{n+m+1} > 0$, $\bar{z}_{n+m+1} = 0$, it can be verified that $\big((\bar{w}_1, \dots, \bar{w}_{n+m}), (\bar{z}_1, \dots, \bar{z}_{n+m})\big)$ is a complementary solution for the LCP $\left(\begin{Bmatrix} c^T \\ -b \end{Bmatrix}, \begin{Bmatrix} D & -A^T \\ A & 0 \end{Bmatrix} \right)$, that is, it is a KKT solution for (2.67) and $\bar{x} = (\bar{z}_1, \dots, \bar{z}_n)^T$ is a KKT point for (2.67).

On the other hand, if $\bar{w}_{n+m+1} = 0$ and $\bar{z}_{n+m+1} > 0$ in the terminal complementary BFS, the basic variables are affine functions of the large positive parameter q_{n+m+1}. Let $\bar{x} = (\bar{z}_1, \dots, \bar{z}_n)^T$, $\bar{y} = (\bar{z}_{n+1}, \dots, \bar{z}_{n+m})$. It can be verified that $Q(\bar{x}) = \frac{1}{2}(c\bar{x} + b^T y) - \frac{1}{2} q_{n+m+1} \bar{z}_{n+m+1}$ and as q_{n+m+1} tends to $+\infty$, this diverges to $-\infty$. Hence in this case, $Q(x)$ is unbounded below on $\mathbf{K}$, and a feasible half-line along which $Q(x)$ diverges to $-\infty$ can be obtained by letting the parameter q_{n+m+1} tend to $+\infty$ in the solution $\bar{x}$.

When D is not PSD, it is possible for (2.67) to have some KKT points, even when $Q(x)$ is unbounded below on $\mathbf{K}$. Thus in this case the fact that this algorithm has terminated with a KKT point of (2.67) is no guarantee that $Q(x)$ is bounded below on $\mathbf{K}$.

2.9.3 Computing a Global Minimum, or Even a Local Minimum in Nonconvex Programming Problems May be Hard

Consider the smooth nonlinear program (NLP)

$$\begin{array}{ll} \text{minimize} & \theta(x) \\ \text{subject to} & g_i(x) \geqq 0, \quad i = 1 \text{ to } m \end{array} \tag{2.72}$$

where each of the functions is a real valued function defined on $\mathbf{R}^n$ with high degrees of differentiability. (2.72) is convex NLP if $\theta(x)$ is convex and $g_i(x)$ are concave for all i, nonconvex NLP, otherwise.

A global minimum for (2.72) is a feasible solution $\bar{x}$ for it satisfying $\theta(x) \geqq \theta(\bar{x})$ for all feasible solutions x of the problem. See Section 10.2. For a convex NLP, under some constraint qualifications (see Appendix 4) necessary and sufficient optimality conditions are known. Given a feasible solution satisfying the constraint qualification, using these optimality conditions, it is possible to check efficiently whether that point is a (global) optimum slution of the problem or not.

For a smooth nonconvex nonlinear program, the problem of computing a global minimum, or checking whether a given feasible solution is a global minimum, are hard problems in general. To establish these facts mathematically, consider the subset sum problem, a hard problem in discrete optimization, which is known to be $\mathcal{NP}$-complete (see reference [8.12] for a complete discussion of $\mathcal{NP}$-completeness): given postive integers $d_0, d_1, \dots, d_n$; is there a solution to

$$\sum_{j=1}^{n} d_j y_j = d_0$$

$$y_j = 0 \text{ or } 1 \text{ for all } j$$

Now consider the quadratic programming problem (QP)

$$\begin{array}{ll} \text{minimize} & \left(\sum_{j=1}^{n} d_j y_j - d_0 \right)^{\mathbf{2}} + \sum_{j=1}^{n} y_j (1 - y_j) \\ \text{subject to} & 0 \leqq y_j \leqq 1, \quad j = 1 \text{ to } n \ . \end{array}$$

Because of the second term in the objective function, QP is a nonconvex quadratic programming problem. Clearly, the subset-sum problem given above has a feasible solution iff the global minimum objective value in QP is zero. Since the problem of checking whether the subset-sum problem is $\mathcal{NP}$-complete, computing the global minimum for QP, a very special and simple case of a smooth nonconvex NLP, is an $\mathcal{NP}$-hard problem (see reference [8.12] for a complete discussion of $\mathcal{NP}$-hardness). This shows that in general, the problem of computing a global minimum in a smooth nonconvex NLP may be a hard problem. See also Section 10.3 where some of the outstanding difficult problems in mathematics have been formulated as those of finding global minima in smooth nonconvex NLPs (for example, there we show that the well known **Fermat's last Theorem** in number theory, unresolved since 1637 AD, can be posed as the problem of checking whether the global minimum objective value in a smooth nonconvex NLP, (10.1), is zero or greater than zero).

Since the problem of computing a global minimum in a nonconvex NLP is a hard problem, we will now study the question whether it is at least possible to compute a local minimum for such a problem by an efficient algorithm.

For nonconvex NLPs, under constraint qualifications, some necessary conditions for a local minimum are known (see Section 10.2 for the definitions of a local minimum, and Appendix 4 for a discussion of necessary conditions for a local minimum) and there are some sufficient conditions for a point to be a local minimum. But there are no simple conditions known, which are both necessary and sufficient for a given point to be a local minimum. The **complexity** of checking whether a given feasible solution is a local minimum in a nonconvex NLP, is not usually addressed in the literature. Many textbooks in NLP, when they discuss algorithms, leave the reader with the impression that these algorithms converge to a global minimum in convex NLPs, and to a local minimum in nonconvex NLPs. The documentations distributed for many professional NLP software packages also create the same impression. This impression could be quite erroneous, in the general case. In this section we study this problem by examining the computational complexity of determining whether a given feasible solution is not a local minimum, and that of determining whether the objective function is not bounded below on the set of feasible solutions, in smooth continuous variable, nonconvex NLPs. For this purpose, we use the very special instance of an nonconvex quadratic programming problem studied in K. G. Murty and S. N. Kabadi [10.32] with integer data, which may be considered as the simplest nonconvex NLP. It turns out that the questions of determining whether a given feasible solution is not a local minimum in this problem, and to check whether the objective function is not bounded below in this problem, can both be studied using the discrete techniques of computational complexity theory, and in fact these questions are $\mathcal{NP}$-complete problems (see reference [8.12] for definition of $\mathcal{NP}$-completeness). This clearly shows that in general, it is a hard problem to check whether a given feasible solution in a nonconvex NLP is even a local minimum, or to check whether the objective function is bouned below. This indicates the following: when a nonlinear programming algorithm is applied on a nonconvex NLP, unless it is proved that it converges to a point satisfying some known sufficient condition for a local minimum, claims that it leads to a local minimum are hard to verify in the worst case. Also, in continuous variable smooth nonconvex minimization, even the down-to-earth goal of guaranteeing that a local minimum will be obtained by the algorithm (as opposed to the lofty goal of finding the global minimum) may be hard to attain.

We review the known optimality conditions for a given feasible solution $\bar{x}$ to (2.72) to be a local minimum. Let $\mathbf{J} = \{i : g_i(\bar{x}) = 0\}$. Optimality conditions are derived under the assumption that some constraint qualifications (CQ, see Appendix 4) are satisfied at $\bar{x}$, which we assume.

First Order Necessary Conditions for $\bar{x}$ to be a Local Minimum for (2.72)

There must exist a $\bar{\mu}_{\mathbf{J}} = (\bar{\mu}_i : i \in \mathbf{J})$ such that

$$\begin{aligned} &\nabla\theta(\bar{x}) - \sum_{i \in \mathbf{J}} \bar{\mu}_i \nabla g_i(\bar{x}) = 0 \\ &\bar{\mu}_i \geqq 0, \quad \text{for all } i \in \mathbf{J} \ . \end{aligned} \tag{2.73}$$

Given the feasible solution $\bar{x}$, it is possible to check whether these conditions hold, efficiently, using Phase I of the simplex method for linear programming.

Second Order Necessary Conditions for $\bar{x}$ to be a Local Minimum for (2.72)

These conditions include (2.73). Given $\bar{\mu}_{\mathbf{J}}$ satisfying (2.73) together with $\bar{x}$, let $L(x, \bar{\mu}_{\mathbf{J}}) = \theta(x) - \sum_{i \in \mathbf{J}} \bar{\mu}_i g_i(x)$. In addition to (2.73) these conditions require

$$y^T H y \geqq 0, \text{ for all } y \in \{y : \nabla g_i(\bar{x}) y = 0 \text{ for each } i \in \mathbf{J}\} \tag{2.74}$$

where H is the Hessian matrix of $L(x, \bar{\mu}_{\mathbf{J}})$ with respect to x at $x = \bar{x}$. Condition (2.74) requires the solution of a quadratic program involving only equality constraints, which can be solved efficiently. It is equivalent to checking the positive semidefiniteness of a matrix which can be carried out efficiently using Gaussian pivot steps (see Section 1.3.1).

Sufficient Conditions for $\bar{x}$ to be a Local Minimum for (2.72)

Given the feasible solution $\bar{x}$, and $\bar{\mu}_{\mathbf{J}}$ which together satisfy (2.73), the most general known sufficient optimality condition states that if

$$y^T H y > 0 \text{ for all } y \in \mathbf{T}_1 \tag{2.75}$$

where $\mathbf{T}_1 = \big\{y : y \neq 0 \text{ and } \nabla g_i(\bar{x}) y = 0 \text{ for each } i \in \{i : i \in \mathbf{J} \text{ and } \bar{\mu}_i > 0\}, \text{ and } \nabla g_i(\bar{x}) y \geqq 0 \text{ for each } i \in \{i : i \in \mathbf{J} \text{ and } \bar{\mu}_i = 0\}\big\}$, then $\bar{x}$ is a local minimum for (2.72). Unfortunately, when H is not positive semidefinite, the problem of checking whether (2.75) holds, leads to a nonconvex QP, which, as we will see later, may be hard to solve.

Aside from the question of the difficulty of checking whether (2.75) holds, we can verify that the gap between conditions (2.74) and (2.75) is very wide, particulary when the set $\{i : i \in \mathbf{J} \text{ and } \bar{\mu}_i = 0\} \neq \emptyset$. In this case, condition (2.74) may hold, and even if we are able to check (2.75), if it is not satisfied, we are unable to determine whether $\bar{x}$ is a local minimum for (2.72) with present theory.

Now we will use a simple indefinite QP, related to the problem of checking whether the sufficient optimality condition (2.75) holds, to study the following questions :

i) Given a smooth nonconvex NLP and a feasible solution for it, can we check whether it is a local minimum or not efficiently ?

ii) At least in the simple case when the constraints are linear, can we check efficiently whether the objective function is bounded below or not on the set of feasible solutions ?

Let D be an integer square symmetric matrix of order n. The problem of checking whether D is not PSD involves the question

$$\text{“is there an } x \in \mathbf{R}^n \text{ satisfying } x^T D x < 0 \text{ ?”} \tag{2.76}$$

This can be answered with an effort of at most n Gaussian pivot steps, by the techniques discussed in Section 1.3.1. This leads to an $\mathcal{O}(n^3)$ algorithm for this problem. At the termination of this algorithm, it is in fact possible to actually produce a vector x satisfying $x^T D x < 0$, if the answer to (2.76) is in the affirmative.

All PSD matrices are copositive, but a matrix which is not PSD may be copositive. Testing whether the given matrix D is not copositive involves the question

$$\text{“is there an } x \geqq 0 \text{ satisfying } x^T D x < 0 \text{ ?”} \tag{2.77}$$

If D is not PSD, no efficient algorithm for this question is known (the computational complexity of the enumerative method of Section 2.9.1 grows exponentially with n in the worst case). In fact we show later that this question is $\mathcal{NP}$-complete. To study this question, we are naturally lead to the NLP

$$\begin{array}{ll} \text{minimize} & Q(x) = x^T D x \\ \text{subject to} & x \geqq 0 \end{array} \tag{2.78}$$

We will show that this problem is an $\mathcal{NP}$-hard problem.

We assume that D is not PSD. So $Q(x)$ is nonconvex and (2.78) is a nonconvex NLP. It can be considered the **simplest nonconvex NLP**. We consider the following decision problems.

Problem 1: Is $x = 0$ not a local minimum for (2.78) ?

Problem 2: Is $Q(x)$ not bounded below on the set of feasible solutions of (2.78) ?

Clearly, the answer to problem 2 is in the affirmative iff the answer to problem 1 is. We will show that both these problems are $\mathcal{NP}$-complete. To study problem 1, we can replace (2.78) by the NLP

$$\begin{array}{ll} \text{minimize} & Q(x) = x^T D x \\ \text{subject to} & 0 \leqq x_j \leqq 1, \quad j = 1 \text{ to } n \end{array} \tag{2.79}$$

Lemma 2.15 *The decision problem “is there an $\bar{x}$ feasible to (2.79) which satisfies $Q(\bar{x}) < 0$”, is in the class $\mathcal{NP}$ (see [8.12] for the definition of the class $\mathcal{NP}$ of decision problems).*

Proof. Given an x feasible to (2.79), to check whether $Q(x) < 0$, can be done by computing $Q(x)$ which takes $\mathcal{O}(n^2)$ time. Also, if the answer to the problem is in the affirmative, an optimum solution $\bar{x}$ of (2.79) satisfies $Q(\bar{x}) < 0$. There is a linear complementarity problem (LCP) corresponding to (2.79) and an optimum solution for (2.79) must correspond to a BFS for this LCP. Since there are only a finite number of BFSs for an LCP, and they are all rational vectors, a nondeterministic algorithm can find one of them satisfying $Q(x) < 0$, if it exists, in polynomial time. Hence, this problem is in the class $\mathcal{NP}$.

□

Lemma 2.16 *The optimum objective value in (2.79) is either 0 or $\leqq -2^{-\mathbf{L}}$ where* L *is the size of D, (i.e., the total number of binary digits in all the data in D).*

Proof. Since the set of feasible solutions of (2.79) is a compact set and $Q(x)$ is continuous, (2.79) has an optimum solution. The necessary optimality conditions for (2.79) lead to the following LCP

$$\begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} D & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ e \end{pmatrix} \tag{2.80}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} \geqq 0\,, \qquad \begin{pmatrix} x \\ y \end{pmatrix} \geqq 0\,, \tag{2.81}$$

$$\begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} = 0 \tag{2.82}$$

It can be verified that whenever (u, v, x, y) satisfies (2.80), (2.81) and (2.82), $x^T D x = -e^T y$, a linear function, where e is the column vector of all 1's in $\mathbf{R}^n$. There exists an optimum solution of (2.79) which is a BFS of (2.80), (2.81). By the results under the ellipsoid algorithm (see, for example Chapter 8 in this book, or Chapter 15 in [2.26]), in every BFS of (2.80), (2.81), each y_j is either 0 or $\geqq 2^{-\mathbf{L}}$. If the optimum objective value in (2.79) is not zero, it must be < 0, and this together with the above facts implies that an optimum solution x or (2.79) corresponds to a BFS (u, v, x, y) of (2.80), (2.81) in which $-e^T y < 0$. All these facts clearly imply that the optimum objective value in (2.79) is either 0 or $\leqq -2^{-\mathbf{L}}$.

□

We now make a list of several decision problems, some of which we have already seen, and some new ones which we need for establishing our results.

Problem 3: Is there an $x \geqq 0$ satisfying $Q(x) < 0$?

Problem 4: For any positive integer a_0, is there an $x \in \mathbf{R}^n$ satisfying $e^T x = a_0$, $x \geqq 0$ and $Q(x) < 0$?

Now consider a subset sum problem with data d_0; $d_1, \ldots, d_n$, which are all positive integers. Let γ be a positive integer $> 4 \left(d_0 \left(\sum_{j=1}^{n} d_j \right) \right)^{\mathbf{2}} n^{\mathbf{3}}$. Let l be the size of this subset sum problem, that is, the total number of binary digits in all the data for the problem. Let ε be a positive rational number $< 2^{-\mathbf{nl^2}}$. The subset sum problem is :

Problem 5: Subset sum problem: Is there a $y = (y_j) \in \mathbf{R}^n$ satisfying $\sum_{j=1}^n d_j y_j = d_0$, $0 \leqq y_j \leqq 1$, $j = 1$ to n, and y integer vector ?

We now define several functions involving nonnegative variables $y = (y_1, \ldots, y_n)^T$ and $s = (s_1, \ldots, s_n)^T$, related to the subset sum problem.

$$
\begin{aligned}
f_1(y,s) &= \left(\sum_{j=1}^n d_j y_j - d_0\right)^{\mathbf{2}} + \gamma\left(\sum_{j=1}^n (y_j + s_j - 1)^{\mathbf{2}}\right) + \sum_{j=1}^n y_j s_j \\
&= \left(\sum_{j=1}^n d_j y_j\right)^{\mathbf{2}} + \sum_{j=1}^n y_j s_j + \gamma \sum_{j=1}^n (y_j + s_j)^{\mathbf{2}} \\
&\quad - 2d_0\left(\sum_{j=1}^n d_j y_j\right) + 2\gamma \sum_{j=1}^n (y_j + s_j) + n\gamma + d_0^{\mathbf{2}} \\
f_2(y,s) &= f_1(y,s) + 2d_0\left(\sum_{j=1}^n d_j y_j (1 - y_j)\right) \\
&= \left(\sum_{j=1}^n d_j y_j\right)^{\mathbf{2}} + \gamma \sum_{j=1}^n (y_j + s_j)^{\mathbf{2}} + \sum_{j=1}^n y_j s_j \\
&\quad - 2d_0\left(\sum_{j=1}^n d_j y_j^{\mathbf{2}}\right) + 2\gamma \sum_{j=1}^n (y_j + s_j) + n\gamma + d_0^{\mathbf{2}} \\
f_3(y,s) &= \left(\sum_{j=1}^n d_j y_j\right)^{\mathbf{2}} + \gamma \sum_{j=1}^n (y_j + s_j)^{\mathbf{2}} + \sum_{j=1}^n y_j s_j \\
&\quad - 2d_0\left(\sum_{j=1}^n d_j y_j^{\mathbf{2}}\right) + d_0^{\mathbf{2}} - n\gamma \\
f_4(y,s) &= \left(\sum_{j=1}^n d_j y_j\right)^{\mathbf{2}} + \gamma \sum_{j=1}^n (y_j + s_j)^{\mathbf{2}} + \sum_{j=1}^n y_j s_j \\
&\quad - 2d_0 \sum_{j=1}^n d_j y_j^{\mathbf{2}} + \left(\frac{d_0^{\mathbf{2}} - n\gamma}{n^{\mathbf{2}}}\right)\left(\sum_{j=1}^n (y_j + s_j)\right)^{\mathbf{2}} \\
f_5(y,s) &= f_4(y,s) - \left(\frac{\varepsilon}{n^{\mathbf{2}}}\right)\left(\sum_{j=1}^n (y_j + s_j)\right)^{\mathbf{2}}
\end{aligned}
$$

Let $\mathbf{P} = \{(y, s) : y \geqq 0, s \geqq 0, \sum_{j=1}^{n}(y_j + s_j) = n\}$. Consider the following additional decision problems

Problem 6: Is there a $(y, s) \in \mathbf{P}$ satisfying $f_1(y, s) \leqq 0$?

Problem 7: Is there a $(y, s) \in \mathbf{P}$ satisfying $f_2(y, s) \leqq 0$?

Problem 8: Is there a $(y, s) \in \mathbf{P}$ satisfying $f_4(y, s) \leqq 0$?

Problem 9: Is there a $(y, s) \in \mathbf{P}$ satisfying $f_5(y, s) < 0$?

Theorem 2.18 *Problem 4 is an $\mathcal{NP}$-hard problem (see [8.11] for the definitions of an $\mathcal{NP}$-hard problem).*

Proof. Since $f_1(y, s)$ is a sum of nonnegative terms whenever $(y, s) \in \mathbf{P}$, if $(\bar{y}, \bar{s}) \in \mathbf{P}$ satisfies $f_1(y, s) \leqq 0$, then we must have $f_1(\bar{y}, \bar{s}) = 0$, this clearly implies from the definition of $f_1(y, s)$, that the following conditions must hold.

$$\sum_{j=1}^{n} d_j \bar{y}_j = d_0, \quad \bar{y}_j \bar{s}_j = 0 \text{ and } \bar{y}_j + \bar{s}_j = 1, \text{ for all } j = 1 \text{ to } n \ .$$

These conditions clearly imply that $\bar{y}$ is a solution of the subset sum problem and that the answer to problem 5 is in the affirmative. Conversely if $\hat{y} = (\hat{y}_j)$ is a solution to the subset sum problem, define $\hat{s} = (\hat{s}_j)$ where $\hat{s}_j = 1 - \hat{y}_j$ for each $j = 1$ to n, and it can be verified that $f_1(\hat{y}, \hat{s}) = 0$. This verifies that problems 5 and 6 are equivalent.

Whenever $\bar{y}$ is a 0–1 vector, we have $\bar{y}_j = \bar{y}_j^2$ for all j, and this implies that $f_1(\bar{y}, s) = f_2(\bar{y}, s)$ for any s. So, from the above arguments, we see that if $(\bar{y}, \bar{s}) \in \mathbf{P}$ satisfies $f_1(\bar{y}, \bar{s}) \leqq 0$, then $f_1(\bar{y}, \bar{s}) = f_2(\bar{y}, \bar{s}) = 0$. If $0 \leqq y_j \leqq 1$, we have $2d_0 d_j y_j(1-y_j) \geqq 0$. If $(y, s) \in \mathbf{P}$, and $y_j > 1$, then $\frac{\gamma}{2}(y_j + s_j - 1)^2 + 2d_0 d_j y_j(1 - y_j) \geqq 0$, since γ is large (from the definition of γ). Using this and the definitions of $f_1(y, s)$, $f_2(y, s)$, it can be verified that for $(y, s) \in \mathbf{P}$, if $f_2(y, s) \leqq 0$ then $f_1(y, s) \leqq 0$ too. These facts imply that problems 6 and 7 are equivalent.

Clearly, problems 7 and 8 are equivalent.

From the definition of ε (since it is sufficiently small) and using Lemma 2.16, one can verify that problems 8 and 9 are equivalent.

Problem 9 is a special case of problem 4. Since problem 5 is $\mathcal{NP}$-complete, from the above chain of arguments we conclude that problem 4 is $\mathcal{NP}$-hard.

□

Theorem 2.19 *Problem 4 is $\mathcal{NP}$-complete.*

Proof. The answer to problem 4 is in the affirmative iff the answer to the decision problem in the statement of Lemma 2.15 is in the affirmative. So, from Lemma 2.15 we conlcude that problem 4 is in $\mathcal{NP}$. From Theorem 2.18, this shows that problem 4 is $\mathcal{NP}$-complete.

□

Theorem 2.20 *Problem 3 is $\mathcal{NP}$-complete.*

Proof. Problems 3 and 4 are clearly equivalent, this result follows from Theorem 2.19. ∎

Theorem 2.21 *Both problems 1 and 2 are $\mathcal{NP}$-complete.*

Proof. Problems 1 and 2 are both equivalent to problem 3, so this result follows from Theorem 2.20. ∎

Theorem 2.22 *Given an integer square matrix D, the decision problem "is D not copositive ?" is $\mathcal{NP}$-complete.*

Proof. The decision problem "is D not copositive ?" is equivalent to problem 1, hence this result follows from Theorem 2.21. ∎

Can We Check Local Minimality Efficiently In Unconstrained Minimization Problems ?

Let $\theta(x)$ be a real valued smooth function defined on $\mathbf{R}^n$. Consider the unconstrained problem

$$\text{minimize} \quad \theta(x) \, . \tag{2.83}$$

A necessary condition for a given point $\bar{x} \in \mathbf{R}^n$ to be a local minimum for (2.83) is (see Appendix 4)

$$\nabla\theta(\bar{x}) = 0, \quad H(\theta(\bar{x})) \text{ is PSD} \tag{2.84}$$

where $H(\theta(\bar{x}))$ is the Hessian matrix (the matrix of second order partial derivatives) of $\theta(x)$ at $\bar{x}$. A sufficient condition for $\bar{x}$ to be a local minimum for (2.83) is

$$\nabla\theta(\bar{x}) = 0, \quad H(\theta(\bar{x})) \text{ is positive definite.} \tag{2.85}$$

Both conditions (2.84) and (2.85) can be checked very efficiently. If (2.84) is satisfied, but (2.85) is violated, there are no simple conditions known to check whether or not $\bar{x}$ is a local minimum for (2.83). Here, we investigate the complexity of checking whether or not a given point $\bar{x}$ is a local minimum for (2.83), and that of checking whether $\theta(x)$ is bounded below or not over $\mathbf{R}^n$.

As before, let $D = (d_{ij})$ be an integer square symmetric matrix of order n. Consider the unconstrained problem,

$$\text{minimize } h(u) = (u_1^{\mathbf{2}}, \ldots, u_n^{\mathbf{2}}) D (u_1^{\mathbf{2}}, \ldots, u_n^{\mathbf{2}})^T \tag{2.86}$$

Clearly, (2.86) is an instance of the general unconstrained minimization problem (2.83). Consider the following decision problems.

Problem 10: Is $\bar{u} = 0$ not a local minimum for (2.86) ?

Problem 11: Is $h(u)$ not bounded below on $\mathbf{R}^n$?

We have, for $i, j = 1$ to n

$$\frac{\partial h(u)}{\partial u_j} = 4u_j\big((u_1^{\mathbf{2}}, \ldots, u_n^{\mathbf{2}})D_{.j}\big)$$
$$\frac{\partial^2 h(u)}{\partial u_i \partial u_j} = 8u_i u_j d_{ij}, \quad i \neq j$$
$$\frac{\partial^2 h(u)}{\partial u_j^2} = 4(u_1^{\mathbf{2}}, \ldots, u_n^{\mathbf{2}})D_{.j} + 8u_j^{\mathbf{2}} d_{jj}$$

where $D_{.j}$ is the j^{th} column vector of D. So, $\bar{u} = 0$ satisfies the necessary conditions for being a local minimum for (2.86), but not the sufficient condition given in (2.85).

Using the transformation $x_j = u_j^{\mathbf{2}}$, $j = 1$ to n, we see that (2.86) is equivalent to (2.78). So problem 1 and 10 are equivalent. Likewise, problems 2 and 11 are equivalent. By Theorem 2.21, we conclude that both problems 10 and 11 are $\mathcal{NP}$-hard. Thus, even in the unconstrained minimization problem, to check whether the objective function is not bounded below, and to check whether a given point is not a local minimum, may be hard problems in general. This also shows that the problem of checking whether a given smooth nonlinear function (even a polynomial) is or is not locally convex at a given point, may be a hard problem in general.

What Are Suitable Goals for Algorithms in Nonconvex NLP ?

Much of nonlinear programming literature stresses that the goal for algorithms in nonconvex NLPs should be to obtain a local minimum. Our results here show that in general, this may be hard to guarantee.

Many nonlinear programming algorithms are iterative in nature, that is, beginning with a initial point x^0, they obtain a sequence of points $\{x^r : r = 0, 1, \ldots\}$. For some of the algorithms, under certain conditions, it can be shown that the sequence converges to a KKT point for the original problem, (a KKT point is a feasible solution at which the first order necessary conditions for a local minimum, (2.73), hold). Unfortunately, there is no guarantee that a KKT point will be a local minimum, and our results point out that in general, checking whether or not it is a local minimum may be a hard problem.

Some algorithms have the property that the sequence of points obtained is actually a descent sequence, that is, either the objective function, or a measure of the infesibility of the current solution to the problem, or some merit function or criterion function which is a combination of both, strictly decreases along the sequence. Given x^r, these algorithms generate a $y^r \neq 0$ such that the direction $x^r + \lambda y^r$, $\lambda \geqq 0$, is a descent direction for the functions discussed above. The next point in the sequence x^{r+1} is usually taken to be the point which minimizes the objective or criterion function on the half-line $\{x^r + \lambda y^r : \lambda \geqq 0\}$, obtained by using a line minimization algorithm. On general nonconvex problems, these methods suffer from the same difficulties, they cannot theoretically guarantee that the point obtained at termination is even a local minimum. However, it seems reasonable to expect that a solution obtained through a descent process is more likely to be a local minimum, than a solution obtained purely based on necessary optimality conditions. Thus a suitable goal for algorithms for non-convex NLPs seems to be a descent sequence converging to a KKT point. Algorithms, such as the sequential quadratic programming methods discussed in Section 1.3.6, and those discussed in Chapter 10, reach this goal.

2.10 Exercises

2.5 Let $\theta(x)$ be a convex function defined on $\mathbf{R}^n$, which is known to be unbounded below on $\mathbf{R}^n$. Does there exist a half-line along which $\theta(x)$ diverges to $-\infty$? Either prove that it does, or construct a counterexample. Does the answer change if $\theta(x)$ is known to be a differentiable convex function ?

2.6 Consider the problem

$$\begin{array}{ll} \text{Minimize} & \theta(x) \\ \text{Subject to} & Ax \geqq b \end{array}$$

where A is a matrix of order $m \times n$, and $\theta(x)$ is a convex function. Suppose it is known that $\theta(x)$ is unbounded below in this problem. Does there exist a feasible half-line along which $\theta(x)$ diverges to $-\infty$? Either prove that it does, or construct a counterexample. Does the answer change if $\theta(x)$ is a differentiabl convex function ?

2.7 If the data in the LCP (q, M) satisfies

i) $M + M^T \geqq 0$, and

ii) $q - M^T z \geqq 0, \quad z \geqq 0$ is feasible,

prove that the complementary pivot algorithm will terminate with a solution when applied on the LCP (q, M).
(Philip C. Jones [2.16])

2.8 Let $\mathbf{G}_j$ be the set $\{(w, z) : w - Mz = q,\ w \geqq 0,\ z \geqq 0,\ w_i z_i = 0 \text{ for all } i \neq j\}$, and let $\mathbf{G} = \bigcup(\mathbf{G}_j : j = 1 \text{ to } n)$. If M is PSD or a P-matrix, prove that $\mathbf{G}$ is a connected subset of $\mathbf{R}^n$. If (q, M) is the LCP corresponding to the following quadratic program, show that $\mathbf{G}$ is not connected.

$$\begin{array}{ll} \text{minimize} & cx + \frac{1}{2} x^T D x \\ \text{subject to} & 0 \leqq x \leqq u \end{array}$$

where $D = \begin{pmatrix} -2 & -3 & -3 \\ -3 & -5 & -1 \\ -1 & -1 & 4 \end{pmatrix}$, $u = \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$, $c^T = \begin{pmatrix} 4 \\ 3 \\ 5 \end{pmatrix}$.

(W. P. Hallman and I. Kaneko [2.15])

2.9 Prove that the complementary pivot algorithm will process the LCP (q, M) if M is a Z-matrix.
(R. Saigal [2.32])

2.10 Let $\{A_{.1}, \ldots, A_{.n-1}\}$ be a linearly independent set of column vectors in $\mathbf{R}^n$. Let $\S = \{y^1, \ldots, y^r\}$ be another finite set of column vectors in $\mathbf{R}^n$, and let $b \in \mathbf{R}^n$ be another given column vector. It is required to choose $A_{.n} \in \S$ so that the minimum distance from b to $\text{Pos}\{A_{.1}, \ldots, A_{.n}\}$ is a small as possible. Develop an efficient algorithm for doing it.

2.11 Let

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

Show that the LCP $(\hat{q}, \widehat{M})$ has a solution. However, show that all the variants of the complementary pivot algorithm discussed in this Chapter are unable to find a solution to this LCP $(\hat{q}, \widehat{M})$.

2.12 Let (P) be a linear programming problem, and (Q) the corresponding linear complementary problem as obtained in Section 1.2. It has been suggested that the sequence of solutions generated when the LCP, (Q), is solved by the complementary pivot method, is the same as the sequence of solutions generated when the LP, (P), is solved by the self-dual parametric algorithm (see Section 8.13 of [2.26]). Discuss, and examine the similarities between the self-dual parametric algorithm applied to (P) and the complementary pivot method applied on (Q).

2.13 Let

$$M = \begin{pmatrix} 2 & 2 & 1 & 2 \\ 3 & 3 & 2 & 3 \\ -2 & 1 & 5 & -2 \\ 1 & -2 & 1 & 2 \end{pmatrix}, \qquad q = \begin{pmatrix} -4 \\ -6 \\ 4 \\ 4 \end{pmatrix}.$$

i) Prove that M is strictly copositive.
ii) Show that the LCP (q, M) has an infinite number of complementary feasible solutions.

2.14 Given a square matrix M of order n, let $\mathbf{K}(M)$ denote the union of all the complementary cones in $\mathcal{C}(M)$. Prove that $\mathbf{K}(M)$ is convex iff $\mathbf{K}(M) = \{q : q + Mz \geqq 0$, for some $z \geqq 0\}$.
(B. C. Eaves [2.8])

2.15 Let $a_1, \ldots, a_n$, b be positive integers satisfying $b > \max\{a_1, \ldots, a_n\}$. Let

$$q(n+2) = (a_1, \ldots, a_n, -b, b)^T$$

$$M(n+2) = \begin{pmatrix} & 0 & 0 \\ -I_n & \vdots & \vdots \\ & 0 & 0 \\ e_n^T & -1 & 0 \\ -e_n^T & 0 & -1 \end{pmatrix}$$

where I_n is the identity matrix of order n, and e_n^T is the row vector in $\mathbf{R}^n$ all the entries in which are "1". Consider the LCP $(q(n+2), M(n+2))$ of order $n+2$. Are any of the algorithms discussed in this chapter able to process this LCP ? Why ? If not, develop an algorithm for solving this LCP using the special structure of the matrix M.

2.16 Consider the quadratic program

$$\begin{array}{ll} \text{minimize} & -x_1 - 2x_2 + \frac{1}{2}(2x_1^2 + 4x_1x_2 + 4x_2^2) \\ \text{subject to} & 3x_1 - 2x_2 - x_3 \qquad = 2 \\ & -x_1 + 2x_2 \qquad - x_4 = 6 \\ & x_j \geqq 0 \quad \text{for all } j \,. \end{array}$$

Formulate this program as an LCP of order 4 and write down this LCP clearly. Does a solution of this LCP lead to a solution of this quadratic program ? Why ?

It is required to solve this LCP using the variant of complementary pivot method in which the column vector of the artificial variable is $(1, 2, 2, 6)^T$. Obtain the canonical tableau corresponding to the initial almost complementary basic vector, and then carry out exactly one more pivot step in this algorithm.

2.17 Suppose $B \geqq 0$, and the linear programs

i) Maximize $c^T x$; subject to $Ax \leqq b$, $x \geqq 0$ and

ii) Minimize $b^T y$; subject to $(A+B)^T y \geqq c$, $y \geqq 0$

have finite optimum solutions. Show that the complementary pivot algorithm terminates with a complementary feasible solution for the LCP (q, M) with

$$q = \begin{pmatrix} -c \\ b \end{pmatrix}, \qquad M = \begin{pmatrix} 0 & (A+B)^T \\ -A & 0 \end{pmatrix}.$$

(G. B. Dantzig and A. S. Manne [2.6])

2.18 Let $\mathbf{\Gamma}$ be a nonenmpty closed convex subset of $\mathbf{R}^n$. For each $x \in \mathbf{R}^n$ let $P_\Gamma(x)$ denote the nearest point in $\mathbf{\Gamma}$ to x in terms of the usual Euclidean distance. Prove the following :

$$\text{(i)} \quad ||P_\Gamma(x) - y||^2 \leqq ||x - y||^2 \text{ for all } x \in \mathbf{R}^n, y \in \mathbf{\Gamma}\,.$$

$$\text{(ii)} \quad ||P_\Gamma(x) - P_\Gamma(y)||^2 \leqq ||x - y||^2 \text{ for all } x, y \in \mathbf{R}^n\,.$$

(Y. C. Cheng [3.6])

2.19 Let G and H be symmetric PSD matrices of order n and m respectively. Consider the following quadratic programs :

$$\begin{array}{ll} \text{maximize} & cx - \frac{1}{2}x^T Gx - \frac{1}{2}y^T Hy \\ \text{subject to} & Ax - Hy \leqq b \\ & x \geqq 0 \end{array}$$

and

$$\begin{array}{ll} \text{minimize} & b^T y + \frac{1}{2}x^T Gx + \frac{1}{2}y^T Hy \\ \text{subject to} & Gx + A^T y \geqq c^T \\ & y \geqq 0 \end{array}$$

Prove that if both the problems are feasible, then each has an optimal solution, and the optimum objective values are equal; moreover, the optimal solutions can be taken to be the same.
(R. W. Cottle [2.5] and W. S. Dorn [2.7])

2.20 Let M be a nondegenerate square matrix of order n. Let $d \in \mathbf{R}^n$, $d > 0$ be such that for every $\mathbf{J} \subset \{1, \ldots, n\}$, if $d_{\mathbf{J}} = (d_j : j \in \mathbf{J})$, $M_{\mathbf{JJ}} = (m_{ij} : i, j \in \mathbf{J})$, then $(M_{\mathbf{JJ}})^{-1} d_{\mathbf{J}} \geqq 0$. Then prove that if the LCP (q, M) is solved by the variant of the complementary pivot algorithm discussed in Section 2.3.3 with $-d$ as the original column vector for the artificial variable z^0, it will terminate with a solution of the LCP after at most $(n + 1)$ pivot steps.
(J. S. Pang and R. Chandrasekaran [8.18])

2.21 Consider the process of solving the LCP (q, M) by the complementary pivot algorithm. Prove that the value of the artificial variable z_0 decreases as the algorithm progresses, whenever M is either a PSD matrix or a P-matrix or a P_0-matrix, until termination occurs.
(R. W. Cottle [4.5] and B. C. Eaves [2.8])

2.22 Consider the process of solving the LCP (q, M) by the variant of the complementary pivot algorithm discussed in Section 2.3.3 with the column vector $d > 0$ as the initial column vector associated with the artificial variable z_0. Prove that in this process, there exists no secondary ray for all $d > 0 > q$ iff M is an $L_\star$-matrix. Using this prove that the variant of the complementary pivot algorithm discussed in Section 2.3.3 with the lexico minimum ratio rule for the dropping variable section in each step, will always terminate with a complementary solution for all q, no matter what $d > 0$ is used, iff M is an $L_\star$-matrix.
(B. C. Eaves [2.8])

2.23 Consider the convex quadratic programming problem

$$\begin{array}{ll} \text{minimize} & Q(x) = cx + \frac{1}{2}x^T Dx \\ \text{subject} & Ax \geqq b \\ & x \geqq 0 \end{array}$$

where D is a symmetric PSD matrix. If the problem has alternate optimum solution prove the following :

(i) the set of optimum solutions is a convex set,

(ii) $(y-x)^T D(y-x) = 0$ and actually $(y-x)^T D = 0$ for every pair of optimum solutions x and y, of the problem,

(iii) the gradient vector of $Q(x)$, $\nabla Q(x)$ is a constant on the set of optimum solutions,

(iv) the set of optimum solutions is the intersection of the constraint set with some linear manifold.

(M. Frank and P. Wolfe [10.14])

2.24 Let A^1, B^1, two given matrices of orders $m \times n$ each, be the loss matrices in a bimatrix game problem. Prove that the problem of computing a Nash equilibrium strategy pair of vectors for this bimatrix game, can be posed as the LCP (q, M), where

$$q = \begin{pmatrix} -e_m \\ e_n \end{pmatrix}, \qquad M = \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix}$$

where $A > 0$ and $B < 0$. Prove (use Lemma 2.8) that the complementary pivot algorithm will terminate with a solution when applied on this LCP.

(B. C. Eaves [2.8])

2.25 Consider the LCP (q, M) of order n. Let $\mathbf{C}_1$ be the set of feasible solutions of the system

$$\begin{aligned} w - Mz &= q \\ w,\ z &\geqq 0 \\ w_j z_j &= 0, \quad j = 2 \text{ to } n. \end{aligned}$$

If q is nondegenerate in the LCP (q, M) (i.e., if in every solution (w, z) of the system of linear equations "$w - Mz = q$", at least n variables are nonzero) prove that $\mathbf{C}_1$ is a disjoint union of edge paths. What happens to this result if q is degenerate ?

2.26 In Merrill's algorithm for computing a Kakutani fixed point discussed in Section 2.7.8, we defined the piecewise linear map in the top layer of the special triangulation of $\mathbf{R}^n \times [0,1]$ by defining for any vertex $V = \begin{pmatrix} v \\ 1 \end{pmatrix}$, $f(V) = \begin{pmatrix} f(v) \\ 1 \end{pmatrix}$ where $f(v)$ is an arbitrary point chosen from the set $\mathbf{F}(v)$. Examine the advantages that could be gained by defining $f(v)$ to be the nearest point (in terms of the usual Euclidean distance) in the set $\mathbf{F}(v)$ to v.

2.27 Let M, q be given matrices of orders $n \times n$ and $n \times 1$ respectively. If $y^T My + y^T q$ is bounded below on the set $\{y : y \geqq 0\}$, prove that the LCP (q, M) has a complementary solution, and that a complementary solution can be obtained by applying the complementary pivot algorithm on the LCP of order $(n+1)$ with data

$$q = \begin{pmatrix} q \\ q_{n+1} \end{pmatrix}, \qquad M = \begin{pmatrix} M & e \\ -e^T & 0 \end{pmatrix}$$

where $q_{n+1} > 0$, with the initial column vector associated with the artificial variable z_0 to be $(-1, \ldots, -1, 0) \in \mathbf{R}^{n+1}$.
(B. C. Eaves [2.8])

2.28 Consider the general quadratic program (2.67). If $Q(x)$ is unbounded below on the set of feasible solutions $\mathbf{K}$ of this problem, prove that there exists a feasible half-line through an extreme point of $\mathbf{K}$ along which $Q(x)$ diverges to $-\infty$.
(B. C. Eaves [2.9])

2.29 Let M be a given square matrix of order n. Let $\{B_{.1}, \ldots, B_{.r}\}$ be a given set of column vectors in $\mathbf{R}^n$. It is required to check whether $x^T Mx$ is $\geqq 0$ for all $x \in \text{Pos}\{B_{.1}, \ldots, B_{.r}\}$. Transform this into the problem of checking the copositivity of a matrix.

Can the problem of checking whether $x^T Mx$ is $\geqq 0$ for all $x \in \{x : Ax \geqq 0\}$ where A is a given matrix of order $m \times n$, be also transformed into the problem of checking the copositivity of a matrix ? How ?

2.30 (Research Problem) Application to pure 0-1 Integer Programming

Consider the pure 0-1 integer programming problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = b \\ & Dx \geqq d \\ & x_j = 0 \text{ or } 1 \text{ for all } j \end{array}$$

where $x \in \mathbf{R}^n$, and c, A, b, D, d are the data in the problem. In the interval $0 \leqq x_j \leqq 1$, the function $x_j(1-x_j)$ is non-negative, and is zero iff x_j is either 0 or 1. Using this we can transfom the above discrete problem into a continuous variable optimization by a penalty transformation as given below

$$\begin{array}{ll} \text{minimize} & cx + \alpha\left(\sum_{j=1}^{n} x_j(1-x_j)\right) \\ \text{subject to} & Ax = b \\ & Dx \geqq d \\ & 0 \leqq x_j \leqq 1, \; j = 1 \text{ to } n \end{array}$$

where α is a large positive penalty parameter. This is now a quadratic programming problem (unfortunately, it is a concave minimization problem and may have lots of local minima, in fact it can be verified that every integer feasible solution is a local minima for this problem). Check whether any of the algorithm for LCP discussed here are useful to approach the integer program through the LCP formulation of the above quadratic program.

2.31 Consider the system

$$\begin{aligned} w - Mz &= q \\ w, z &\geqq 0 \end{aligned}$$

where M is a given square matrix of order n. Let $\mathbf{C}_1$ be the set of feasible solutions of this problem satisfying the additional conditions

$$w_j z_j = 0, \quad j = 2 \text{ to } n \ .$$

Assuming that q is nondegenerate in this system (i.e., that in every solution (w, z) of the system of equations "$w - Mz = q$", at last n variables are non-zero), study whether $\mathbf{C}_1$ can contain an edge path terminating with extreme half-lines at both ends, when M is a copositive plus matrix.

2.32 (Research Problem) : Consider the general quadratic programming problem (2.67) of Section 2.9.2, and let $\mathbf{K}$ be its set of feasible solutions.

Develop necessary and sufficient conditions for $Q(x)$ to be unbounded below on $\mathbf{K}$. Develop an efficient procedure to check whether $Q(x)$ is unbounded below on $\mathbf{K}$.

In (2.67), the objective function is said to be strongly unbounded below, if it remains unbounded below whatever the vector c may be, as long as all the other data in the problem remains unchanged. Develop necessary and sufficient conditions for and an efficient procedure to check this strong unboundedness.

Extend the enumeration procedure for solving the general quadratic programming problem under the assumption of a bounded feasible set discussed in Section 2.9, to the case when $\mathbf{K}$ is unbounded.

The method discussed in Section 2.9 for solving this problem, may be viewed as a total enumeration method (enumerating over all the faces of $\mathbf{K}$). Develop an efficient method for computing a lower bound for $Q(x)$ on $\mathbf{K}$, and using it, develop a branch and bound method for solving this problem (this will be an efficient partial enumeration method). (See B. C. Eaves [2.9] for some useful information on this problem.)

2.33 Let M be a square matrix of order n which is $D + E$ where

D is symmetric and copositive plus

E is copositive.

Let $q \in \mathbf{R}_n$. If the system $Dx - E^T y \geqq -q$, $y \geqq 0$ is feasible, prove that the complementary pivot algorithm will terminate with a solution when applied on the LCP (q, M).
(P. C. Jones [2.17])

2.34 Let $(\bar{w}, \bar{z})$ be the solution of the LCP (q, M).

i) If M is PSD, prove that $\bar{z}^T q \leqq 0$.
ii) If the LCP (q, M) comes from an LP prove that $\bar{z}^T q = 0$.

2.35 Prove that if M is a copositive plus matrix of order n, and $q \in \mathbf{R}^n$ then the optimum objective value in the following quadratic program is zero, if the problem has a feasible solution.

$$\begin{array}{ll} \text{minimize} & Q(x) = x^T(Mx + q) \\ \text{subject to} & Mx + q \geqq 0 \\ & \quad x \qquad \geqq 0 \end{array}$$

2.36 In Section 2.9.2, we have seen that if a quadratic function $Q(x)$ is bounded below on a convex polyhedron, then $Q(x)$ has a finite global minimum point on that polyhedron. Does this result hold for a general polynomial function ?
(Hint: Examine the fourth degree polynomial function $f(x) = x_1^2 + (x_1 x_2 - 1)^2$ defined over $\mathbf{R}^2$).
(L. M. Kelly)

2.37 Apply the Complementary pivot method on the LCP with the following data.

a) $$q = \begin{pmatrix} -4 \\ -5 \\ -1 \end{pmatrix}, \quad M = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

b) $$q = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 2 & 0 \\ -2 & -1 & 0 \\ -1 & -3 & -1 \end{pmatrix}$$

c) $$q = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}, \quad M = \begin{pmatrix} -1 & 2 & -2 \\ 2 & -1 & 2 \\ -2 & 2 & -1 \end{pmatrix}.$$

Verify that (z_1, z_2, z_3) is a complementary feasible basic vector for (c).

Also, solve (a) by the variant of the complementary pivot method discussed in Section 2.4.

2.11 References

2.1 I. Adler, "The Expected Number of Pivots Needed to Solve Parametric Linear Programs and the Efficiency of the Self-dual Simplex Method", Department of Industrial Engineering and Operations Research, University of California, Berkeley, California (May 1983).

2.2 V. Aggarval, "On the Generation of all Equilibrium Points for Bimatrix Games Through the Lemke-Howson Algorithm", *Mathematical Programming*, 4 (1973) 233–234.

2.3 F. A. Al-Khayyal, "Linear, Quadratic and Bilinear Programming Approaches to the Linear Complementary Problem", Industrial and Systems Engineering Report No.J–83–5, Georgia Institute of Technology, Atlanta, Georgia (September 1983).

2.4 J. C. G. Boot, "Quadratic Programming", North-Holland, Amsterdam, The Netherlands (1964).

2.5 R. W. Cottle, "Symmetric Dual Quadratic Programs", *Quarterly of Applied Mathematics*, 21 (1963), 237–243.

2.6 G. B. Dantzig & A. Manne, "A Complementarity Algorithm for an Optimal Capital Path with Invariant Proportions", *Journal of Economic Theory*, 9 (1974) 312–323.

2.7 W. S. Dorn, "Duality in Quadratic Programming", *Quaterly of Applied Mathematics*, 18 (1960) 155–162.

2.8 B. C. Eaves, "The Linear Complementarity Problem", Management Science, 17, 9 (May 1971) 612–634.

2.9 B. C. Eaves, "On Quadratic Programming", Management Science, 17, 11 (July 1971) 698–711.

2.10 B. C. Eaves & H. Scarf, "The solution of Systems of Piecewise Linear Equations", *Mathematics of Operations Research*, 1 (1976) 1–27.

2.11 J. J. Evers, "More With Lemke Complementarity Algorithm", *Mathematical Programming*, 15 (1978) 214–219.

2.12 P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders & M. H. Wright, "A Range-space Method for Quadratic Programming with Bounds and General Constraints", Technical Report SOL 82–15, Department of Operations Research, Stanford University, Stanford, California (October 1982).

2.13 P. E. Gill & W. Murray, "Numerically Stable Methods for Quadratic Programming", *Mathematical Programming*, 14 (1978) 349–372.

2.14 F. J. Gould & J. W. Tolle, "A Unified Approach to Complementarity in Optimization", *Discrete Mathematics*, 7 (1974) 225–271.

2.15 W. P. Hallman & I. Kaneko, "On the Connectedness of the Set of Almost Complementary Paths in a Linear Complementarity Problem", *Mathematical Programming*, 16 (1979) 384–385.

2.16 P. C. Jones, "Computing an Optimal Invariant Capital Stock", *SIAM Journal of Algebraic and Discrete Methods*, 3 (No.2)(1982) 145–150.

2.17 P. C. Jones, "Even more with the Lemke complementary algorithm", Technical Report, Department of IEMS, Northwestern University, Evanston, IL 60201, December 1984.

2.18 E. L. Keller, "Quadratic Optimization and Linear Complementarity", Ph.D. Dissertation, University of Michigan (1969).

2.19 E. L. Keller, "The General Quadratic Optimization Problem", *Mathematical Programming*, 5 (1973) 311–337.

2.20 M. M. Kostreva, "Cycling in Linear Complementarity Problems", *Mathematical Programming*, 16 (1979) 127–130.

2.21 C. E. Lemke, "Bimatrix Equilibrium Points and Mathematical Programming", *Management Science*, 11 (1965) 681–689.

2.22 C. E. Lemke, "Some Pivot Schemes for the Linear Complementarity Problem", *Mathematical Programming Study*, 7 (1978) 15–35.

2.23 R. K. Mueller, "A Method for Solving the Indefinite Quadratic Programming Problem", *Management Science (Theory)*, 16 (No.5)(1970) 333–339.

2.24 K. G. Murty, "On the Finite Nature of Quadratic Programming: A Note on R. K. Mueller's Paper", Technical Report 71–5, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan (1971).

2.25 K. G. Murty, "Algorithm for Finding all the Feasible Complementary Bases for a Linear Complementarity Problem", Technical Report No. 72–2, Department IOE, University of Michigan, Ann Arbor, Michigan (February 1972).

2.26 K. G. Murty, *Linear Programming*, John Wiley & Sons, New York (1983).

2.27 H. Nishino, M. Kojima & T. Sekine, "An Extension of Lemke's Method to the Piecewise Linear Complementarity Problem", *SIAM Journal on Applied Mathematics*, 31 (1976) 600–613.

2.28 T. D. Parsons, "A Combinatorial Approach to Convex Quadratic Programming", Ph. D. Dissertation, Department of Mathematics, Princeton University (1966).

2.29 F. J. Pereira R., "On Characterizations of Copostive Matrices", Technical Report No. 72–8, OR House, Stanford University, Stanford, California (May 1972).

2.30 A. Ravindran, "A Comparison of the Primal Simplex and Complementary Pivot Methods for Linear Programming", *Naval Research Logistics Quarterly*, 20 (No.1) (1973) 95–100.

2.31 R. Saigal, "On Some Average Results for Linear Complementarity Problems", Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois (August 1983).

2.32 R. Saigal, "Lemke's Algorithm and a Special Linear Complementarity Problem", *Opsearch* 8 (1971) 201–208.

2.33 L. S. Shapley, "A Note on the Lemke-Howson Algorithm", *Mathematical Programming Study 1* (1974) 175–189.

2.34 M. J. Todd, "Extensions of Lemke's Algorithm for the Linear Complementarity Problem", *Journal of Optimization Theory and Applications*, 20 (1976) 397–416.

2.35 M. J. Todd, "Linear and Quadratic Programming in Oriented Matroids", Technical Report No.565, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York (1983).

2.36 M. J. Todd, "Polynomial Expected Behavior of a Pivoting Algorithm for Linear Complementarity and Linear Programming Problems", Technical Report No.595, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York (1983).

2.37 C. Van de Panne, "A Complementary Variant of Lemke's Method for the Linear Complementarity Problem", *Mathematical Programming*, 7 (1974) 283–310.

2.38 L. Van der Heyden, "A Variable Dimension Algorithm for the Linear Complementarity Problem", *Mathematical Programming*, 19 (1980) 328–346.

2.39 L. Van der Heyden, "On a Variable Dimension Algorithm for the Linear Complementarity Problem", Cowles Foundation Discussion Paper No.689 (January 1984).

References on Fixed Point Computing Methods

2.40 E. Allgower & K. Georg, "Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations", *SIAM Review*, 22 (1980) 28–85.

2.41 M. N. Broadie, "Computational Experience with the Octahedral Algorithm and Related Results", Department of OR, Stanford University, Stanford, California (March 1983).

2.42 S. N. Chow, J. Mallet-Paret & J. A. Torke, "Finding Zeros of Maps: Homotopy Methods that are Constructive with Probability One", *Math. Comput.*; 32, pp. 887–899, 1978.

2.43 B. C. Eaves, "Computing Kakutani Fixed Points", *SIAM Journal of Applied Mathematics*, 21, pp. 236–244, 1971.

2.44 B. C. Eaves, "Homotopies for Computation of Fixed Points", *Mathematical Programming* 3 (1972) 1–22.

2.45 B. C. Eaves, *A course in triangulations for solving equations with deformations,* Lecture Notes in Economics and Mathematical Systems 293, Springer Verlag, New York, 1984.

2.46 B. C. Eaves, F. J. Gould, H. O. Peitgen & M. J. Todd (editors), *Homotopy Methods and Global Convergence*, Plenum Press, New York (1983).

2.47 B. C. Eaves & R. Saigal, "Homotopies for Computation of Fixed Points on Unbounded Regions", *Mathematical Programming*, 3 (1972) 225–237.

2.48 C. B. Garcia & W. I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria,* Prentice-Hall, Englewood Cliffs, New Jersey (1981).

2.49 F. J. Gould & J. W. Tolle, *Complementary Pivoting on a Pseudo-manifold Structure with Applications in the Decision Sciences,* Heldermann Verlag, Berlin (1983).

2.50 S. Kakutani, "A Generalization of Brouwer's Fixed Point Theorem", *Duke Mathematical Journal*, 8 (1941) 457–459.

2.51 S. Karamardian & C. B. Garcia (editors), *Fixed Point Algorithms and Applications,* Academic Press, New York (1977).

2.52 R. B. Kellog, T. Y. Li & J. Yorke, "A Constructive Proof of the Brouwer Fixed Point Theorem and Computational Results", *SIAM Journal on Numerical Analysis*, 13, pp. 473–483, 1976.

2.53 M. Kojima & Y. Yamamoto, "A unified Approach to the Implementation of Several Restart Fixed Point Algorithms and a New Variable Dimension Algorithm", *Mathematical Programming* 28 (1984) 288–328.

2.54 H. W. Kuhn, "Approximate Search for Fixed Points", *Computing Methods in Optimization Problems*, 2, Academic Press, New York (1969).

2.55 H. J. Luethi, "A Simplicial Approximation of a Solution for the Nonlinear Complementarity Problem", *Mathematical Programming*, 9 (1975) 278–293.

2.56 O. L. Mangasarian, "Equivalence of the Complementarity Problem to a System of Nonlinear Equations", *SIAM Journal of Applied Mathematics*, 31 (No.1) (1976) 89–92.

2.57 O. H. Merrill, "A Summary of Techniques for Computing Fixed Points of Continuous Mappings", *Mathematical Topics in Economic Theory and Computation*, SIAM, Philadelphia, Pennsylvania (1972).

2.58 O. H. Merrill, "Application and Extensions of an Algorithm that Computes Fixed Points of Upper Semicontinuous Point to Set Mappings", Ph. D. Dissertation, Department of Industrial and Operations Engineering, The University of Michigan (1972). Available from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan, 48106, USA, under number 73–6876.

2.59 A. Netravali & R. Saigal, "Optimum Quantizer Design Using a Fixed Point Algorithm", *The Bell System Technical Journal*, 55 (No.9) (November 1976) 1423–1435.

2.60 J. M. Ortega & C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).

2.61 S. M. Robinson (Editor), *Analysis and Computation of Fixed Points,* Academic Press, (1980).

2.62 R. Saigal, D. Solow & L. Wolsey, "A Comparative Study of Two Algorithms that Compute Fixed Points in Unbounded Regions", VIII Internationsl Symposium of Mathematical Programming, Stanford University, Stanford, California (August 1973).

2.63 R. Saigal, "Fixed Point Coumputing Methods", in J. Belzer, A. G. Holzman and A. Kent (editors) *Encyclopedia of Computer Science and Technology*, Volume 8, Marcel Dekker, Inc., New York (1976) 213–235.

2.64 R. Saigal, "On the Convergence Rate of Algorithm for Solving Equations that are Based on Methods of Complementary Pivoting", *Mathematics of Operations Research*, 2 (1977) 108–124.

2.65 R. Saigal & M. J. Todd, "Efficient Acceleration Techniques for Fixed Point Algorithms", *SIAM Journal on Numerical Mathematics*, 15 (1978) 997–1007.

2.66 R. Saigal, "A Homotopy for Solving Large, Sparse and Structured Fixed Point Problems", *Mathematics of Operations Research*, 8 (No.4) (1983) 557–578.

2.67 R. Saigal, "Computational Complexity of a Piecewise Linear Homotopy Algorithm", *Mathematical Programming*, 28 (1984) 164–173.

2.68 H. E. Scarf, "The Approximation of Fixed Points of Continuous Mappings", *SIAM Journal of Applied Mathematics*, 15 (1967) 1328–1343.

2.69 H. E. Scarf (in collaboration with T. Hansen), *The Computation of Economic Equilibria,* Yale University Press, New Haven, Connecticut (1973).

2.70 D. Solow, "Homeomorphisms of Triangulations with Applications to Computing Fixed Points", *Mathematical Programming*, 20, 2, pp. 213–224, March 1981.

2.71 M. J. Todd, "On Triangulations for Computing Fixed Points", *Mathematical Programming*, 10 (1976a) 322–346.

2.72 M. J. Todd, *The Computation of Fixed Points and Applications,* Lecture Notes in Economics and Mathematical Systems 124, Springer Verlag, New York (1976b).

2.73 M. J. Todd, "Exploiting Structure in Piecewise-Linear Homotopy Algorithms for Solving Equations", *Mathematical Programming*, 18 (1980) 233–247.

2.74 M. J. Todd, "On the Computational Complexity of Piecewise Linear Homotopy Algorithms", *Mathematical Programming*, 24 (1982) 216–224.

2.75 M. J. Todd, "J': A New Triangulation of $\mathbf{R}^n$", *SIAM Journal on Algebraic and Discrete Methods*, 5 No.2 (June 1984) 244–254.

2.76 M. J. Todd, "Fat Triangulations, or solving certain nonconvex matrix optimization problems", *Mathematical Programming*, 31 (1985) 123–136.

2.77 G. Van der Laan & A. J. J. Talman, "A Class of Simplicial Restart Fixed Point Algorithms Without an Extra Dimension", *Mathematical Programming*, 20 (1981) 33–48.

2.78 L. T. Watson, "A Globally Convergent Algorithm for Computing Fixed Points of C^2 Maps", *Applied Mathematics and Computation*, 5, pp. 297–311, 1979.

2.79 R. J. Wilmuth, "The Computation of Fixed Points", Ph. D. Dissertation, Department of OR, Stanford University, Stanford, California (1973).

2.80 A. H. Wright, "The Octahedral Algorithm: A New Simplicial Fixed Point Algorithm", *Mathematical Programming*, 21 (1981) 47–69.

References on Available Computer Codes

2.81 M. N. Broadie, "Octasolve User's Guide", Department of OR, Stanford University, Stanford, Californis (1983).

2.82 R. L. Crane, K. E. Hillstrom & M. Minkoff, "Solution of the General Nonlinear Programming Problem with Subroutine VMCON", ANL80–64, Argonne National Laboratory, Illinois 60439 (July 1980).

2.83 P. E. Gill, W. Murray, M. A. Saunders & M. H. Wright, "User's Guide for SOL/QPSOL", Technical Report SOL 83–7, Department of OR, Stanford University, Stanford, California (1983).

2.84 P. E. Gill, W. Murray, M. A. Saunders & M. H. Wright, "User's Guide for SOL/NPSOL: A Fortran Package for Nonlinear Programming", Technical Report SOL 83–12, Department of OR, Stanford University, Stanford, California (July 1983).

2.85 B. A. Murtagh & M. A. Saunders, "MINOS: A Large Scale Nonlinear Programming System for Problems with Linear Constraints, User's Guide", Technical Report SOL 77–9, Department of OR, Stanford University, Stanford, California 94305 (February 1977). Also, see "MINOS 5.0 User's Guide", Technical Report SOL 83–20, (December 1983).

2.86 R. Saigal, "Efficient Algorithms for Computing Fixed Points When Mappings May be Separable, A Computer Program and User's Guide", Department of Industrial Engineering, Northwestern University, Evanston, Illinois (1979).

2.87 K. Schittkowski, "User's Guide for the Nonlinear Programming Code NLPQL", Institut fuer Informatik, Universitaet Stuttgart, Azenbergstr. 12, D-7000 Stuttgart 1, Federal Republic of Germany, 1983.

2.88 M. J. Todd, "PLALGO: A Fortran Implementation of a Piecewise-linear Homotopy Algorithm for Solving Systems of Nonlinear Equations", Technical Report No.452, School of OR and IE, Cornell University, Ithaca, New York (1981).

2.89 J. Tomlin, "Robust Implementation of Lemke's Method for the Linear Complementarity Problem", *Mathematical Programming Study* 7 (1978) 55–60.

2.90 J. Tomlin, "User's Guide to LCPL, A Program for Solving Linear Complementarity Problems by Lemke's Method", Technical Report SOL76–16, Department of Operations Research, Stanford University, Stanford, California (1976).

2.91 L. T. Watson, S. C. Billups & A. P. Morgan, "HOMPACK: A suite of codes for globally convergent homotopy algorithms", Technical Report 85–34, Industrial and Operations Engineering Department, The University of Michigan, Ann Arbor, Michigan 48109, USA, November 1985.

References on Convexity and Convex Programming

2.92 R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey (1970).

2.93 R. T. Rockafellar, "The Theory of Subgradients and Its Applications on Problems of Optimization: Convex and Nonconvex Functions", Heldermann Verlag, Berlin (1981).

2.94 J. Stoer & C. Witzgall, "Convexity and Optimization in Finite Dimensions I", Springer Verlag, Berlin (1970).