# Linear Complementarity and Mathematical (Non-linear) Programming

Joel Friedman, Dept. of Math., UBC

April 3, 1998

## 1 Introduction

The goal of these notes is to give a quick introduction to convex quadratic programming and the tools needed to solve it. We assume that the reader is familiar with the dictionary approach to the simplex method. Tableaux can be used for everything done here, but they look a little less intuitive to us. Hence we will stick with dictionaries.

A standard form for a linear program is maximize $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. In quadratic programming we subtract from $\mathbf{c}^T\mathbf{x}$ a quadratic function of $\mathbf{x}$. When this quadratic term is convex, i.e. takes on only non-negative values, then the problem becomes much easier to solve than otherwise.

Both linear programming and (convex) quadratic programming can be reduced to the *linear complementarity problem*. In addition to this, the linear complementarity problem is easy to explain, easy to solve under certain conditions, and is therefore the starting point of these notes. We then discuss the *Karush-Kuhn-Tucker conditions* which reduce (convex) quadratic programming to the linear complementarity problem.

Throughout these notes all vectors will be column vectors unless otherwise indicated. The only exception to this rule will be gradients; if $f\colon \mathbf{R}^n \to \mathbf{R}$ is any differentiable function, then

$$\nabla f = [\partial f/\partial x_1 \quad \cdots \quad \partial f/\partial x_n]$$

is a row vector[1]. All dot products will be indicated by writing transposes, e.g. the dot product of $\mathbf{x}$ and $\mathbf{y}$ will be written $\mathbf{x}^T\mathbf{y}$ or $\mathbf{y}^T\mathbf{x}$.

[1] We think of the gradients as acting (via dot product) on column vectors, which means they must be row vectors (unless we want to write transpose signs everywhere).

## 2 The Linear Complementarity Problem

The *linear complementarity problem* is given a $\mathbf{q} \in \mathbf{R}^p$ and a $p \times p$ matrix, $\mathbf{M}$, to find $\mathbf{w}, \mathbf{z} \in \mathbf{R}^p$ satisfying

$$\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z}, \qquad \mathbf{w}\,\mathbf{z} = \mathbf{0}, \qquad \mathbf{w}, \mathbf{z} \geq \mathbf{0}. \tag{2.1}$$

Here $\mathbf{w}\,\mathbf{z}$ is interpreted as componentwise multiplication, so that $\mathbf{w}\,\mathbf{z} = \mathbf{0}$ means that for each $i$ we have $w_i z_i = 0$, i.e. at least one of $w_i, z_i$ is zero. (Notice that the $\mathbf{w}, \mathbf{z}$ have no relationship to the objective variables $w, z$ that we used in linear programming.) We will give two important types of problems which can be viewed as special cases of this linear complementarity problem.

By complementary slackness, we know that linear programming can be seen as a special case of equation 2.1. Namely, maximizing $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ is equivalent to solving

$$\mathbf{s} = \mathbf{b} - \mathbf{A}\mathbf{x}, \quad \mathbf{u} = -\mathbf{c} + \mathbf{A}^T\mathbf{y}, \quad \mathbf{u}\mathbf{x} = \mathbf{0}, \quad \mathbf{s}\mathbf{y} = \mathbf{0}$$

$$\text{and} \quad \mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{u} \geq \mathbf{0},$$

where $\mathbf{s}, \mathbf{u}$ are the primal slack and dual slack variables. Hence our linear program is equivalent to solving equation 2.1 for

$$\mathbf{w} = \begin{bmatrix} \mathbf{u} \\ \mathbf{s} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix},$$

$$\text{and} \quad \mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{bmatrix}.$$

The important second problem which is a special case of the linear complementarity problem is that of *convex quadratic programming*. We will explain it in detail later in this note. Roughly speaking, this problem is like linear programming, except that the objective function (which is minimized, not maximized) is allowed to be a *convex, quadratic* function of the variables.

## 3 An Algorithm for Linear Complementarity

In this section we describe an algorithm for the linear complementarity problem, due to Lemke and Howson. This algorithm may not work, but in the next section we will give conditions on $\mathbf{M}$ which ensure that the algorithm works; these conditions on $\mathbf{M}$ will be satisfied in a number of important cases.

To describe our algorithm, we view

$$\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z}$$

as a dictionary for the basic variables, $\mathbf{w}$, in terms of the nonbasic variables, $\mathbf{z}$. If $\mathbf{q} \geq \mathbf{0}$, then this dictionary is feasible, i.e. the corresponding basic feasible solution ($\mathbf{z} = \mathbf{0}$ and $\mathbf{w} = \mathbf{q}$) is non-negative, and we are done. If not, then we begin a two phase algorithm.

Phase I of the algorithm involves adding an auxiliary variable, $z_0$, and involves one pivot. Namely, we modify the dictionary to

$$\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z} + \mathbf{1}z_0,$$

where $\mathbf{1}$ is the column vector all of whose entries are 1. Now we make $z_0$ enter the basis and choose the leaving variable from $\mathbf{w}$ so as to make the new dictionary feasible (this means that all the variables, including $z_0$, must be non-negative). This is the end of phase I.

Let us explain the goal of phase II. Our algorithm will end if we reach a dictionary with the following properties: (1) $z_0$ is nonbasic, and (2) for each $i = 1, \ldots, p$ we have either $z_i$ or $w_i$ is a nonbasic variable. Indeed, condition (1) will ensure that $z_0 = 0$ in the corresponding BFS, and so $\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z}$ holds; condition (2) will ensure that $\mathbf{w}\mathbf{z} = \mathbf{0}$. We call a dictionary satisfying conditions (1) and (2) a *terminal* dictionary. We call a dictionary *balanced* if it satisfies condition (2), i.e. if for each $i$ at least one of $w_i, z_i$ is nonbasic.

To arrive at a terminal dictionary, phase II pivots through balanced, feasible dictionaries. This is done as follows. If we are at a non-terminal dictionary, then $z_0$ is basic, and some variable, either a $w_i$ or a $z_i$ has just left the dictionary. Which variable should enter the dictionary on the next iteration? Only if $w_i$ or $z_i$ enters can we be assured that the new dictionary will be balanced; since we don't want to return to immediately return to same dictionary twice, we insist that if $w_i$ left then $z_i$ must enter on the next iteration, and that if $z_i$ left then $w_i$ must enter.

To better summarize the Lemke-Howson algorithm, we shall say that $w_i$ is *the complement* of $z_i$ and vice versa.

**Summary of the Lemke-Howson algorithm:** In phase I we add an auxiliary variable $z_0$ and pivot once to have $z_0$ enter and to have the new dictionary feasible. In phase II we repeatedly pivot, always maintaining feasibility in our dictionaries, and always taking the entering variable being the complement of the previously leaving variable. If at some point $z_0$ leaves the dictionary, we are done.

We illustrate this method on the problem:

$$\text{maximize } 2x_1 + x_2 \qquad \text{s.t. } x_1 + x_2 \leq 3, \qquad x_1, x_2 \geq 0.$$

We have the primal and dual dictionaries:

$$x_3 = 3 - x_1 - x_2 \qquad \begin{aligned} u_1 &= \ -2 + u_3 \\ u_2 &= \ -1 + u_3 \end{aligned}$$

where the $u_i$ are the renumbered dual variables. Setting

$$w_1 = u_1, w_2 = u_2, w_3 = x_3, z_1 = x_1, z_2 = x_2, z_3 = u_3$$

we get a linear complementarity problem in the notation as before. However, since $x_i$ and $u_i$ are complements, we will keep the $x$'s and $u$'s in the following calculation (the reader who likes can rewrite the equations below with $w$'s and $z$'s).

Entering phase I we add the auxiliary $z_0$ to get:

$$\begin{aligned} u_1 &= -2 + z_0 + u_3 \\ u_2 &= -1 + z_0 + u_3 \\ x_3 &= 3 + z_0 - x_1 - x_2 \end{aligned}$$

So $z_0$ enters and $u_1$ leaves, to get

$$\begin{aligned} z_0 &= 2 + u_1 - u_3 \\ u_2 &= 1 + u_1 \\ x_3 &= 5 + u_1 - u_3 - x_1 - x_2 \end{aligned}$$

Now we enter phase II. Since $u_1$ left, $x_1$ now enters and we find $x_3$ leaves:

$$\begin{aligned} z_0 &= 2 + u_1 - u_3 \\ u_2 &= 1 + u_1 \\ x_1 &= 5 + u_1 - u_3 - x_3 - x_2 \end{aligned}$$

Since $x_3$ left, $u_3$ enters; we find $z_0$ leaves and we are done:

$$\begin{aligned} u_3 &= 2 + u_1 - z_0 \\ u_2 &= 1 + u_1 \\ x_1 &= 3 + z_0 - x_3 - x_2 \end{aligned}$$

Our optimal solution to original primal LP is $(x_1, x_2) = (3, 0)$, and the optimal solution to the dual is $y_1 = 2$ (since $y_1 = u_3$).

**Discussion Question 3.1** In the final dictionary, the basic $x$'s were written only in terms of nonbasic $x$'s and $z_0$; similarly for the $u$'s. Do you think that this is always the case?

**Pitfalls of the Lemke-Howson algorithm.** Just like in the simplex method, there are a few potential problems with the Lemke-Howson algorithm. We divide them into three: (1) degeneracies, (2) cycling without degeneracies, and (3) a situation in which no variable leaves a dictionary. In the next section we will deal with the latter two problems in detail. Here we summarize the results and deal completely with the first problem.

By a degeneracies we mean a situation in which $\mathbf{q}$ has a zero, or in which the choice of leaving variable is ambiguous (leading to a dictionary with a zero for the constant term of one variable). While this is not necessarily a problem, it is undesirable because it is hard to make sure you won't cycle when degeneracies occur. One solution is to use the perturbation method just as in linear programming, i.e. add $\epsilon, \epsilon^2, \ldots$ to the constant terms of the dictionary. The same argument used in linear programming shows that such a dictionary can never degenerate. This gives one way to completely take care of degeneracies. We will give an example at the end of this section.

Once we eliminate the degeneracies, it turns out that cycling is impossible. However, this is true for reasons that are very different than in linear programming; we discuss these reasons in the next section.

Finally, it may happen that no variable can leave the dictionary, and the algorithm stops. Under a certain condition on $\mathbf{M}$, this will imply that the linear complementarity problem has no solution.

**Definition 3.2** $\mathbf{M}$ *is said to be* copositive *if* $\mathbf{x}^T\mathbf{M}\mathbf{x} \geq 0$ *for all* $\mathbf{x} \geq 0$. $\mathbf{M}$ *is said to be* copositive-plus *if it is copositive and if* $(\mathbf{M} + \mathbf{M}^T)\mathbf{x} = 0$ *for all* $\mathbf{x} \geq 0$ *such that* $\mathbf{x}^T\mathbf{M}\mathbf{x} = 0$ *and* $\mathbf{M}\mathbf{x} \geq 0$.

This the next section we will prove:

**Theorem 3.3** *If* $\mathbf{M}$ *is copositive-plus, then the Lemke-Howson algorithm can only stop with no leaving variable if the linear complementarity problem has no solution*

**Proposition 3.4** *If* $\mathbf{M}$ *comes from complementary slackness in a linear program, then* $\mathbf{M}$ *is copositive-plus.*

We will also see that the $\mathbf{M}$ arising from convex quadratic programming is also copositive-plus. This fact and the above proposition follow easily from:

**Proposition 3.5** *Let* $\mathbf{M}$ *be positive semidefinite; i.e.* $\mathbf{x}^T\mathbf{M}\mathbf{x} \geq 0$ *for all* $\mathbf{x}$. *(Note:* $\mathbf{M}$ *is* not *assumed to be symmetric*[2]*.) Then* $\mathbf{M}$ *is copositive-plus*

This proposition follows easily from the spectral theorem[3] applied to $\mathbf{M} + \mathbf{M}^T$.

We finish with our degeneracy example, from the problem

$$\text{minimize } x_1 + x_2 \qquad \text{s.t. } x_1 + x_2 \leq 2.$$

[2] $\mathbf{M}$ is symmetric if $\mathbf{M}^T = \mathbf{M}$.

[3] The spectral theorem says that any symmetric matrix has a purely real orthonormal diagonalization.

We get primal and dual dictionaries

$$\begin{aligned} x_3 &= 2 - x_1 - x_2 + z_0 \\ u_1 &= -1 + u_3 + z_0 \\ u_2 &= -1 + u_3 + z_0 \end{aligned}$$

We see that if $z_0$ enters then either $u_1$ or $u_2$ can leave, a degeneracy. So before pivoting we add $\epsilon$'s to the equations:

$$\begin{aligned} x_3 &= 2 + \epsilon - x_1 - x_2 + z_0 \\ u_1 &= -1 + \epsilon^2 + u_3 + z_0 \\ u_2 &= -1 + \epsilon^3 + u_3 + z_0 \end{aligned}$$

Now $z_0$ enters and $u_2$ leaves, giving

$$\begin{aligned} x_3 &= 3 + \epsilon - \epsilon^3 - x_1 - x_2 - u_3 + u_2 \\ u_1 &= \epsilon^2 - \epsilon^3 + u_2 \\ z_0 &= 1 - \epsilon^3 - u_3 + u_2 \end{aligned}$$

Now $x_2$ enters (since $u_2$ previously left), so that $x_3$ leaves:

$$\begin{aligned} x_2 &= 3 + \epsilon - \epsilon^3 - x_1 - x_3 - u_3 + u_2 \\ u_1 &= \epsilon^2 - \epsilon^3 + u_2 \\ z_0 &= 1 - \epsilon^3 - u_3 + u_2 \end{aligned}$$

Now $u_3$ enters and $z_0$ leaves:

$$\begin{aligned} x_2 &= 2 + \epsilon - 2\epsilon^3 - x_1 - x_3 + z_0 \\ u_1 &= \epsilon^2 - \epsilon^3 + u_2 \\ u_3 &= 1 - \epsilon^3 - z_0 + u_2 \end{aligned}$$

Taking $\epsilon \to 0$ we get the solution $\mathbf{x} = [0\ 2\ 0]^T$, $\mathbf{u} = [0\ 0\ 1]^T$.

# 4 More on the Lemke-Howson Algorithm

We will now discuss (1) cycling without degeneracies, and (2) stopping with no leaving variable in the Lemke-Howson algorithm. To understand cycling we invoke the "museum principle," otherwise known as the "train principle."

Imagine that you are in a museum, and you start visiting its rooms. The museum has the following properties:

- all its rooms are labelled "stop" or "continue,"

- all "continue" rooms have at most two doors,
- the room you start in is a "continue" room with only one door, and
- the museum has only finitely many rooms.

Assume your room visiting satisfies:

- if you reach a "stop" room then your visiting stops,
- if you reach a "continue" room with one door then your visiting stops, and
- if you reach a "continue" room with two doors then your visiting continues, leaving the room through the other door (i.e. the one through which you did not enter).

The "museum principle" says that you will never cycle, and that in a finite number of room changes your visiting will stop. It is pretty easy to convince yourself of this fact (see exercise 1).

It remains to verify that phase II of the Lemke-Howson algorithm is like this museum room visiting.

Let's consider the museum properties. Consider the museum whose rooms are the balanced, feasible dictionaries, and whose doors are pivots[4]. Clearly there are only finitely many rooms. A room is labelled "stop" if it is a terminal dictionary, all others are labelled "continue." So a "continue" dictionary is one with $z_0$ being basic; such a dictionary has exactly one pair of complementary variables which are both nonbasic; such a dictionary has at most two "doors," namely each of the pair of complementary variables can enter the dictionary in at most one way (in the absence of degeneracies).

Our last claim is that we begin phase II in a room with only one door. Indeed, in phase I, initially $z_0$ appeared as nonbasic with a coefficient of one in each dictionary equation. So if $w_i$ left the dictionary at the end of phase I, it follows that it appears with a one in every dictionary equation in the beginning of phase II. Hence there is no pivot or door in which $w_i$ enters and we arrive at a feasible dictionary (i.e. $w_i$ will be negative in each BFS arising from a dictionary in which $w_i$ enters). Hence our initial room has only one door.

Our visiting properties are clearly satisfied. It follows that phase II never cylces (unless degeneracies are present).

We conclude this section with the rather tedious proof that $\mathbf{M}$ being copositive-plus implies that our algorithm stops with no possible leaving variable exactly when the linear complementarity problem is not feasible.

If no leaving variable is possible, then any non-negative value, $t$, of the entering value gives a solution to

$$\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z} + \mathbf{1}z_0, \qquad \mathbf{w}\,\mathbf{z} = \mathbf{0}, \qquad \mathbf{w}, \mathbf{z} \geq \mathbf{0}, \qquad z_0 \geq 0.$$

[4]By a *pivot* we mean the act of having one variable enter the basis and another leave.

As $t$ varies, these solutions vary linearly, and hence we have solutions:

$$\mathbf{w} = \mathbf{w}^* + t\mathbf{w}^h, \qquad \mathbf{z} = \mathbf{z}^* + t\mathbf{z}^h, \qquad z_0 = z_0^* + tz_0^h,$$

and clearly at least one of $\mathbf{w}^h, \mathbf{z}^h, z_0^h$ is non-zero. From the non-negativity of $\mathbf{w}, \mathbf{z}, z_0$ and the vanishing of $\mathbf{wz}$ we conclude:

$$\mathbf{w}^*, \mathbf{w}^h, \mathbf{z}^*, \mathbf{z}^h \geq \mathbf{0}, \qquad z_0^*, z_0^h \geq 0, \qquad \mathbf{w}^h\mathbf{z}^h = \mathbf{w}^h\mathbf{z}^* = \mathbf{w}^*\mathbf{z}^h = \mathbf{w}^*\mathbf{z}^* = \mathbf{0}.$$

And from $\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z} + \mathbf{1}z_0$ we conclude

$$\mathbf{w}^* = \mathbf{q} + \mathbf{M}\mathbf{z}^* + \mathbf{1}z_0^*, \qquad \mathbf{w}^h = \mathbf{M}\mathbf{z}^h + \mathbf{1}z_0^h.$$

Clearly it suffices to establish the following claims, we which now do:

1. $z_0^* \neq 0$,
2. $\mathbf{z}^h \neq \mathbf{0}$ (assuming non-degeneracy at the phase I pivot and no cycling)
3. $z_0^h = 0$ and ${\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^h = 0$,
4. $\mathbf{M}\mathbf{z}^h \geq \mathbf{0}$ and ${\mathbf{z}^h}^T\mathbf{M} \leq \mathbf{0}^T$,
5. ${\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^* = 0$ and ${\mathbf{z}^h}^T\mathbf{q} < 0$,
6. there is $\mathbf{v} \geq \mathbf{0}$ with $\mathbf{v}^T\mathbf{q} < 0$ and $\mathbf{v}^T\mathbf{M} \leq \mathbf{0}$,
7. the previous claim makes $\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z}$ with $\mathbf{w}, \mathbf{z} \geq \mathbf{0}$ infeasible.

For the first claim, if $z_0^* = 0$ then our algorithm would be done. For the second claim, notice that $\mathbf{z}^h = \mathbf{0}$ implies that $\mathbf{w}^h = \mathbf{1}z_0^h$ with $z_0^h \neq 0$, and from $\mathbf{w}^h\mathbf{z}^* = \mathbf{0}$ we conclude $\mathbf{z}^* = 0$. So as $t$ increases, both $\mathbf{w}(t) = \mathbf{w}^* + t\mathbf{w}^h$ and $z_0(t) = z_0^* + tz_0^h$ increase ($\mathbf{w}(t)$ in every component); hence all the $\mathbf{z}$ variables are nonbasic in this dictionary and one the $\mathbf{w}$ variables, say $w_i$, is nonbasic and entering ($z_0^* \neq 0$ so $z_0$ is basic); assuming non-degeneracy at the phase I pivot, we see that this must be the first phase II dictionary. But this can't be the first phase II step, for $w_i$ is *entering*, and so we have cycled.

For the thrird claim, we have

$$0 = {\mathbf{z}^h}^T\mathbf{w}^h = {\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^h + {\mathbf{z}^h}^T\mathbf{1}z_0^h.$$

The copositivity of $\mathbf{M}$ implies

$${\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^h = 0, \qquad \text{and thus } {\mathbf{z}^h}^T\mathbf{1}z_0^h = 0.$$

Since $\mathbf{z}^h \neq \mathbf{0}$ we have ${\mathbf{z}^h}^T\mathbf{1} > 0$, and hence $z_0^h = 0$ and we have the third claim.

For the fourth claim, we have $z_0^h = 0$ implies that $\mathbf{M}\mathbf{z}^h = \mathbf{w}^h$ which is $\geq \mathbf{0}$. Then the copositivity-plus of $\mathbf{M}$ gives $\mathbf{M}\mathbf{z}^h + \mathbf{M}^T\mathbf{z}^h = \mathbf{0}$ and hence $\mathbf{M}^T\mathbf{z}^h = -\mathbf{M}\mathbf{z}^h = -\mathbf{w}^h \leq \mathbf{0}$ and so $\mathbf{M}^T\mathbf{z}^h \leq \mathbf{0}$ or ${\mathbf{z}^h}^T\mathbf{M} \leq \mathbf{0}^T$, which is the fourth claim. For the fifth claim we have

$$0 = {\mathbf{z}^*}^T\mathbf{w}^h = {\mathbf{z}^*}^T\mathbf{M}\mathbf{z}^h = {\mathbf{z}^*}^T(-\mathbf{M}^T\mathbf{z}^h) = -{\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^*,$$

and also

$$0 = {\mathbf{z}^h}^T\mathbf{w}^* = {\mathbf{z}^h}^T\mathbf{q} + {\mathbf{z}^h}^T\mathbf{M}\mathbf{z}^* + {\mathbf{z}^h}^T\mathbf{1}z_0^* = {\mathbf{z}^h}^T\mathbf{q} + {\mathbf{z}^h}^T\mathbf{1}z_0^* > {\mathbf{z}^h}^T\mathbf{q}$$

since ${\mathbf{z}^h}^T\mathbf{1}$ and $z_0^*$ are positive.

The sixth claim holds with $\mathbf{v} = \mathbf{z}^h$. The seventh claim follows since $\mathbf{w} \geq \mathbf{0}$ implies $\mathbf{v}^T\mathbf{w} \geq 0$, while

$$\mathbf{v}^T\mathbf{w} = \mathbf{v}^T\mathbf{q} + \mathbf{v}^T\mathbf{M}\mathbf{z},$$

the first summand being negative, the second non-positive.

# 5 Mathematical (Non-linear) Programming

In these notes we now focus on the problem

$$\text{minimize } f(\mathbf{x}) \quad \text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{5.1}$$

where $f\colon \mathbf{R}^n \to \mathbf{R}$ and $\mathbf{g}\colon \mathbf{R}^n \to \mathbf{R}^m$ are arbitrary functions; in other words, $\mathbf{g}$ represents $m$ constraints, and both $\mathbf{g}$ and the objective $f$ are allowed to be non-linear functions. This is an example of a *mathematical program*, i.e. a general optimization problem which is possibly non-linear. Notice that, as most authors do, we *minimize* our objective function in the problem's usual form.

We will be interested in a special case of this, where $\mathbf{g}$ is linear and $f$ is quadratic; this is known as *quadratic programming*.

**Example 5.1** Consider our standard LP: max $\mathbf{c}^T\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. The "max $\mathbf{c}^T\mathbf{x}$" is essentially the same as "min $-\mathbf{c}^T\mathbf{x}$," so here $f(\mathbf{x}) = -\mathbf{c}^T\mathbf{x}$. The two sets of constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ can be written as $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ where

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ -\mathbf{x} \end{bmatrix}.$$

In summary, our standard LP is the same as the mathematical program in equation 5.1 with

$$f(\mathbf{x}) = -\mathbf{c}^T\mathbf{x} \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ -\mathbf{x} \end{bmatrix}$$

Note that $f$ and $\mathbf{g}$ are linear functions of $\mathbf{x}$.

**Example 5.2** We wish to invest in a portfolio (i.e. collection) of three stocks. Let $x_1, x_2, x_3$ be the proportion of the investment invested in each the three stocks; we have

$$x_1 + x_2 + x_3 \leq 1 \qquad x_1, x_2, x_3 \geq 0.$$

If the *expected rates of returns* of the stocks are $r_1, r_2, r_3$, then our portfolio, $P$, will have an expected rate of return

$$r_P = r_1 x_1 + r_2 x_2 + r_3 x_3 = \mathbf{r}^T \mathbf{x};$$

our portfolio's *risk*, $\sigma_P$, is given by

$$\sigma_P^2 = \mathbf{x}^T \mathbf{S} \mathbf{x},$$

where

$$\mathbf{S} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

is the "variance-covariance" matrix, which we can assume is positive semidefinite (i.e. $\mathbf{x}^T \mathbf{S} \mathbf{x} \geq 0$ for all $\mathbf{x}$). On problem connected with the portfolio selection problem is to minimize the risk for a given rate of return, $r_0$. This is just

$$\begin{aligned} \min \mathbf{x}^T \mathbf{S} \mathbf{x}, & \quad \text{s.t.} \\ -x_1, -x_2, -x_3 & \leq 0, \\ x_1 + x_2 + x_3 - 1 & \leq 0, \\ r_1 x_1 + r_2 x_2 + r_3 x_3 - r_0 & \leq 0, \\ -(r_1 x_1 + r_2 x_2 + r_3 x_3 - r_0) & \leq 0. \end{aligned}$$

Noticed that we have expressed the constraint $\mathbf{r}^T \mathbf{x} = r_0$ as two inequalities, $\mathbf{r}^T \mathbf{x} - r_0 \leq 0$ and $\mathbf{r}^T \mathbf{x} - r_0 \geq 0$ or $-\mathbf{r}^T \mathbf{x} + r_0 \leq 0$. We get an instance of equation 5.1 with

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{S} \mathbf{x}, \qquad g_i(\mathbf{x}) = -x_i \text{ for } i = 1, 2, 3,$$

$$g_4(\mathbf{x}) = x_1 + x_2 + x_3 - 1, \qquad g_5(\mathbf{x}) = r_1 x_1 + r_2 x_2 + r_3 x_3 - r_0, \qquad g_6(\mathbf{x}) = -g_5(\mathbf{x}).$$

**Example 5.3** We consider the portfolio selection problem above, with $n$ stocks instead of three. We similarly get the problem

$$\min \mathbf{x}^T \mathbf{S} \mathbf{x}, \qquad \text{s.t.}$$

$$-\mathbf{x} \leq \mathbf{0}, \qquad \mathbf{1}^T \mathbf{x} - 1 \leq 0, \qquad \mathbf{r}^T \mathbf{x} - r_0 \leq 0, \qquad -\mathbf{r}^T \mathbf{x} + r_0 \leq 0.$$

We get an instance of equation 5.1 with

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{S} \mathbf{x}, \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} -\mathbf{x} \\ \mathbf{1}^T \mathbf{x} - 1 \\ \mathbf{r}^T \mathbf{x} - r_0 \\ -\mathbf{r}^T \mathbf{x} + r_0 \end{bmatrix}.$$

**Example 5.4** In the previous example, we got $n+3$ constraints for the problem with $n$ stocks. By using substitution with $\mathbf{r}^T\mathbf{x} = r_0$, we may eliminate one variable and get two fewer constraints (at the cost of complicating things slightly).

# 6 The Karush-Kuhn-Tucker Conditions

Consider the mathematical program

$$\text{minimize } f(\mathbf{x}), \quad \text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \tag{6.1}$$

We say that $\mathbf{x_0}$ is a *(constrained) local minimum* for the above program if $f(\mathbf{x_0}) \leq f(\mathbf{x})$ for all $\mathbf{x}$ sufficiently close to $\mathbf{x_0}$ satisfying $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. Assume that $f$ and the $g_i$ are differentiable. One can easily show (and we shall do so in the next section):

**Theorem 6.1** *Let $\mathbf{x_0}$ be a local minimum of equation 6.1. Then there exist non-negative $u_0, \ldots, u_m$ not all zero such that*

$$u_0\nabla f(\mathbf{x_0}) + u_1\nabla g_1(\mathbf{x_0}) + \cdots + u_m\nabla g_m(\mathbf{x_0}) = \mathbf{0} \tag{6.2}$$

*and such that for each $i$ with $g_i(\mathbf{x_0}) < 0$, we have $u_i = 0$.*

For any feasible $\mathbf{x}$, we say that $g_i$ is *active at* $\mathbf{x}$ if $g_i(\mathbf{x}) = 0$. The inactive constraints are not relevant for local considerations. It is not surprizing, therefore, to find that the gradients of the inactive constraints don't play a role in equation 6.2.

**Remark 6.2** It is also true that if we replace some of the $g_i(\mathbf{x}) \leq 0$ constrainsts by equality constraints $g_i(\mathbf{x}) = 0$, then the same theorem holds except that those corresponding $u_i$ can also be negative. This generalized form of theorem 6.1 clearly includes the classical theory of Lagrange multipliers.

Now we consider equation 6.2 more carefully. If $u_0 = 0$ in this equation, then the equation does not involve $f$. Consequently the equation says nothing about $f$; rather, it tells us something about the constraints alone. It turns out that in many important cases, we will know that we can take $u_0 \neq 0$; therefore we can assume $u_0 = 1$.

**Definition 6.3** *Any feasible point, $\mathbf{x_0}$ (a local minimum or not) satisfying equation 6.2 with $u_0 = 1$ (and $u_i \geq 0$ with equality at the inactive constraints) is called a* Karush-Kuhn-Tucker (KKT) point*.*

Being a KKT point is therefore equivalent to the *KKT conditions*

$$\mathbf{g}(\mathbf{x_0}) \leq \mathbf{0}, \quad \nabla f(\mathbf{x_0}) + \mathbf{u}^T\nabla\mathbf{g}(\mathbf{x_0}) = \mathbf{0} \text{ for a } \mathbf{u} \geq \mathbf{0}, \quad \mathbf{u}\,\mathbf{g}(\mathbf{x_0}) = \mathbf{0}.$$

Notice that by our conventions, $\nabla f(\mathbf{x_0})$ is a row vector; since $\mathbf{g}$ is a column vector, this means that $\nabla\mathbf{g}(\mathbf{x_0})$ is a matrix whose $i$-th row is $\nabla g_i(\mathbf{x_0})$.

In the next section we will outline a proof of the following theorem.

**Theorem 6.4** *If $\mathbf{g}(\mathbf{x})$ depends linearly on $\mathbf{x}$ (i.e. $\mathbf{g}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$), then a local minimum of equation 6.1 must be a KKT point. The same conclusion holds if the $g_i(\mathbf{x})$ are any convex functions such that there exists a feasible point at which all constraints are inactive. The same conclusion holds if for any feasible $\mathbf{x}$, the $\nabla g_i(\mathbf{x})$ of the active $g_i$ are linearly independent.*

In this note it is the case of linear constraints that is of interest in the examples we present. Hence the above theorem suffices for our needs.

**Example 6.5** Our standard linear program is equivalent to

$$f(\mathbf{x}) = -\mathbf{c}^T\mathbf{x}, \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ -\mathbf{x} \end{bmatrix}$$

The three KKT conditions are:

1. $\mathbf{g}(\mathbf{x_0}) \leq \mathbf{0}$, which amounts to feasibility of $\mathbf{x_0}$;
2. $\nabla f(\mathbf{x_0}) + \mathbf{u}\nabla\mathbf{g}(\mathbf{x_0}) = \mathbf{0}$, which is just

   $$-\mathbf{c}^T + \mathbf{u}^T \begin{bmatrix} \mathbf{A} \\ -\mathbf{I} \end{bmatrix} = \mathbf{0}$$

   where $\mathbf{I}$ is the identity matrix; writing $\mathbf{u}^T = [\mathbf{u_d}^T | \mathbf{u_s}^T]$ and taking transposes yields the equivalent

   $$\mathbf{u_s} = -\mathbf{c} + \mathbf{A}^T\mathbf{u_d},$$

   which is just the dual equations; the condition $\mathbf{u} \geq \mathbf{0}$ is just dual feasibility;
3. $\mathbf{u}\,\mathbf{g}(\mathbf{x_0}) = \mathbf{0}$, which amounts to

   $$\mathbf{u_d}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0} \qquad \text{and} \qquad \mathbf{u_s}\mathbf{x} = \mathbf{0},$$

   which is just complementary slackness.

Hence the KKT conditions for a linear program is exactly complementary slackness.

**Example 6.6** Now let us add a quadratic term to $f(\mathbf{x})$ in the last example, namely we take:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{S}\mathbf{x} - \mathbf{c}^T\mathbf{x}, \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ -\mathbf{x} \end{bmatrix}$$

There is no loss in generality in assuming that $\mathbf{S}$ is symmetric (i.e. $\mathbf{S} = \mathbf{S}^T$). Only the second KKT condition is modified; it now reads

$$2\mathbf{x}^T\mathbf{S} - \mathbf{c}^T + \mathbf{u}^T \begin{bmatrix} \mathbf{A} \\ -\mathbf{I} \end{bmatrix} = \mathbf{0}.$$

This yields the modified equation for the "dual slack" variables

$$\mathbf{u_s} = -\mathbf{c} + 2\mathbf{S}\mathbf{x} + \mathbf{A}^T\mathbf{u_d}.$$

**Example 6.7** Consider the program for $\mathbf{x} \in \mathbf{R}^2$

$$\text{maximize } x_1 \qquad \text{s.t. } 0 \le x_2 \le -x_1^3,$$

in other words

$$f(\mathbf{x}) = -x_1, \qquad g_1(\mathbf{x}) = -x_2, \qquad g_2(\mathbf{x}) = x_2 + x_1^3.$$

It is easy to see that $\mathbf{0}$ is the unique global minimum of this program. However, $\mathbf{0}$ is not a KKT point since

$$\nabla f(\mathbf{0}) = [-1\ 0], \qquad \nabla g_1(\mathbf{0}) = [0\ -1], \qquad \nabla g_2(\mathbf{0}) = [0\ 1].$$

Of course, the curves $g_1(\mathbf{x}) = 0$ and $g_2(\mathbf{x}) = 0$ intersect "very badly" (in a cusp) at $\mathbf{x} = \mathbf{0}$.

**Example 6.8** Consider a program where $g_2(\mathbf{x}) = -g_1(\mathbf{x})$; in other words, one of your constraints is the equality $g_1(\mathbf{x}) = 0$, which you reduce to two inequalities: $g_1(\mathbf{x}) \le 0$ and $-g_1(\mathbf{x}) \le 0$. Then every point satisfies equation 6.2 with $u_1 = u_2 = 1$ and the other $u_i$'s zero (since $\nabla g_2 = -\nabla g_1$). So equation 6.2 has very little content in this case, and it is only when we insist on $u_0 \ne 0$ that we get something interesting.

Note that in this case equation 6.2 may give us something interesting if we left the equality $g_1(\mathbf{x}) = 0$ as an equality, forgot about $g_2$, and used remark 6.2; then $u_2 \nabla g_2(\mathbf{x_0})$ would disappear from equation 6.2 but $u_1$ would be allowed to be any real value. Then there would be no trivial $\mathbf{u} \ne \mathbf{0}$ satisfying equation 6.2.

This last example goes to show that sometimes it is better not to write an equality as two inequalities.

# 7 More on the Karush-Kuhn-Tucker Conditions

In this section we indicate the proofs of the results of the previous section.

Theorem 6.1 can be proven in two simple steps:

**Proposition 7.1** *Let $\mathbf{x_0}$ be a local minimum of equation 6.1. There can exist no $\mathbf{y} \in \mathbf{R}^p$ such that $\mathbf{y}^T \nabla f(\mathbf{x_0}) < 0$ and $\mathbf{y}^T \nabla g_i(\mathbf{x_0}) < 0$ for those $i$ satisfying $g_i(\mathbf{x_0}) = 0$.*

**Proof** Calculus shows that for small $\epsilon > 0$ we have $\mathbf{x_0} + \epsilon \mathbf{y}$ is feasible, and $f(\mathbf{x_0} + \epsilon \mathbf{y}) < f(\mathbf{x_0})$, which is impossible.

□

**Proposition 7.2** *Let $\mathbf{v}_0, \ldots, \mathbf{v}_m \in \mathbf{R}^p$ be such that there exists no $\mathbf{y} \in \mathbf{R}^p$ with $\mathbf{y}^T \mathbf{v}_i < 0$ for all $i$. Then there is a non-zero $\mathbf{u} \ge \mathbf{0}$ with*

$$u_0 \mathbf{v}_0 + \cdots + u_m \mathbf{v}_m = \mathbf{0}.$$

**Proof** Consider the linear program: maximize 0 (yes 0...) subject to $\mathbf{y}^T\mathbf{v}_i \leq -1$ for all $i$, viewed as a linear program in $\mathbf{y}$ with $\mathbf{v}_i$ given. This LP cannot be feasible. Putting it into standard form by introducing variables $\mathbf{x}, \mathbf{z} \geq \mathbf{0}$ with $\mathbf{y} = \mathbf{x} - \mathbf{z}$, this is the LP: maximize 0 subject to $\mathbf{x}^T\mathbf{v}_i - \mathbf{z}^T\mathbf{v}_i \leq -1$. Its dual is minimize $-u_0 - \cdots - u_m$ subject to

$$\begin{bmatrix} \mathbf{v}_0 & \cdots & \mathbf{v}_m \\ -\mathbf{v}_0 & \cdots & -\mathbf{v}_m \end{bmatrix} \mathbf{u} \geq \mathbf{0}, \qquad \mathbf{u} \geq \mathbf{0}.$$

The above conditions on $\mathbf{u}$ are the same as

$$u_0\mathbf{v}_0 + \cdots + u_m\mathbf{v}_m = \mathbf{0}, \qquad \mathbf{u} \geq \mathbf{0}.$$

Since the dual is feasible (for we can take $\mathbf{u} = \mathbf{0}$), and since the primal is infeasible, it must be the case that the dual is unbounded. Hence there is a $\mathbf{u}$ satisfying the above conditions with $-u_0 - \cdots - u_m$ as small as we like; taking $-u_0 - \cdots - u_m$ to be any negative number produced a $\mathbf{u} \neq \mathbf{0}$ with the desired properties.

$\square$

The remark about replacing some constraints by equalities follows from the implicit function theorem. Namely, if the $g_i(\mathbf{x}) = 0$ constraints have linearly dependent gradients at $\mathbf{x_0}$, then the desired equation is trivially satisfied. However, if they are linearly independent, then we can apply the implicit function theorem to get a "nice" level set of the equality constraints near $\mathbf{x_0}$ and then apply the above two propositions on this level set.

An analogue of proposition 7.2 offers strong insight into the KKT conditions. Namely we can similarly show:

**Proposition 7.3** *Let $\mathbf{v}_0, \ldots, \mathbf{v}_m \in \mathbf{R}^p$ be such that there exists no $\mathbf{y} \in \mathbf{R}^p$ with $\mathbf{y}^T\mathbf{v}_0 < 0$ and $\mathbf{y}^T\mathbf{v}_i \leq 0$ for all $i \geq 1$. Then there is a non-zero $\mathbf{u} \geq \mathbf{0}$ with*

$$u_0\mathbf{v}_0 + \cdots + u_m\mathbf{v}_m = \mathbf{0} \qquad \text{and} \qquad u_0 > 0.$$

**Corollary 7.4** *A point $\mathbf{x_0}$ is a KKT point iff there exists no $\mathbf{y} \in \mathbf{R}^p$ such that $\mathbf{y}^T\nabla f(\mathbf{x_0}) < 0$ and $\mathbf{y}^T\nabla g_i(\mathbf{x_0}) \leq 0$ for those $i$ satisfying $g_i(\mathbf{x_0}) = 0$.*

To better express the above corollary and to outline a proof of theorem 6.4 we make the following definitions.

**Definition 7.5** *A $\mathbf{y} \in \mathbf{R}^p$ is a* feasible seeming direction *with respect to (a feasible) $\mathbf{x_0}$ if $\mathbf{y}^T\nabla g_i(\mathbf{x_0}) \leq 0$ for those $i$ satisfying $g_i(\mathbf{x_0}) = 0$.*

**Definition 7.6** *Let $\mathbf{c}(t)$ be an $\mathbf{R}^p$ valued function of $t$ defined in a neighbourhood of $t = 0$. We say that $\mathbf{c}$* represents (the direction) $\mathbf{y}$ at $\mathbf{x_0}$, *if $\mathbf{c}(0) = \mathbf{x_0}$ and if $\mathbf{c}$ is differentiable at $t = 0$ and $\mathbf{c}'(0) = \mathbf{y}$.*

So a "feasible seeming direction" is a direction (or a vector), $\mathbf{y}$, such that curves representing this direction seem like they will be feasible, since $g_i(\mathbf{x_0}) \le 0$ for each $i$ and $\mathbf{y}^T \nabla g_i(\mathbf{x_0}) \le 0$ for $i$ associated to active constraints. The problem, however, is that if $\mathbf{y}^T \nabla g_i(\mathbf{x_0}) = 0$ for an active $i$, the representing curves may be infeasible (they are only feasible to "first and second order").

**Definition 7.7** *We say that* $\mathbf{y}$ *can be* feasibly represented *at* $\mathbf{x_0}$ *if there is a curve,* $\mathbf{c}$*, that represents* $\mathbf{y}$ *at* $\mathbf{x_0}$ *and that* $\mathbf{c}(t)$ *is feasible for* $t > 0$*.*

For example, in example 6.7, the direction $[1\ 0]^T$ at $[1\ 0]^T$ is seemingly feasible but not feasibly representable.

Theorem 6.4 follows easily from the following observation:

**Proposition 7.8** *Let* $\mathbf{g}$ *be such that any feasible seeming direction at a feasible point is feasibly representable. Then any local minimum is a KKT point.*

The hypothesis in this proposition is known as the "constraint qualification" of Kuhn and Tucker.

# 8 Convex Programming

A function, $f(\mathbf{x})$ is *convex* if

$$f\Big(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}\Big) \le \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y})$$

for all $0 \le \alpha \le 1$ and $\mathbf{x}, \mathbf{y}$ in the domain[5] of $f$. For a twice differentiable function, $f(x)$, of one variable, this amounts to $f''(x) \ge 0$, and for $f(\mathbf{x})$ of any number of variables this amounts to the Hessian being positive semidefinite.

In particular, a quadratic function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{S}\mathbf{x} - \mathbf{c}^T\mathbf{x}$ is convex iff $\mathbf{x}^T\mathbf{S}\mathbf{x} \ge 0$ for all $\mathbf{x}$. So

$$4x_1^2 + 5x_2^2 + 6x_3^2, \qquad x_1^2 + 3x_1x_2 + 10x_2^2,$$

are convex, quadratic functions, but

$$-x_1^2 + x_2^2, \qquad 4x_1x_2, \qquad x_1^2 + 3x_1x_2 + x_2^2$$

are not.

Quadratic programming becomes hard, in a sense, when the quadratic objective function, $f(\mathbf{x})$, fails to be convex. It is not hard to see why: consider the problem

$$\text{minimize } f(\mathbf{x}) = -x_1^2 - \cdots - x_n^2, \qquad \text{s.t. } -a \le x_i \le b$$

[5] We are therefore assuming that if $\mathbf{x}, \mathbf{y}$ are in the domain of $f$, then so is $\alpha\mathbf{x} + (1-\alpha)\mathbf{y}$, i.e. that the domain of $f$ is convex.

with $0 < a < b$. This program has $2^n$ local minima, namely where each $x_i$ is either $-a$ or $b$. Each of these $2^n$ local minima satisfy the KKT conditions. However, only $\mathbf{x} = [b\ b\ \ldots\ b]^T$ is a global minimum.

If we want to use the KKT conditions to solve a quadratic program, we can most easily do so when *any* KKT point is a global minimum. This works when the objective is convex:

**Theorem 8.1** *Let $f$ and $g_1, \ldots, g_m$ be convex functions. Then any KKT point for*

$$\textit{minimize } f(\mathbf{x}) \quad \textit{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

*is a global minimum for the above program.*

**Proof** (Outline) Let $\mathbf{x_0}$ be a KKT point and let $\mathbf{y}$ be any feasible point. Considering the line $\mathbf{x}(t) = (1-t)\mathbf{x_0} + t\mathbf{y}$ in $t$ it is easy to show that $\mathbf{x}(0)$ is a global minimum for the above mathematical program restricted to $\mathbf{x}(t)$. (In essence we reduce the theorem to the one dimensional case, which is easy.)

# 9 Quadratic Programming

Let us return example 6.6, where we minimize $f(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ where

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{S}\mathbf{x} - \mathbf{c}^T\mathbf{x}, \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ -\mathbf{x} \end{bmatrix}$$

We saw that the KKT conditions amount to: (1) feasibility, namely $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$, (2) a "dual slack" variable equation:

$$\mathbf{u_s} = -\mathbf{c} + 2\mathbf{S}\mathbf{x} + \mathbf{A}^T\mathbf{u_d},$$

and (3) "complementary slackness" type conditions:

$$\mathbf{u_d}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0} \qquad \text{and} \qquad \mathbf{u_s}\mathbf{x} = \mathbf{0}.$$

We may form this as a linear complementarity problem with

$$\mathbf{w} = \mathbf{q} + \mathbf{M}\mathbf{z}, \qquad \mathbf{w}\,\mathbf{z} = \mathbf{0}, \qquad \mathbf{w}, \mathbf{z} \geq \mathbf{0},$$

where we set $\mathbf{x_s} = \mathbf{b} - \mathbf{A}\mathbf{x}$ and

$$\mathbf{w} = \begin{bmatrix} \mathbf{u_s} \\ \mathbf{x_s} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{u_d} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix},$$

$$\text{and} \quad \mathbf{M} = \begin{bmatrix} 2\mathbf{S} & \mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{bmatrix}.$$

As we said before, any solution of this linear complementarity problem will be a KKT point and hence, provided that $f$ is convex, it will be a global minimum. So we turn our attention to the case where $f$ is convex, i.e. where $\mathbf{S}$ (assumed symmetric) is positive semidefinite, i.e. $\mathbf{x}^T\mathbf{S}\mathbf{x} \geq 0$ for all $\mathbf{x}$.

**Proposition 9.1** *If $S$ is positive semidefinite, then $\mathbf{M}$ is positive semidefinite and hence copositive plus.*

**Proof** The $-\mathbf{A}$ and the $\mathbf{A}^T$ in $\mathbf{M}$ cancel in computing $\mathbf{z}^T\mathbf{M}\mathbf{z}$, i.e.

$$\mathbf{z}^T\mathbf{M}\mathbf{z} = \mathbf{z}^T \begin{bmatrix} 2\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{z},$$

and the matrix on the right is clearly positive semidefinite exactly when $\mathbf{S}$ is.

□

We may therefore use the Lemke-Howson algorithm and the KKT conditions to solve any convex quadratic program.

**Example 9.2** Consider the problem of minimizing $f(\mathbf{x}) = (x_1-1)^2+(x_2-2)^2$ subject to $x_1 + x_2 \leq 1$ and the $x_i$'s being non-negative. This is a form of the above quadratic program with

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{c} = [2\ 4]^T, \qquad \mathbf{A} = [1\ 1], \qquad \mathbf{b} = [1].$$

We get the initial dictionary plus auxiliary $z_0$ being

$$\begin{aligned} u_1 &= -2 + 2x_1 + u_3 + z_0 \\ u_2 &= -4 + 2x_2 + u_3 + z_0 \\ x_3 &= 1 - x_1 - x_2 + z_0 \end{aligned}$$

We get $z_0$ enters and $u_2$ leaves, yielding:

$$\begin{aligned} u_1 &= 2 + 2x_1 - 2x_2 + u_2 \\ z_0 &= 4 - 2x_2 - u_3 + u_2 \\ x_3 &= 5 - x_1 - 3x_2 - u_3 + u_2 \end{aligned}$$

Then $x_2$ enters and $u_1$ leaves, yielding

$$\begin{aligned} x_2 &= 1 + x_1 - (1/2)u_1 + (1/2)u_2 \\ z_0 &= 2 - 2x_1 + u_1 - u_3 \\ x_3 &= 2 - 4x_1 + (3/2)u_1 - u_3 - (1/2)u_2 \end{aligned}$$

The $x_1$ enters and $x_3$ leaves, yielding

$$\begin{aligned} z_0 &= 1 + (1/2)x_3 + (1/4)u_1 + (1/4)u_2 - (1/2)u_3 \\ x_1 &= (1/2) - (1/4)x_3 + (3/8)u_1 - (1/8)u_2 - (1/4)u_3 \\ x_2 &= (3/2) - (1/4)x_3 - (1/8)u_1 + (3/8)u_2 - (1/4)u_3 \end{aligned}$$

Finally $u_3$ enters and $z_0$ leaves, yielding

$$\begin{aligned} x_1 &= 0 + (1/2)z_0 - (1/2)x_3 + (1/4)u_1 - (1/4)u_2 \\ x_2 &= 1 + (1/2)z_0 - (1/2)x_3 - (1/4)u_1 + (1/4)u_2 \\ u_3 &= 2 - 2z_0 + x_3 + (1/2)u_1 + (1/2)u_2 \end{aligned}$$

We see that the optimal solution is $(x_1, x_2) = (0, 1)$.

# 10 General Duality Theory

The KKT conditions that part of duality theory may carry over to any mathematical program. This is indeed true, and we will give a very brief introduction as to how this is done.

We consider, as usual, the mathematical program

$$\text{minimize } f(\mathbf{x}) \quad \text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}.$$

We define the *Lagrangian*, $L(\mathbf{u})$, as the function[6]

$$L(\mathbf{u}) = \min_{\mathbf{x} \in \mathbf{R}^n} L(\mathbf{x}, \mathbf{u}) \qquad \text{where} \quad L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \mathbf{u}^T \mathbf{g}(\mathbf{x}).$$

The *dual problem* becomes

$$\text{maximize } L(\mathbf{u}), \text{ s.t. } \mathbf{u} \geq \mathbf{0}.$$

It is easy to see that if the maximum of the dual problem is $d$, and the minimum of the original (primal) mathematical program is $v$, then $d \leq v$. $v - d$ is referred to as the *duality gap*.

The duality theory of linear programming is generalized by the above duality theory, as is easy to check. For feasible linear programs, the duality gap is zero. The propositions stated below give a further indication of how the above duality theory resembles that in linear programming.

Recall that $\mathbf{x}$ is *feasible* if $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$; we say that $\mathbf{u}$ is *feasible* if $\mathbf{u} \geq \mathbf{0}$.

[6] Sometimes authors will also restrict $\mathbf{x}$ to lie in a set $X \subset \mathbf{R}^n$, in addition to the constraints on $\mathbf{x}$ placed by $\mathbf{g}$. In this case all of duality theory works, with slight modifications. For example, $L(\mathbf{u})$ would be defined as the minimum of $L(\mathbf{x}, \mathbf{u})$ with $\mathbf{x} \in X$.

**Proposition 10.1** *If for some feasible* $\mathbf{u}^*$ *and* $\mathbf{x}^*$ *we have* $L(\mathbf{u}^*) = f(\mathbf{x}^*)$*, then* $\mathbf{x}^*$ *is an optimal solution.*

**Proposition 10.2** *If feasible* $\mathbf{u}^*$ *and* $\mathbf{x}^*$ *satisfy* $f(\mathbf{x}^*) + \mathbf{u}^{*T}\mathbf{g}(\mathbf{x}^*) = L(\mathbf{u}^*)$ *and* $\mathbf{u}^*\,\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$*, then* $\mathbf{x}^*$ *is an optimal solution.*

In both of these propositions, the hypotheses imply that the duality gap is zero. Furthermore, that fact that the duality gap is zero is, in a sense, what makes these propositions work.

# 11 Exercises

**Exercise 1** Convince yourself that the museum principle is true. [Hint: assume that you cylce: consider the first room you visit twice.]

**Exercise 2** Consider the infeasible linear program

$$\max x_1 + 2x_2, \qquad \text{s.t. } x_1 + 2x_2 \leq -1, \qquad \text{and } x_1, x_2 \geq 0.$$

Use complementary slackness to write down a linear complementarity version of this problem. Use the Lemke-Howson algorithm to show that the linear complementarity problem has no feasible solution.

**Exercise 3** Consider the problem

$$\max x_1 + x_2, \qquad \text{s.t. } -x_1 + x_2 \leq 0, \qquad \text{and } x_1, x_2 \geq 0.$$

Use complementary slackness to write down a linear complementarity version of this problem. Perform the Lemke-Howson algorithm, using the perturbation method to avoid degeneracies.

**Exercise 4** Minimize $(x_1 - 4)^2 + (x_2 - 4.5)^2$ subject to the $x_i$ being non-negative and $x_1 + x_2 \leq 1$, using the KKT conditions and the Lemke-Howson algorithm.

# 12 Answers to the Exercises

**Solution 1** Let $R$ be the room you first visit twice. $R$ must be a "continue" room (or you would have stopped when you first visited it), and so it has at most two doors. We claim all these doors are used during the first visit to $R$. Indeed, if $R$ was the initial room, then it had only one door (and you used this door). If $R$ was not the initial door, then you used one door to enter it, and a different door to leave it.

Now let $S$ be the room you visit just before you visit $R$ for the second time. Your door of entry from $S$ to $R$ must be a new (never used) door, since $S$ has never been visited before. But this contradicts the claim in the previous paragraph.

Hence we cannot cycle, i.e. we cannot visit a room more than once. Hence our visiting stops after a finite number of steps (i.e. room visits).

**Solution 2** We have primal and dual (relabelled) dictionaries:

$$x_3 = -1 - x_1 - 2x_2, \qquad \begin{aligned} u_1 &= \ -1 + u_3, \\ u_2 &= \ -2 + 2u_3. \end{aligned}$$

We add the auxiliary variable $z_0$:

$$\begin{aligned} u_1 &= -1 + z_0 + u_3, \\ u_2 &= -2 + z_0 + 2u_3, \\ x_3 &= -1 + z_0 - x_1 - 2x_2. \end{aligned}$$

So $z_0$ enters and $u_2$ leaves:

$$\begin{aligned} u_1 &= 1 - u_3 + u_2, \\ z_0 &= 2 - 2u_3 + u_2, \\ x_3 &= 1 - x_1 - 2x_2 - 2u_3 + u_2. \end{aligned}$$

Since $u_2$ left previously, $x_2$ now enters and hence $x_3$ leaves:

$$\begin{aligned} u_1 &= 1 - u_3 + u_2, \\ z_0 &= 2 - 2u_3 + u_2, \\ x_2 &= (1/2) - (1/2)x_1 - (1/2)x_3 - u_3 + (1/2)u_2. \end{aligned}$$

Since $x_3$ left previously, $u_3$ now enters and hence $x_2$ leaves:

$$\begin{aligned} u_1 &= (1/2) + (1/2)x_1 + (1/2)x_3 + x_2 + (1/2)u_2, \\ z_0 &= 1 + x_1 + x_3 + 2x_2, \\ u_3 &= (1/2) - (1/2)x_1 - (1/2)x_3 - x_2 + (1/2)u_2. \end{aligned}$$

Now $u_2$ enters, but no variable leaves. Therefore the problem is infeasible.

**Solution 3** Our initial dictionary is

$$\begin{aligned} u_1 &= -1 - u_3 + z_0 \\ u_2 &= -1 + u_3 + z_0 \\ x_3 &= 0 + x_1 - x_2 + z_0 \end{aligned}$$

We have two degeneracies here— first of all, the constant in the $x_3$ row is zero; second, the constants in the $u_1, u_2$ rows are both $-1$, and when $z_0$ enters there will be a tie for which variable leaves. So to be safe we add $\epsilon$'s to the dictionary:

$$\begin{aligned} u_1 &= -1 + \epsilon - u_3 + z_0 \\ u_2 &= -1 + \epsilon^2 + u_3 + z_0 \\ x_3 &= \epsilon^3 + x_1 - x_2 + z_0 \end{aligned}$$

So as $z_0$ enters, $u_2$ leaves, yielding

$$\begin{aligned} u_1 &= \epsilon - \epsilon^2 - 2u_3 + u_2 \\ z_0 &= 1 - \epsilon^2 - u_3 + u_2 \\ x_3 &= 1 - \epsilon^2 + \epsilon^3 + x_1 - x_2 - u_3 + u_2 \end{aligned}$$

So $x_2$ enters, and $x_3$ leaves, yielding

$$\begin{aligned} u_1 &= \epsilon - \epsilon^2 - 2u_3 + u_2 \\ z_0 &= 1 - \epsilon^2 - u_3 + u_2 \\ x_2 &= 1 - \epsilon^2 + \epsilon^3 + x_1 - x_3 - u_3 + u_2 \end{aligned}$$

Now $u_3$ enters, and $u_1$ leaves, yielding

$$\begin{aligned} u_3 &= (\epsilon - \epsilon^2)/2 - (1/2)u_1 + (1/2)u_2 \\ z_0 &= 1 - (\epsilon/2) - (\epsilon^2/2) + (1/2)u_1 + (1/2)u_2 \\ x_2 &= 1 - (\epsilon/2) - (\epsilon^2/2) + \epsilon^3 + (1/2)u_1 + (1/2)u_2 + x_1 - x_3 \end{aligned}$$

Then $x_1$ enters but nothing leaves; we conclude that the complementarity problem is infeasible (and so either the original primal or dual problem is infeasible; in this case it is clearly the dual, since the primal is clearly unbounded).

**Solution 4** This is a form of the above quadratic program with

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{c} = [8\ 9]^T, \qquad \mathbf{A} = [1\ 1], \qquad \mathbf{b} = [1].$$

We get the initial dictionary plus auxiliary $z_0$ being

$$\begin{aligned} u_1 &= -8 + 2x_1 + u_3 + z_0 \\ u_2 &= -9 + 2x_2 + u_3 + z_0 \\ x_3 &= 1 - x_1 - x_2 + z_0 \end{aligned}$$

We get $z_0$ enters and $u_2$ leaves, yielding:

$$\begin{aligned} u_1 &= 1 + 2x_1 - 2x_2 + u_2 \\ z_0 &= 9 - 2x_2 - u_3 + u_2 \\ x_3 &= 10 - x_1 - 3x_2 - u_3 + u_2 \end{aligned}$$

Then $x_2$ enters and $u_1$ leaves, yielding

$$\begin{aligned}
x_2 &= (1/2) + x_1 - (1/2)u_1 + (1/2)u_2 \\
z_0 &= 8 - 2x_1 + u_1 - u_3 \\
x_3 &= (17/2) - 4x_1 + (3/2)u_1 - u_3 - (1/2)u_2
\end{aligned}$$

The $x_1$ enters and $x_3$ leaves, yielding

$$\begin{aligned}
z_0 &= (15/4) + (1/2)x_3 + (1/4)u_1 + (1/4)u_2 - (1/2)u_3 \\
x_1 &= (17/4) - (1/4)x_3 + (3/8)u_1 - (1/8)u_2 - (1/4)u_3 \\
x_2 &= (21/4) - (1/4)x_3 - (1/8)u_1 + (3/8)u_2 - (1/4)u_3
\end{aligned}$$

Finally $u_3$ enters and $z_0$ leaves, yielding

$$\begin{aligned}
x_1 &= (1/4) + (1/2)z_0 - (1/2)x_3 + (1/4)u_1 - (1/4)u_2 \\
x_2 &= (3/4) + (1/2)z_0 - (1/2)x_3 - (1/4)u_1 + (1/4)u_2 \\
u_3 &= (15/2) - 2z_0 + x_3 + (1/2)u_1 + (1/2)u_2
\end{aligned}$$

We see that the optimal solution is $(x_1, x_2) = (1/4, 3/4)$.