Chapter 4

Convex Optimization

4.1 Introduction

4.1.1 Mathematical Optimization

The problem of mathematical optimization is to minimize a non-linear cost function $f_0(x)$ subject to inequality constraints $f_i(x) \leq 0, i = 1, \ldots, m$ and equality constraints $h_i(x) = 0, i = 1, \ldots, p$. $x = (x_1, \ldots, x_n)$ is a vector of variables involved in the optimization problem. The general framework of a non-linear optimization problem is outlined in (4.1).

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p \\ \text{variable } x = (x_1, \ldots, x_n) & & \end{array} \tag{4.1}$$

It is obviously very useful and arises throughout engineering, statistics, estimation and numerical analysis. In fact there is the tautology that 'everything is an optimization problem', though the tautology does not convey anything useful. The most important thing to note first is that the optimization problem is extremely hard in general. The solution and method is very much dependent on the property of the objective function as well as properties of the functions involved in the inequality and equality constraints. There are no good methods for solving the general non-linear optimization problem. In practice, you have to make some compromises, which usually translates to finding locally optimial solutions efficiently. But then you get only suboptimial solutions, unless you are willing to do global optimizations, which is for most applications too expensive.

There are important exceptions for which the situation is much better; the global optimum in some cases can be found efficiently and relaibly. Three best known exceptions are

1. least-squares
2. linear programming
3. convex optimization problems - more or less the most general class of problems that can be solved efficiently.

Least squares and linear programming have been around for quite some time and are very special types of convex optimization problems. Convex programming was not appreciated very much until last 15 years. It has drawn attention more recently. In fact many combinatorial optimization problems have been identified to be convex optimization problems. There are also some exceptions besides convex optimization problems, such as singular value decomposition (which corresponds to the problem of finding the best rank-k approximation to a matrix, under the Frobenius norm) *etc.*, which has an exact global solution.

We will first introduce some general optimization principles. We will subsequently motivate the specific class of optimization problems called convex optimization problems and define convex sets and functions. Next, the theory of lagrange multipliers will be motivated and duality theory will be introduced. As two specific and well-studied examples of convex optimization, techniques for least squares and linear programming will be discussed to contrast them against generic convex optimization. Finally, we will dive into techniques for solving general convex optimization problems.

4.1.2 Some Topological Concepts in $\Re^n$

The definitions of some basic topological concepts in $\Re^n$ could be helpful in the discussions that follow.

Definition 12 [Balls in $\Re^n$]: *Consider a point $\mathbf{x} \in \Re^n$. Then the closed ball around $\mathbf{x}$ of radius ϵ is defined as*

$$\mathcal{B}[\mathbf{x}, \epsilon] = \{\mathbf{y} \in \Re^n | ||\mathbf{y} - \mathbf{x}|| \leq \epsilon\}$$

Likewise, the open ball around $\mathbf{x}$ of radius ϵ is defined as

$$\mathcal{B}(\mathbf{x}, \epsilon) = \{\mathbf{y} \in \Re^n | ||\mathbf{y} - \mathbf{x}|| < \epsilon\}$$

For the 1-D case, open and closed balls degenerate to open and closed intervals respectively.

Definition 13 [Boundedness in $\Re^n$]: *We say that a set $\mathcal{S} \subset \Re^n$ is bounded when there exists an $\epsilon > 0$ such that $\mathcal{S} \subseteq \mathcal{B}[0, \epsilon]$.*

In other words, a set $\mathcal{S} \subseteq \Re^n$ is bounded means that there exists a number $\epsilon > 0$ such that for all $\mathbf{x} \in \mathcal{S}$, $||\mathbf{x}|| \leq \epsilon$.

Definition 14 [Interior and Boundary points]: *A point $\mathbf{x}$ is called an interior point of a set $\mathcal{S}$ if there exists an $\epsilon > 0$ such that $\mathcal{B}(\mathbf{x}, \epsilon) \subseteq \mathcal{S}$.*

In other words, a point $\mathbf{x} \in \mathcal{S}$ is called an interior point of a set $\mathcal{S}$ if there exists an open ball of non-zero radius around $\mathbf{x}$ such that the ball is completely contained within $\mathcal{S}$.

Definition 15 [Interior of a set]: *Let $\mathcal{S} \subseteq \Re^n$. The set of all points lying in the interior of $\mathcal{S}$ is denoted by $int(\mathcal{S})$ and is called the interior of $\mathcal{S}$. That is,*

$$int(\mathcal{S}) = \{\mathbf{x} | \exists \epsilon > 0 \ s.t. \ \mathcal{B}(\mathbf{x}, \epsilon) \subset \mathcal{S}\}$$

In the $1-$D case, the open interval obtained by excluding endpoints from an interval $\mathcal{I}$ is the interior of $\mathcal{I}$, denoted by $int(\mathcal{I})$. For example, $int([a, b]) = (a, b)$ and $int([0, \infty)) = (0, \infty)$.

Definition 16 [Boundary of a set]: *Let $\mathcal{S} \subseteq \Re^n$. The boundary of $\mathcal{S}$, denoted by $bnd(\mathcal{S})$ is defined as*

$$bnd(\mathcal{S}) = \left\{\mathbf{y} | \forall \ \epsilon > 0, \ \mathcal{B}(\mathbf{y}, \epsilon) \cap \mathcal{S} \neq \emptyset \text{ and } \mathcal{B}(\mathbf{y}, \epsilon) \cap \mathcal{S}^C \neq \emptyset\right\}$$

For example, $bnd([a, b]) = \{a, b\}$.

Definition 17 [Open Set]: *Let $\mathcal{S} \subseteq \Re^n$. We say that $\mathcal{S}$ is an open set when, for every $\mathbf{x} \in \mathcal{S}$, there exists an $\epsilon > 0$ such that $\mathcal{B}(\mathbf{x}, \epsilon) \subset \mathcal{S}$.*

The simplest examples of an open set are the open ball, the empty set $\emptyset$ and $\Re^n$. Further, arbitrary union of opens sets is open. Also, finite intersection of open sets is open. The interior of any set is always open. It can be proved that a set $\mathcal{S}$ is open if and only if $int(\mathcal{S}) = \mathcal{S}$.

The complement of an open set is the closed set.

Definition 18 [Closed Set]: *Let $\mathcal{S} \subseteq \Re^n$. We say that $\mathcal{S}$ is a closed set when $\mathcal{S}^C$ (that is the complement of $\mathcal{S}$) is an open set.*

The closed ball, the empty set $\emptyset$ and $\Re^n$ are three simple examples of closed sets. Arbitrary intersection of closed sets is closed. Furthermore, finite union of closed sets is closed.

Definition 19 [Closure of a Set]: *Let $\mathcal{S} \subseteq \Re^n$. The closure of $\mathcal{S}$, denoted by $closure(\mathcal{S})$ is given by*

$$closure(\mathcal{S}) = \{\mathbf{y} \in \Re^n | \forall \ \epsilon > 0, \mathcal{B}(\mathbf{y}, \epsilon) \cap \mathcal{S} \neq \emptyset\}$$

Loosely speaking, the closure of a set is the smallest closed set containing the set. The closure of a closed set is the set itself. In fact, a set $\mathcal{S}$ is closed if and only if $closure(\mathcal{S}) = \mathcal{S}$. A bounded set can be defined in terms of a closed set; a set $\mathcal{S}$ is bounded if and only if it is contained inside a closed set. A relationship between the interior, boundary and closure of a set $\mathcal{S}$ is $closure(\mathcal{S}) = int(\mathcal{S}) \cup bnd(\mathcal{S})$.

4.1.3 Optimization Principles for Univariate Functions

Maximum and Minimum values of univariate functions

Let f be a function with domain $\mathcal{D}$. Then f has an *absolute maximum* (or global maximum) value at point $c \in \mathcal{D}$ if

$$f(x) \leq f(c),\ \forall x \in \mathcal{D}$$

and an *absolute minimum* (or global minimum) value at $c \in \mathcal{D}$ if

$$f(x) \geq f(c),\ \forall x \in \mathcal{D}$$

If there is an open interval $\mathcal{I}$ containing c in which $f(c) \geq f(x),\ \forall x \in \mathcal{I}$, then we say that $f(c)$ is a *local maximum value* of f. On the other hand, if there is an open interval $\mathcal{I}$ containing c in which $f(c) \leq f(x),\ \forall x \in \mathcal{I}$, then we say that $f(c)$ is a *local minimum value* of f. If $f(c)$ is either a local maximum or local minimum value of f in an open interval $\mathcal{I}$ with $c \in \mathcal{I}$, the $f(c)$ is called a *local extreme value* of f.

The following theorem gives us the first derivative test for local extreme value of f, when f is differentiable at the extremum.

Theorem 39 *If $f(c)$ is a local extreme value and if f is differentiable at $x = c$, then $f'(c) = 0$.*

Proof: Suppose $f(c) \geq f(x)$ for all x in an open interval $\mathcal{I}$ containing c and that $f'(c)$ exists. Then the difference quotient $\frac{f(c+h)-f(c)}{h} \leq 0$ for small $h \geq 0$ (so that $c + h \in \mathcal{I}$). This inequality remains true as $h \to 0$ from the right. In the limit, $f'(c) \leq 0$. Also, the difference quotient $\frac{f(c+h)-f(c)}{h} \geq 0$ for small $h \leq 0$ (so that $c + h \in \mathcal{I}$). This inequality remains true as $h \to 0$ from the left. In the limit, $f'(c) \geq 0$. Since $f'(c) \leq 0$ as well as $f'(c) \geq 0$, we must have $f'(c) = 0$[1]. □

The *extreme value theorem* is one of the most fundamental theorems in calculus concerning continuous functions on closed intervals. It can be stated as:

Theorem 40 *A continuous function $f(x)$ on a closed and bounded interval $[a, b]$ attains a minimum value $f(c)$ for some $c \in [a, b]$ and a maximum value $f(d)$ for some $d \in [a, b]$. That is, a continuous function on a closed, bounded interval attains a minimum and a maximum value.*

We must point out that either or both of the values c and d may be attained at the end points of the interval $[a, b]$. Based on theorem (39), the extreme value theorem can extended as:

Theorem 41 *A continuous function $f(x)$ on a closed and bounded interval $[a, b]$ attains a minimum value $f(c)$ for some $c \in [a, b]$ and a maximum value $f(d)$ for some $d \in [a, b]$. If $a < c < b$ and $f'(c)$ exists, then $f'(c) = 0$. If $a < d < b$ and $f'(d)$ exists, then $f'(d) = 0$.*

[1] By virtue of the *squeeze* or *sandwich theorem*

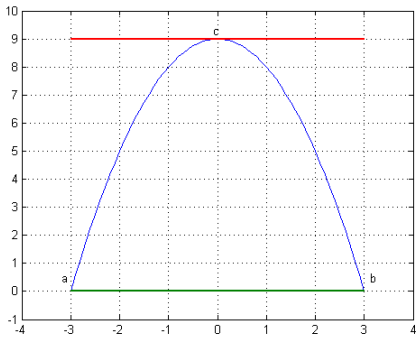

Figure 4.1: Illustration of Rolle's theorem with $f(x) = 9 - x^2$ on the interval $[-3, +3]$. We see that $f'(0) = 0$.

Next, we state the Rolle's theorem.

Theorem 42 *If f is continuous on $[a, b]$ and differentiable at all $x \in (a, b)$ and if $f(a) = f(b)$, then $f'(c) = 0$ for some $c \in (a, b)$.*

Figure 4.1 illustrates Rolle's theorem with an example function $f(x) = 9-x^2$ on the interval $[-3, +3]$.

The *mean value theorem* is a generalization of the Rolle's theorem, though we will use the Rolle's theorem to prove it.

Theorem 43 *If f is continuous on $[a, b]$ and differentiable at all $x \in (a, b)$, then there is some $c \in (a, b)$ such that, $f'(c) = \frac{f(b)-f(a)}{b-a}$.*

Proof: Define $g(x) = f(x) - \frac{f(b)-f(a)}{b-a}(x - a)$ on $[a, b]$. We note rightaway that $g(a) = g(b)$ and $g'(x) = f'(x) - \frac{f(b)-f(a)}{b-a}$. Applying Rolle's theorem on $g(x)$, we know that there exists $c \in (a, b)$ such that $g'(c) = 0$. Which implies that $f'(c) = \frac{f(b)-f(a)}{b-a}$. □

Figure 4.2 illustrates the mean value theorem for $f(x) = 9 - x^2$ on the interval $[-3, 1]$. We observe that the tanget at $x = -1$ is parallel to the secant joining -3 to 1. One could think of the *mean value theorem* as a slanted version of Rolle's theorem. A natural corollary of the mean value theorem is as follows:

Corollary 44 *Let f be continuous on $[a, b]$ and differentiable on (a, b) with $m \leq f'(x) \leq M$, $\forall x \in (a, b)$. Then, $m(x - t) \leq f(x) - f(t) \leq M(x - t)$, if $a \leq t \leq x \leq b$.*

Let $\mathcal{D}$ be the domain of function f. We define

1. the linear approximation of a differentiable function $f(x)$ as $L_a(x) = f(a) + f'(a)(x - a)$ for some $a \in \mathcal{D}$. We note that $L_a(x)$ and its first derivative at a agree with $f(a)$ and $f'(a)$ respectively.

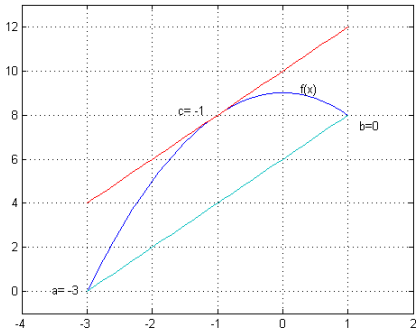

Figure 4.2: Illustration of mean value theorem with $f(x) = 9 - x^2$ on the interval $[-3, 1]$. We see that $f'(-1) = \frac{f(1)-f(-3)}{4}$.

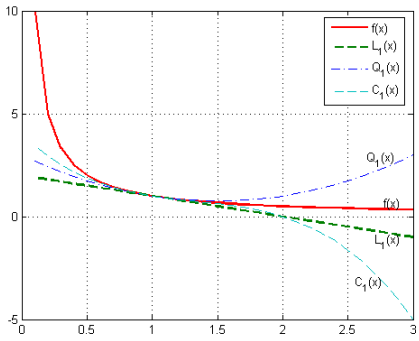

Figure 4.3: Plot of $f(x) = \frac{1}{x}$, and its linear, quadratic and cubic approximations.

2. the quadratic approximatin of a twice differentiable function $f(x)$ as the parabola $Q_a(x) = f(a) + f'(a)(x - a) + \frac{1}{2}f''(a)(x - a)^2$. We note that $Q_a(x)$ and its first and second derivatives at a agree with $f(a)$, $f'(a)$ and $f''(a)$ respectively.

3. the cubic approximation of a thrice differentiable function $f(x)$ is $C_a(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{6}f'''(a)(x-a)^3$. $C_a(x)$ and its first, second and third derivatives at a agree with $f(a)$, $f'(a)$, $f''(a)$ and $f'''(a)$ respectively.

The coefficient[2] of x^2 in $Q_a(x)$ is $\frac{1}{2}f''(a)$. Figure 4.3 illustrates the linear, quadratic and cubic approximations to the function $f(x) = \frac{1}{x}$ with $a = 1$.

[2]The parabola given by $Q_a(x)$ is strictly convex if $f''(a) > 0$ and is strictly concave if $f''(a) < 0$. Strict convexity for functions of single variable will be defined on page 224.

In general, an n^{th} degree polynomial approximation of a function can be found. Such an approximation will be used to prove a generalization of the mean value theorem, called the *Taylor's theorem.*

Theorem 45 *The Taylor's theorem states that if f and its first n derivatives $f', f'', \ldots, f^{(n)}$ are continuous on the closed interval $[a, b]$, and differentiable on (a, b), then there exists a number $c \in (a, b)$ such that*

$$f(b) = f(a)+f'(a)(b-a)+\frac{1}{2!}f''(a)(b-a)^2+\ldots+\frac{1}{n!}f^{(n)}(a)(b-a)^n+\frac{1}{(n+1)!}f^{(n+1)}(c)(b-a)^{n+1}$$

Proof: Define

$$p_n(x) = f(a) + f'(a)(x-a) + \frac{1}{2!}f''(a)(x-a)^2 + \ldots + \frac{1}{n!}f^{(n)}(a)(x-a)^n$$

and

$$\phi_n(x) = p_n(x) + \Gamma(x-a)^{n+1}$$

The polynomials $p_n(x)$ as well as $\phi_n(x)$ and their first n derivatives match f and its first n derivatives at $x = a$. We will choose a value of Γ so that

$$f(b) = p_n(b) + \Gamma(b-a)^{n+1}$$

This requires that $\Gamma = \frac{f(b)-p_n(b)}{(b-a)^{n+1}}$. Define the function $g(x) = f(x) - \phi_n(x)$ that measures the difference between function f and the approximating function $\phi_n(x)$ for each $x \in [a, b]$.

- Since $g(a) = g(b) = 0$ and since g and g' are both continuous on $[a, b]$, we can apply the Rolle's theorem to conclude that there exists $c_1 \in [a, b]$ such that $g'(c_1) = 0$.
- Similarly, since $g'(a) = g'(c_1) = 0$, and since g' and g'' are continuous on $[a, c_1]$, we can apply the Rolle's theorem to conclude that there exists $c_2 \in [a, c_1]$ such that $g''(c_2) = 0$.
- In this way, Rolle's theorem can be applied successively to $g'', g''', \ldots, g^{(n+1)}$ to imply the existence of $c_i \in (a, c_{i-1})$ such that $g^{(i)}(c_i) = 0$ for $i = 3, 4, \ldots, n+1$. Note however that $g^{(n+1)}(x) = f^{(n+1)}(x) - 0 - (n+1)!\Gamma$ which gives us another representation 'of Γ as $\frac{f^{(n+1)}(c_{n+1})}{(n+1)!}$.

Thus,

$$f(b) = f(a)+f'(a)(b-a)+\frac{1}{2!}f''(a)(b-a)^2+\ldots+\frac{1}{n!}f^{(n)}(a)(b-a)^n+\frac{f^{(n+1)}(c_{n+1})}{(n+1)!}(x-a)^{n+1}$$

□

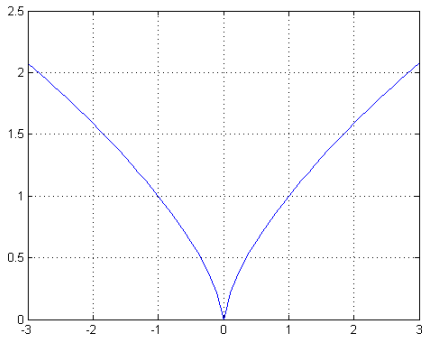

Figure 4.4: The mean value theorem can be violated if $f(x)$ is not differentiable at even a single point of the interval. Illustration on $f(x) = x^{2/3}$ with the interval $[-3, 3]$.

Note that if f fails to be differentiable at even one number in the interval, then the conclusion of the mean value theorem may be false. For example, if $f(x) = x^{2/3}$, then $f'(x) = \frac{2}{3\sqrt[3]{x}}$ and the theorem does not hold in the interval $[-3, 3]$, since f is not differentiable at 0 as can be seen in Figure 4.4.

We will introduce some definitions at this point:

- A function f is said to be *increasing* on an interval $\mathcal{I}$ in its domain $\mathcal{D}$ if $f(t) < f(x)$ whenever $t < x$.
- The function f is said to be *decreasing* on an interval $\mathcal{I} \in \mathcal{D}$ if $f(t) > f(x)$ whenever $t < x$.

These definitions help us derive the following theorem:

Theorem 46 *Let $\mathcal{I}$ be an interval and suppose f is continuous on $\mathcal{I}$ and differentiable on $int(\mathcal{I})$. Then:*

1. *if $f'(x) > 0$ for all $x \in int(\mathcal{I})$, then f is increasing on $\mathcal{I}$;*
2. *if $f'(x) < 0$ for all $x \in int(\mathcal{I})$, then f is decreasing on $\mathcal{I}$;*
3. *if $f'(x) = 0$ for all $x \in int(\mathcal{I})$, iff, f is constant on $\mathcal{I}$.*

Proof: Let $t \in \mathcal{I}$ and $x \in \mathcal{I}$ with $t < x$. By virtue of the mean value theorem, $\exists c \in (t, x)$ such that $f'(c) = \frac{f(x)-f(t)}{x-t}$.

- If $f'(x) > 0$ for all $x \in int(\mathcal{I})$, $f'(c) > 0$, which implies that $f(x) - f(t) > 0$ and we can conclude that f is increasing on $\mathcal{I}$.
- If $f'(x) < 0$ for all $x \in int(\mathcal{I})$, $f'(c) < 0$, which implies that $f(x) - f(t) < 0$ and we can conclude that f is decreasing on $\mathcal{I}$.

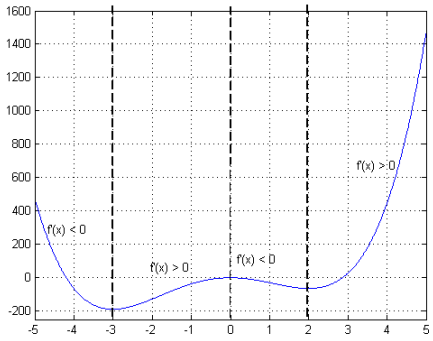

Figure 4.5: Illustration of the increasing and decreasing regions of a function $f(x) = 3x^4 + 4x^3 - 36x^2$

- If $f'(x) = 0$ for all $x \in int(\mathcal{I})$, $f'(c) = 0$, which implies that $f(x) - f(t) = 0$, and since x and t are arbitrary, we can conclude that f is constant on $\mathcal{I}$.

□

Figure 4.5 illustrates the intervals in $(-\infty, \infty)$ on which the function $f(x) = 3x^4 + 4x^3 - 36x^2$ is decreasing and increasing. First we note that $f(x)$ is differentiable everywhere on $(-\infty, \infty)$ and compute $f'(x) = 12(x^3 + x^2 - 6x) = 12(x-2)(x+3)x$, which is negative in the intervals $(-\infty, -3]$ and $[0, 2]$ and positive in the intervals $[-3, 0]$ and $[2, \infty)$. We observe that f is decreasing in the intervals $(-\infty, -3]$ and $[0, 2]$ and while it is increasing in the intervals $[-3, 0]$ and $[2, \infty)$.

There is a related sufficient condition for a function f to be increasing/decreasing on an interval $\mathcal{I}$, stated through the following theorem:

Theorem 47 *Let $\mathcal{I}$ be an interval and suppose f is continuous on $\mathcal{I}$ and differentiable on $int(\mathcal{I})$. Then:*

1. *if $f'(x) \geq 0$ for all $x \in int(\mathcal{I})$, and if $f'(x) = 0$ at only finitely many $x \in \mathcal{I}$, then f is increasing on $\mathcal{I}$;*

2. *if $f'(x) \leq 0$ for all $x \in int(\mathcal{I})$, and if $f'(x) = 0$ at only finitely many $x \in \mathcal{I}$, then f is decreasing on $\mathcal{I}$.*

For example, the derivative of the function $f(x) = 6x^5 - 15x^4 + 10x^3$ vanishes at 0, and 1 and $f'(x) > 0$ elsewhere. So $f(x)$ is increasing on $(-\infty, \infty)$.

Are the sufficient conditions for increasing and decreasing properties of $f(x)$ in theorem 46 also necesssary? It turns out that it is not the case. Figure 4.6 shows that for the function $f(x) = x^5$, though $f(x)$ is increasing in $(-\infty, \infty)$, $f'(0) = 0$.

In fact, we have a slightly different necessary condition for an increasing or decreasing function.

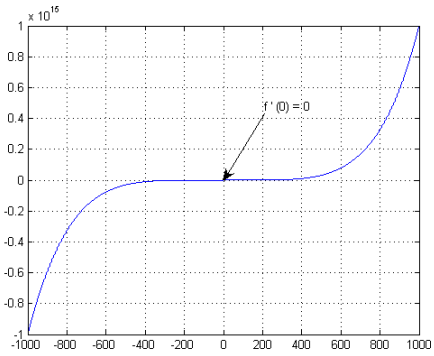

Figure 4.6: Plot of $f(x) = x^5$, illustrating that though the function is increasing on $(-\infty, \infty)$, $f'(0) = 0$.

Theorem 48 *Let $\mathcal{I}$ be an interval, and suppose f is continuous on $\mathcal{I}$ and differentiable in $int(\mathcal{I})$. Then:*

1. *if f is increasing on $\mathcal{I}$, then $f'(x) \geq 0$ for all $x \in int(\mathcal{I})$;*

2. *if f is decreasing on $\mathcal{I}$, then $f'(x) \leq 0$ for all $x \in int(\mathcal{I})$.*

Proof: Suppose f is increasing on $\mathcal{I}$, and let $x \in int(\mathcal{I})$. Them $\frac{f(x+h)-f(x)}{h} > 0$ for all h such that $x+h \in int(\mathcal{I})$. This implies that $f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h} \geq 0$. For the case when f is decreasing on $\mathcal{I}$, it can be similarly proved that $f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h} \leq 0$. □

Next, we define the concept of *critical number*, which will help us derive the general condition for local extrema.

Definition 20 [Critical number]: *A number c in the domain $\mathcal{D}$ of f is called a critical number of f if either $f'(c) = 0$ or $f'(c)$ does not exist.*

The general condition for local extrema is stated in the next theorem; it extends the result in theorem 39 to general non-differentiable functions.

Theorem 49 *If $f(c)$ is a local extreme value, then c is a critical number of f.*

That the converse of theorem 49 does not hold is illustrated in Figure 4.6; 0 is a critical number ($f'(0) = 0$), although $f(0)$ is not a local extreme value. Then, given a critical number c, how do we discern whether $f(c)$ is a local extreme value? This can be answered using the *first derivative test*:

Procedure 1 [First derivative test]: *Let c be an isolated critical number of f. Then,*

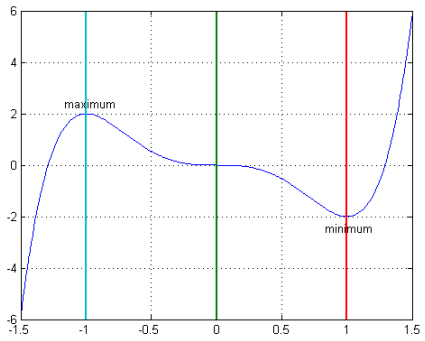

Figure 4.7: Example illustrating the derivative test for function $f(x) = 3x^5 - 5x^3$.

1. *$f(c)$ is a local minimum if $f(x)$ is decreasing in an interval $[c-\epsilon_1, c]$ and increasing in an interval $[c, c+\epsilon_2]$ with $\epsilon_1, \epsilon_2 > 0$, or (but not equivalently), the sign of $f'(x)$ changes from negative in $[c-\epsilon_1, c]$ to positive in $[c, c+\epsilon_2]$ with $\epsilon_1, \epsilon_2 > 0$.*
2. *$f(c)$ is a local maximum if $f(x)$ is increasing in an interval $[c-\epsilon_1, c]$ and decreasing in an interval $[c, c+\epsilon_2]$ with $\epsilon_1, \epsilon_2 > 0$, or (but not equivalently), the sign of $f'(x)$ changes from positive in $[c-\epsilon_1, c]$ to negative in $[c, c+\epsilon_2]$ with $\epsilon_1, \epsilon_2 > 0$.*
3. *If $f'(x)$ is positive in an interval $[c-\epsilon_1, c]$ and also positive in an interval $[c, c-\epsilon_2]$, or $f'(x)$ is negative in an interval $[c-\epsilon_1, c]$ and also negative in an interval $[c, c-\epsilon_2]$ with $\epsilon_1, \epsilon_2 > 0$, then $f(c)$ is not a local extremum.*

As an example, the function $f(x) = 3x^5 - 5x^3$ has the derivative $f'(x) = 15x^2(x+1)(x-1)$. The critical points are 0, 1 and -1. Of the three, the sign of $f'(x)$ changes at 1 and -1, which are local minimum and maximum respectively. The sign does not change at 0, which is therefore not a local supremum. This is pictorially depicted in Figure 4.7 As another example, consider the function

$$f(x) = \begin{cases} -x & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Then,

$$f'(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x > 0 \end{cases}$$

Note that $f(x)$ is discontinuous at $x = 0$, and therefore $f'(x)$ is not defined at $x = 0$. All numbers $x \geq 0$ are critical numbers. $f(0) = 0$ is a local minimum, whereas $f(x) = 1$ is a local minimum as well as a local maximum $\forall x > 0$.

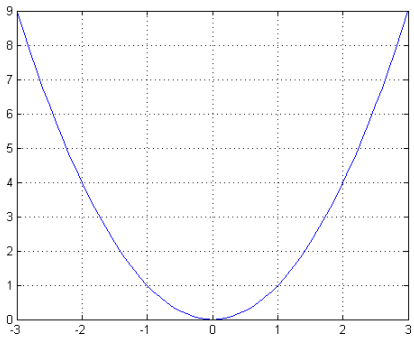

Figure 4.8: Plot for the strictly convex function $f(x) = x^2$ which has $f''(x) = 2 > 0,\ \forall x$.

Strict Convexity and Extremum

We define strictly convex and concave functions as follows:

1. A differentiable function f is said to be *strictly convex* (or *strictly concave up*) on an open interval $\mathcal{I}$, *iff*, $f'(x)$ is increasing on $\mathcal{I}$. Recall from theorem 46, the graphical interpretation of the first derivative $f'(x)$; $f'(x) > 0$ implies that $f(x)$ is increasing at x. Similarly, $f'(x)$ is increasing when $f''(x) > 0$. This gives us a sufficient condition for the strict convexity of a function:

   Theorem 50 *If at all points in an open interval $\mathcal{I}$, $f(x)$ is doubly differentiable and if $f''(x) > 0,\ \forall x \in \mathcal{I}$, then the slope of the function is always increasing with x and the graph is strictly convex. This is illustrated in Figure 4.8.*

   On the other hand, if the function is strictly convex and doubly differentiable in $\mathcal{I}$, then $f''(x) \geq 0,\ \forall x \in \mathcal{I}$.

   There is also a slopeless interpretation of strict convexity as stated in the following theorem:

   Theorem 51 *A differentiable function f is strictly convex on an open interval $\mathcal{I}$, iff*

$$f(ax_1 + (1-a)x_2) < af(x_1) + (1-a)f(x_2) \tag{4.2}$$

   whenver $x_1, x_2 \in \mathcal{I}$, $x_1 \neq x_2$ and $0 < a < 1$.

Proof: First we will prove the necessity. Suppose f' is increasing on $\mathcal{I}$. Let $0 < a < 1$, $x_1, x_2 \in \mathcal{I}$ and $x_1 \neq x_2$. Without loss of generality assume that $x_1 < x_2$[3]. Then, $x_1 < ax_1 + (1-a)x_2 < x_2$ and therefore $ax_1 + (1-a)x_2 \in \mathcal{I}$. By the mean value theorem, there exist s and t with $x_1 < s < ax_1 + (1-a)x_2 < t < x_2$, such that $f(ax_1 + (1-a)x_2) - f(x_1) = f'(s)(x_2 - x_1)(1-a)$ and $f(x_2) - f(ax_1 + (1-a)x_2) = f'(t)(x_2 - x_1)a$. Therefore,

$$\begin{aligned} (1-a)f(x_1) - f(ax_1 + (1-a)x_2) + af(x_2) &= \\ a\left[f(x_2) - f(ax_1 + (1-a)x_2)\right] - (1-a)\left[f(ax_1 + (1-a)x_2) - f(x_1)\right] &= \\ a(1-a)(x_2 - x_1)\left[f'(t) - f'(s)\right] & \end{aligned}$$

Since $f(x)$ is strictly convex on $\mathcal{I}$, $f'(x)$ is increasing $\mathcal{I}$ and therefore, $f'(t) - f'(s) > 0$. Moreover, $x_2 - x_1 > 0$ and $0 < a < 1$. This implies that $(1-a)f(x_1) - f(ax_1 + (1-a)x_2) + af(x_2) > 0$, or equivalently, $f(ax_1 + (1-a)x_2) < af(x_1) + (1-a)f(x_2)$, which is what we wanted to prove in 4.2.

Next, we prove the sufficiency. Suppose the inequality in 4.2 holds. Therefore,

$$\lim_{a \to 0} \frac{f(x_2 + a(x_1 - x_2)) - f(x_2)}{a} \leq f(x_1) - f(x_2)$$

that is,

$$f'(x_2)(x_1 - x_2) \leq f(x_1) - f(x_2) \tag{4.3}$$

Similarly, we can show that

$$f'(x_1)(x_2 - x_1) \leq f(x_2) - f(x_1) \tag{4.4}$$

Adding the left and right hand sides of inequalities in (4.3) and (4.4), and multiplying the resultant inequality by -1 gives us

$$\left(f'(x_2) - f'(x_1)\right)(x_2 - x_1) \geq 0 \tag{4.5}$$

Using the mean value theorem, $\exists z = x_1 + t(x_2 - x_1)$ for $t \in (0,1)$ such that

[3]For the case $x_2 < x_1$, the proof is very similar.

$$f(x_2) - f(x_1) = f'(z)(x_2 - x_1) \tag{4.6}$$

Since 4.5 holds for any $x_1, x_2 \in \mathcal{I}$, it also hold for $x_2 = z$. Therefore,

$$(f'(z) - f'(x_1))(x_2 - x_1) = \frac{1}{t}(f'(z) - f'(x_1))(z - x_1) \geq 0$$

Additionally using 4.6, we get

$$f(x_2) - f(x_1) = (f'(z) - f'(x_1))(x_2 - x_1) + f'(x_1)(x_2 - x_1) \geq f'(x_1)(x_2 - x_1) \tag{4.7}$$

Suppose equality holds in 4.5 for some $x_1 \neq x_2$. Then equality holds in 4.7 for the same x_1 and x_2. That is,

$$f(x_2) - f(x_1) = f'(x_1)(x_2 - x_1) \tag{4.8}$$

Applying 4.7 we can conclude that

$$f(x_1) + af'(x_1)(x_2 - x_1) \leq f(x_1 + a(x_2 - x_1)) \tag{4.9}$$

From 4.2 and 4.8, we can derive that

$$f(x_1 + a(x_2 - x_1)) < (1 - a)f(x_1) + af(x_2) = f(x_1) + af'(x_1)(x_2 - x_1) \tag{4.10}$$

However, equations 4.9 and 4.10 contradict each other. Therefore, equality in 4.5 cannot hold for any $x_1 \neq x_2$, implying that

$$(f'(x_2) - f'(x_1))\,(x_2 - x_1) > 0$$

that is, $f'(x)$ is increasing and therefore f is convex on $\mathcal{I}$. □

2. A differentiable function f is said to be *strictly concave* on an open interval $\mathcal{I}$, *iff*, $f'(x)$ is decreasing on $\mathcal{I}$. Recall from theorem 46, the graphical interpretation of the first derivative $f'(x)$; $f'(x) < 0$ implies that $f(x)$ is decreasing at x. Similarly, $f'(x)$ is monotonically decreasing when $f''(x) > 0$. This gives us a sufficient condition for the concavity of a function:

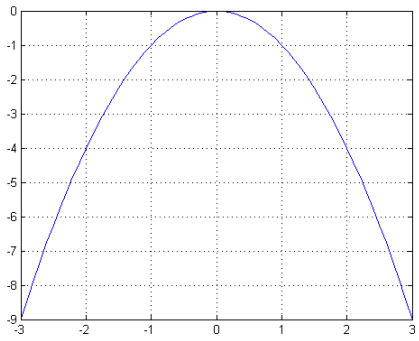

Figure 4.9: Plot for the strictly convex function $f(x) = -x^2$ which has $f''(x) = -2 < 0,\ \forall x$.

Theorem 52 *If at all points in an open interval $\mathcal{I}$, $f(x)$ is doubly differentiable and if $f''(x) < 0,\ \forall x \in \mathcal{I}$, then the slope of the function is always decreasing with x and the graph is strictly concave. This is illustrated in Figure 4.9.*

On the other hand, if the function is strictly concave and doubly differentiable in $\mathcal{I}$, then $f''(x) \leq 0,\ \forall x \in \mathcal{I}$.

There is also a slopeless interpretation of concavity as stated in the following theorem:

Theorem 53 *A differentiable function f is strictly concave on an open interval $\mathcal{I}$, iff*

$$f(ax_1 + (1-a)x_2) > af(x_1) + (1-a)f(x_2) \tag{4.11}$$

whenver $x_1, x_2 \in \mathcal{I},\ x_1 \neq x_2$ and $0 < a < 1$.

The proof is similar to that for theorem 51.

Figure 4.10 illustrates a function $f(x) = x^3 - x + 2$, whose slope decreases as x increases to 0 ($f''(x) < 0$) and then the slope increases beyond $x = 0$ ($f''(x) > 0$). The point 0, where the $f''(x)$ changes sign is called the *inflection point*; the graph is strictly concave for $x < 0$ and strictly convex for $x > 0$. Along similar lines, we can diagnose the function $f(x) = \frac{1}{20}x^5 - \frac{7}{12}x^4 + \frac{7}{6}x^3 - \frac{15}{2}x^2$; it is strictly concave on $(-\infty, -1]$ and $[3, 5]$ and strictly convex on $[-1, 3]$ and $[5, \infty]$. The inflection points for this function are at $x = -1$, $x = 3$ and $x = 5$.

The *first derivative test* for local extrema can be restated in terms of strict convexity and concavity of functions.

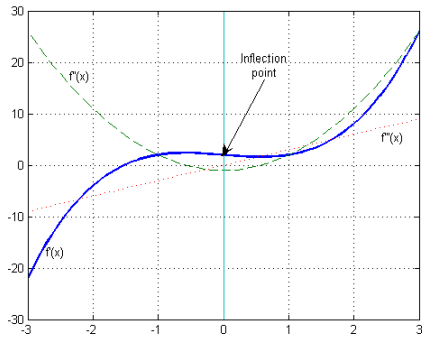

Figure 4.10: Plot for $f(x) = x^3 + x + 2$, which has an inflection point $x = 0$, along with plots for $f'(x)$ and $f''(x)$.

Procedure 2 [First derivative test in terms of strict convexity]: *Let c be a critical number of f and $f'(c) = 0$. Then,*

1. *$f(c)$ is a local minimum if the graph of $f(x)$ is strictly convex on an open interval containing c.*
2. *$f(c)$ is a local maximum if the graph of $f(x)$ is strictly concave on an open interval containing c.*

If the second derivative $f''(c)$ exists, then the strict convexity conditions for the critical number can be stated in terms of the sign of of $f''(c)$, making use of theorems 50 and 52. This is called the *second derivative test.*

Procedure 3 [Second derivative test]: *Let c be a critical number of f where $f'(c) = 0$ and $f''(c)$ exists.*

1. *If $f''(c) > 0$ then $f(c)$ is a local minimum.*
2. *If $f''(c) < 0$ then $f(c)$ is a local maximum.*
3. *If $f''(c) = 0$ then $f(c)$ could be a local maximum, a local minimum, neither or both. That is, the test fails.*

For example,

- If $f(x) = x^4$, then $f'(0) = 0$ and $f''(0) = 0$ and we can see that $f(0)$ is a local minimum.
- If $f(x) = -x^4$, then $f'(0) = 0$ and $f''(0) = 0$ and we can see that $f(0)$ is a local maximum.
- If $f(x) = x^3$, then $f'(0) = 0$ and $f''(0) = 0$ and we can see that $f(0)$ is neither a local minimum nor a local maximum. $(0, 0)$ is an inflection point in this case.

- If $f(x) = x + 2\sin x$, then $f'(x) = 1 + 2\cos x$. $f'(x) = 0$ for $x = \frac{2\pi}{3}, \frac{4\pi}{3}$, which are the critical numbers. $f''\left(\frac{2\pi}{3}\right) = -2\sin\frac{2\pi}{3} = -\sqrt{3} < 0 \Rightarrow f\left(\frac{2\pi}{3}\right) = \frac{2\pi}{3} + \sqrt{3}$ is a local maximum value. On the other hand, $f''\left(\frac{4\pi}{3}\right) = \sqrt{3} > 0 \Rightarrow f\left(\frac{4\pi}{3}\right) = \frac{4\pi}{3} - \sqrt{3}$ is a local minimum value.
- If $f(x) = x + \frac{1}{x}$, then $f'(x) = 1 - \frac{1}{x^2}$. The critical numbers are $x = \pm 1$. Note that $x = 0$ is not a critical number, even though $f'(0)$ does not exist, because 0 is not in the domain of f. $f''(x) = \frac{2}{x^3}$. $f''(-1) = -2 < 0$ and therefore $f(-1) = -2$ is a local maximum. $f''(1) = 2 > 0$ and therefore $f(1) = 2$ is a local minimum.

Global Extrema on Closed Intervals

Recall the extreme value theorem (theorem 40). An outcome of the extreme value theorem is that

- if either of c or d lies in (a, b), then it is a critical number of f;
- else each of c and d must lie on one of the boundaries of $[a, b]$.

This gives us a procedure for finding the maximum and minimum of a continuous function f on a closed bounded interval $\mathcal{I}$:

Procedure 4 [Finding extreme values on closed, bounded intervals]:
1. *Find the critical points in int($\mathcal{I}$).*
2. *Compute the values of f at the critical points and at the endpoints of the interval.*
3. *Select the least and greatest of the computed values.*

For example, to compute the maximum and minimum values of $f(x) = 4x^3 - 8x^2 + 5x$ on the interval $[0, 1]$, we first compute $f'(x) = 12x^2 - 16x + 5$ which is 0 at $x = \frac{1}{2}, \frac{5}{6}$. Values at the critical points are $f(\frac{1}{2}) = 1$, $f(\frac{5}{6}) = \frac{25}{27}$. The values at the end points are $f(0) = 0$ and $f(1) = 1$. Therefore, the minimum value is $f(0) = 0$ and the maximum value is $f(1) = f(\frac{1}{2}) = 1$.

In this context, it is relevant to discuss the one-sided derivatives of a function at the endpoints of the closed interval on which it is defined.

Definition 21 [One-sided derivatives at endpoints]: *Let f be defined on a closed bounded interval $[a, b]$. The (right-sided) derivative of f at $x = a$ is defined as*

$$f'(a) = \lim_{h \to 0^+} \frac{f(a+h) - f(a)}{h}$$

Similarly, the (left-sided) derivative of f at $x = b$ is defined as

$$f'(b) = \lim_{h \to 0^-} \frac{f(b+h) - f(b)}{h}$$

Essentially, each of the one-sided derivatives defines one-sided slopes at the endpoints. Based on these definitions, the following result can be derived.

Theorem 54 *If f is continuous on $[a,b]$ and $f'(a)$ exists as a real number or as $\pm\infty$, then we have the following necessary conditions for extremum at a.*

- *If $f(a)$ is the maximum value of f on $[a,b]$, then $f'(a) \leq 0$ or $f'(a) = -\infty$.*
- *If $f(a)$ is the minimum value of f on $[a,b]$, then $f'(a) \geq 0$ or $f'(a) = \infty$.*

If f is continuous on $[a,b]$ and $f'(b)$ exists as a real number or as $\pm\infty$, then we have the following necessary conditions for extremum at b.

- *If $f(b)$ is the maximum value of f on $[a,b]$, then $f'(b) \geq 0$ or $f'(b) = \infty$.*
- *If $f(b)$ is the minimum value of f on $[a,b]$, then $f'(b) \leq 0$ or $f'(b) = -\infty$.*

The following theorem gives a useful procedure for finding extrema on closed intervals.

Theorem 55 *If f is continuous on $[a,b]$ and $f''(x)$ exists for all $x \in (a,b)$. Then,*

- *If $f''(x) \leq 0,\ \forall x \in (a,b)$, then the minimum value of f on $[a,b]$ is either $f(a)$ or $f(b)$. If, in addition, f has a critical number $c \in (a,b)$, then $f(c)$ is the maximum value of f on $[a,b]$.*
- *If $f''(x) \geq 0,\ \forall x \in (a,b)$, then the maximum value of f on $[a,b]$ is either $f(a)$ or $f(b)$. If, in addition, f has a critical number $c \in (a,b)$, then $f(c)$ is the minimum value of f on $[a,b]$.*

The next theorem is very useful for finding global extrema values on open intervals.

Theorem 56 *Let $\mathcal{I}$ be an open interval and let $f''(x)$ exist $\forall x \in \mathcal{I}$.*

- *If $f''(x) \geq 0,\ \forall x \in \mathcal{I}$, and if there is a number $c \in \mathcal{I}$ where $f'(c) = 0$, then $f(c)$ is the global minimum value of f on $\mathcal{I}$.*
- *If $f''(x) \leq 0,\ \forall x \in \mathcal{I}$, and if there is a number $c \in \mathcal{I}$ where $f'(c) = 0$, then $f(c)$ is the global maximum value of f on $\mathcal{I}$.*

For example, let $f(x) = \frac{2}{3}x - \sec x$ and $\mathcal{I} = (\frac{-\pi}{2}, \frac{\pi}{2})$. $f'(x) = \frac{2}{3} - \sec x \tan x = \frac{2}{3} - \frac{\sin x}{\cos^2 x} = 0 \Rightarrow x = \frac{\pi}{6}$. Further, $f''(x) = -\sec x(\tan^2 x + \sec^2 x) < 0$ on $(\frac{-\pi}{2}, \frac{\pi}{2})$. Therefore, f attains the maximum value $f(\frac{\pi}{6}) = \frac{\pi}{9} - \frac{2}{\sqrt{3}}$ on $\mathcal{I}$.

As another example, let us find the dimensions of the cone with minimum volume that can contain a sphere with radius R. Let h be the height of the cone and r the radius of its base. The objective to be minimized is the volume $f(r,h) = \frac{1}{3}\pi r^2 h$. The constraint betwen r and h is shown in Figure 4.11; the traingle AEF is similar to traingle ADB and therefore, $\frac{h-R}{R} = \frac{\sqrt{h^2+r^2}}{r}$. Our

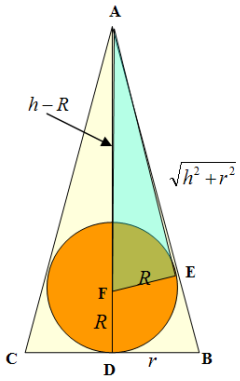

Figure 4.11: Illustrating the constraints for the optimization problem of finding the cone with minimum volume that can contain a sphere of radius R.

first step is to reduce the volume formula to involve only one of r[24] or h. The algebra involved will be the simplest if we solved for h. The constraint gives us $r^2 = \frac{R^2 h}{h-2R}$. Substituting this expression for r^2 into the volume formula, we get $g(h) = \frac{\pi R^2}{3}\frac{h^2}{(h-2R)}$ with the domain given by $\mathcal{D} = \{h|2R < h < \infty\}$. Note that $\mathcal{D}$ is an open interval. $g' = \frac{\pi R^2}{3}\frac{2h(h-2R)-h^2}{(h-2R)^2} = \frac{\pi R^2}{3}\frac{h(h-4R)}{(h-2R)^2}$ which is 0 in its domain $\mathcal{D}$ if and only if $h = 4R$. $g'' = \frac{\pi R^2}{3}\frac{2(h-2R)^3-2h(h-4R)(h-2R)^2}{(h-2R)^4} = \frac{\pi R^2}{3}\frac{2(h^2-4Rh+4R^2-h^2+4Rh)}{(h-2R)^3} = \frac{\pi R^2}{3}\frac{8R^2}{(h-2R)^3}$, which is greater than 0 in $\mathcal{D}$. Therefore, g (and consequently f) has a unique minimum at $h = 4R$ and correspondingly, $r^2 = \frac{R^2 h}{h-2R} = 2R^2$.

4.1.4 Optimization Principles for Multivariate Functions

Directional derivative and the gradient vector

Consider a function $f(\mathbf{x})$, with $\mathbf{x} \in \Re^n$. We start with the concept of the direction at a point $\mathbf{x} \in \Re^n$. We will represent a vector by $\mathbf{x}$ and the k^{th} component of $\mathbf{x}$ by x_k. Let $\mathbf{u}^k$ be a unit vector pointing along the k^{th} coordinate axis in $\Re^n$; $u_k^k = 1$ and $u_j^k = 0$, $\forall j \neq k$ An arbitrary direction vector $\mathbf{v}$ at $\mathbf{x}$ is a vector in $\Re^n$ with unit norm (*i.e.*, $||\mathbf{v}|| = 1$) and component v_k in the direction of $\mathbf{u}^k$. Let $f : \mathcal{D} \to \Re$, $\mathcal{D} \subseteq \Re^n$ be a function.

Definition 22 [Directional derivative]: *The directional derivative of $f(\mathbf{x})$ at $\mathbf{x}$ in the direction of the unit vector $\mathbf{v}$ is*

$$D_{\mathbf{v}} f(\mathbf{x}) = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h} \tag{4.12}$$

[4]Since r appears in the volume formula only in terms of r^2.

provided the limit exists.

As a special case, when $\mathbf{v} = \mathbf{u}^k$ the directional derivative reduces to the partial derivative of f with respect to x_k.

$$D_{\mathbf{u}^k} f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_k}$$

Theorem 57 *If $f(\mathbf{x})$ is a differentiable function of $\mathbf{x} \in \Re^n$, then f has a directional derivative in the direction of any unit vector $\mathbf{v}$, and*

$$D_{\mathbf{v}} f(\mathbf{x}) = \sum_{k=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_k} v_k \tag{4.13}$$

Proof: Define $g(h) = f(\mathbf{x} + \mathbf{v}h)$. Now:

- $g'(0) = \lim_{h \to 0} \frac{g(0+h) - g(0)}{h} = \lim_{h \to 0} \frac{f(\mathbf{x}+h\mathbf{v}) - f(\mathbf{x})}{h}$, which is the expression for the directional derivative defined in equation 4.12. Thus, $g'(0) = D_{\mathbf{v}} f(\mathbf{x})$.
- By definition of the chain rule for partial differentiation, we get another expression for $g'(0)$; $g'(0) = \sum_{k=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_k} v_k$

Therefore, $g'(0) = D_{\mathbf{v}} f(\mathbf{x}) = \sum_{k=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_k} v_k$ □

The theorem works if the function is differentiable at the point, else it is not predictable. The above theorem leads us directly to the idea of the gradient. We can see that the right hand side of (4.13) can be realized as the dot product of two vectors, *viz.*, $\left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^T$ and $\mathbf{v}$. Let us denote $\frac{\partial f(\mathbf{x})}{\partial x_i}$ by $f_{x_i}(\mathbf{x})$. Then we assign a name to the special vector discovered above.

Definition 23 [Gradient Vector]: *If f is differentiable function of $\mathbf{x} \in \Re^n$, then the gradient of $f(\mathbf{x})$ is the vector function $\nabla f(\mathbf{x})$, defined as:*

$$\nabla f(\mathbf{x}) = [f_{x_1}(\mathbf{x}),\ f_{x_2}(\mathbf{x}), \ldots,\ f_{x_n}(\mathbf{x})]$$

The directional derivative of a function f at a point $\mathbf{x}$ in the direction of a unit vector $\mathbf{v}$ can be now written as

$$D_{\mathbf{v}} f(\mathbf{x}) = \nabla^T f(\mathbf{x}).\mathbf{v} \tag{4.14}$$

What does the gradient $\nabla f(\mathbf{x})$ tell you about the function $f(\mathbf{x})$? We will illustrate with some examples. Consider the polynomial $f(x,y,z) = x^2y + z\sin xy$ and the unit vector $\mathbf{v}^T = \frac{1}{\sqrt{3}}[1,1,1]^T$. Consider the point $p_0 = (0,1,3)$. We will compute the directional derivative of f at p_0 in the direction of $\mathbf{v}$. To do this, we first compute the gradient of f in general: $\nabla f = \left[2xy + yz\cos xy,\ x^2 + xz\cos xy,\ \sin xy\right]^T$. Evaluating the gradient at a specific point p_0, $\nabla f(0,1,3) = [3,\ 0,\ 0]^T$. The directional derivative at p_0 in the direction $\mathbf{v}$ is $D_{\mathbf{v}}f(0,1,3) = [3,0,0].\frac{1}{\sqrt{3}}[1,1,1]^T = \sqrt{3}$. This directional derivative is the rate of change of f at p_0 in the direction $\mathbf{v}$; it is positive indicating that the function f increases at p_0 in the direction $\mathbf{v}$. All our ideas about first and second derivative in the case of a single variable carry over to the directional derivative.

As another example, let us find the rate of change of $f(x,y,z) = e^{xyz}$ at $p_0 = (1,2,3)$ in the direction from $p_1 = (1,2,3)$ to $p_2 = (-4,6,-1)$. We first construct a unit vector from p_1 to p_2; $\mathbf{v} = \frac{1}{\sqrt{57}}[-5,4,-4]$. The gradient of f in general is $\nabla f = [yze^{xyz},\ xze^{xyz},\ xye^{xyz}] = e^{xyz}[yz,\ xz,\ xy]$. Evaluating the gradient at a specific point p_0, $\nabla f(1,2,3) = e^6\,[6,3,2]^T$. The directional derivative at p_0 in the direction $\mathbf{v}$ is $D_{\mathbf{u}}f(1,2,3) = e^6[6,3,2].\frac{1}{\sqrt{57}}[-5,4,-4]^T = e^6\frac{-26}{\sqrt{57}}$. This directional derivative is negative, indicating that the function f decreases at p_0 in the direction from p_1 to p_2.

While there exist infinitely many direction vectors $\mathbf{v}$ at any point $\mathbf{x}$, there is a unique gradient vector $\nabla f(\mathbf{x})$. Since we seperated $D_{\mathbf{v}}f(\mathbf{x})$ as the dot prouduct of $\nabla f(\mathbf{x})$ with $\mathbf{v}$, we can study $\nabla f(\mathbf{x})$ independently. What does the gradient vector tell us? We will state a theorem to answer this question.

Theorem 58 *Suppose f is a differentiable function of $\mathbf{x} \in \Re^n$. The maximum value of the directional derivative $D_{\mathbf{v}}f(\mathbf{x})$ is $||\nabla f(\mathbf{x}||$ and it is so when $\mathbf{v}$ has the same direction as the gradient vector $\nabla f(\mathbf{x})$.*

Proof: The *cauchy schwartz inequality* when applied in the eucledian space states that $|\mathbf{x}^T.\mathbf{y}| \leq ||\mathbf{x}||.||\mathbf{y}||$ for any $\mathbf{x},\mathbf{y} \in \Re^n$, with equality holding *iff* $\mathbf{x}$ and $\mathbf{y}$ are linearly dependent. The inequality gives upper and lower bounds on the dot product between two vectors; $-||\mathbf{x}||.||\mathbf{y}|| \leq \mathbf{x}^T.\mathbf{y} \leq ||\mathbf{x}||.||\mathbf{y}||$. Applying these bounds to the right hand side of 4.14 and using the fact that $||\mathbf{v}|| = 1$, we get

$$-||\nabla f(\mathbf{x})|| \leq D_{\mathbf{v}}f(\mathbf{x}) = \nabla^T f(\mathbf{x}).\mathbf{v} \leq ||\nabla f(\mathbf{x})||$$

with equality holding *iff* $\mathbf{v} = k\nabla f(\mathbf{x})$ for some $k \geq 0$. Since $||\mathbf{v}|| = 1$, equality can hold *iff* $\mathbf{v} = \frac{\nabla f(\mathbf{x})}{||\nabla f(\mathbf{x})||}$. □

The theorem implies that the maximum rate of change of f at a point $\mathbf{x}$ is given by the norm of the gradient vector at $\mathbf{x}$. And the direction in which the rate of change of f is maximum is given by the unit vector $\frac{\nabla f(\mathbf{x}}{||\nabla f(\mathbf{x}||}$.

An associated fact is that the minimum value of the directional derivative $D_{\mathbf{v}}f(\mathbf{x})$ is $-||\nabla f(\mathbf{x}||$ and it occurs when $\mathbf{v}$ has the opposite direction of the gradient vector, *i.e.*, $-\frac{\nabla f(\mathbf{x}}{||\nabla f(\mathbf{x}||}$. This fact is often used in numerical analysis when one is trying to minimize the value of very complex functions. The method

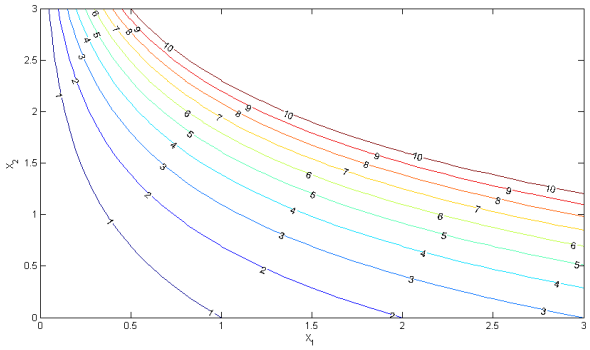

Figure 4.12: 10 level curves for the function $f(x_1, x_2) = x_1 e^{x_2}$.

of steepest descent uses this result to iteratively choose a new value of $\mathbf{x}$ by traversing in the direction of $-\nabla f(\mathbf{x})$.

Consider the function $f(x_1, x_2) = x_1 e^{x_2}$. Figure 4.12 shows 10 level curves for this function, corresponding to $f(x_1, x_2) = c$ for $c = 1, 2, \ldots, 10$. The idea behind a level curve is that as you change $\mathbf{x}$ along any level curve, the function value remains unchanged, but as you move $\mathbf{x}$ across level curves, the function value changes.

We will define the concept of a hyperplane next, since it will be repeatedly referred to in the sequel.

Definition 24 [Hyperplane]: *A set of points $\mathcal{H} \subseteq \Re^n$ is called a hyperplane if there exists a vector $\mathbf{v} \in \Re^n$ and a point $\mathbf{q} \in \Re^n$ such that*

$$\forall\ \mathbf{p} \in \mathcal{H}, (\mathbf{p} - \mathbf{q})^T \mathbf{v} = 0$$

or in other words, $\forall \mathbf{p} \in \mathcal{H}, \mathbf{p}^T\mathbf{v} = \mathbf{q}^T\mathbf{v}$. This is the equation of a hyperplane orthogonal to vector $\mathbf{v}$ and passing through point $\mathbf{q}$. The space spanned by vectors in the hyperplane $\mathcal{H}$ which are orthogonal to vector v, forms the orthogonal complement of the space spanned by v.

Hyperplane $\mathcal{H}$ can also be equivalently defined as the set of points $\mathbf{p}$ such that $\mathbf{p}^T\mathbf{v} = c$ for some $c \in \Re$ and some $\mathbf{v} \in \Re^n$, with $c = \mathbf{q}^T\mathbf{v}$ in our definition. (This definition will be referred to at a later point.)

What if $D_{\mathbf{v}} f(\mathbf{x})$ turns out to be 0? What can we say about $\nabla f(\mathbf{x})$ and $\mathbf{v}$? There is a useful theorem in this regard.

Theorem 59 *Let $f : \mathcal{D} \to \Re$ with $\mathcal{D} \in \Re^n$ be a differentiable function. The gradient ∇f evaluated at $\mathbf{x}^*$ is orthogonal to the tangent hyperplane (tangent line in case $n = 2$) to the level surface of f passing through $\mathbf{x}^*$.*

Proof: Let $\mathcal{K}$ be the range of f and let $k \in \mathcal{K}$ such that $f(\mathbf{x}^*) = k$. Consider the level surface $f(\mathbf{x}) = k$. Let $\mathbf{r}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$ be a curve on the level surface, parametrized by $t \in \Re$, with $\mathbf{r}(0) = \mathbf{x}^*$. Then, $f(x(t), y(t), z(t)) = k$. Applying the chain rule

$$\frac{df(\mathbf{r}(t))}{dt} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \frac{dx_i(t)}{dt} = \nabla^T f(\mathbf{x}(t)) \frac{d\mathbf{r}(t)}{dt} = 0$$

For $t = 0$, the equations become

$$\nabla^T f(\mathbf{x}^*) \frac{d\mathbf{r}(0)}{dt} = 0$$

Now, $\frac{d\mathbf{r}(t)}{dt}$ represents any tangent vector to the curve through $\mathbf{r}(t)$ which lies completely on the level surface. That is, the tangent line to any curve at $\mathbf{x}^*$ on the level surface containing $\mathbf{x}^*$, is orthogonal to $\nabla f(\mathbf{x}^*)$. Since the tangent hyperplane to a surface at any point is the hyperplane containing all tangent vectors to curves on the surface passing through the point, the gradient is perpendicular to the tangent hyperplane to the level surface passing through that point. The equation of the tangent hyperplane is given by $(\mathbf{x}-\mathbf{x}^*)^T \nabla f(\mathbf{x}^*) = 0$. □

Recall from elementary calculus, that the normal to a plane can be found by taking the cross product of any two vectors lying within the plane. The gradient vector at any point on the level surface of a function is normal to the tangent hyperplane (or tangent line in the case of two variables) to the surface at the same point, but can however be conveniently obtained using the partial derivatives of the function at that point.

We will use some illustrative examples to study these facts.

1. Consider the same plot as in Figure 4.12 with a gradient vector at $(2, 0)$ as shown in Figure 4.13. The gradient vector $[1,\ 2]^T$ is perpendicular to the tangent hyperplane to the level curve $x_1 e^{x_2} = 2$ at $(2, 0)$. The equation of the tangent hyperplane is $(x_1 - 2) + 2(x_2 - 0) = 0$ and it turns out to be a tangent line.

2. The level surfaces for $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ are shown in Figure 4.14. The gradient at $(1, 1, 1)$ is orthogonal to the tangent hyperplane to the level surface $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 = 3$ at $(1, 1, 1)$. The gradient vector at $(1, 1, 1)$ is $[2,\ 2,\ 2]^T$ and the tanget hyperplane has the equation $2(x_1 - 1) + 2(x_2 - 1) + 2(x_3 - 1) = 0$, which is a plane in $3D$. On the other hand, the dotted line in Figure 4.15 is not orthogonal to the level surface, since it does not coincide with the gradient.

3. Let $f(x_1, x, x_3) = x_1^2 x_2^3 x_3^4$ and consider the point $\mathbf{x}^0 = (1, 2, 1)$. We will find the equation of the tangent plane to the level surface through $\mathbf{x}^0$. The level surface through $\mathbf{x}^0$ is determined by setting f equal to its value evaluated at $\mathbf{x}^0$; that is, the level surface will have the equation $x_1^2 x_2^3 x_3^4 = 1^2 2^3 1^4 = 8$. The gradient vector (normal to tangent plane) at

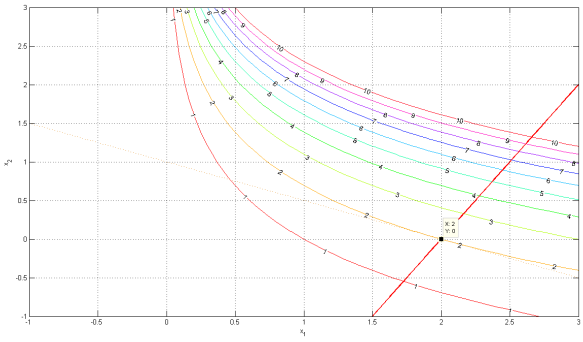

Figure 4.13: The level curves from Figure 4.12 along with the gradient vector at $(2, 0)$. Note that the gradient vector is perpenducular to the level curve $x_1 e^{x_2} = 2$ at $(2, 0)$.

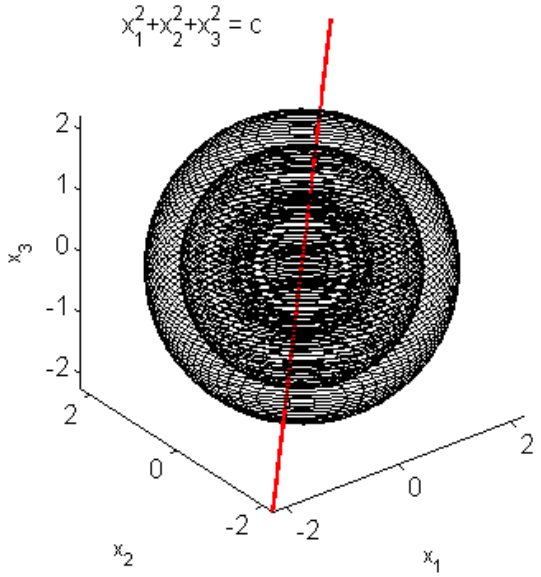

Figure 4.14: 3 level surfaces for the function $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ with $c = 1, 3, 5$. The gradient at $(1, 1, 1)$ is orthogonal to the level surface $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 = 3$ at $(1, 1, 1)$.

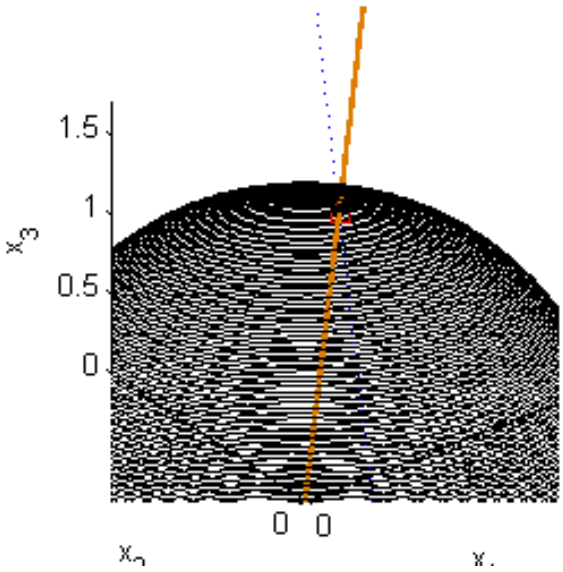

Figure 4.15: Level surface $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 = 3$. The gradient at $(1, 1, 1)$, drawn as a bold line, is perpendicular to the tangent plane to the level surface at $(1, 1, 1)$, whereas, the dotted line, though passing through $(1, 1, 1)$ is not perpendicular to the same tangent plane.

$(1,2,1)$ is $\nabla f(x_1,x_2,x_3)|_{(1,2,1)} = [2x_1x_2^3x_3^4,\ 3x_1^2x_2^2x_3^4, 4x_1^2x_2^3x_3^3]^T\big|_{(1,2,1)} = [16,\ 12,\ 32]^T$. The equation of the tangent plane at $\mathbf{x}^0$, given the normal vector $\nabla f(\mathbf{x}^0)$ can be easily written down: $\nabla f(\mathbf{x}^0)^T.[\mathbf{x}-\mathbf{x}^0] = 0$ which turns out to be $16(x_1-1)+12(x_2-2)+32(x_3-1)=0$, a plane in $3D$.

4. Consider the function $f(x,y,z) = \frac{x}{y+z}$. The directional derivative of f in the direction of the vector $\mathbf{v} = \frac{1}{\sqrt{14}}[1,\ 2,\ 3]$ at the point $x^0 = (4,1,1)$ is $\nabla^T f\big|_{(4,1,1)} \cdot \frac{1}{\sqrt{14}}[1,\ 2,\ 3]^T = \left[\frac{1}{y+z},\ -\frac{x}{(y+z)^2},\ -\frac{x}{(y+z)^2}\right]\Big|_{(4,1,1)} \cdot \frac{1}{\sqrt{14}}[1,\ 2,\ 3]^T = \left[\frac{1}{2},\ -1,\ -1\right] \cdot \frac{1}{\sqrt{14}}[1,\ 2,\ 3]^T = -\frac{9}{2\sqrt{14}}$. The directional derivative is negative, indicating that the function decreases along the direction of $\mathbf{v}$. Based on theorem 58, we know that the maximum rate of change of a function at a point $\mathbf{x}$ is given by $||\nabla f(\mathbf{x})||$ and it is in the direction $\frac{\nabla f(\mathbf{x})}{||\nabla f(\mathbf{x})||}$. In the example under consideration, this maximum rate of change at $\mathbf{x}^0$ is $\frac{3}{2}$ and it is in the direction of the vector $\frac{2}{3}\left[\frac{1}{2},\ -1,\ -1\right]$.

5. Let us find the maximum rate of change of the function $f(x,y,z) = x^2y^3z^4$ at the point $\mathbf{x}^0 = (1,1,1)$ and the direction in which it occurs. The gradient at $\mathbf{x}^0$ is $\nabla^T f\big|_{(1,1,1)} = [2,\ 3,\ 4]$. The maximum rate of change at $\mathbf{x}^0$ is therefore $\sqrt{29}$ and the direction of the corresponding rate of change is $\frac{1}{\sqrt{29}}[2,\ 3,\ 4]$. The minimum rate of change is $-\sqrt{29}$ and the corresponding direction is $-\frac{1}{\sqrt{29}}[2,\ 3,\ 4]$.

6. Let us determine the equations of (a) the tangent plane to the paraboloid $\mathcal{P} : x_1 = x_2^2 + x_3^2 + 2$ at $(-1,1,0)$ and (b) the normal line to the tangent plane. To realize this as the level surface of a function of three variables, we define the function $f(x_1,x_2,x_3) = x_1 - x_2^2 - x_3^2$ and find that the paraboloid $\mathcal{P}$ is the same as the level surface $f(x_1,x_2,x_3) = -2$. The normal to the tangent plane to $\mathcal{P}$ at $\mathbf{x}^0$ is in the direction of the gradient vector $\nabla f(\mathbf{x}^0) = [1,-2,0]^T$ and its parametric equation is $[x_1,\ x_2,\ x_3] = [-1+t,\ 1-2t,\ 0]$. The equation of the tangent plane is therefore $(x_1+1) - 2(x_2-1) = 0$.

We can embed the graph of a function of n variables as the 0-level surface of a function of $n+1$ variables. More concretely, if $f : \mathcal{D} \to \Re,\ \mathcal{D} \subseteq \Re^n$ then we define $F : \mathcal{D}' \to \Re,\ \mathcal{D}' = \mathcal{D} \times \Re$ as $F(\mathbf{x},z) = f(\mathbf{x}) - z$ with $\mathbf{x} \in \mathcal{D}'$. The function f then corresponds to a single level surface of F given by $F(\mathbf{x},z) = 0$. In other words, the $0-$level surface of F gives back the graph of f. The gradient of F at any point $(\mathbf{x},z)$ is simply, $\nabla F(\mathbf{x},z) = [f_{x_1}, f_{x_2}, \ldots, f_{x_n}, -1]$ with the first n components of $\nabla F(\mathbf{x},z)$ given by the n components of $\nabla f(\mathbf{x})$. We note that the level surface of F passing through point $(\mathbf{x}^0, f(\mathbf{x}^0))$ is its 0-level surface, which is essentially the surface of the function $f(\mathbf{x})$. The equation of the tangent hyperplane to the $0-$level surface of F at the point $(\mathbf{x}^0, f(\mathbf{x}^0))$ (that is, the tangent hyperplane to $f(\mathbf{x})$ at the point $\mathbf{x}_0$), is $\nabla F(\mathbf{x}^0, f(\mathbf{x}^0))^T.[\mathbf{x}-\mathbf{x}^0, z - f(\mathbf{x}^0)]^T = 0$. Substituting appropriate expression for $\nabla F(\mathbf{x}^0)$, the equation of the tangent plane can be written as

$$\left(\sum_{i=1}^{n} f_{x_i}(\mathbf{x}^0)(x_i - x_i^0)\right) - \big(z - f(\mathbf{x}^0)\big) = 0$$

or equivalently as,

$$\left(\sum_{i=1}^{n} f_{x_i}(\mathbf{x}^0)(x_i - x_i^0)\right) + f(\mathbf{x}^0) = z$$

As an example, consider the paraboloid, $f(x_1, x_2) = 9 - x_1^2 - x_2^2$, the corresponding $F(x_1, x_2, z) = 9 - x_1^2 - x_2^2 - z$ and the point $x^0 = (\mathbf{x}^0, z) = (1, 1, 7)$ which lies on the 0-level surface of F. The gradient $\nabla F(x_1, x_2, z)$ is $[-2x_1,\ -2x_2,\ -1]$, which when evaluated at $x^0 = (1, 1, 7)$ is $[-2,\ -2,\ -1]$. The equation of the tangent plane to f at x^0 is therefore given by $-2(x_1 - 1) - 2(x_2 - 1) + 7 = z$.

Recall from theorem 39 that for functions of single variable, at local extreme points, the tangent to the curve is a line with a constant component in the direction of the function and is therefore parallel to the x-axis. If the function is is differentiable at the extreme point, then the derivative must vanish. This idea can be extended to functions of multiple variables. The requirement in this case turns out to be that the tangent plane to the function at any extreme point must be parallel to the plane $z = 0$. This can happen if and only if the gradient ∇F is parallel to the $z-$axis at the extreme point, or equivalently, the gradient to the function f must be the zero vector at every extreme point.

We will formalize this discussion by first providing the definitions for local maximum and minimum as well as absolute maximum and minimum values of a function of n variables.

Definition 25 [Local maximum]: *A function f of n variables has a local maximum at $\mathbf{x}^0$ if $\exists \epsilon > 0$ such that $\forall\ ||\mathbf{x} - \mathbf{x}^0|| < \epsilon$. $f(\mathbf{x}) \leq f(\mathbf{x}^0)$. In other words, $f(\mathbf{x}) \leq f(\mathbf{x}^0)$ whenever $\mathbf{x}$ lies in some circular disk around $\mathbf{x}^0$.*

Definition 26 [Local minimum]: *A function f of n variables has a local minimum at $\mathbf{x}^0$ if $\exists \epsilon > 0$ such that $\forall\ ||\mathbf{x} - \mathbf{x}^0|| < \epsilon$. $f(\mathbf{x}) \geq f(\mathbf{x}^0)$. In other words, $f(\mathbf{x}) \geq f(\mathbf{x}^0)$ whenever $\mathbf{x}$ lies in some circular disk around $\mathbf{x}^0$.*

These definitions are exactly analogous to the definitions for a function of single variable. Figure 4.16 shows the plot of $f(x_1, x_2) = 3x_1^2 - x_1^3 - 2x_2^2 + x_2^4$. As can be seen in the plot, the function has several local maxima and minima.

We will next state a theorem fundamental to determining the locally extreme values of functions of multiple variables.

Theorem 60 *If $f(\mathbf{x})$ defined on a domain $\mathcal{D} \subseteq \Re^n$ has a local maximum or minimum at $\mathbf{x}^*$ and if the first-order partial derivatives exist at $\mathbf{x}^*$, then $f_{x_i}(\mathbf{x}^*) = 0$ for all $1 \leq i \leq n$.*

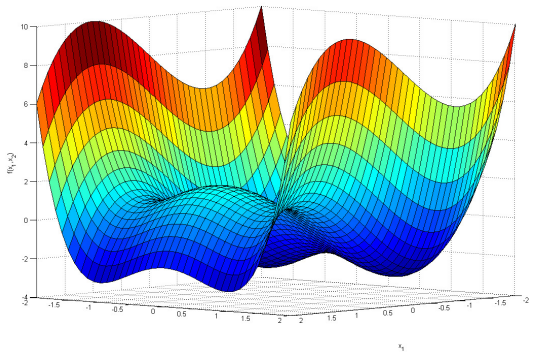

Figure 4.16: Plot of $f(x_1, x_2) = 3x_1^2 - x_1^3 - 2x_2^2 + x_2^4$, showing the various local maxima and minima of the function.

Proof: The idea behind this theorem can be stated as follows. The tangent hyperplane to the function at any extreme point must be parallel to the plane $z = 0$. This can happen if and only if the gradient $\nabla F = [\nabla^T f,\ -1]^T$ is parallel to the $z-$axis at the extreme point. Or equivalently, the gradient to the function f must be the zero vector at every extreme point, *i.e.*, $f_{x_i}(\mathbf{x}^*) = 0$ for $1 \leq i \leq n$.

To formally prove this theorem, consider the function $g_i(x_i) = f(x_1^*, x_2^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, \ldots, x_n^*)$. If f has a local extremum at $\mathbf{x}^*$, then each function $g_i(x_i)$ must have a local extremum at x_i^*. Therefore $g_i^{'}(x_i^*) = 0$ by theorem 39. Now $g_i^{'}(x_i^*) = f_{x_i}(\mathbf{x}^*)$ so $f_{x_i}(\mathbf{x}^*) = 0$. □

Applying theorem 60 to the function $f(x_1, x_2) = 9 - x_1^2 - x_2^2$, we require that at any extreme point $f_{x_1} = -2x_1 = 0 \Rightarrow x_1 = 0$ and $f_{x_2} = -2x_2 = 0 \Rightarrow x_2 = 0$. Thus, f indeed attains its maximum at the point $(0, 0)$ as shown in Figure 4.17.

Definition 27 [Critical point]: *A point $\mathbf{x}^*$ is called a critical point of a function $f(\mathbf{x})$ defined on $\mathcal{D} \subseteq \Re^n$ if*

1. *If $f_{x_i}(\mathbf{x}^*) = 0$, for $1 \leq i \leq n$.*
2. *OR $f_{x_i}(\mathbf{x}^*)$ fails to exist for any $1 \leq i \leq n$.*

A procedure for computing all critical points of a function f is:

1. Compute f_{x_i} for $1 \leq i \leq n$.
2. Determine if there are any points where any one of f_{x_i} fails to exist. Add such points (if any) to the list of critical points.
3. Solve the system of equations $f_{x_i} = 0$ simultaneously. Add the solution points to the list of saddle points.

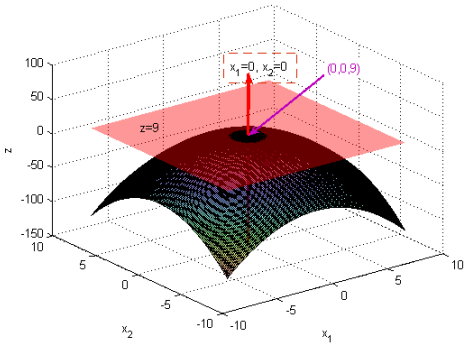

Figure 4.17: The paraboloid $f(x_1, x_2) = 9 - x_1^2 - x_2^2$ attains its maximum at $(0, 0)$. The tanget plane to the surface at $(0, 0, f(0, 0))$ is also shown, and so is the gradient vector ∇F at $(0, 0, f(0, 0))$.

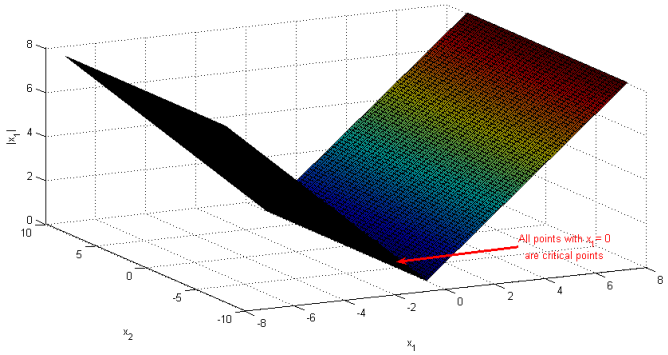

Figure 4.18: Plot illustrating critical points where derivative fails to exist.

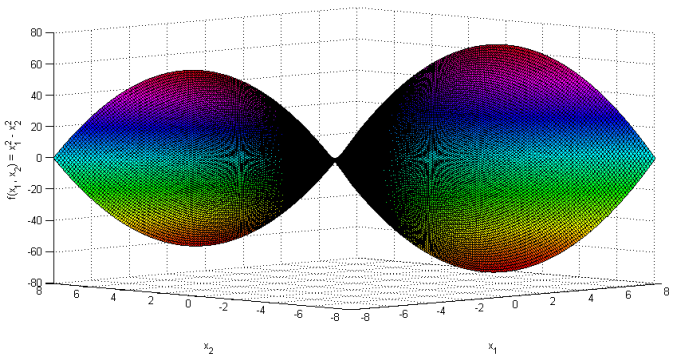

Figure 4.19: The hyperbolic paraboloid $f(x_1, x_2) = x_1^2 - x_2^2$, which has a saddle point at $(0, 0)$.

As an example, for the function $f(x_1, x_2) = |x_1|$, f_{x_1} does not exist for $(0, s)$ for any $s \in \Re$ and all of them are critical points. Figure 4.18 shows the corresponding 3−D plot.

Is the converse of theorem 60 true? That is, if you find an $\mathbf{x}^*$ that satisifes $f_{x_i}(\mathbf{x}^*) =$ for all $1 \leq i \leq n$, is it necessary that $\mathbf{x}^*$ is an extreme point? The answer is no. In fact, points that violate the converse of theorem 60 are called saddle points.

Definition 28 [Saddle point]: *A point $\mathbf{x}^*$ is called a saddle point of a function $f(\mathbf{x})$ defined on $\mathcal{D} \subseteq \Re^n$ if $\mathbf{x}^*$ is a critical point of f but $\mathbf{x}^*$ does not correspond to a local maximum or minimum of the function.*

We saw the example of a saddle point in Figure 4.7, for the case $n = 1$. The *inflection point* for a function of single variable, that was discussed earlier, is the analogue of the saddle point for a function of multiple variables. An example for $n = 2$ is the hyperbolic paraboloid[5] $f(x_1, x_2) = x_1^2 - x_2^2$, the graph of which is shown in Figure 4.19. The hyperbolic paraboloid opens up on x_1-axis (Figure 4.20 and down on x_2-axis (Figure 4.21) and has a saddle point at $(0, 0)$.

To get working on figuring out how to find the maximum and minimum of a function, we will take some examples. Let us find the critical points of $f(x_1, x_2) = x_1^2 + x_2^2 - 2x_1 - 6x_2 + 14$ and classify the critical point. This function is a polyonomial function and is differentiable everywhere. It is a paraboloid that is shifted away from origin. To find its critical points, we will solve $f_{x_1} = 2x_1 - 2 = 0$ and $f_{x_2} = 2x_2 - 6 = 0$, which when solved simultaneously, yield a single critical point $(1, 3)$. For a simple example like this, the function f can be rewritten as $f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 3)^2 + 4$, which implies that $f(x_1, x_2) \geq 4 = f(1, 3)$. Therefore, $(1, 3)$ is indeed a local minimum (in fact a global minimum) of $f(x_1, x_2)$.

[5]The hyperbolic paraboloid is shaped like a *saddle* and can have a critical point called the saddle point.

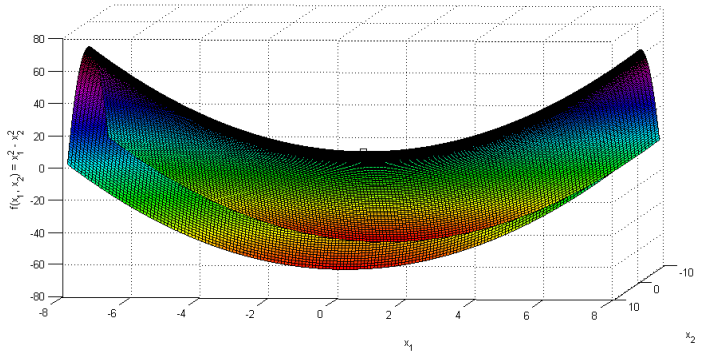

Figure 4.20: The hyperbolic paraboloid $f(x_1, x_2) = x_1^2 - x_2^2$, when viewed from the x_1 axis is concave up.

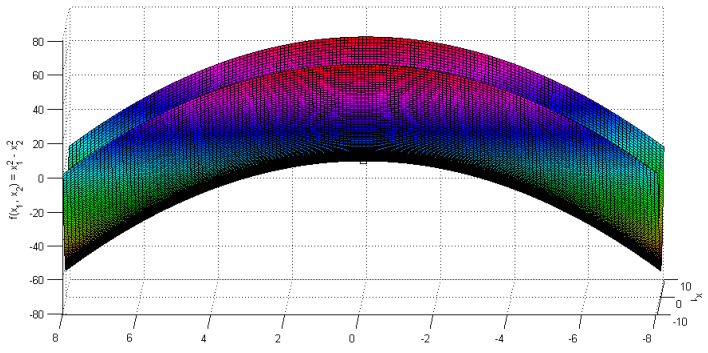

Figure 4.21: The hyperbolic paraboloid $f(x_1, x_2) = x_1^2 - x_2^2$, when viewed from the x_2 axis is concave down.

However, it is not always so easy to determine if a critical point is a point of local extreme value. To understand this, consider the function $f(x_1, x_2) = 2x_1^3 + x_1x_2^2 + 5x_1^2 + x_2^2$. The system of equations to be solved are $f_{x_1} = 6x_1^2 + x_2^2 + 10x_1 = 0$ and $f_{x_2} = 2x_1x_2 + 2x_2 = 0$. From the second equation, we get either $x_2 = 0$ or $x_1 = -1$. Using these values one at a time in the first equation, we get values for the other variables. The critical points are: $(0, 0)$, $(-\frac{5}{3}, 0)$, $(-1, 2)$ and $(-1, -2)$. Which of these critical points correspond to extreme values of the function? Since f does not have a quadratic form, it is not easy to find a lower bound on the function as in the previous example. However, we can make use of the taylor series expansion for single variable to find polynomial expansions of functions of n variables. The following theorem gives a systematic method, similar to the second derivative test for functions of single variable, for finding maxima and minima of functions of multiple variables.

Theorem 61 *Let $f : \mathcal{D} \rightarrow \Re$ where $\mathcal{D} \subseteq \Re^n$. Let $f(\mathbf{x})$ have continuous partial derivatives and continuous mixed partial derivatives in an open ball $\mathcal{R}$ containing a point $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = 0$. Let $\nabla^2 f(\mathbf{x})$ denote an $n \times n$ matrix of mixed partial derivatives of f evaluated at the point $\mathbf{x}$, such that the ij^{th} entry of the matrix is $f_{x_i x_j}$. The matrix $\nabla^2 f(\mathbf{x})$ is called the Hessian matrix. The Hessian matrix is symmetric[6]. Then,*

- *If $\nabla^2 f(\mathbf{x}^*)$ is positive definite, $\mathbf{x}^*$ is a local minimum.*
- *If $\nabla^2 f(\mathbf{x}^*)$ is negative definite (that is if $-\nabla^2 f(\mathbf{x}^*)$ is positive definite), $\mathbf{x}^*$ is a local maximum.*

Proof: Since the mixed partial derivatives of f are continuous in an open ball containing $\mathcal{R}$ containing $\mathbf{x}^*$ and since $\nabla^2 f(\mathbf{x}^*) \succ 0$, it can be shown that there exists an $\epsilon > 0$, with $\mathcal{B}(\mathbf{x}^*, \epsilon) \subseteq \mathcal{R}$ such that for all $||\mathbf{h}|| < \epsilon$, $\nabla^2 f(\mathbf{x}^* + \mathbf{h}) \succ 0$. Consider an increment vector $\mathbf{h}$ such that $(\mathbf{x}^* + \mathbf{h}) \in \mathcal{B}(\mathbf{x}^*, \epsilon)$. Define $g(t) = f(\mathbf{x}^* + t\mathbf{h}) : [0, 1] \rightarrow \Re$. Using the chain rule,

$$g'(t) = \sum_{i=1}^{n} f_{x_i}(\mathbf{x}^* + t\mathbf{h}) \frac{dx_i}{dt} = \mathbf{h}^T . \nabla f(\mathbf{x}^* + t\mathbf{h})$$

Since f has continuous partial and mixed partial derivatives, g' is a differentiable function of t and

$$g''(t) = \mathbf{h}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{h}) \mathbf{h}$$

Since g and g' are continous on $[0, 1]$ and g' is differentiable on $(0, 1)$, we can make use of the Taylor's theorem (45) with $n = 1$ and $a = 0$ to obtain:

$$g(1) = g(0) + g'(0) + \frac{1}{2} g''(c)$$

[6] By Clairauts Theorem, if the partial and mixed derivatives of a function are continuous on an open region containing a point $\mathbf{x}^*$, then $f_{x_i x_j}(\mathbf{x}^*) = f_{x_j x_i}(\mathbf{x}^*)$, for all $i, j \in [1, n]$.

for some $c \in (0, 1)$. Writing this equation in terms of f gives

$$f(\mathbf{x}^* + \mathbf{h}) = f(\mathbf{x}^*) + \mathbf{h}^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\mathbf{h}^T \nabla^2 f(\mathbf{x}^* + c\mathbf{h})\mathbf{h}$$

We are given that $\nabla f(\mathbf{x}^*) = 0$. Therefore,

$$f(\mathbf{x}^* + \mathbf{h}) - f(\mathbf{x}^*) = \frac{1}{2}\mathbf{h}^T \nabla^2 f(\mathbf{x}^* + c\mathbf{h})\mathbf{h}$$

The presence of an extremum of f at $\mathbf{x}^*$ is determined by the sign of $f(\mathbf{x}^* + \mathbf{h}) - f(\mathbf{x}^*)$. By virtue of the above equation, this is the same as the sign of $H(c) = \mathbf{h}^T \nabla^2 f(\mathbf{x}^* + c\mathbf{h})\mathbf{h}$. Because the partial derivatives of f are continuous in $\mathcal{R}$, if $H(0) \neq 0$, the sign of $H(c)$ will be the same as the sign of $H(0) = \mathbf{h}^T \nabla^2 f(\mathbf{x}^*)\mathbf{h}$ for $\mathbf{h}$ with sufficiently small components (*i.e.*, since the function has continuous partial and mixed partial derivatives at ($\mathbf{x}^*$, the hessian will be positive in some small neighborhood around ($\mathbf{x}^*$). Therefore, if $\nabla^2 f(\mathbf{x}^*)$ is positive definite, we are guaranteed to have $H(0)$ positive, implying that f has a local minimum at $\mathbf{x}^*$. Similarly, if $-\nabla^2 f(\mathbf{x}^*)$ is positive definite, we are guaranteed to have $H(0)$ negative, implying that f has a local maximum at $\mathbf{x}^*$. □

Theorem 61 gives sufficient conditions for local maxima and minima of functions of multiple variables. Along similar lines of the proof of theorem 61, we can prove necessary conditions for local extrema in theorem 62.

Theorem 62 *Let $f : \mathcal{D} \to \Re$ where $\mathcal{D} \subseteq \Re^n$. Let $f(\mathbf{x})$ have continuous partial derivatives and continuous mixed partial derivatives in an open region $\mathcal{R}$ containing a point $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = 0$. Then,*

- *If $\mathbf{x}^*$ is a point of local minimum, $\nabla^2 f(\mathbf{x}^*)$ must be positive semi-definite.*
- *If $\mathbf{x}^*$ is a point of local maximum, $\nabla^2 f(\mathbf{x}^*)$ must be negative semi-definite (that is, $-\nabla^2 f(\mathbf{x}^*)$ must be positive semi-definite).*

The following corollary of theorem 62 states a sufficient condition for a point to be a saddle point.

Corollary 63 *Let $f : \mathcal{D} \to \Re$ where $\mathcal{D} \subseteq \Re^n$. Let $f(\mathbf{x})$ have continuous partial derivatives and continuous mixed partial derivatives in an open region $\mathcal{R}$ containing a point $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = 0$. If $\nabla^2 f(\mathbf{x}^*)$ is neither positive semi-definite nor negative semi-definite (that is, some of its eigenvalues are positive and some negative), then $\mathbf{x}^*$ is a saddle point.*

Thus, for a function of more than one variable, the second derivative test generalizes to a test based on the eigenvalues of the function's Hessian matrix at the stationary point. Based on theorem 61, we will derive the second derivative test for determining extreme values of a function of two variables.

Theorem 64 *Let the partial and second partial derivatives of $f(x_1, x_2)$ be continuous on a disk with center (a, b) and suppose $f_{x_1}(a, b) = 0$ and $f_{x_2}(a, b) = 0$ so that (a, b) is a critical point of f. Let $D(a, b) = f_{x_1x_1}(a, b)f_{x_2x_2}(a, b) - [f_{x_1x_2}(a, b)]^2$. Then*[7]*,*

- *If $D > 0$ and $f_{x_1x_1}(a, b) > 0$, then $f(a, b)$ is a local minimum.*
- *Else if $D > 0$ and $f_{x_1x_1}(a, b) < 0$, then $f(a, b)$ is a local maximum.*
- *Else if $D < 0$ then (a, b) is a saddle point.*

Proof: Recall the definition of positive definiteness; a matrix is positive definite if all its eigenvalues are positive. For the 2×2 matrix $\nabla^2 f$ in this problem, the product of the eigenvalues is $det(\nabla^2 f) = f_{x_1x_1}(a, b)f_{x_2x_2}(a, b) - [f_{x_1x_2}(a, b)]^2$ and the sum of the eigenvalues is $f_{x_1x_1}(a, b) + f_{x_2x_2}(a, b)$. Now:

- If $det(\nabla^2 f(a, b)) > 0$ and if additionally $f_{x_1x_1}(a, b) > 0$ (or equivalently, $f_{x_2x_2}(a, b) > 0$), the product as well as the sum of eigenvalues will be positive, implying that the eigenvalues are positive and therefore $\nabla^2 f(a, b)$ is positive definite, According to theorem 61, this is a sufficient condition for $f(a, b)$ to be a local minimum.
- If $det(\nabla^2 f(a, b)) > 0$ and if additionally $f_{x_1x_1}(a, b) < 0$ (or equivalently, $f_{x_2x_2}(a, b) < 0$), the product of the eigenvalue is positive whereas the sum is negative, implying that the eigenvalues are negative and therefore $\nabla^2 f(a, b)$ is negative definite, According to theorem 61, this is a sufficient condition for $f(a, b)$ to be a local maximum.
- If $det(\nabla^2 f(a, b)) < 0$, the eigenvalues must have opposite signs, implying that the $\nabla^2 f(a, b)$ is neither positive semi-definite nor negative-semidefinite. By corollary 63, this is a sufficient condition for $f(a, b)$ to be a saddle point.

□

We saw earlier that the critical points for $f(x_1, x_2) = 2x_1^3 + x_1x_2^2 + 5x_1^2 + x_2^2$ are $(0, 0)$, $(-\frac{5}{3}, 0)$, $(-1, 2)$ and $(-1, -2)$. To determine which of these correspond to local extrema and which are saddle, we first compute compute the partial derivatives of f:

$f_{x_1x_1}(x_1, x_2) = 12x_1 + 10$
$f_{x_2x_2}(x_1, x_2) = 2x_1 + 2$
$f_{x_1x_2}(x_1, x_2) = 2x_2$

Using theorem 64, we can verify that $(0, 0)$ corresponds to a local minimum, $(-\frac{5}{3}, 0)$ corresponds to a local maximum while $(-1, 2)$ and $(-1, -2)$ correspond to saddle points. Figure 4.22 shows the plot of the function while pointing out the four critical points.

[7] D here stands for the discriminant.

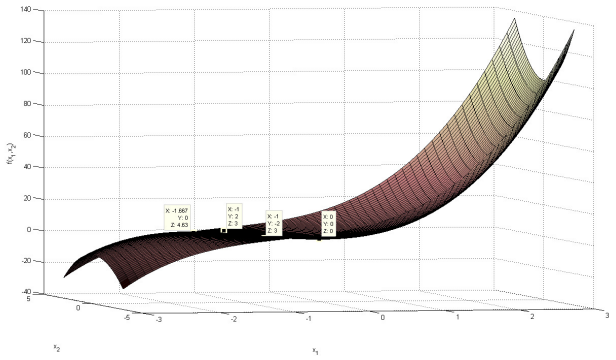

Figure 4.22: Plot of the function $2x_1^3 + x_1x_2^2 + 5x_1^2 + x_2^2$ showing the four critical points.

We will take some more examples:

1. Consider a significantly harder function $f(x, y) = 10x^2y - 5x^2 - 4y^2 - x^4 - 2y^4$. Let us find and classify its critical points. The gradient vector is $\nabla f(x, y) = [20xy - 10x - 4x^3, \quad 10x^2 - 8y - 8y^3]$. The critical points correspond to solutions of the simultaneous set of equations

$$\begin{aligned} 20xy - 10x - 4x^3 &= 0 \\ 10x^2 - 8y - 8y^3 &= 0 \end{aligned} \tag{4.15}$$

One of the solutions corresponds to solving the system $-8y^3 + 42y - 25 = 0$[8] and $10x^2 = 50y - 25$, which have four real solutions[9], *viz.*, $(0.8567, 0.646772)$, $(-0.8567, 0.646772)$, $(2.6442, 1.898384)$, and $(-2.6442, 1.898384)$. Another real solution is $(0, 0)$. The mixed partial derivatives of the function are

$$\begin{aligned} f_{xx} &= 20y - 10 - 12x^2 \\ f_{xy} &= 20x \\ f_{yy} &= -8 - 24y^2 \end{aligned} \tag{4.16}$$

Using theorem 64, we can verify that $(2.6442, 1.898384)$ and $(-2.6442, 1.898384)$ correspond to local maxima whereas $(0.8567, 0.646772)$ and $(-0.8567, 0.646772)$ correspond to saddle points. This is illustrated in Figure 4.23.

[8]Solving this using matlab without proper scaling could give you complex values. With proper scaling of the equation, you should get $y = -2.545156$ or $y = 0.646772$ or $y = 1.898384$.

[9]The values of x corresponding to $y = -2.545156$ are complex

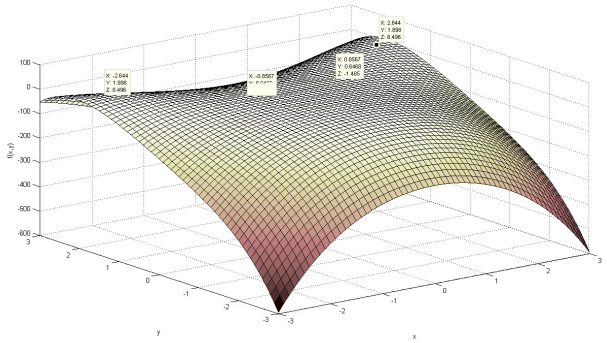

Figure 4.23: Plot of the function $10x^2y - 5x^2 - 4y^2 - x^4 - 2y^4$ showing the four critical points.

2. The function $f(x, y) = x \sin y$ has the gradient vector $[\sin y, \quad x \cos y]$. The critical points correspond to the solutions to the simultaneous set of equations

$$\begin{aligned} \sin y &= 0 \\ x \cos y &= 0 \end{aligned} \tag{4.17}$$

The critical points are[10] $(0, n\pi)$ for $n = 0, \pm 1, \pm 2, \ldots$. The mixed partial derivatives of the function are

$$\begin{aligned} f_{xx} &= 0 \\ f_{xy} &= \cos y \\ f_{yy} &= -x \sin y \end{aligned} \tag{4.18}$$

which tell us that the discriminant function $D = -\cos^2 y$ is always negative. Therefore, all the critical points turn out to be saddle points. This is illustrated in Figure 4.24.

Along similar lines of the single variable case, we next define the global maximum and minimum.

Definition 29 [Global maximum]: *A function f of n variables, with domain $\mathcal{D} \subseteq \Re^n$ has an absolute or global maximum at $\mathbf{x}^0$ if $\forall\ \mathbf{x} \in \mathcal{D}$, $f(\mathbf{x}) \leq f(\mathbf{x}^0)$.*

[10] Note that the *cosine* does not vanish wherever the *sin* vanishes.

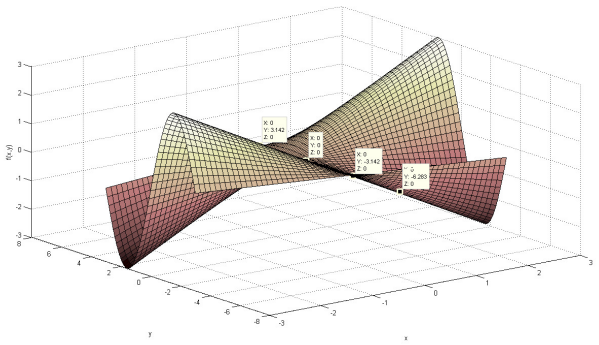

Figure 4.24: Plot of the function $x \sin y$ illustrating that all critical points are saddle points.

Definition 30 [Global minimum]: *A function f of n variables, with domain $\mathcal{D} \subseteq \Re^n$ has an absolute or global minimum at $\mathbf{x}^0$ if $\forall\ \mathbf{x} \in \mathcal{D}$, $f(\mathbf{x}) \geq f(\mathbf{x}^0)$.*

We would like to find the absolute maximum and minimum values of a function of multiple variables in a closed interval, along similar lines of the method yielded by theorem 41 for functions of single variable. The procedure was to evaluate the value of the function at the critical points as well as the end points of the interal and determine the absolute maximum and minimum values by scanning this list. To generalize the idea to function of multiple variables, we point out that the analogue of finding the value of the function at the boundaries of closed interval in the single variable case is to find the function value along the boundary curve, which reduces the evaluation of a function of multiple variables to evaluating a function of a single variable. Recall from the definitions on page 214 that a closed set in $\Re^n$ is a set that contains its boundary points (analogous to closed interval in $\Re$) while a bounded set in $\Re^n$ is a set that is contained inside a closed ball, $\mathcal{B}[\mathbf{0}, \epsilon]$. An example bounded set is $\left\{(x_1, x_2, x_3) | x_1^2 + x_2^2 + x_3^2 \leq 1\right\}$. An example unbounded set is $\{(x_1, x_2, x_3) | x_1 > 1, x_2 > 1, x_3 > 1\}$. Based on these definitions, we can state the extreme value theorem for a function of n variables.

Theorem 65 *Let $f : \mathcal{D} \to \Re$ where $\mathcal{D} \subseteq \Re^n$ is a closed bounded set and f be continuous on $\mathcal{D}$. Then f attains an absolute maximum and absolute minimum at some points in $\mathcal{D}$.*

The theorem implies that whenever a function of n variables is restricted to a bounded space, it has an absolute maximum and an absolute minimum. Following theorem 60, we note that the locally extreme values of a function occur at its critical points. By the very definition of local extremum, it cannot occur at the boundary point of $\mathcal{D}$. Since every absolute extremum is also a

Figure 4.25: The region bounded by the points $(0,3)$, $(2,0)$, $(0,0)$ on which we consider the maximum and minimum of the function $f(x,y) = 1 + 4x - 5y$.

local extremum, the absolute maximum and minimum of a function on a closed, bounded set will either happen at the critical points or at the boundary. The procedure for finding the absolute maximum and minimum of a function on a closed bounded set is outlined below and is similar to the procedure 4 for a function of single variable continuous on a closed and bounded interval:

Procedure 5 [Finding extreme values on closed, bounded sets]: *To find the absolute maximum and absolute minimum of a continuous function f on a closed bounded set $\mathcal{D}$;*

- *evaluate f at the critical points of f on $\mathcal{D}$*
- *find the extreme values of f on the boundary of $\mathcal{D}$*
- *the largest of the values found above is the absolute maximum, and the smallest of them is the absolute mininum.*

We will take some examples to illustrate procedure 5.

1. Consider the function $f(x,y) = 1 + 4x - 5y$ defined on the region $\mathcal{R}$ bounded by the points $(0,3)$, $(2,0)$, $(0,0)$. The region $\mathcal{R}$ is shown in Figure 4.25 and is bounded by three line segments

   - $\mathbf{B}_1$: $x = 0$, $0 \leq y \leq 3$
   - $\mathbf{B}_2$: $y = 0$, $0 \leq x \leq 2$
   - and $\mathbf{B}_3$: $y = 3 - \frac{3}{2}x$, $0 \leq x \leq 2$.

   The linear function $f(x,y) = 1 + 4x - 5y$ has no critical points, since $\nabla f(x,y) = [4, \ \ 5]^T$ is defined everywhere, though it cannot disappear at any point. In fact, linear functions have no critical points and the extreme values are always assumed at the boundaries; this forms the basis of linear programming. We will find the extreme values on the boundaries.

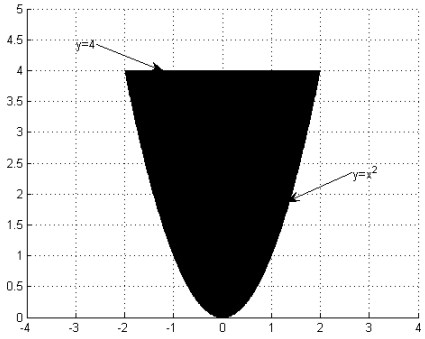

Figure 4.26: The region $\mathcal{R}$ bounded by $y = x^2$ and $y = 4$ on which we consider the maximum and minimum of the function $f(x, y) = 1 - xy - x - y$.

- On $\mathbf{B}_1$, $f(x, y) = f(0, y) = 1 - 5y$, for $y \in [0, 3]$. This is a single variable extreme value problem for a continuous function. Its largest value is assumed at $y = 0$ and equals 1 while the smallest value is assumed at $y = 3$ and equals -14.
- On $\mathbf{B}_2$, $f(x, y) = f(x, 0) = 1 + 4x$, for $x \in [0, 2]$. This is again a single variable extreme value problem for a continuous function. Its largest value is assumed at $x = 2$ and equals 9 while the smallest value is assumed at $x = 0$ and equals 1.
- On $\mathbf{B}_3$, $f(x, y) = 1 + 4x - 5\,(3 - (3/2)x) = -14 + (23/2)x$, for $x \in [0, 2]$. This is also a single variable extreme value problem for a continuous function. Its largest value is assumed at $x = 2$ and equals 9 while the smallest value is assumed at $x = 0$ and equals -14.

Thus, the absolute maximum is attained by f at $(2, 0)$ while the absolute minimum is attained at $(0, 3)$. Both extrema are at the vertices of the polygon (triangle) This example illustrates the general procedure for determining the absolute maximum and minimum of a function on a closed, bounded set. However, the problem can become very hard in practice as the function f gets complex.

2. Let us look at a harder problem. Let us find the absolute maximum and the absolute minimum of the function $f(x, y) = 1 - xy - x - y$ on the region $\mathcal{R}$ bounded by $y = x^2$ and $y = 4$. This is not a linear function any longer. The region $\mathcal{R}$ is shown in Figure 4.26 and is bounded by

   - $\mathbf{B}_1$: $y = x^2$, $-2 \leq x \leq 2$
   - $\mathbf{B}_2$: $y = 4$, $-2 \leq x \leq 2$

   Since $f(x, y) = 1 - xy - x - y$ is differentiable everywhere, the critical point of f is characterized by $\nabla f(x, y) = [-y - 1,\ x - 1]^T = \mathbf{0}$, that is

$x = -1, \quad y = -1$. However, this point does not lie in $\mathcal{R}$ and hence, there are no critical points, in $\mathcal{R}$. Along similar lines of the previous problem, we will find the extreme values of f on the boundaries of $\mathcal{R}$.

- On $\mathbf{B}_1$, $f(x, y) = 1 - x^3 - x - x^2$, for $x \in [-2, 2]$. This is a single variable extreme value problem for a continuous function. Its critical points correspond to solutions of $3x^2 + 2x + 1 = 0$. However, this equation has no real solutions[11] and therefore, the function's extreme values are only at the boundary points; the minimum value -13 is attained at $x = 2$ and the maximum value 7 is attained at $x = -2$.
- On $\mathbf{B}_2$, $f(x, y) = 1 - 4x - x - 4 = -3 - 5x$, for $x \in [-2, 2]$. This is again a single variable extreme value problem for a continuous function. It has no critical points and extreme values correspond to the boundary points; its maximum value 7 is assumed at $x = -12$ while the minimum value -13 is assumed at $x = 2$.

Thus, the absolute maximum value 7 is attained by f at $(-2, 4)$ while the absolute minimum value -13 is attained at $(2, 4)$.

3. Consider the same problem as the previous one, with a slightly different objective function, $f(x, y) = 1 + xy - x - y$. The critical point of f is characterized by $\nabla f(x, y) = [y - 1, \quad x - 1]^T = \mathbf{0}$, that is $x = 1, \quad y = 1$. This lies within $\mathcal{R}$ and f takes the value 0 at $(1, 1)$. Next, we find the extreme values of f on the boundaries of $\mathcal{R}$.

   - On $\mathbf{B}_1$, $f(x, y) = 1 + x^3 - x - x^2$, for $x \in [-2, 2]$. Its critical points correspond to solutions of $3x^2 - 2x - 1 = 0$. Its solutions are $x = 1$ and $x = -\frac{1}{3}$. The function values corresponding to these points are $f(1, 1) = 0$ and $f(-1/3, 1/9) = 32/27$. At the boundary points, the function assumes the values $f(-2, 4) = -9$ and $f(2, 4) = 3$. Thus, the maximum value on $\mathbf{B}_1$ is $f(2, 4) = 3$ and the minimum value is $f(-2, 4) = -9$.
   - On $\mathbf{B}_2$, $f(x, y) = 1 + 4x - x - 4 = -3 + 3x$, for $x \in [-2, 2]$. It has no critical points and extreme values correspond to the boundary points; At the boundary points, the function assumes the values $f(-2, 4) = -9$ and $f(2, 4) = 3$, which correspond to the minimum and maximum values respectively of f on $\mathbf{B}_2$.

   Thus, the absolute maximum value 3 is attained by f at $(2, 4)$ while the absolute minimum value -9 is attained at $(-2, 4)$.

4.1.5 Absolute extrema and Convexity

Theorem 61 specified a sufficient condition for the local minimum of a differentiable function with continuous partial and mixed partial derivatives, while

[11] The complex solutions are $x = -\frac{1}{3} + i\frac{1}{3}\sqrt{2}$ and $x = -\frac{1}{3} - i\frac{1}{3}\sqrt{2}$.

theorem 62 specified a necessary condition for the same. Can these conditions be extended to globally optimal solutions? The answer is that the extensions to globally optimal solutions can be made for a specific class of optimization problems called convex optimization problems. In the next section we introduce the concept of convex sets and convex functions, enroute to discussing convex optimization.

4.2 Convex Optimization Problem

A function $f(.)$ is called convex if its value at the scalar combination of two points x and y is less than the same scalar combination of the function at the two points. In other words, $f(.)$ is convex if and only if:

$$\begin{aligned} &f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \\ &\text{if } \alpha + \beta = 1, \alpha \geq 0, \beta \geq 0 \end{aligned} \tag{4.19}$$

For a convex optimization problem, the objective function $f(x)$ as well as the inquality functions $g_i(x), i = 1, \ldots, m$ are convex. The equality constraints are linear, *i.e.*, of the form, $Ax = b$.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array} \tag{4.20}$$

Least squares and linear programming are special cases of convex optimization problems. Like in the case of linear programming, there are no analytical solutions for convex optimization problems. But they can be solved reliably, efficiently and optimally. There are not many well developed software for the general class of convex optimization problems, though there are several software packages in matlab, C, *etc.*, and many free softwares as well. The computation time is polynomial but more complicated to be expressed exactly because the computation time depends on the cost of validating the function values and their derivates. Modulo that, computation time for convex optimization problems is similar to that for linear programming problems.

To pose pratical problems as convex optimization problems is more difficult than to recognize least squares and linear programs. There exist many techniques to reformulate problems in the convex form. However, surprisingly, many problems in practice can be solved via convex optimization.

4.2.1 Why Convex Optimization?

We will see in this sequel, that generic convex programs, under mild computability and boundedness assumptions, are computationally tractable. Many convex

programs admit theoretically and practically efficient solution methods. Convex optimization admits *duality theory*, which can be used to quantitatively establish the quality of an approximate solution. Even though duality may not yield a closed-form solution, it often facilitates nontrivial reformulations of the problem. Duality theory also comes handy in confirming if an approximate solution is optimal.

In contrast to this, rarely does it happen that a global solution can be efficiently found for nonconvex optimization programs[12]. For most nonconvex programs, there are no sound techniques for certifying the global optimalality of approximate solutions or estimating how non-optimal an approximate solution is.

4.2.2 History

Numerical optimization started in the 1940s with the development of the simplex method for linear programming. The next obvious extension to linear programming was by replacing the linear cost function with a quadratic cost function. The linear inequality constraints were however maintained. This first extension took place in the 1950s. We can expect that the next step would have been to replace the linear constraints with quadratic constraints. But it did not really happen that way. On the other hand, around the end of the 1060s, there was another non-linear, convex extension of linear programming called *geometric programming*. Geometric programming includes linear programming as a special case. Nothing more happened until the beginning of the 1990s. The beginning of the 1990s was marked by a big explosion of activities in the area of convex optimizations, and development really picked up. Researches formulated different and more general classes of convex optimization problems that are known as semidefinite programming, second-order cone programming, quadratically constrained quadratic programming, sum-of-squares programming, *etc.*

The same happened in terms of applications. Since 1990s, applications have been investigated in many different areas. One of the first application areas was control, and the optimization methods that were investigated included semidefinite programming for certain control problem. Geometric programming had been around since late 1960s and it was applied extensively to circuit design problems. Quadratic programming found application in machine learning problem formulations such as support vector machines. Semi-definite programming relaxations found use in combinatorial optimization. There were many other interesting applications in different areas such as image processing, quantum information, finance, signal processing, communications, *etc.*

This first look at the activities involving applications of optimization clearly indicates that a lot a of development took place around the 1990s. Further, people extended interior-point methods (which were already known for linear

[12]Optimization problems such as singular value decomposition are some few exceptions to this.

programming since 1984[13]) to non-linear convex optimization problems. A highlight in this area was the work of Nesterov and Nemirovski who extended Karmarkar's work to polynomial-time interior-point methods for nonlinear convex programming in their book published in 1994, though the work actually took place in 1990. As a result, people started looking at non-linear convex optimization in a special way; instead of treating non-linear convex optimization as a special case of non-linear optimization, they looked at it as an extension of linear programming which can be solved almost with the same efficiency. Once people started looking at applications of non-linear convex optimization, they discovered many!

We will begin with a background on convex sets and functions. Convex sets and functions constitute the basic theory for the entire area of convex optimization. Next, we will discuss some standard problem classes and some recently studied problem classes such as semi-definite programming and cone programming. FInally, we will look at applications.

4.2.3 Affine Set

Definition 31 [Affine Set]: *A set $\mathcal{A}$ is called affine if the line connecting any two distinct points in the set is completely contained within $\mathcal{A}$. Mathematically, the set $\mathcal{A}$ is called affine if*

$$\forall\ \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{A}, \quad \theta \in \Re \quad \Rightarrow \quad \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{A} \tag{4.21}$$

Theorem 66 *The solution set of the system of linear equations $A\mathbf{x} = \mathbf{b}$ is an affine set.*

Proof: Suppose $\mathbf{x}_1$ and $\mathbf{x}_2$ are solutions to the system $A\mathbf{x} = \mathbf{b}$ with $\mathbf{x}_1 \neq \mathbf{x}_2$. Then, $A(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2) = \theta\mathbf{b} + (1-\theta)\mathbf{b} = \mathbf{b}$. Thus, $\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{A}$, implying that the solution set of the system $A\mathbf{x} = \mathbf{b}$ is an affine set. □

In fact, converse of theorem 66 is also true; any affine set can be expressed as the solution set of a system of linear equations $A\mathbf{x} = \mathbf{b}$.

4.2.4 Convex Set

Definition 32 [Convex Set]: *A set $\mathcal{C}$ is called convex if the line segment connecting any two points in the set is completely contained within $\mathcal{C}$. Else $\mathcal{C}$ is called concave. That is,*

$$\forall\ \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C} \quad 0 \leq \theta \leq 1 \quad \Rightarrow \quad \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{C} \tag{4.22}$$

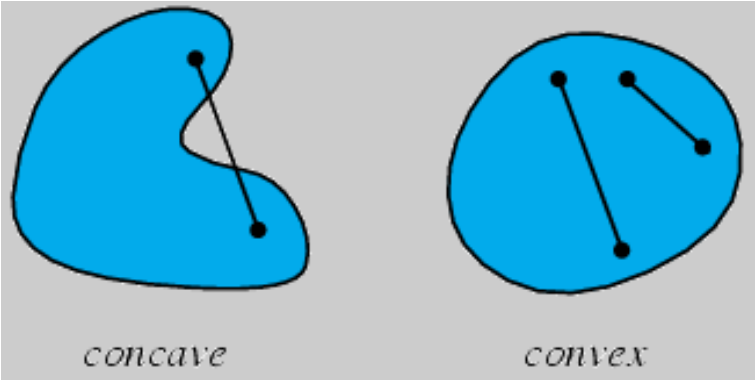

Figure 4.27: Examples of convex and non-convex sets.

Figure 4.27 shows examples of convex and non-convex (concave) sets. Since an affine set contains any line passing through two distinct points in the set, it also contains any line segment connecting two points in the set. Thus, an affine set is our first example of a convex set.

A set $\mathcal{C}$ is a convex cone if it is covex and additionally, for every point $\mathbf{x} \in \mathcal{C}$, all non-negative multiples of $\mathbf{x}$ are also in $\mathcal{C}$. In other words,

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in C \quad \theta_1, \theta_2 \geq 0 \quad \Rightarrow \quad \theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 \in \mathcal{C} \tag{4.23}$$

Combinations of the form $\theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2$ for $\theta_1 \geq 0, \theta_2 \geq 0$ are called conic combinations. We will state a related definition next - that of the convex hull of a set of points.

Definition 33 [Convex Hull]: *A convex combination of the set of points* $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ *is any point* $\mathbf{x}$ *of the form*

$$\mathbf{x} = \sum_{i=1}^{k} \theta_i \mathbf{x}_i \quad \textit{with} \sum_{i=1}^{k} \theta_i = 1 \quad \textit{and } \theta_i \geq 0 \tag{4.24}$$

The convex hull $conv(\mathcal{S})$ *of the set of points* $\mathcal{S}$ *is the set of all convex combinations of points in* $\mathcal{S}$*. The convex hull of a convex set* $\mathcal{S}$ *is* $\mathcal{S}$ *itself.*

[13] The first practical polynomial time algorithm for linear programming by Karmarkar (1984) involved interior-point methods.

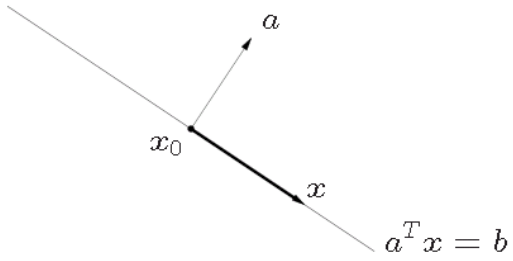

Figure 4.28: Example of a hyperplane in $\Re^2$.

4.2.5 Examples of Convex Sets

We will look at simple but important examples of convex sets. We will also look at some operations that preserve convexity.

A *hyperplane* is the most common example of a convext set. A hyperplane is the set of solutions to a linear system of equations of the form $a^T\mathbf{x} = b$ with $a \neq 0$ and was defined earlier in definition 24. A *half space* is a solution set over the linear inequality $a^T\mathbf{x} \leq b, a \neq 0$. The hyperplane $a^T\mathbf{x} = b$ bounds the half-space from one side.

Formally,

Hyperplane: $\{\mathbf{x}|a^T\mathbf{x} = b, a \neq 0\}$. Figure 4.28 shows an example hyperplane in $\Re^2$. a is the normal vector.

Halfspace: $\{\mathbf{x}|a^T\mathbf{x} \leq b, a \neq 0\}$. Figure 4.29 shows an example half-space in $\Re^2$.

The hyperplane is convex and affine, whereas the halfspace is merely convex and not affine.

Another simple example of a convex set is a *closed ball* in $\Re^n$ with radius r and center $\mathbf{x}_c$ which is an n-dimensional vector.

$$\mathcal{B}[\mathbf{x}_c, r] = \{\mathbf{x}_c + r\mathbf{u} \mid ||\mathbf{u}||_2 \leq 1\}$$

where $\mathbf{u}$ is a vector with norm less than or equal to 1. The open ball $\mathcal{B}(\mathbf{x}_c, r)$ is also convex. Replacing r with a non-singular square matrix A, we get an *ellipsoid* given by

$$\{\mathbf{x}_c + A\mathbf{u} \mid ||\mathbf{u}||_2 \leq 1\}$$

which is also a convex set. Another equivalent representation of the ellipsoid can be obtained by observing that for any point $\mathbf{x}$ in the ellipsoid, $||A^{-1}(\mathbf{x}-\mathbf{x}_c)||_2 \leq 1$, that is $(\mathbf{x} - \mathbf{x}_c)^T(A^{-1})^T A^{-1}(\mathbf{x} - \mathbf{x}_c) \leq 1$. Since $(A^{-1})^T = (A^T)^{-1}$ and

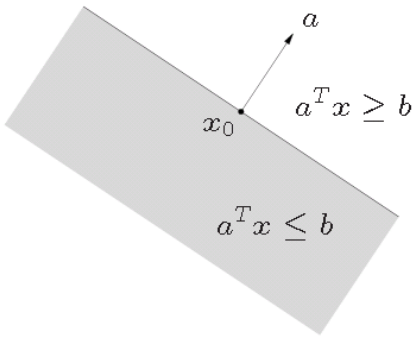

Figure 4.29: Example of a half-space in $\Re^2$.

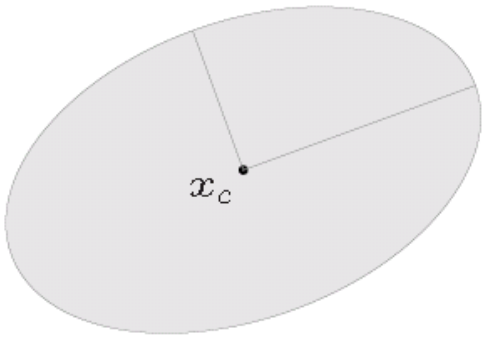

Figure 4.30: Example of a ellipsoid in $\Re^2$.

$A^{-1}B^{-1} = (BA)^{-1}$, the ellipsoid can be equivalently defined as $\left\{\mathbf{x}|(\mathbf{x}-\mathbf{x}_c)^T P^{-1}(\mathbf{x}-\mathbf{x}_c) \leq 1\right\}$ where $P = (AA^T)$ is a symmetric matrix. Furthermore, P is positive definite, since A is non-singular (*c.f.* page 208).

Matrix A determines the size of the ellipsoid; the eigenvalue λ_i of A determines the length of the i^{th} semi-axis of the ellipsoid (see page number 206). The ellipsoid is another example of a convex set and is a generalization of the eucledian ball. Figure 4.30 illustrates an ellipsoid in $\Re^2$.

A *norm ball* is a ball with an arbitrary norm. A norm ball with center $\mathbf{x}_c$ and radius r is given by

$$\{\mathbf{x} \mid ||\mathbf{x} - \mathbf{x}_c|| \leq r\}$$

By the definition of the norm, a ball in that norm will be convex. The norm ball with the $\infty-$norm corresponds to a square in $\Re^2$, while the norm ball with the $1-$norm in $\Re^2$ corresponds to the same square rotated by $45°$. The norm ball is convex for all norms.

The definition of cone can be extended to any arbitrary norm to define a

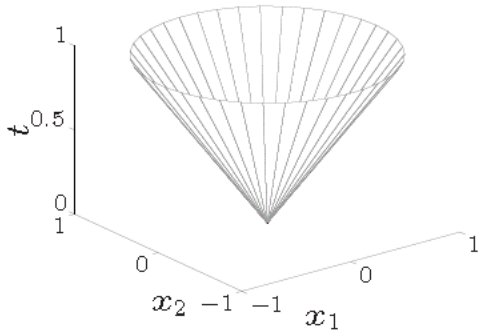

Figure 4.31: Example of a cone in $\Re^2$.

norm cone. The set of all pairs $(\mathbf{x}, t)$ satisfying $||\mathbf{x}|| \leq t$, *i.e.*,

$$\{(\mathbf{x}, t) \mid ||\mathbf{x}|| \leq t\}$$

is called a norm cone

When the norm is the eucledian norm, the cone (which looks like an ice-cream cone) is called the *second order cone*. Norm cones are always convex. Figure 4.31 shows a cone in $\Re^2$. In general, the cross section of a norm cone has the shape of a norm ball with the same norm. The norm cone for the $\infty-$norm is a square pyramid in $\Re^3$ and the cone for $1-$norm in $\Re^3$ is the same square pyramid rotated by $45°$.

A *polyhedron* is another convex set which is given as the solution set of a finite set of linear equalities and inequalities. In matrix form, the inequalities can be stated as

$$\begin{array}{ll} A\mathbf{x} \preceq \mathbf{b} & A \in \Re^{m \times n} \\ C\mathbf{x} = \mathbf{d} & C \in \Re^{p \times n} \end{array} \tag{4.25}$$

where $\preceq$ stands for component-wise inequality of the form $\leq$[14]. A polyhedron can also be represented as the intersection of a finite number of halfspaces and hyperplanes. Figure 4.32 depicts a typical polyhedron in $\Re^2$. An affine set is a special type of polyhedron.

A last simple example is the positive semi-definite cone. Let $\mathcal{S}^n$ be the set of all symmetric $n \times n$ matrices and $\mathcal{S}^n_+ \subset \mathcal{S}^n$ be the set of all positive semi-definite $n \times n$ matrices. The set $\mathcal{S}^n_+$ is a convex cone and is called the *positive semi-definite cone*. Consider a positive semi-definite matrix S in $\Re^2$. Then S must of the form

[14]The component-wise inequality corresponds to a generalized inequality $\preceq_K$ with $K = \Re^n_+$.

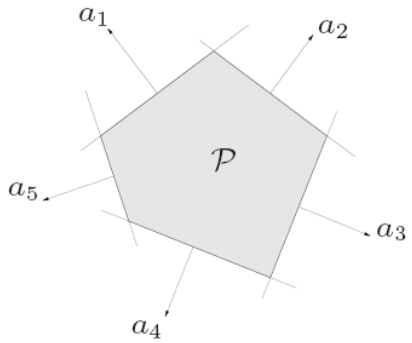

Figure 4.32: Example of a polyhedron in $\Re^2$.

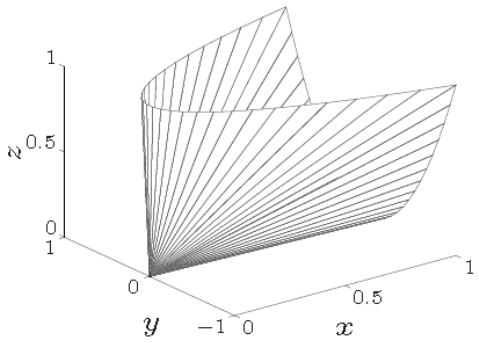

Figure 4.33: Example of a positive semidefinite cone in $\Re^2$.

$$S = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \tag{4.26}$$

We can represent the space of matrices $\mathcal{S}_+^2$ of the form $S \in \mathcal{S}_+^2$ as a three dimensional space with non-negative x, y and z coordinates and a non-negative determinant. This space corresponds to a cone as shown in Figure 4.33.

4.2.6 Convexity preserving operations

In practice if you want to establish the convexity of a set $\mathcal{C}$, you could either

1. prove it from first principles, *i.e.*, using the definition of convexity or
2. prove that $\mathcal{C}$ can be built from simpler convex sets through some basic operations which preserve convexity.

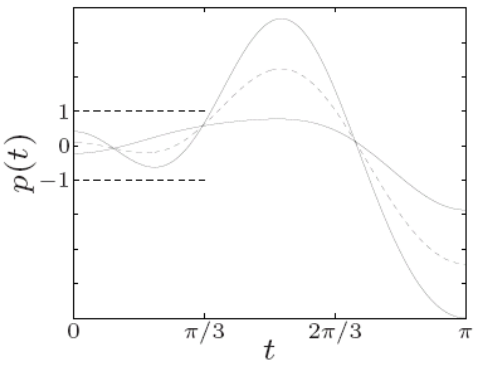

Figure 4.34: Plot for the function in (4.28)

Some of the important operations that preserve complexity are:

Intersection

The intersection of any number of convex sets is convex[15]. Consider the set $\mathcal{S}$:

$$\mathcal{S} = \left\{ \mathbf{x} \in \Re^n \mid |p(t)| \leq 1 \ for \ |t| \leq \frac{\pi}{3} \right\} \tag{4.27}$$

where

$$p(t) = x_1 \cos t + x_2 \cos 2t + \ldots + x_m \cos mt \tag{4.28}$$

Any value of t that satisfies $|p(t)| \leq 1$, defines two regions, *viz.*,

$$\Re^{\leq}(t) = \{\mathbf{x} \mid x_1 \cos t + x_2 \cos 2t + \ldots + x_m \cos mt \leq 1\}$$

and

$$\Re^{\geq}(t) = \{\mathbf{x} \mid x_1 \cos t + x_2 \cos 2t + \ldots + x_m \cos mt \geq -1\}$$

Each of the these regions is convex and for a given value of t, the set of points that may lie in $\mathcal{S}$ is given by

$$\Re(t) = \Re^{\leq}(t) \cap \Re^{\geq}(t)$$

This set is also convex. However, not all the points in $\Re(t)$ lie in $\mathcal{S}$, since the points that lie in $\mathcal{S}$ satisfy the inequalities for every value of t. Thus, $\mathcal{S}$ can be given as:

$$\mathcal{S} = \cap_{|t| \leq \frac{\pi}{3}} \Re(t)$$

[15] Exercise: Prove.

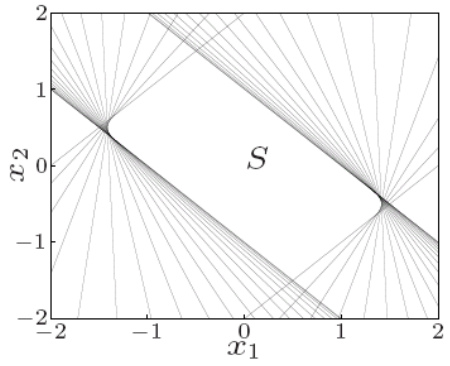

Figure 4.35: Illustration of the closure property for $\mathcal{S}$ defined in (4.27), for $m = 2$.

Affine transform

An affine transform is one that preserves

- Collinearity between points, *i.e.*, three points which lie on a line continue to be collinear after the transformation.
- Ratios of distances along a line, *i.e.*, for distinct colinear points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, $\frac{||\mathbf{p}_2 - \mathbf{p}_1||}{||\mathbf{p}_3 - \mathbf{p}_2||}$ is preserved.

An affine transformation or affine map between two vector spaces $f : \Re^n \to \Re^m$ consists of a linear transformation followed by a translation:

$$\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$$

where $A \in \Re^{n \times m}$ and $\mathbf{b} \in \Re^m$. In the finite-dimensional case each affine transformation is given by a matrix A and a vector $\mathbf{b}$.

The image and pre-image of convex sets under an affine transformation defined as

$$f(\mathbf{x}) = \sum_i^n x_i a_i + b$$

yield convex sets[16]. Here a_i is the i^{th} row of A. The following are examples of convex sets that are either images or inverse images of convex sets under affine transformations:

1. the solution set of linear matrix inequality ($A_i, B \in \mathcal{S}^m$)

$$\{\mathbf{x} \in \Re^n \mid x_1 A_1 + \ldots + x_n A_n \preceq B\}$$

[16] Exercise: Prove.

is a convex set. Here $A \preceq B$ means $B - A$ is positive semi-definite[17]. This set is the inverse image under an affine mapping of the positive semi-definite cone. That is, $f^{-1}(cone) = \left\{\mathbf{x} \in \Re^n \mid B - (x_1A_1 + \ldots + x_nA_n) \in \mathcal{S}_+^m\right\} = \{\mathbf{x} \in \Re^n | B \geq (x_1A_1 + \ldots + x_nA_n)\}$.

2. hyperbolic cone ($P \in \mathcal{S}_+^n$), which is the inverse image of the norm cone $\mathcal{C}_{m+1} = \{(\mathbf{z}, u) | ||\mathbf{z}|| \leq u, u \geq 0, \mathbf{z} \in \Re^m\} = \left\{(\mathbf{z}, u) | \mathbf{z}^T\mathbf{z} - u^2 \leq 0,\ u \geq 0, \mathbf{z} \in \Re^m\right\}$ is a convex set. The inverse image is given by $f^{-1}(\mathcal{C}_{m+1}) = \left\{\mathbf{x} \in \Re^n \mid \left(A\mathbf{x}, \mathbf{c}^T\mathbf{x}\right) \in \mathcal{C}_{m+1}\right\} = \left\{\mathbf{x} \in \Re^n | \mathbf{x}^TA^TA\mathbf{x} - (\mathbf{c}^T\mathbf{x})^2 \leq 0\right\}$. Setting, $P = A^TA$, we get the equation of the hyperbolic cone:

$$\left\{\mathbf{x} \mid \mathbf{x}^TP\mathbf{x} \leq (\mathbf{c}^T\mathbf{x})^2, \mathbf{c}^T\mathbf{x} \geq 0\right\}$$

Perspective and linear-fractional functions

The perspective function $P : \Re^{n+1} \to \Re^n$ is defined as follows:

$$\begin{array}{ll} P : \Re^{n+1} \to \Re^n \text{ such that} & \\ P(x,t) = x/t & \textbf{dom } P = \{(x,t) \mid t > 0\} \end{array} \tag{4.29}$$

The linear-fractional function f is a generalization of the perspective function and is defined as: $\Re^n \to \Re^m$:

$$\begin{array}{ll} f : \Re^n \to \Re^m \text{ such that} & \\ f(\mathbf{x}) = \frac{A\mathbf{x}+\mathbf{b}}{\mathbf{c}^T\mathbf{x}+d} & \textbf{dom } f = \{\mathbf{x} \mid \mathbf{c}^T\mathbf{x} + d > 0\} \end{array} \tag{4.30}$$

The images and inverse images of convex sets under perspective and linear-fractional functions are convex[18].

Consider the linear-fractional function $f = \frac{1}{x_1+x_2+1}x$. Figure **??** shows an example convex set. Figure **??** shows the image of this convex set under the linear-fractional function f.

Supporting Hyperplane Theorem

On page 4.1.4, we introduced the concept of the hyperplane. For disjoint convex sets, we state the *separating hyperplane theorem*.

Theorem 67 *If $\mathcal{C}$ and $\mathcal{D}$ are disjoint convex sets, i.e., $\mathcal{C} \cap \mathcal{D} = \phi$, then there exists $\mathbf{a} \neq \mathbf{0}$, with a $b \in \Re$ such that*
$\mathbf{a}^T\mathbf{x} \leq \mathbf{b}$ *for* $\mathbf{x} \in \mathcal{C}$,
$\mathbf{a}^T\mathbf{x} \geq \mathbf{b}$ *for* $\mathbf{x} \in \mathcal{D}$.
That is, the hyperplane $\left\{\mathbf{x} | \mathbf{a}^T\mathbf{x} = \mathbf{b}\right\}$ separates $\mathcal{C}$ and $\mathcal{D}$. The seperating hyperplane need not be unique though.

[17] The inequality induced by positive semi-definiteness corresponds to a generalized inequality $\preceq_K$ with $K = \mathcal{S}_+^n$.
[18] Exercise: Prove.

Proof: We first note that the set $\mathcal{S} = \{\mathbf{x} - \mathbf{y} | \mathbf{x} \in \mathcal{C}, \mathbf{y} \in \mathcal{D}\}$ is convex, since it is the sum of two convex sets. Since $\mathcal{C}$ and $\mathcal{D}$ are disjoint, $\mathbf{0} \notin \mathcal{S}$. Consider two cases:

1. Suppose $\mathbf{0} \notin closure(\mathcal{S})$. Let $\mathcal{E} = \{0\}$ and $\mathcal{F} = closure(\mathbf{S})$. Then, the euclidean distance between $\mathcal{E}$ and $\mathcal{F}$, defined as

   $dist(\mathcal{E}; \mathcal{F}) = inf\{||\mathbf{u} - \mathbf{v}||_2 | \mathbf{u} \in \mathcal{E}, \mathbf{v} \in \mathcal{F}\}$

   is positive, and there exists a point $\mathbf{f} \in \mathcal{F}$ that achieves the minimum distance, i.e., $||\mathbf{f}||_2 = dist(\mathcal{E}, \mathcal{F})$. Define $\mathbf{a} = \mathbf{f}$, $\mathbf{b} = ||\mathbf{f}||_2$. Then $\mathbf{a} \neq \mathbf{0}$ and the affine function $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} - b = \mathbf{f}^T(\mathbf{x} - \frac{1}{2}\mathbf{f})$ is nonpositive on $\mathcal{E}$ and nonnegative on $\mathcal{F}$, *i.e.*, that the hyperplane $\left\{\mathbf{x} | \mathbf{a}^T\mathbf{x} = b\right\}$ separates $\mathcal{E}$ and $\mathcal{F}$. Thus, $\mathbf{a}^T(\mathbf{x} - \mathbf{y}) > 0$ for all $\mathbf{x} - \mathbf{y} \in \mathcal{S} \subseteq closure(\mathcal{S})$, which implies that, $\mathbf{a}^T\mathbf{x} \geq \mathbf{a}^T\mathbf{y}$ for all $\mathbf{x} \in \mathcal{C}$ and $\mathbf{y} \in \mathcal{D}$.

2. Suppose, $0 \in closure(\mathcal{S})$. Since $0 \notin \mathcal{S}$, it must be in the boundary of $\mathcal{S}$.

   - If $\mathcal{S}$ has empty interior, it must lie in an affine set of dimension less than n, and any hyperplane containing that affine set contains $\mathcal{S}$ and is a hyperplane. In other words, $\mathcal{S}$ is contained in a hyperplane $\left\{\mathbf{z} | \mathbf{a}^T\mathbf{z} = b\right\}$, which must include the origin and therefore $b = 0$. In other words, $\mathbf{a}^T\mathbf{x} = \mathbf{a}^T\mathbf{y}$ for all $\mathbf{x} \in \mathcal{C}$ and all $\mathbf{y} \in \mathcal{D}$ gives us a trivial separating hyperplane.
   - If $\mathcal{S}$ has a nonempty interior, consider the set

     $\mathcal{S}_{-\epsilon} = \{\mathbf{z} | B(\mathbf{z}, \epsilon) \subseteq \mathcal{S}\}$

     where $B(\mathbf{z}, \epsilon)$ is the Euclidean ball with center $\mathbf{z}$ and radius $\epsilon > 0$. $\mathcal{S}_{-\epsilon}$ is the set $\mathcal{S}$, shrunk by ϵ. $closure\left(\mathcal{S}_{-\epsilon}\right)$ is closed and convex, and does not contain $\mathbf{0}$, so as argued before, it is separated from $\{\mathbf{0}\}$ by atleast one hyperplane with normal vector $\mathbf{a}(\epsilon)$ such that

     $\mathbf{a}(\epsilon)^T\mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathcal{S}_\epsilon$

     Without loss of generality assume $||\mathbf{a}(\epsilon)||_2 = 1$. Let ϵ_k, for $k = 1, 2, \ldots$ be a sequence of positive values of ϵ_k with $\lim_{k \to \infty} \epsilon_k = 0$. Since $||\mathbf{a}(\epsilon_k)||_2 = 1$ for all k, the sequence $\mathbf{a}(\epsilon_k)$ contains a convergent subsequence, and let $\overline{\mathbf{a}}$ be its limit. We have

     $\mathbf{a}(\epsilon_k)^T\mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathcal{S}_{-\epsilon_k}$

     and therefore $\overline{\mathbf{a}}^T\mathbf{z} \geq 0$ for all $\mathbf{z} \in interior(\mathcal{S})$, and $\overline{\mathbf{a}}^T\mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathcal{S}$, which means

     $\overline{\mathbf{a}}^T\mathbf{x} \geq \overline{\mathbf{a}}^T\mathbf{y}$ for all $\mathbf{x} \in \mathcal{C}$, and $\mathbf{y} \in \mathcal{D}$.

□

Theorem 59 stated that the gradient evaluated at a point on a level set is orthogonal to the tangent hyperplane to the level set at that point. We now state the definition of a supporting hyperplane, which is special type of tangent hyperplane.

Definition 34 [Supporting Hyperplane]: *The supporting hyperplane to a set $\mathcal{C}$ at a boundary point $\mathbf{x}_0$ is defined as* $\left\{\mathbf{x} | \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\mathbf{x}_0,\ \mathbf{a} \neq \mathbf{0},\ \mathbf{a}^T\mathbf{y} \leq \mathbf{a}^T\mathbf{x}_0,\ \forall\ \mathbf{y} \in \mathcal{C}\right\}$

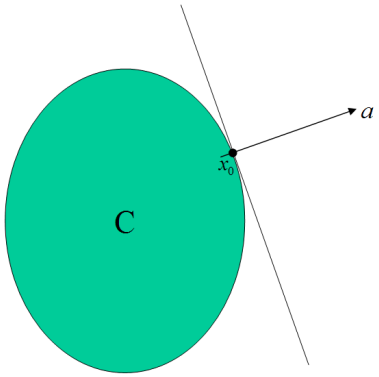

Figure 4.36: Example of a supporting hyperplane.

Figure 4.36 shows a supporting hyperplane at the point $\mathbf{x}_0$ on the boundary of a convex set $\mathcal{C}$.

For convex sets, there is an important theorem regarding supporting hyperplanes.

Theorem 68 *If the set $\mathcal{C}$ is convex, then there exists a supporting hyperplane at every boundary point of $\mathcal{C}$. As in the case of the seperating hyperplane, the supporting hyperplane need not be unique.*

Proof: If the interior of $\mathcal{C}$ is nonempty, the result follows immediately by applying the separating hyperplane theorem to the sets $\{\mathbf{x}_0\}$ and $interior(\mathcal{C})$. If the interior of $\mathcal{C}$ is empty, then $\mathcal{C}$ must lie in an affine set of dimension less than n, and any hyperplane containing that affine set contains $\mathcal{C}$ and $\mathbf{x}_0$, and is a (trivial) supporting hyperplane. □

4.2.7 Convex Functions

Definition 35 [Convex Function]: *A function $f : \mathcal{D} \rightarrow \Re$ is convex if $\mathcal{D}$ is a convex set and*

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) \quad \forall\ \mathbf{x}, \mathbf{y} \in \mathcal{D} \quad 0 \leq \theta \leq 1 \,(4.31)$$

Figure 4.37 illustrates an example convex function. A function $f : \mathcal{D} \rightarrow \Re$ is strictly convex if $\mathcal{D}$ is convex and

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) < \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y})) \quad \forall\ \mathbf{x}, \mathbf{y} \in \mathcal{D} \quad 0 \leq \theta \leq 1 (4.32)$$

Figure 4.37: Example of convex function.

A function $f : \mathcal{D} \to \Re$ is called uniformly or strongly convex if $\mathcal{D}$ is convex and there exists a constant $c > 0$ such that

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y})) - \tfrac{1}{2}c\theta(1-\theta)||\mathbf{x}-\mathbf{y}|| \quad \forall\ \mathbf{x},\mathbf{y} \in \mathcal{D} \quad 0 \leq \theta \leq 1 (4.33)$$

A function $f : \Re^n \to \Re$ is said to be concave if the function $-f$ is convex. Examples of convex functions on the set of reals $\Re$ as well as on $\Re^n$ and $\Re^{m\times n}$ are shown in Table 4.1. Examples of concave functions on the set of reals $\Re$ are shown in Table 4.2. If a function is both convex and concave, it must be affine, as can be seen in the two tables.

Function type	**Domain**	**Additional Constraints**
The affine function: $ax + b$	$\Re$	Any $a, b \in \Re$
The exponential function: e^{ax}	$\Re$	Any $a \in \Re$
Powers: x^α	$\Re_{++}$	$\alpha \geq 1$ or $\alpha \leq 1$
Powers of absolute value: $\|x\|^p$	$\Re$	$p \geq 1$
Negative entropy: $x \log x$	$\Re_{++}$	
Affine functions of vectors: $\mathbf{a}^T\mathbf{x} + b$	$\Re^n$	
p-norms of vectors: $\|\|\mathbf{x}\|\|_p = \left(\sum_{i=1}^{n} \|x_i\|^p\right)^{1/p}$	$\Re^n$	$p \geq 1$
inf norms of vectors: $\|\|\mathbf{x}\|\|_\infty = \max_k \|x_k\|$	$\Re^n$	
Affine functions of matrices: $tr(A^T X) + b = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}X_{ij} + b$	$\Re^{m\times n}$	
Spectral (maximum singular value) matrix norm: $\|\|X\|\|_2 = \sigma_{max}(X) = (\lambda_{max}(X^T X))^{1/2}$	$\Re^{m\times n}$	

Table 4.1: Examples of convex functions on $\Re$, $\Re^n$ and $\Re^{m\times n}$.

4.2.8 Convexity and Global Minimum

One of the most fundamental and useful chracteristics of convex functions is that any point of local minimum point for a convex function is also a point of global minimum.

Function type	Domain	Additional Constraints
The affine function: $ax + b$	$\Re$	Any $a, b \in \Re$
Powers: x^α	$\Re_{++}$	$0 \leq \alpha \leq 1$
logarithm: $\log x$	$\Re_{++}$	

Table 4.2: Examples of concave functions on $\Re$.

Theorem 69 *Let $f : \mathcal{D} \rightarrow \Re$ be a convex function on a convex domain $\mathcal{D}$. Any point of locally minimum solution for f is also a point of its globally minimum solution.*

Proof: Suppose $\mathbf{x} \in \mathcal{D}$ is a point of local minimum and let $\mathbf{y} \in \mathcal{D}$ be a point of global minimum. Thus, $f(\mathbf{y}) < f(\mathbf{x})$. Since $\mathbf{x}$ corresponds to a local minimum, there exists an $\epsilon > 0$ such that

$$\forall\ \mathbf{z} \in \mathcal{D},\ ||\mathbf{z} - \mathbf{x}|| \leq \epsilon \Rightarrow f(\mathbf{z}) \geq f(\mathbf{x})$$

Consider a point $\mathbf{z} = \theta\mathbf{y} + (1-\theta)\mathbf{x}$ with $\theta = \frac{\epsilon}{2||\mathbf{y}-\mathbf{x}||}$. Since $\mathbf{x}$ is a point of local minimum (in a ball of radius ϵ), and since $f(\mathbf{y}) < f(\mathbf{x})$, it must be that $||\mathbf{y} - \mathbf{x}|| > \epsilon$. Thus, $0 < \theta < \frac{1}{2}$ and $\mathbf{z} \in \mathcal{D}$. Furthermore, $||\mathbf{z} - \mathbf{x}|| = \frac{\epsilon}{2}$. Since f is a convex function

$$f(\mathbf{z}) \leq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y})$$

Since $f(\mathbf{y}) < f(\mathbf{x})$, we also have

$$\theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) < f(\mathbf{x})$$

The two equations imply that $f(\mathbf{z}) < f(\mathbf{x})$, which contradicts our assumption that $\mathbf{x}$ corresponds to a point of local minimum. That is f cannot have a point of local minimum, which does not coincide with the point $\mathbf{y}$ of global minimum. □

Since any locally minimum point for a convex function also corresponds to its global minimum, we will drop the qualifiers 'locally' as well as 'globally' while referring to the points corresponding to minimum values of a convex function. For any stricly convex function, the point corresponding to the gobal minimum is also unique, as stated in the following theorem.

Theorem 70 *Let $f : \mathcal{D} \rightarrow \Re$ be a strictly convex function on a convex domain $\mathcal{D}$. Then f has a unique point corresponding to its global minimum.*

Proof: Suppose $\mathbf{x} \in \mathcal{D}$ and $\mathbf{y} \in \mathcal{D}$ with $\mathbf{y} \neq \mathbf{x}$ are two points of global minimum. That is $f(\mathbf{x}) = f(\mathbf{y})$ for $\mathbf{y} \neq \mathbf{x}$. The point $\frac{\mathbf{x}+\mathbf{y}}{2}$ also belongs to the convex set $\mathcal{D}$ and since f is strictly convex, we must have

$$f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) < \frac{1}{2}f(\mathbf{x}) + \frac{1}{2}f(\mathbf{y}) = f(\mathbf{x})$$

which is a contradiction. Thus, the point corresponding to the minimum of f must be unique. □

In the following section, we state some important properties of convex functions, including relationships between convex functions and convex sets, and first and second order conditions for convexity. We will also draw relationships between the definitions of convexity and strict convexity stated here, with the definitions on page 224 for the single variable case.

4.2.9 Properties of Convex Functions

We will first extend the domain of a convex function to all $\Re^n$, while retaining its convexity and preserving its value in the domain.

Definition 36 [Extended value extension]: *If $f : \mathcal{D} \to \Re$, with $\mathcal{D} \subseteq \Re^n$ is a convex function, then we define its extended-valued extension $\widetilde{f} : \Re^n \to \Re$ as*

$$\widetilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{D} \\ \infty & \text{if } \mathbf{x} \notin \mathcal{D} \end{cases} \tag{4.34}$$

In what follows, we will assume if necessary, that all convex functions are implicitly extended to the domain $\Re^n$. A useful technique for verifying the convexity of a function is to investigate its convexity, by restricting the function to a line and checking for the convexity of a function of single variable. This technique is hinged on the following theorem.

Theorem 71 *A function $f : \mathcal{D} \to \Re$ is (strictly) convex if and only if the function $\phi : \mathcal{D}_\phi \to \Re$ defined below, is (strictly) convex in t for every $\mathbf{x} \in \Re^n$ and for every $\mathbf{h} \in \Re^n$*

$$\phi(t) = f(\mathbf{x} + t\mathbf{h})$$

with the domain of ϕ given by $\mathcal{D}_\phi = \{t | \mathbf{x} + t\mathbf{h} \in \mathcal{D}\}$.

Proof: We will prove the necessity and sufficiency of the convexity of ϕ for a convex function f. The proof for necessity and sufficiency of the strict convexity of ϕ for a strictly convex f is very similar and is left as an exercise.

Proof of Necessity: Assume that f is convex. And we need to prove that $\phi(t) = f(\mathbf{x} + t\mathbf{h})$ is also convex. Let $t_1, t_2 \in \mathcal{D}_\phi$ and $\theta \in [0, 1]$. Then,

$$\begin{aligned} \phi(\theta t_1 + (1-\theta)t_2) &= f\left(\theta(\mathbf{x} + t_1\mathbf{h}) + (1-\theta)(\mathbf{x} + t_2\mathbf{h})\right) \\ \leq \theta f\left((\mathbf{x} + t_1\mathbf{h})\right) + (1-\theta) f\left((\mathbf{x} + t_2\mathbf{h})\right) &= \theta\phi(t_1) + (1-\theta)\phi(t_2) \end{aligned} \tag{4.35}$$

Thus, ϕ is convex.

Proof of Sufficiency: Assume that for every $\mathbf{h} \in \Re^n$ and every $\mathbf{x} \in \Re^n$, $\phi(t) = f(\mathbf{x} + t\mathbf{h})$ is convex. We will prove that f is convex. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$. Take, $\mathbf{x} = \mathbf{x}_1$ and $\mathbf{h} = \mathbf{x}_2 - \mathbf{x}_1$. We know that $\phi(t) = f\left(\mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)\right)$ is convex, with $\phi(1) = f(\mathbf{x}_2)$ and $\phi(0) = f(\mathbf{x}_1)$. Therefore, for any $\theta \in [0, 1]$

$$f\left(\theta\mathbf{x}_2 + (1-\theta)\mathbf{x}_1\right) = \phi(\theta)$$
$$\leq \theta\phi(1) + (1-\theta)\phi(0) \leq \theta f(\mathbf{x}_2) + (1-\theta)f(\mathbf{x}_1) \tag{4.36}$$

This implies that f is convex. □

Next, we will draw the parallel between convex sets and convex functions by introducing the concept of the *epigraph* of a function.

Definition 37 [Epigraph]: *Let $\mathcal{D} \subseteq \Re^n$ be a nonempty set and $f : \mathcal{D} \rightarrow \Re$. The set $\{(\mathbf{x}, f(\mathbf{x})|\mathbf{x} \in \mathcal{D}\}$ is called graph of f and lies in $\Re^{n+1}$. The epigraph of f is a subset of $\Re^{n+1}$ and is defined as*

$$epi(f) = \{(\mathbf{x}, \alpha)|f(\mathbf{x}) \leq \alpha,\ \mathbf{x} \in \mathcal{D},\ \alpha \in \Re\} \tag{4.37}$$

In some sense, the epigraph is the set of points lying above the graph of f. Similarly, the hypograph of f is a subset of $\Re^{n+1}$, lying below the graph of f and is defined by

$$hyp(f) = \{(\mathbf{x}, \alpha)|f(\mathbf{x}) \geq \alpha,\ \mathbf{x} \in \mathcal{D},\ \alpha \in \Re\} \tag{4.38}$$

There is a one to one correspondence between the convexity of function f and that of the set $epi(f)$, as stated in the following theorem.

Theorem 72 *Let $\mathcal{D} \subseteq \Re^n$ be a nonempty convex set, and $f : \mathcal{D} \rightarrow \Re$. Then f is convex if and only if $epi(f)$ is a convex set.*

Proof: Let f be convex. For any $(\mathbf{x}_1, \alpha_1) \in epi(f)$ and $(\mathbf{x}_2, \alpha_2) \in epi(f)$ and any $\theta \in (0, 1)$,

$$f(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1-\theta)f(\mathbf{x}_2)) \leq \theta\alpha_1 + (1-\theta)\alpha_2$$

Since $\mathcal{D}$ is convex, $\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{D}$. Therefore, $(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2, \theta\alpha_1 + (1-\theta)\alpha_2) \in epi(f)$. Thus, $epi(f)$ is convex if f is convex. This proves the necessity part.

To prove sufficiency, assume that $epi(f)$ is convex. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$. So, $(\mathbf{x}_1, f(\mathbf{x}_1)) \in epi(f)$ and $(\mathbf{x}_2, f(\mathbf{x}_2)) \in epi(f)$. Since $epi(f)$ is convex, for $\theta \in (0, 1)$,

$$(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2, \theta\alpha_1 + (1-\theta)\alpha_2) \in epi(f)$$

which implies that $f(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1-\theta)f(\mathbf{x}_2))$ for any $\theta \in (0, 1)$. This proves the sufficiency. □

There is also a correspondence between the convexity of a function and the convexity of its *sublevel sets*.

Definition 38 [Sublevel Sets]: *Let $\mathcal{D} \subseteq \Re^n$ be a nonempty set and $f : \mathcal{D} \rightarrow \Re$. The set*

$$L_\alpha(f) = \{\mathbf{x} | \mathbf{x} \in \mathcal{D},\ f(\mathbf{x}) \leq \alpha\}$$

is called the $\alpha-$sub-level set of f.

The correspondence between the convexity of f and its $\alpha-$sub-level set is stated in the following theorem. Unlike the correspondence with the epigraph, the correspondence with the $\alpha-$sub-level set is not one to one.

Theorem 73 *Let $\mathcal{D} \subseteq \Re^n$ be a nonempty convex set, and $f : \mathcal{D} \rightarrow \Re$ be a convex function. Then $L_\alpha(f)$ is a convex set for any $\alpha \in \Re$.*

Proof: Consider $\mathbf{x}_1, \mathbf{x}_2 \in L_\alpha(f)$. Then by definition of the level set, $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$, $f(\mathbf{x}_1) \leq \alpha$ and $f(\mathbf{x}_2) \leq \alpha$. From convexity of $\mathcal{D}$ it follows that for all $\theta \in (0, 1)$, $\mathbf{x} = \theta\mathbf{x}_1 + (1 - \theta)\mathbf{x}_2 \in \mathcal{D}$. Moreover, since f is also convex,

$$f(\mathbf{x}) \leq \theta f(\mathbf{x}_1) + (1 - \theta)f(\mathbf{x}_2) \leq \theta\alpha + (1 - \theta)\alpha = \alpha$$

which implies that $\mathbf{x} \in L_\alpha(f)$. Thus, $L_\alpha(f)$ is a convex set. □The converse of this theorem does not hold. To illustrate this, consider the function $f(\mathbf{x}) = \frac{x_2}{1+2x_1^2}$. The 0-sublevel set of this function is $\{(x_1, x_2) \mid x_2 \leq 0\}$, which is convex. However, the function $f(\mathbf{x})$ itself is not convex.

An important property of a convex function is that it is continuous in the interior of its domain.

Theorem 74 *Let $f : \mathcal{D} \rightarrow \Re$ be a convex function with $\mathcal{D} \subseteq \Re^n$ being a convex set. Let $\mathcal{S} \subset \mathcal{D}$ be an open convex set. Then f is continuous on $\mathcal{S}$.*

Proof: Let us consider a point $\mathbf{x}_0 \in \mathcal{S}$. Since $\mathcal{S}$ is an open convex set, we can find $n + 1$ points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n+1}$ such that the interior of the convex hull

$$\mathcal{C} = \left\{ \mathbf{x} | \mathbf{x} = \sum_{i=1}^{n+1} a_i \mathbf{x}_i,\ a_i \geq 0,\ \sum_{1}^{n+1} a_i = 1 \right\}$$

is not empty and $\mathbf{x}_0 \in interior(\mathcal{C})$. Let $M = \max\limits_{1 \leq i \leq n+1} f(x_i)$. Then, for any $\mathbf{x} = \sum\limits_{i=1}^{n+1} a_i \mathbf{x}_i \in \mathcal{C}$,

$$f(\mathbf{x}) = f\left(\sum_{i=1}^{n+1} a_i \mathbf{x}_i \right) \leq \sum_{i=1}^{n+1} a_i f(\mathbf{x}_i) \leq M$$

Since $\mathbf{x}_0 \in \mathcal{C}$, there exists a $\delta > 0$ such that $B(\mathbf{x}_0, \delta) \subset \mathcal{C}$, where, $B(\mathbf{x}_0, \delta) = \{\mathbf{x} | ||\mathbf{x} - \mathbf{x}_0|| \leq \delta\}$. Therefore, $\mathbf{x}_0$ can be expressed as a convex combination of $(\mathbf{x}_0 + \theta\mathbf{h}$ and $\mathbf{x}_0 - \mathbf{h}$ for some $\mathbf{h} \in B(\mathbf{x}_0, \delta)$ and some $\theta \in [0, 1]$.

$$\mathbf{x}_0 = \frac{1}{1 + \theta}(\mathbf{x}_0 + \theta\mathbf{h}) + \frac{\theta}{1 + \theta}(\mathbf{x}_0 - \mathbf{h})$$

Since f is convex on $\mathcal{C}$,

$$f(\mathbf{x}_0) \leq \frac{1}{1+\theta} f(\mathbf{x}_0 + \theta\mathbf{h}) + \frac{\theta}{1+\theta} f(\mathbf{x}_0 - \mathbf{h})$$

From this, we can conclude that

$$f(\mathbf{x}_0 + \theta\mathbf{h}) - f(\mathbf{x}_0) \geq \theta\left(f(\mathbf{x}_0 - f(\mathbf{x}_0 - \mathbf{h})\right) \geq -\theta\left(M - f(\mathbf{x}_0)\right) \tag{4.39}$$

On the other hand,

$$f(\mathbf{x}_0 + \theta\mathbf{h}) \leq \theta f(\mathbf{x}_0 + \mathbf{h}) + (1-\theta) f(\mathbf{x}_0)$$

which implies that

$$f(\mathbf{x}_0 + \theta\mathbf{h}) - f(\mathbf{x}_0) \leq \theta\left(f(\mathbf{x}_0 + \mathbf{h}) - f(\mathbf{x}_0) \leq \theta\left(M - f(\mathbf{x}_0)\right)\right. \tag{4.40}$$

From equations 4.39 and 4.40, we can infer that

$$|f(\mathbf{x}_0 + \theta\mathbf{h}) - f(\mathbf{x}_0)| \leq \theta |f(\mathbf{x}_0) - M|$$

For a given $\epsilon > 0$, select $\delta' \leq \delta$ such that $\delta' |f(\mathbf{x}_0) - M| \leq \epsilon\delta$. Then $\mathbf{d} = \theta\mathbf{h}$ with $||\mathbf{h}|| = \delta$, implies that $d \in B(\mathbf{x}_0, \delta)$ and $f(\mathbf{x}_0 + \mathbf{d}) - f(\mathbf{x}_0)| \leq \epsilon$. This proves the theorem. □

Analogous to the definition of increasing functions introduced on page number 220, we next introduce the concept of monotonic functions. This concept is very useful for characterization of a convex function.

Definition 39 *Let* $\mathbf{f} : \mathcal{D} \to \Re^n$ *and* $\mathcal{D} \subseteq \Re^n$. *Then*

1. $\mathbf{f}$ *is monotone on* $\mathcal{D}$ *if for any* $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$,

$$\left(\mathbf{f}(\mathbf{x}_1) - \mathbf{f}(\mathbf{x}_2)\right)^T (\mathbf{x}_1 - \mathbf{x}_2) \geq 0 \tag{4.41}$$

2. $\mathbf{f}$ *is strictly monotone on* $\mathcal{D}$ *if for any* $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$ *with* $\mathbf{x}_1 \neq \mathbf{x}_2$,

$$\left(\mathbf{f}(\mathbf{x}_1) - \mathbf{f}(\mathbf{x}_2)\right)^T (\mathbf{x}_1 - \mathbf{x}_2) > 0 \tag{4.42}$$

3. $\mathbf{f}$ *is uniformly or strongly monotone on* $\mathcal{D}$ *if for any* $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$, *there is a constant* $c > 0$ *such that*

$$\left(\mathbf{f}(\mathbf{x}_1) - \mathbf{f}(\mathbf{x}_2)\right)^T (\mathbf{x}_1 - \mathbf{x}_2) \geq c||\mathbf{x}_1 - \mathbf{x}_2||^2 \tag{4.43}$$

First-Order Convexity Conditions

The first order convexity condition for differentiable functions is provided by the following theorem:

Theorem 75 *Let $f : \mathcal{D} \rightarrow \Re$ be a differentiable convex function on an open convex set $\mathcal{D}$. Then:*

1. *f is convex if and only if, for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$,*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \tag{4.44}$$

2. *f is strictly convex on $\mathcal{D}$ if and only if, for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, with $\mathbf{x} \neq \mathbf{y}$,*

$$f(\mathbf{y}) > f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \tag{4.45}$$

3. *f is strongly convex on $\mathcal{D}$ if and only if, for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$,*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}c||\mathbf{y} - \mathbf{x}||^2 \tag{4.46}$$

for some constant $c > 0$.

Proof:

Sufficiency: The proof of sufficiency is very similar for all the three statements of the theorem. So we will prove only for statement (4.44). Suppose (4.44) holds. Consider $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$ and any $\theta \in (0, 1)$. Let $\mathbf{x} = \theta\mathbf{x}_1 + (1 - \theta)\mathbf{x}_2$. Then,

$$\begin{aligned} f(\mathbf{x}_1) &\geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{x}_1 - \mathbf{x}) \\ f(\mathbf{x}_2) &\geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{x}_2 - \mathbf{x}) \end{aligned} \tag{4.47}$$

Adding $(1 - \theta)$ times the second inequality to θ times the first, we get,

$$\theta f(\mathbf{x}_1) + (1 - \theta) f(\mathbf{x}_2) \geq f(\mathbf{x})$$

which proves that $f(\mathbf{x})$ is a convex function. In the case of strict convexity, strict inequality holds in (4.47) and it follows through. In the case of strong convexity, we need to additionally prove that

$$\theta\frac{1}{2}c||\mathbf{x} - \mathbf{x}_1||^2 + (1 - \theta)\frac{1}{2}c||\mathbf{x} - \mathbf{x}_2||^2 = \frac{1}{2}c\theta(1 - \theta)||\mathbf{x}_2 - \mathbf{x}_1||^2$$

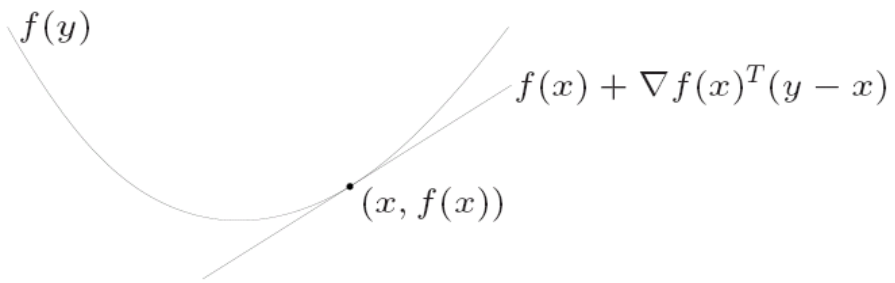

Figure 4.38: Figure illustrating Theorem 75.

Necessity: Suppose f is convex. Then for all $\theta \in (0,1)$ and $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$, we must have

$$f(\theta \mathbf{x}_2 + (1-\theta)\mathbf{x}_1) \leq \theta f(\mathbf{x}_2) + (1-\theta) f(\mathbf{x}_1)$$

Thus,

$$\nabla^T f(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) = \lim_{\theta \to 0} \frac{f\left(\mathbf{x}_1 + \theta(\mathbf{x}_2 - \mathbf{x}_1)\right) - f\left(\mathbf{x}_1\right)}{\theta} \leq f(\mathbf{x}_2) - f(\mathbf{x}_1)$$

This proves necessity for (4.44). The necessity proofs for (4.45) and (4.46) are very similar, except for a small difference for the case of strict convexity; the strict inequality is not preserved when we take limits. Suppose equality does hold in the case of strict convexity, that is for a strictly convex function f, let

$$f(\mathbf{x}_2) = f(\mathbf{x}_1) + \nabla^T f(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) \tag{4.48}$$

for some $\mathbf{x}_2 \neq \mathbf{x}_1$. Because f is stricly convex, for any $\theta \in (0,1)$ we can write

$$f\left(\theta \mathbf{x}_1 + (1-\theta)\mathbf{x}_2\right) = f\left(\mathbf{x}_2 + \theta(\mathbf{x}_1 - \mathbf{x}_2)\right) < \theta f(\mathbf{x}_1) + (1-\theta) f(\mathbf{x}_2) \tag{4.49}$$

Since (4.44) is already proved for convex functions, we use it in conjunction with (4.48), and (4.49), to get

$$f(\mathbf{x}_2) + \theta \nabla^T f(\mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2) \leq f\left(\mathbf{x}_2 + \theta(\mathbf{x}_1 - \mathbf{x}_2)\right) < f(\mathbf{x}_2) + \theta \nabla^T f(\mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)$$

which is a contradiction. Thus, equality can never hold in (4.44) for any $\mathbf{x}_1 \neq \mathbf{x}_2$. This proves the necessity of (4.45). □

The geometrical interpretation of theorem 75 is that at any point, the linear approximation based on a local derivative gives a lower estimate of the function, *i.e.* the convex function always lies above the supporting hyperplane at that point. This is pictorially depicted in Figure 4.38. There are some implications of theorem 75 for strongly convex functions. We state them next.

Definition 40 [Some corollaries of theorem 75 for strongly convex functions]: *For a fixed* $\mathbf{x}$*, the right hand side of the inequality (4.46) is a convex quadratic function of* $\mathbf{y}$*. Thus, the critical point of the RHS should correspond to the minimum value that the RHS could take. This yields another lower bound on* $f(\mathbf{y})$*.*

$$f(\mathbf{y}) \geq f(\mathbf{x}) - \frac{1}{2c}||\nabla f(\mathbf{x})||_2^2 \tag{4.50}$$

Since this holds for any $\mathbf{y} \in \mathcal{D}$*, we have*

$$\min_{\mathbf{y}\in\mathcal{D}} f(\mathbf{y}) \geq f(\mathbf{x}) - \frac{1}{2c}||\nabla f(\mathbf{x})||_2^2 \tag{4.51}$$

which can be used to bound the suboptimality of a point $\mathbf{x}$ *in terms of* $||\nabla f(\mathbf{x})||_2$*. This bound comes handy in theoretically understanding the convergence of gradient methods. If* $\widehat{\mathbf{y}} = \min_{\mathbf{y}\in\mathcal{D}} f(\mathbf{y})$*, we can also derive a bound on the distance between any point* $\mathbf{x} \in \mathcal{D}$ *and the point of optimality* $\widehat{\mathbf{y}}$*.*

$$||\mathbf{x} - \widehat{\mathbf{y}}||_2 \leq \frac{2}{c}||\nabla f(\mathbf{x})||_2 \tag{4.52}$$

Theorem 75 motivates the definition of the *subgradient* for non-differentiable convex functions, which has properties very similar to the gradient vector.

Definition 41 [Subgradient]: *Let* $f : \mathcal{D} \to \Re$ *be a convex function defined on a convex set* $\mathcal{D}$*. A vector* $\mathbf{h} \in \Re^n$ *is said to be a subgradient of* f *at the point* $\mathbf{x} \in \mathcal{D}$ *if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{h}^T(\mathbf{y} - \mathbf{x})$$

for all $\mathbf{y} \in \mathcal{D}$*. The set of all such vectors is called the subdifferential of* f *at* $\mathbf{x}$*.*

For a differentiable convex function, the gradient at point $\mathbf{x}$ is the only subgradient at that point. Most properties of differentiable convex functions that hold in terms of the gradient also hold in terms of the subgradient for non-differentiable convex functions. Theorem 75 gives a very simple optimality criterion for a differentiable function f.

Theorem 76 *Let* $f : \mathcal{D} \to \Re$ *be a convex function defined on a convex set* $\mathcal{D}$*. A point* $\mathbf{x} \in \mathcal{D}$ *corresponds to a minimum if and only if*

$$\nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \geq 0$$

for all $\mathbf{y} \in \mathcal{D}$*.*

If $\nabla f(\mathbf{x})$ is nonzero, it defines a supporting hyperplane to $\mathcal{D}$ at the point $\mathbf{x}$. Theorem 77 implies that for a differentiable convex function defined on an open set, every critical point must be a point of (global) minimum.

Theorem 77 *Let $f : \mathcal{D} \to \Re$ be differentiable and convex on an open convex domain $\mathcal{D} \subseteq \Re^n$. Then $\mathbf{x}$ is a critical point of f if and only if it is a (global) minimum.*

Proof: If $\mathbf{x}$ is a global minimum, it is a local minimum and by theorem 60, it must be a critical point and therefore $\nabla f(\mathbf{x}) = 0$. Conversely, let $\nabla f(\mathbf{x}) = 0$, By theorem 75, we know that for all $\mathbf{y} \in \mathcal{D}$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x})$$

Substituting $\nabla f(\mathbf{x}) = 0$ in this inequality, we get for all $\mathbf{y} \in \mathcal{D}$,

$$f(\mathbf{y}) \geq f(\mathbf{x})$$

That is, $\mathbf{x}$ corresponds to a (global) minimum. □

Based on the definition of monotonic functions in definition 39, we show the relationship between convexity of a function and monotonicity of its gradient in the next theorem.

Theorem 78 *Let $f : \mathcal{D} \to \Re$ with $\mathcal{D} \subseteq \Re^n$ be differentiable on the convex set $\mathcal{D}$. Then,*

1. *f is convex on $\mathcal{D}$ if and only if is its gradient ∇f is monotone. That is, for all $\mathbf{x}, \mathbf{y} \in \Re$*

$$\left(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right)^T (\mathbf{x} - \mathbf{y}) \geq 0 \tag{4.53}$$

2. *f is strictly convex on $\mathcal{D}$ if and only if is its gradient ∇f is strictly monotone. That is, for all $\mathbf{x}, \mathbf{y} \in \Re$ with $\mathbf{x} \neq \mathbf{y}$,*

$$\left(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right)^T (\mathbf{x} - \mathbf{y}) > 0 \tag{4.54}$$

3. *f is uniformly or strongly convex on $\mathcal{D}$ if and only if is its gradient ∇f is uniformly monotone. That is, for all $\mathbf{x}, \mathbf{y} \in \Re$,*

$$\left(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right)^T (\mathbf{x} - \mathbf{y}) \geq c||\mathbf{x} - \mathbf{y}||^2 \tag{4.55}$$

for some constant $c > 0$.

Proof:

Necessity: Suppose f is uniformly convex on $\mathcal{D}$. Then from theorem 75, we know that for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) - \frac{1}{2}c||\mathbf{y} + \mathbf{x}||^2$$
$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla^T f(\mathbf{y})(\mathbf{x} - \mathbf{y}) - \frac{1}{2}c||\mathbf{x} + \mathbf{y}||^2$$

Adding the two inequalities, we get (4.55). If f is convex, the inequalities hold with $c = 0$, yielding (4.54). If f is strictly convex, the inequalities will be strict, yielding (4.54).

Sufficiency: Suppose ∇f is monotone. For any fixed $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, consider the function $\phi(t) = f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))$. By the mean value theorem applied to $\phi(t)$, we should have for some $t \in (0, 1)$,

$$\phi(1) - \phi(0) = \phi'(t) \tag{4.56}$$

Letting $\mathbf{z} = \mathbf{x} + t(\mathbf{y} - \mathbf{x})$, (4.56) translates to

$$f(\mathbf{y}) - f(\mathbf{x}) = \nabla^T f(\mathbf{z})(\mathbf{y} - \mathbf{x}) \tag{4.57}$$

Also, by definition of monotonicity of ∇f, (from (4.53)),

$$(\nabla f(\mathbf{z}) - \nabla f(\mathbf{x}))^T (\mathbf{y} - \mathbf{x}) = \frac{1}{t} (\nabla f(\mathbf{z}) - \nabla f(\mathbf{x}))^T (\mathbf{z} - \mathbf{x}) \geq 0 \tag{4.58}$$

Combining (4.57) with (4.58), we get,

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{x}) &= (\nabla f(\mathbf{z}) - f(\mathbf{x}))^T (\mathbf{y} - \mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \\ &\geq \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \end{aligned} \tag{4.59}$$

By theorem 75, this inequality proves that f is convex. Strict convexity can be similarly proved by using the strict inequality in (4.58) inherited from strict monotonicity, and letting the strict inequality follow through to (4.59). For the case of strong convexity, from (4.55), we have

$$\begin{aligned} \phi'(t) - \phi'(0) &= (\nabla f(\mathbf{z}) - f(\mathbf{x}))^T (\mathbf{y} - \mathbf{x}) \\ = \frac{1}{t} (\nabla f(\mathbf{z}) - f(\mathbf{x}))^T (\mathbf{z} - \mathbf{x}) &\geq \frac{1}{t}c||\mathbf{z} - \mathbf{x}||^2 = ct||\mathbf{y} - \mathbf{x}||^2 \end{aligned} \tag{4.60}$$

Therefore,

$$\phi(1) - \phi(0) - \phi'(0) = \int_0^1 [\phi'(t) - \phi'(0)]dt \geq \frac{1}{2}c||\mathbf{y} - \mathbf{x}||^2 \tag{4.61}$$

which translates to

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}c||\mathbf{y} - \mathbf{x}||^2$$

By theorem 75, f must be strongly convex. □

Second Order Condition

For twice continuously differentiable convex functions the convexity condition can be characterized as follows.

Theorem 79 *A twice differential function $f : \mathcal{D} \to \Re$ for a nonempty open convex set $\mathcal{D}$*

1. *is convex if and only if its domain is convex and its Hessian matrix is positive semidefinite at each point in $\mathcal{D}$. That is*

$$\nabla^2 f(\mathbf{x}) \succeq 0 \quad \forall\ \mathbf{x} \in \mathcal{D} \tag{4.62}$$

2. *is strictly convex if its domain is convex and its Hessian matrix is positive definite at each point in $\mathcal{D}$. That is*

$$\nabla^2 f(\mathbf{x}) \succ 0 \quad \forall\ \mathbf{x} \in \mathcal{D} \tag{4.63}$$

3. *is uniformly convex if and only if its domain is convex and its Hessian matrix is uniformly positive definite at each point in $\mathcal{D}$. That is, for any $\mathbf{v} \in \Re^n$ and any $\mathbf{x} \in \mathcal{D}$, there exists a $c > 0$ such that*

$$\mathbf{v}^T \nabla^2 f(\mathbf{x})\mathbf{v} \geq c||\mathbf{v}||^2 \tag{4.64}$$

In other words

$$\nabla^2 f(\mathbf{x}) \succeq cI_{n\times n}$$

where $I_{n\times n}$ is the $n \times n$ identity matrix and $\succeq$ corresponds to the positive semidefinite inequality. That is, the function f is strongly convex iff $\nabla^2 f(\mathbf{x}) - cI_{n\times n}$ is positive semidefinite, for all $\mathbf{x} \in \mathcal{D}$ and for some constant $c > 0$, which corresponds to the positive minimum curvature of f.

Proof: We will prove only the first statement in the theorem; the other two statements are proved in a similar manner.

Necessity: Suppose f is a convex function, and consider a point $\mathbf{x} \in \mathcal{D}$. We will prove that for any $\mathbf{h} \in \Re^n$, $\mathbf{h}^T \nabla^2 f(\mathbf{x}) \mathbf{h} \geq 0$. Since f is convex, by theorem 75, we have

$$f(\mathbf{x} + t\mathbf{h}) \geq f(\mathbf{x}) + t\nabla^T f(\mathbf{x})\mathbf{h} \tag{4.65}$$

Consider the function $\phi(t) = f(\mathbf{x} + t\mathbf{h})$ considered in theorem 71, defined on the domain $\mathcal{D}_\phi = [0, 1]$. Using the chain rule,

$$\phi'(t) = \sum_{i=1}^{n} f_{x_i}(\mathbf{x} + t\mathbf{h}) \frac{dx_i}{dt} = \mathbf{h}^T . \nabla f(\mathbf{x} + t\mathbf{h})$$

Since f has partial and mixed partial derivatives, ϕ' is a differentiable function of t on $\mathcal{D}_\phi$ and

$$\phi''(t) = \mathbf{h}^T \nabla^2 f(\mathbf{x} + t\mathbf{h})\mathbf{h}$$

Since ϕ and ϕ' are continous on $\mathcal{D}_\phi$ and ϕ' is differentiable on $int(\mathcal{D}_\phi)$, we can make use of the Taylor's theorem (45) with $n = 3$ to obtain:

$$\phi(t) = \phi(0) + t.\phi'(0) + t^2 . \frac{1}{2}\phi''(0) + O(t^3)$$

Writing this equation in terms of f gives

$$f(\mathbf{x} + t\mathbf{h}) = f(\mathbf{x}) + t\mathbf{h}^T \nabla f(\mathbf{x}) + t^2 \frac{1}{2} h^T \nabla^2 f(\mathbf{x})\mathbf{h} + O(t^3)$$

In conjunction with (4.65), the above equation implies that

$$\frac{t^2}{2} h^T \nabla^2 f(\mathbf{x})\mathbf{h} + O(t^3) \geq 0$$

Dividing by t^2 and taking limits as $t \to 0$, we get

$$h^T \nabla^2 f(\mathbf{x})\mathbf{h} \geq 0$$

Sufficiency: Suppose that the Hessian matrix is positive semidefinite at each point $\mathbf{x} \in \mathcal{D}$. Consider the same function $\phi(t)$ defined above with $\mathbf{h} = \mathbf{y} - \mathbf{x}$ for $\mathbf{y}, \mathbf{x} \in \mathcal{D}$. Applying Taylor's theorem (45) with $n = 2$ and $a = 0$, we obtain,

$$\phi(1) = \phi(0) + t.\phi'(0) + t^2 . \frac{1}{2}\phi''(c)$$

for some $c \in (0, 1)$. Writing this equation in terms of f gives

$$f(\mathbf{x}) = f(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla f(\mathbf{y}) + \frac{1}{2}(\mathbf{x} - \mathbf{y})^T \nabla^2 f(\mathbf{z})(\mathbf{x} - \mathbf{y})$$

where $\mathbf{z} = \mathbf{y} + c(\mathbf{x} - \mathbf{y})$. Since $\mathcal{D}$ is convex, $\mathbf{z} \in \mathcal{D}$. Thus, $\nabla^2 f(\mathbf{z}) \succeq 0$. It follows that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla f(\mathbf{y})$$

By theorem 75, the function f is convex. □

Examples of differentiable/twice differentiable convex functions, along with the value of their respective gradients/hessians are tabulated in Table 4.3.

Function type	**Constraints**	**Gradient/Hessian**
Quadratic : $\frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$	$A \succeq 0$	$\nabla^2 f(\mathbf{x}) = P$
Quadratic over linear: $\frac{x^2}{y} \geq 0$	$y > 0$	$\nabla^2 f(x,y) = \frac{2}{y^3}\begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix}$
Log-sum-exp: $\log \sum_{k=1}^{n} exp(x_k)$		$\nabla^2 f(x) = \frac{1}{(\mathbf{1}^T\mathbf{z})^2}\left((\mathbf{1}^T\mathbf{z})\ \boldsymbol{diag}(\mathbf{z}) - \mathbf{z}\mathbf{z}^T\right)$ where $\mathbf{z} = [e^{x_1}, e^{x_1}, \ldots, e^{x_n}]$
Negative Geometric mean: $-\left(\prod_{k=1}^{n} x_k\right)^{\frac{1}{n}}$	$\mathbf{x} \in \Re^n_{++}$	$\nabla^2 f(x) = \frac{\prod_{i=1}^{n} n x_i^{1/n}}{n^2}\left(n\ \boldsymbol{diag}(\frac{1}{x_1^2}, \ldots, \frac{1}{x_n^2}) - qq^T\right)$

Table 4.3: Examples of twice differentiable convex functions on $\Re$.

4.2.10 Convexity Preserving Operations on Functions

In practice if you want to establish the convexity of a function f, you could either

1. Prove it from first principles, i.e., using the definition of convexity or
2. If f is twice differentiable, show that $\nabla^2 f(x) \succeq 0$
3. Show that f is obtained from simple convex functions by operations that preserve complexity. Following are operations on functions that preserve complexity (proofs omitted, since they are trivial):

   - **Nonnegative weighted sum:** $f = \sum_{i=1}^{n} \alpha_i f_i$ is convex if each f_i for $1 \leq i \leq n$ is convex and $\alpha_i \geq 0, 1 \leq i \leq n$.
   - **Composition with affine function:** $f(Ax + b)$ is convex if f is convex. For example:
     - The log barrier for linear inequalities, $f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$, is convex since $-\log(x)$ is convex.
     - Any norm of an affine function, $f(x) = ||Ax + b||$, is convex.
   - **Pointwise maximum:** If $f_1, f_2, \ldots, f_m$ are convex, then $f(x) = max\{f_1(x), f_2(x), \ldots, f_m(x)\}$ is also convex, For example:

  - Sum of r largest components of $\mathbf{x} \in \Re^n$ $f(\mathbf{x}) = x_{[1]} + x_{[2]} + \ldots + x_{[r]}$, where $x_{[1]}$ is the i^{th} largest component of $\mathbf{x}$, is a convex function.

- **Pointwise supremum:** If $f(x, y)$ is convex in x for every $y \in \mathcal{S}$, then $g(x) = \sup_{y \in \mathcal{S}} f(x, y)$ is convex. For example:
  - The function that returns the maximum eigenvalue of a symmetric matrix X, *viz.*, $\lambda_{max}(X) = \sup_{y \in \mathcal{S}} f(x, y)$ is a convex function of the symmetrix matrix X.
- **Composition with functions:** Let $h : \Re^k \rightarrow \Re$ with $h(x) = \infty, \forall\ x \notin \mathbf{dom}\ h$ and $g : \Re^n \rightarrow \Re^k$. Define $f(x) = h(g(x))$. f is convex if
  - g_i is convex, h is convex and nondecreasing in each argument
  - or g_i is concave, h is convex and nonincreasing in each argument

  Some examples illustrating this property are:
  - $exp\ g(x)$ is convex if g is convex
  - $\sum_{i=1}^{m} \log g_i(x)$ is concave if g_i are concave and positive
  - $\log \sum_{i=1}^{m} \exp g_i(x)$ is convex if g_i are convex
  - $1/g(x)$ is convex if g is concave and positive
- **Infimum:** If $f(x, y)$ is convex in (x, y) and $\mathcal{C}$ is a convex set, then $g(x) = \inf_{y \in \mathcal{C}} f(x, y)$ is convex. For example:
  - Let $f(x, \mathcal{S})$ that returns the distance of a point x to a convex set $\mathcal{S}$. That is $f(x, \mathcal{S}) = \inf_{y \in \mathcal{S}} ||x - y||$. Then $f(x, \mathcal{S})$ is a convex.
- **Perspective Function:** The perspective of a function $f : \Re^n \rightarrow \Re$ is the function $g : R^n \times \Re \rightarrow \Re$, $g(x, t) = tf(x/t)$. Function g is convex if f is convex on $\mathbf{dom} g = \{(x, t) | x/t \in \mathbf{dom} f, t > 0\}$. For example,
  - The perspective of $f(x) = x^T x$ is (quadratic-over-linear) function $g(x, t) = \frac{x^T x}{t}$ and is convex.
  - The perspective of negative logarithm $f(x) = -\log x$ is the relative entropy function $g(x, t) = t \log t - t \log x$ and is convex.

4.3 Convex Optimization Problem

Formally, a convex program is defined as

$$\min_{\mathbf{x} \in \mathcal{X}} c^T x \tag{4.66}$$

where $\mathcal{X} \subset \Re^n$ is a convex set and $\mathbf{x}$ is a vector of n optimization or decision variables. In applications, convex optimization programs usually arise in the form:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & A\mathbf{x} = \mathbf{b} \\ \text{variable } \mathbf{x} = (x_1, \ldots, x_n) & \end{array} \tag{4.67}$$

If it is given that the functions $f, g_1, \ldots, g_m$ are convex, by theorem 73, the feasible set $\mathcal{X}$ of this problem, which is the intersection of a finite number of $0-$sub-level sets of convex functions is also convex. Therefore, this problem can be posed as the following convex optimization problem:

$$\begin{array}{ll} \min\limits_{x=(t,u)\in\mathcal{X}} & t \\ & \mathcal{X} = \{(t,u) | f(u) \leq t, g_1(u) \leq 0, g_2(u) \leq 0, \ldots, g_m(u) \leq 0\} \end{array} \tag{4.68}$$

The set $\mathcal{X}$ is convex, and hence the problem in (4.68) is a convex optimization problem. Further, every locally optimal point is also globally optimal. The computation time of algorithms for solving convex optimization problems is roughly proportional to $max\left(n^2, n^2m, C\right)$, where C is the cost of evaluating f, the g_i's and their first and second derivatives. There are many reliable and efficient algorithms for solving convex optimization problems. However, it is often difficult to recognize convex optimization problems in practice.

Examples

Consider the optimization problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{subject to} & g_1(\mathbf{x}) = \frac{x_1}{1+x_2^2} \leq 0 \\ & h(\mathbf{x}) = (x_1 + x_2)^2 = 0 \end{array} \tag{4.69}$$

We note that the optimiation problem above is not a convex problem according to our definition, since g_1 is not convex and h is not affine. However, we note that the feasible set $\{(x_1, x_2) \mid x_1 = -x_2,\ x_1 \leq 0\}$ is convex (recall that the converse of theorem 73 does not hold - the 0-sublevel set of a non convex function can be convex). This problem can be posed as an equivalent (but not identical) convex optimization problem:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{subject to} & x_1 \leq 0 \\ & x_1 + x_2 = 0 \end{array} \tag{4.70}$$

4.4 Duality Theory

Duality is a very important component of nonlinear and linear optimization models. It has a wide spectrum of applications that are very popular. It arises in the basic form of linear programming as well as in interior point methods for linear programming. The duality in linear programming was first observed by Von Neumann, and later formalized by Tucker, Gale and Kuhn. In the first attempt at extending duality beyond linear programs, duals of quadratic programs were next developed. It was subsequently observed that you can always write a dual for any optimization problem and the modern Lagrange-based 'constructive'[19] duality theory followed in the late 1960s.

An extremely popular application of duality happens to be in the quadratic programming for Support Vector Machines. The primal and dual both happen to be convex optimization programs in this case. The Minimax theorem[20], a fundamental theorem of Game Theory, proved by John von Neumann in 1928, is but one instance of the general duality theory. In the consideration of equilibrium in electrical networks, current are 'primal variables' and the potential differences are the 'dual variables'. In models of economic markets, the 'primal' variables are production and consumption levels while the 'dual' variables are prices (of goods, *etc.*). Dual price-based decomposition methods were developed by Danzig. In the case of thrust structures in mechanics, forces are primal variables and the displacements are the dual variables. Dual problems and their solutions are used for proving optimality of solutions, finding near-optimal solutions, analysing how sensitive the solution of the primal is to perturbations in the right hand side of constraints, analysing convergence of algorithms, *etc.*

4.4.1 Lagrange Multipliers

Consider the following quadratic function of $\mathbf{x} \in \Re^n$.

$$F(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} - \mathbf{x}^T \mathbf{b} \tag{4.71}$$

where A is an $n \times n$ square matrix. Consider the unconstrained minimization problem

[19] As we will see, the theory helps us construct duals that are useful in practice.
[20] The name Minimax was invented by Tucker.

$$\min_{\mathbf{x}\in\mathcal{D}} F(\mathbf{x}) \tag{4.72}$$

A locally optimum solution $\hat{\mathbf{x}}$ to this objective can be obtained by setting $\nabla F(\hat{\mathbf{x}}) = \mathbf{0}$. This condition translates to $A\hat{\mathbf{x}} = \mathbf{b}$. A sufficient condition for $\hat{\mathbf{x}}$ to be a point of local minimum is that $\nabla^2 F(\hat{\mathbf{x}}) \succ 0$. This condition holds *iff*, $A \succ 0$, that is, A is a positive definite matrix. Given that $A \succ 0$, A must be invertible (*c.f.* Section 3.12.2) and the unique solution is $\mathbf{x} = A^{-1}\mathbf{b}$.

Now suppose we have a constrained minimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{y}\in\Re^n} & \frac{1}{2}\mathbf{y}^T B\mathbf{y} \\ \text{subject to} & A^T\mathbf{y} = \mathbf{b} \end{array} \tag{4.73}$$

where $\mathbf{y} \in \Re^n$, A is an $n \times m$ matrix, B is an $n \times n$ matrix and $\mathbf{b}$ is a vector of size m. To handle constrained minimization, let us consider minimization of the modified objective function $L(\mathbf{y}, \lambda) = \frac{1}{2}\mathbf{y}^T B\mathbf{y} + \lambda^T(A^T\mathbf{y} - \mathbf{b})$.

$$\min_{\mathbf{y}\in\Re^n, \lambda\in\Re^m} \quad \frac{1}{2}\mathbf{y}^T B\mathbf{y} + \lambda^T(A^T\mathbf{y} - \mathbf{b}) \tag{4.74}$$

The function $L(\mathbf{y}, \lambda)$ is called the lagrangian and involves the lagrange multiplier $\lambda \in \Re^m$. A sufficient condition for optimality of $L(\mathbf{y}, \lambda)$ at a point $L(\mathbf{y}^*, \lambda^*)$ is that $\nabla L(\mathbf{y}^*, \lambda^*) = 0$ and $\nabla^2 L(\mathbf{y}^*, \lambda^*) \succ 0$. For this particular problem:

$$\nabla L(\mathbf{y}^*, \lambda^*) = \begin{bmatrix} B\mathbf{y}^* + A\lambda^* \\ A^T\mathbf{y}^* - \mathbf{b} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$\nabla^2 L(\mathbf{y}^*, \lambda^*) = \begin{bmatrix} B & A \\ A^T & 0 \end{bmatrix} \succ 0$$

The point $(\mathbf{y}^*, \lambda^*)$ must therefore satisfy, $A^T\mathbf{y}^* = \mathbf{b}$ and $A\lambda^* = -B\mathbf{y}^*$. If B is taken to be the identity matrix, $n = 2$ and $m = 1$, the minimization problem (4.73) amounts to finding a point $\mathbf{y}^*$ on a line $a_{11}y_1 + a_{12}y_2 = b$ that is closest to the origin. From geometry, we know that the point on a line closest to the origin is the point of intersection $\mathbf{p}^*$ of a perpendicular from the origin to the line. On the other hand, the solution for the minimum of (4.74), for these conditions coincides with $\mathbf{p}^*$ and is given by:

$$y_1 = \frac{a_{11}b}{(a_{11})^2 + (a_{12})^2}$$
$$y_2 = \frac{a_{12}b}{(a_{11})^2 + (a_{12})^2}$$

That is, for $n = 2$ and $m = 1$, the solution to (4.74) is the same as the solution to (4.72) Can this construction be used to always find optimal solutions to a minimization problem? We will answer this question by first motivating the concept of lagrange multipliers and in Section 4.4.2, we will formalize the lagrangian dual.

Lagrange Multipliers with Equality Constraints

The concept of lagrange multipliers can be attributed to the mathematician Lagrange, who was born in the year 1736 in Turin. He largely worked on mechanics, the calculus of variations probability, group theory, and number theory. He was party to the choice of base 10 for the metric system (rather than 12). We will here give a brief introduction to lagrange multipliers; Section 4.4.2 will discuss the *Karush-Kuhn-Tucker conditions*, which are a generalization of lagrange multipliers.

Consider the equality constrained minimization problem (with $\mathcal{D} \subseteq \Re^n$)

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{D}} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, m \end{array} \tag{4.75}$$

A direct approach to solving this problem is to find a parametrization of the constraints (as in the example on page 230) such that f is expressed in terms of the parameters, to give an unconstrained problem. For example if there is a single constraint of the form $\mathbf{x}^T A\mathbf{x} = k$, and $A \succ 0$, then the coordinate system can be rotated and $\mathbf{x}$ can be rescaled so that we get the constraint $\mathbf{y}'\mathbf{y} = k$. Further, we can substitute with parametrization of the y_i's as

$$\begin{aligned} y_1 &= k \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{n-1} \\ y_2 &= k \sin\theta_1 \sin\theta_2 \ldots \cos\theta_{n-1} \\ &\ldots\ldots\ldots \end{aligned}$$

However, this is not possible for general constraints. The method of lagrange multipliers presents an indirect approach to solving this problem.

Consider a schematic representation of the problem in (4.75) with a single constraint, *i.e.*, $m = 1$ in Figure 4.39. The figure shows some level curves of the function f. The constraint function g_1 is also plotted with dotted lines in the same figure. The gradient of the constraint ∇g_1 is not parallel to the gradient ∇f of the function[21] at $f = 10.4$; it is therefore possible to move along the constraint surface so as to further reduce f. However, as shown in Figure 4.39, ∇g_1 and ∇f are parallel at $f = 10.3$, and any motion along $g_1(\mathbf{x}) = 0$ will

[21] Note that the (negative) gradient at a point is orthogonal to the contour line going through that point. This was proved in Theorem 59.

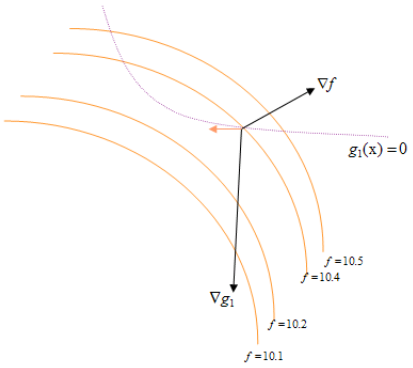

Figure 4.39: At any non-optimal and non-saddle point of the equality constrained problem, the gradient of the constraint will not be parallel to that of the function.

increase f, or leave it unchanged. Hence, at the solution $\mathbf{x}^*$, $\nabla f(\mathbf{x}^*)$ must be proportional to $-\nabla g_1(\mathbf{x}^*)$, yielding, $\nabla f(\mathbf{x}^*) = -\lambda \nabla g_1(\mathbf{x}^*)$, for some constant $\lambda \in \Re$; λ is called a *Lagrange multiplier*. In several problems, the value of λ itself need never be computed and therefore λ is often qualified as the *undetermined* lagrange multiplier.

The necessary condition for an optimum at $\mathbf{x}^*$ for the optimization problem in (4.75) with $m = 1$ can be stated as in (4.76), where the gradient is now $n+1$ dimensional with its last component being a partial derivative with respect to λ.

$$\nabla L(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) + \lambda^* \nabla g_1(\mathbf{x}^*) = 0 \tag{4.76}$$

The solutions to (4.76) are the stationary points of the lagrangian L; they are not necessarily local extrema of L. L is unbounded: given a point $\mathbf{x}$ that doesn't lie on the constraint, letting $\lambda \to \pm\infty$ makes L arbitrarily large or small. However, under certain stronger assumptions, as we shall see in Section 4.4.2, if the *strong Lagrangian principle* holds, the minima of f minimize the Lagrangian globally.

We will extend the necessary condition for optimality of a minimization problem with single constraint to minimization problems with multiple equality constraints (*i.e.*, $m > 1$. in (4.75)). Let $\mathcal{S}$ be the subspace spanned by $\nabla g_i(\mathbf{x})$ at any point $\mathbf{x}$ and let $\mathcal{S}_\perp$ be its orthogonal complement. Let $(\nabla f)_\perp$ be the component of ∇f in the subspace $\mathcal{S}_\perp$. At any solution $\mathbf{x}^*$, it must be true that the gradient of f has $(\nabla f)_\perp = 0$ (*i.e.*, no components that are perpendicular to all of the ∇g_i), because otherwise you could move $\mathbf{x}^*$ a little in that direction (or in the opposite direction) to increase (decrease) f without changing any of the g_i, *i.e.* without violating any constraints. Hence for multiple equality constraints, it must be true that at the solution $\mathbf{x}^*$, the space $\mathcal{S}$ contains the

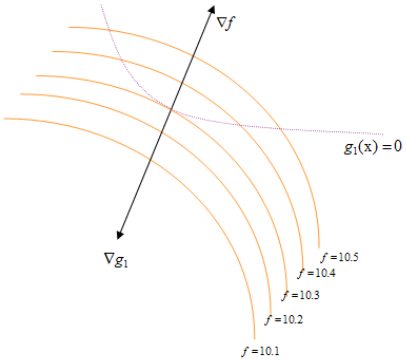

Figure 4.40: At the equality constrained optimum, the gradient of the constraint must be parallel to that of the function.

vector ∇f, *i.e.*, there are some constants λ_i such that $\nabla f(\mathbf{x}^*) = \lambda_i \nabla g_i(\mathbf{x}^*)$. We also need to impose that the solution is on the correct constraint surface (*i.e.*, $g_i = 0$, $\forall i$). In the same manner as in the case of $m = 1$, this can be encapsulated by introducing the Lagrangian $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$, whose gradient with respect to both $\mathbf{x}$, and λ vanishes at the solution.

This gives us the following necessary condition for optimality of (4.75):

$$\nabla L(\mathbf{x}^*, \lambda^*) = \nabla \left(f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \right) = 0 \tag{4.77}$$

Lagrange Multipliers with Inequality Constraints

Instead of a single equality constraint $g_1(\mathbf{x}) = 0$, we could have a single inequality constraint $g_1(\mathbf{x}) \leq 0$. The entire region labeled $g_1(\mathbf{x}) \leq 0$ in Figure 4.41 then becomes feasible. At the solution $\mathbf{x}^*$, if $g_1(\mathbf{x}^*) = 0$, *i.e.*, if the constraint is active, we must have (as in the case of a single equality constraint) that ∇f is parallel to ∇g_1, by the same argument as before. Additionally, it is necessary that the two gradients must point in opposite directions; otherwise a move away from the surface $g_1 = 0$ and into the feasible region would further reduce f. Since we are minimizing f, if the Lagrangian is written as $L = f + \lambda g_1$, we must have $\lambda \geq 0$. Therefore, with an inequality constraint, the sign of λ is important, and $\lambda \geq 0$ becomes a constraint.

However, if the constraint is not active at the solution $\nabla f(\mathbf{x}^*) = 0$, then removing g_1 makes no difference and we can drop it from $L = f + \lambda g_1$, which is equivalent to setting $\lambda = 0$. Thus, whether or not the constraints $g_1 = 0$ are active, we can find the solution by requiring that the gradients of the Lagrangian vanish, and also requiring that $\lambda g_1(\mathbf{x}^*) = 0$. This latter condition is one of the

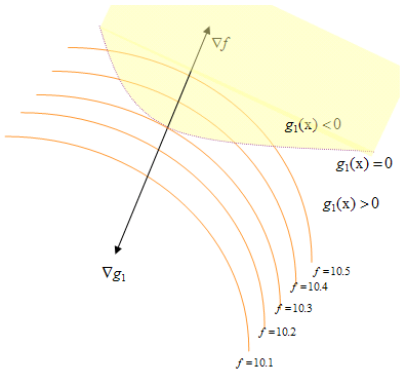

Figure 4.41: At the inequality constrained optimum, the gradient of the constraint must be parallel to that of the function.

important Karush-Kuhn-Tucker conditions of convex optimization theory that can facilitate the search for the solution and will be more formally discussed in Section 4.4.2.

Now consider the general inequality constrained minimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{D}} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \end{array} \tag{4.78}$$

With multiple inequality constraints, for constraints that are active, as in the case of multiple equality constraints, ∇f must lie in the space spanned by the ∇g_i's, and if the Lagrangian is $L = f + \sum_{i=1}^{m} \lambda_i g_i$, then we must additionally have $\lambda_i \geq 0$, $\forall i$ (since otherwise f could be reduced by moving into the feasible region). As for an inactive constraint g_j ($g_j < 0$), removing g_j from L makes no difference and we can drop ∇g_j from $\nabla f = -\sum_{i=1}^{m} \lambda_i \nabla g_i$ or equivalently set $\lambda_j = 0$. Thus, the above KKT condition generalizes to $\lambda_i g_i(\mathbf{x}^*) = 0$, $\forall i$. The necessary condition for optimality of (4.78) is summarily given as

$$\begin{aligned} \nabla L(\mathbf{x}^*, \lambda^*) = \nabla \left(f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \right) = 0 \\ \forall i \quad \lambda_i g_i(\mathbf{x}) = 0 \end{aligned} \tag{4.79}$$

A simple and often useful trick called the *free constraint gambit* is to solve ignoring one or more of the constraints, and then check that the solution satisfies those constraints, in which case you have solved the problem.

4.4.2 The Dual Theory for Constrained Optimization

Consider the general inequality constrained minimization problem in (4.78), restated below.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{D}} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0,\ i = 1, 2, \ldots, m \end{array} \tag{4.80}$$

There are three simple and straightforward steps in forming a dual problem.

1. The first step involves forming the lagrange function by associating a price λ_i, called a lagrange multiplier, with the constraint involving g_i.

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i g_i(\mathbf{x}) = f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x})$$

2. The second step is the construction of the dual function $L^*(\lambda)$ which is defined as:

$$L^*(\lambda) = \min_{\mathbf{x}\in\mathcal{D}} L(\mathbf{x}, \lambda) = \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x})$$

   What makes the theory of duality constructive is when we can solve for L^* efficiently - either in a closed form or some other 'simple' mechanism. If L^* is not easy to evaluate, the duality theory will be less useful.

3. We finally define the dual problem:

$$\begin{array}{ll} \max\limits_{\lambda\in\Re^m} & L^*(\lambda) \\ \text{subject to} & \lambda \geq \mathbf{0} \end{array} \tag{4.81}$$

It can be immediatly proved that the dual problem is a concave maximization problem.

Theorem 80 *The dual function $L^*(\lambda)$ is concave.*

Proof: Consider two values of the dual variables, *viz.*, $\lambda_1 \geq \mathbf{0}$ and $\lambda_2 \geq \mathbf{0}$. Let $\lambda = \theta\lambda_1 + (1-\theta)\lambda_2$ for any $\theta \in [0, 1]$. Then,

$$\begin{aligned} L^*(\lambda) &= \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) + \lambda^T g(\mathbf{x}) \\ &= \min_{\mathbf{x}\in\mathcal{D}} \theta \left[f(\mathbf{x}) + \lambda_1^T g(\mathbf{x})\right] + (1-\theta)\left[f(\mathbf{x}) + \lambda_2^T g(\mathbf{x})\right] \\ &\geq \min_{\mathbf{x}\in\mathcal{D}} \theta \left[f(\mathbf{x}) + \lambda_1^T g(\mathbf{x})\right] + \min_{\mathbf{x}\in\mathcal{D}} (1-\theta)\left[f(\mathbf{x}) + \lambda_2^T g(\mathbf{x})\right] \\ &= \theta L^*(\lambda_1) + (1-\theta) L^*(\lambda_2) \end{aligned}$$

This proves that $L^*(\lambda)$ is a concave function. □

The dual is concave (or the negative of the dual is convex) irrespective of the primal. Solving the dual is therefore always a convex programming problem. Thus, in some sense, the dual is better structured than the primal. However, the dual cannot be drastically simpler than the primal. For example, if the primal is not an LP, the dual cannot be an LP. Similarly, the dual can be quadratic only if the primal is quadratic.

A tricky thing in duality theory is to decide what we call the domain or *ground set* $\mathcal{D}$ and what we call the constraints g_i's. Based on whether constraints are explicitly stated or implicitly stated in the form of the ground set, the dual problem could be very different. Thus, many duals are possible for the given primal.

We will look at two examples to give a flavour of how the duality theory works.

1. We will first look at linear programming.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & -A\mathbf{x}+\mathbf{b}\leq\mathbf{0} \end{array}$$

The lagrangian for this problem is:

$$L(\mathbf{x},\lambda)=\mathbf{c}^T\mathbf{x}+\lambda^T\mathbf{b}-\lambda^TA\mathbf{x}=\mathbf{b}^T\lambda+\left(\mathbf{c}^T-A^T\lambda\right)\mathbf{x}$$

The next step is to get L^*, which we obtain using the first derivative test:

$$L^*(\lambda)=\min_{\mathbf{x}\in\Re^n}\mathbf{b}^T\lambda+\left(\mathbf{c}^T-A\lambda\right)^T\mathbf{x}=\begin{cases}\mathbf{b}^T\lambda & \text{if } A^T\lambda=\mathbf{c}\\ -\infty & \text{if } A^T\lambda\neq\mathbf{c}\end{cases}$$

The function L^* can be thought of as the extended value extension of the same function restricted to the domain $\left\{\lambda|A^T\lambda=\mathbf{c}\right\}$. Therefore, the dual problem can be formulated as:

$$\begin{array}{ll} \max\limits_{\lambda\in\Re^m} & \mathbf{b}^T\lambda \\ \text{subject to} & A^T\lambda=\mathbf{c} \\ & \lambda\geq\mathbf{0} \end{array} \tag{4.82}$$

This is the dual of the standard LP. What if the original LP was the following?

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & -A\mathbf{x}+\mathbf{b}\leq\mathbf{0} \quad \mathbf{x}\geq\mathbf{0} \end{array}$$

Now we have a variety of options based on what constraints are introduced into the ground set (or domain) and what are explicitly treated as constraints. Some working out will convince us that treating $\mathbf{x}\in\Re^n$ as the constraint and the explicit constraints as part of the ground set is a very bad idea. One dual for this problem is the same as (4.82).

2. Let us look at a modified version of the problem in (4.83).

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & \mathbf{c}^T\mathbf{x}-\sum_{i=1}^n \ln x_i \\ \text{subject to} & -A\mathbf{x}+\mathbf{b}=\mathbf{0} \\ & \mathbf{x}>\mathbf{0} \end{array}$$

Typically, when we try to formulate a dual problem, we look for constraints that get in the way of conveniently solving the problem. We first formulate the lagrangian for this problem.

$$L(\mathbf{x},\lambda)=\mathbf{c}^T\mathbf{x}-\sum_{i=1}^n \ln x_i+\lambda^T\mathbf{b}-\lambda^T A\mathbf{x}=\mathbf{b}^T\lambda+\mathbf{x}^T\left(\mathbf{c}-A^T\lambda\right)-\sum_{i=1}^n \ln x_i$$

The domain (or ground set) for this problem is $\mathbf{x}>\mathbf{0}$, which is open.

The expression for L^* can be obtained using the first derivative test, while keeping in mind that L can be made arbitrarily small (tending to $-\infty$) unless $(\mathbf{c}-A^T\lambda)>\mathbf{0}$. This is because, even if one component of $\mathbf{c}-A^T\lambda$ is less than or equal to zero, the value of L can be made arbitrarily small by decreasing the value of the corresponding component of $\mathbf{x}$ in the $\sum_{i=1}^n \ln x_i$ part. Further, the sum $\mathbf{b}^T\lambda+\left(\mathbf{c}-A^T\lambda\right)^T\mathbf{x}-\sum_{i=1}^n \ln x_i$ can be separated out into the individual components of λ_i, and this can be exploited while determining the critical point of L.

$$L^*(\lambda)=\min_{\mathbf{x}>\mathbf{0}} L(\mathbf{x},\lambda)=\begin{cases} \mathbf{b}^T\lambda+n+\sum_{i=1}^n \ln\frac{1}{(\mathbf{c}-A^T\lambda)_i} & \text{if } (\mathbf{c}-A^T\lambda)>\mathbf{0} \\ -\infty & \text{otherwise} \end{cases}$$

Finally, the dual will be

$$\begin{array}{ll} \max\limits_{\lambda\in\Re^m} & \mathbf{b}^T\lambda+n+\sum_{i=1}^n \ln\frac{1}{(\mathbf{c}-A^T\lambda)_i} \\ \text{subject to} & -A^T\lambda+\mathbf{c}>\mathbf{0} \end{array}$$

As noted earlier, the theory of duality remains a theory unless the dual lends itself to some constructive evaluation; not always is the dual a useful form.

The following *Weak duality theorem* states an important relationship between solutions to the primal (4.80) and the dual (4.81) problems.

Theorem 81 *If $p^* \in \Re$ is the solution to the primal problem in (4.80) and $d^* \in \Re$ is the solution to the dual problem in (4.81), then*

$$p^* \geq d^*$$

In general, if $\widehat{\mathbf{x}}$ is any feasible solution to the primal problem (4.80) and $\widehat{\lambda}$ is a feasible solution to the dual problem (4.81), then

$$f(\widehat{\mathbf{x}}) \geq L^*(\widehat{\lambda})$$

Proof: If $\widehat{\mathbf{x}}$ is a feasible solution to the primal problem (4.80) and $\widehat{\lambda}$ is a feasible solution to the dual problem, then

$$f(\widehat{\mathbf{x}}) \geq f(\widehat{\mathbf{x}}) + \widehat{\lambda}^T \mathbf{g}(\widehat{\lambda}) \geq \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x} + \widehat{\lambda}^T \mathbf{g}(\lambda) = L^*(\widehat{\lambda})$$

This proves the second part of the theorem. A direct consequence of this is that

$$p^* = \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) \geq \min_{\lambda \geq \mathbf{0}} L^*(\lambda) = d^*$$

□

The weak duality theorem has some important implications. If the primal problem is unbounded below, that is, $p^* = -\infty$, we must have $d^* = -\infty$, which means that the Lagrange dual problem is infeasible. Conversely, if the dual problem is unbounded above, that is, $d^* = \infty$, we must have $p^* = \infty$, which is equivalent to saying that the primal problem is infeasible. The difference, $p^* - d^*$ is called the duality gap.

In many hard combinatorial optimization problems with duality gaps, we get good dual solutions, which tell us that we are guaranteed of being some k % within the optimal solution to the primal, for some satisfactorily low values of k. This is one of the powerful uses of duality theory; constructing bounds for optimization problems.

Under what conditions can one assert that $d^* = p^*$? The condition $d^* = p^*$ is called *strong duality* and it does not hold in general. It usually holds for convex problems but there are exceptions to that - one of the most typical being that of the semi-definite optimization problem. The semi-definite program (SDP) is defined, with the linear matrix inequality constraint (*c.f.* page 262) as follows:

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \Re^n} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & x_1 A_1 + \ldots + x_n A_n + G \preceq 0 \\ & A\mathbf{x} = \mathbf{b} \end{array} \tag{4.83}$$

Sufficient conditions for strong duality in convex problems are called *constraint qualifications*. One of the most useful sufficient conditions for strong duality is called the *Slaters constraint qualification*.

Definition 42 [Slaters constraint qualification]: *For a convex problem*

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{D}} & f(\mathbf{x}) \\ \textit{subject to} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & A\mathbf{x} = \mathbf{b} \\ \textit{variable } \mathbf{x} = (x_1, \ldots, x_n) & \end{array} \tag{4.84}$$

strong duality holds (that is $d^* = p^*$) if it is strictly feasible. That is,

$$\exists \mathbf{x} \in int(\mathcal{D}) : \quad g_i(\mathbf{x}) < 0 \quad i = 1, 2, \ldots, m \quad A\mathbf{x} = \mathbf{b}$$

However, if any of the g_i's are linear, they do not need to hold with strict inequalities.

Table 4.4 summarizes some optimization problems, their duals and conditions for strong duality. Strong duality also holds for nonconvex problems

Problem type	**Objective Function**	**Constraints**	$L^*(\lambda)$	**Dual constraints**	**Strong duality**
Linear Program	$\mathbf{c}^T\mathbf{x}$	$A\mathbf{x} \leq \mathbf{b}$	$-\mathbf{b}^T\lambda$	$A^T\lambda + \mathbf{c} = \mathbf{0}$ $\lambda \geq \mathbf{0}$	Feasible primal and dual
Quadratic Program	$\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$ for $Q \in \mathcal{S}^n_{++}$	$A\mathbf{x} \leq \mathbf{b}$	$-\frac{1}{2}\left(\mathbf{c} - A^T\lambda\right)^T Q^{-1}\left(\mathbf{c} - A^T\lambda\right) + \mathbf{b}^T\lambda$	$\lambda \geq \mathbf{0}$	Always
Entropy maximization	$x_i \sum_{i=1}^n \ln x_i$	$A\mathbf{x} \leq \mathbf{b}$ $\mathbf{x}^T\mathbf{1} = 1$	$-\mathbf{b}^T\lambda - \mu - e^{-\mu-1}\sum_{i=1}^n e^{-\mathbf{a}_i^T\lambda}$ $\mathbf{a}_i$ is the i^{th} column of A	$\lambda \geq \mathbf{0}$	Primal constraints are satisfied.

Table 4.4: Examples of functions and their duals.

in extremely rare cases. One example of this is minimization of a nonconvex quadratic function over the unit ball.

4.4.3 Geometry of the Dual

We will study the geometry of the dual in the *column space* $\Re^{m+1}$. The column geometry of the dual will require definition of the following set:

$$\mathcal{I} = \{(\mathbf{s}, z) \mid \mathbf{s} \in \Re^m,\ z \in \Re,\ \exists \mathbf{x} \in \mathcal{D}\ with\ g_i(\mathbf{x}) \leq s_i\ \forall 1 \leq i \leq m,\ f(\mathbf{x}) \leq z\}$$

The set $\mathcal{I}$ is a subset of $\Re^{m+1}$, where m is the number of constraints. Consider a plot in two dimensions, for $n = 1$, with s_1 along the $x-$axis and z along the $y-$axis. For every point, $\mathbf{x} \in \mathcal{D}$, we can identify all points (s_1, z) for $s_1 \geq g_1(\mathbf{x})$

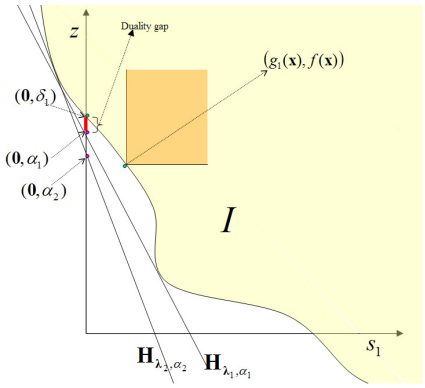

Figure 4.42: Example of the set $\mathcal{I}$ for a single constraint (*i.e.*, for $n = 1$).

and $z \geq f(\mathbf{x})$ and these are points that lie to the right and above the point $(g_1(\mathbf{x}), f(\mathbf{x}))$. An example set $\mathcal{I}$ is shown in Figure 4.42. It turns out that all the intuitions we need are in two dimensions, which makes it fairly convenient to understand the idea. It is straightforward to prove that if the objective function $f(\mathbf{x})$ is convex and each of the constraints $g_i(\mathbf{x}),\ 1 \leq i \leq n$ is a convex function, then $\mathcal{I}$ must be a convex set. Since the feasible region for the primal problem (4.78) is the region in $\mathcal{I}$ with $\mathbf{s} \leq \mathbf{0}$, and since all points above and to the right of a point in $\mathcal{I}$ also belong to $\mathcal{I}$, the solution to the primal problem corresponds to the point in $\mathcal{I}$ with $\mathbf{s} = \mathbf{0}$ and least possible value of z. For example, in Figure 4.42, the solution to the primal corresponds to $(\mathbf{0}, \delta_1)$.

Let us define a hyerplane $\mathcal{H}_{\lambda,\alpha}$, parametrized by $\lambda \in \Re^m$ and $\alpha \in \Re$ as

$$\mathcal{H}_{\lambda,\alpha} = \left\{ (\mathbf{s}, z) \,\middle|\, \lambda^T.\mathbf{s} + z = \alpha \right\}$$

Consider all hyperplanes that lie below $\mathcal{I}$. For example, in the Figure 4.42, both hyperplanes $\mathcal{H}_{\lambda_1,\alpha_1}$ and $\mathcal{H}_{\lambda_2,\alpha_2}$ lie below the set $\mathcal{I}$. Of all hyperplanes that lie below $\mathcal{I}$, consider the hyperplane whose intersection with the line $\mathbf{s} = \mathbf{0}$, corresponds to as high a value of z as possible. This hyperplane must be supporting hyperplane. Incidentally, $\mathcal{H}_{\lambda_1,\alpha_1}$ happens to be such a supporting hyperplane. Its point of intersection $(\mathbf{0}, \alpha_1)$ precisely corresponds to the solution to the dual problem. Let us derive this statement formally after setting up some more notation.

We will define two half-spaces corresponding to $\mathcal{H}_{\lambda,\alpha}$

$$\mathcal{H}^+_{\lambda,\alpha} = \left\{ (\mathbf{s}, z) \,\middle|\, \lambda^T.\mathbf{s} + z \geq \alpha \right\}$$

$$\mathcal{H}^-_{\lambda,\alpha} = \left\{ (\mathbf{s}, z) \,\middle|\, \lambda^T.\mathbf{s} + z \leq \alpha \right\}$$

Let us define another set $\mathcal{L}$ as

$$\mathcal{L} = \left\{ (\mathbf{s}, z) \,\middle|\, \mathbf{s} = \mathbf{0} \right\}$$

Note that $\mathcal{L}$ is essentially the z or function axis. The intersection of $\mathcal{H}_{\lambda,\alpha}$ with $\mathcal{L}$ is the point $(\mathbf{0}, \alpha)$. That is

$$(\mathbf{0}, \alpha) = \mathcal{L} \bigcap \mathcal{H}_{\lambda,\alpha}$$

We would like to manipulate λ and α so that the set $\mathcal{I}$ lies in the half-space $\mathcal{H}^+_{\lambda,\alpha}$ as tightly as possible. Mathematically, we are interested in the problem of maximizing the height of the point of intersection of $\mathcal{L}$ with $\mathcal{H}_{\lambda,\alpha}$ above the $\mathbf{s} = \mathbf{0}$ plane, while ensuring that $\mathcal{I}$ remains a subset of $\mathcal{H}^+_{\lambda,\alpha}$.

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \mathcal{H}^+_{\lambda,\alpha} \supseteq \mathcal{I} \end{array}$$

By definitions of $\mathcal{I}$, $\mathcal{H}^+_{\lambda,\alpha}$ and the subset relation, this problem is equivalent to

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \lambda^T.\mathbf{s} + z \geq \alpha \ \forall (\mathbf{s}, z) \in \mathcal{I} \end{array}$$

Now notice that if $(\mathbf{s}, z) \in \mathcal{I}$, then $(\mathbf{s}', z) \in \mathcal{I}$ for all $\mathbf{s}' \geq \mathbf{s}$. This was also illustrated in Figure 4.42. Thus, we cannot afford to have any component of λ negative; if any of the λ_i's were negative, we could cranck up s_i arbitrarily to violate the inequality $\lambda^T.\mathbf{s} + z \geq \alpha$. Thus, we can add the constraint $\lambda \geq \mathbf{0}$ to the above problem without changing the solution.

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \lambda^T.\mathbf{s} + z \geq \alpha \ \forall (\mathbf{s}, z) \in \mathcal{I} \\ & \lambda \geq \mathbf{0} \end{array}$$

Any equality constraint $h(\mathbf{x}) = 0$ can be expressed using two inequality constraints, *viz.*, $h(\mathbf{x}) \leq 0$ and $-h(\mathbf{x}) \leq 0$. This problem can again be proved to be equivalent to the following problem, using the definition of $\mathcal{I}$ or equivalently, the fact that every point on $\partial\mathcal{I}$ must be of the form $(g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x}), f(\mathbf{x}))$ for some $\mathbf{x} \in \mathcal{D}$.

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \lambda^T.\mathbf{g}(\mathbf{x}) + f(\mathbf{x}) \geq \alpha \ \forall \mathbf{x} \in \mathcal{D} \\ & \lambda \geq \mathbf{0} \end{array}$$

We will remind the reader at this point that $L(\mathbf{x}, \lambda) = \lambda^T.\mathbf{g}(\mathbf{x}) + f(\mathbf{x})$. The above problem is therefore the same as

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & L(\mathbf{x}, \lambda) \geq \alpha \ \forall \mathbf{x} \in \mathcal{D} \\ & \lambda \geq \mathbf{0} \end{array}$$

Since, $L^*(\lambda) = \min_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda)$, we can deal with the equivalent problem

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & L^*(\lambda) \geq \alpha \\ & \lambda \geq \mathbf{0} \end{array}$$

This problem can be restated as

$$\begin{array}{ll} \max & L^*(\lambda) \\ \text{subject to} & \lambda \geq \mathbf{0} \end{array}$$

This is precisely the dual problem. We thus get a geometric interpretation of the dual.

Again referring to Figure 4.42, we note that if the set $\mathcal{I}$ is not convex, there could be a *gap* between the $z-$intercept $(\mathbf{0}, \alpha_1)$ of the best supporting hyperplane $\mathcal{H}_{\lambda_1, \alpha_1}$ and the closest point $(\mathbf{0}, \delta_1)$ of $\mathcal{I}$ on the $z-$axis, which corresponds to the solution to the primal. In fact, when the set $\mathcal{I}$ is not convex, we can never prove that there will be no duality gap. And even when the set $\mathcal{I}$ is convex, bizzaire things can happen; for example, in the case of semi-definite programming, the set $\mathcal{I}$, though convex, is not at all well-behaved and this yields a large duality gap, as shown in Figure 4.43. In fact, the set $\mathcal{I}$ is open from below (the dotted boundary) for a semi-definite program. We could create very simple problems with convex $\mathcal{I}$, for which there are duality gaps. For well-behaved convex functions (as in the case of linear programming), there are no duality gaps. Figure 4.44 illustrates the case of a well-behaved convex program.

4.4.4 Complementary slackness and KKT Conditions

We now state the conditions between the primal and dual optimal points for an arbitrary function. These conditions, called the *Karush-Kuhn-Tucker conditions* (abbreviated as KKT conditions) state a necessary condition for a solution to be optimal with zero duality gap. Consider the following general optimization problem.

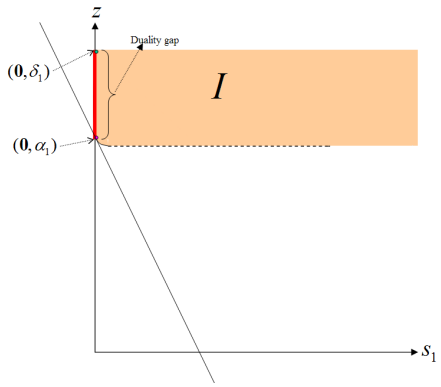

Figure 4.43: Example of the convex set $\mathcal{I}$ for a single constrained semi-definite program.

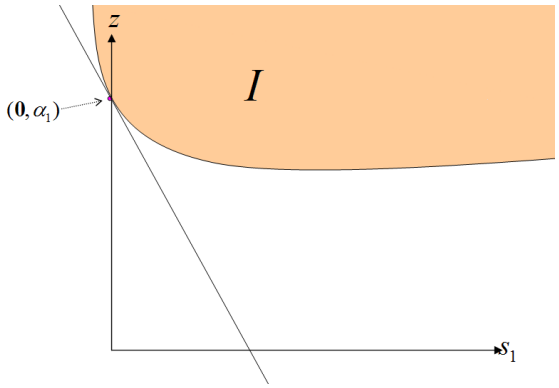

Figure 4.44: Example of the convex set $\mathcal{I}$ for a single constrained well-behaved convex program.

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathcal{D}} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, & i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, & j = 1, \ldots, p \\ \text{variable } \mathbf{x} = (x_1, \ldots, x_n) & & \end{array} \tag{4.85}$$

Suppose that the primal and dual optimal values for the above problem are attained and equal, that is, strong duality holds. Let $\widehat{\mathbf{x}}$ be a primal optimal and $(\widehat{\lambda}, \widehat{\mu})$ be a dual optimal point ($\widehat{\lambda} \in \Re^m, \widehat{\mu} \in \Re^p$). Thus,

$$\begin{aligned} f(\widehat{\mathbf{x}}) &= L^*(\widehat{\lambda}, \widehat{\mu}) \\ &= \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) + \widehat{\lambda}^T \mathbf{g}(\mathbf{x}) + \widehat{\mu}^T \mathbf{h}(\mathbf{x}) \\ &\leq f(\widehat{\mathbf{x}}) + \widehat{\lambda}^T \mathbf{g}(\widehat{\mathbf{x}}) + \widehat{\mu}^T \mathbf{h}(\widehat{\mathbf{x}}) \\ &\leq f(\widehat{\mathbf{x}}) \end{aligned}$$

The last inequality follows from the fact that $\widehat{\lambda} \geq \mathbf{0}$, $\mathbf{g}(\widehat{\mathbf{x}}) \leq \mathbf{0}$, and $\mathbf{h}(\widehat{\mathbf{x}}) = \mathbf{0}$. We can therefore conclude that the two inequalities in this chain must hold with equality. Some of the conclusions that we can draw from this chain of equalities are

1. That $\widehat{\mathbf{x}}$ is a minimizer for $L(\mathbf{x}, \widehat{\lambda}, \widehat{\mu})$ over $\mathbf{x} \in \mathcal{D}$. In particular, if the functions f, $g_1, g_2, \ldots, g_m$ and $h_1, h_2, \ldots, h_p$ are differentiable (and therefore have open domains), the gradient of $L(\mathbf{x}, \widehat{\lambda}, \widehat{\mu})$ must vanish at $\widehat{\mathbf{x}}$, since any point of global optimum must be a point of local optimum. That is,

$$\nabla f(\widehat{\mathbf{x}}) + \sum_{i=1}^{m} \widehat{\lambda}_i \nabla g_i(\widehat{\mathbf{x}}) + \sum_{j=1}^{p} \widehat{\mu}_j \nabla h_j(\widehat{\mathbf{x}}) = \mathbf{0} \tag{4.86}$$

2. That

$$\widehat{\lambda}^T \mathbf{g}(\widehat{\mathbf{x}}) = \sum_{i=1}^{n} \widehat{\lambda}_i g_i(\widehat{\mathbf{x}}) = 0$$

Since each term in this sum is nonpositive, we conclude that

$$\widehat{\lambda}_i g_i(\widehat{\mathbf{x}}) = 0 \; for \; i = 1, 2, \ldots, m \tag{4.87}$$

This condition is called *complementary slackness* and is a necessary condition for strong duality. Complementary slackness implies that the i^{th}

optimal lagrange multiplier is 0 unless the i^{th} inequality constraint is active at the optimum. That is,

$$\begin{aligned} \widehat{\lambda}_i > 0 \quad &\Rightarrow \quad g_i(\widehat{\mathbf{x}}) = 0 \\ g_i(\widehat{\mathbf{x}}) < 0 \quad &\Rightarrow \quad \widehat{\lambda}_i = 0 \end{aligned}$$

Let us further assume that the functions f, $g_1, g_2, \ldots, g_m$ and $h_1, h_2, \ldots, h_p$ are differentiable on open domains. As above, let $\widehat{\mathbf{x}}$ be a primal optimal and $(\widehat{\lambda}, \widehat{\mu})$ be a dual optimal point with zero duality gap. Putting together the conditions in (4.86), (4.87) along with the feasibility conditions for any primal solution and dual solution, we can state the following Karush-Kuhn-Tucker (KKT) necessary conditions for zero duality gap.

$$\begin{array}{lrcll} (1) & \nabla f(\widehat{\mathbf{x}}) + \sum_{i=1}^{m} \widehat{\lambda}_i \nabla g_i(\widehat{\mathbf{x}}) + \sum_{j=1}^{p} \widehat{\mu}_j \nabla h_j(\widehat{\mathbf{x}}) & = & \mathbf{0} & \\ (2) & g_i(\widehat{\mathbf{x}}) & \leq & 0 & i = 1, 2, \ldots, m \\ (3) & \widehat{\lambda}_i & \geq & 0 & i = 1, 2, \ldots, m \\ (4) & \widehat{\lambda}_i g_i(\widehat{\mathbf{x}}) & = & 0 & i = 1, 2, \ldots, m \\ (5) & h_j(\widehat{\mathbf{x}}) & = & 0 & j = 1, 2, \ldots, p \end{array} \quad (4.88)$$

When the primal problem is convex, the KKT conditions are also sufficient for the points to be primal and dual optimal with zero duality gap. If f is convex, g_i are convex and h_j are affine, the primal problem is convex and consequently, the KKT conditions are sufficient conditions for zero duality gap.

Theorem 82 *If the function f is convex, g_i are convex and h_j are affine, then KKT conditions in 4.88 are necessary and sufficient conditions for zero duality gap.*

Proof: The necessity part has already been proved; here we only prove the sufficiency part. The conditions (2) and (5) in (4.88) ensure that $\widehat{\mathbf{x}}$ is primal feasible. Since $\lambda \geq \mathbf{0}$, $L(\mathbf{x}, \widehat{\lambda}, \widehat{\mu})$ is convex in $\mathbf{x}$. Based on condition (1) in (4.88) and theorem 77, we can infer that $\widehat{\mathbf{x}}$ minimizes $L(\mathbf{x}, \widehat{\lambda}, \widehat{\mu})$. We can thus conclude that

$$\begin{aligned} L^*(\widehat{\lambda}, \widehat{\mu}) &= f(\widehat{\mathbf{x}}) + \widehat{\lambda}^T \mathbf{g}(\widehat{\mathbf{x}}) + \widehat{\mu}^T \mathbf{h}(\widehat{\mathbf{x}}) \\ &= f(\widehat{\mathbf{x}}) \end{aligned}$$

In the equality above, we use $h_j(\widehat{\mathbf{x}}) = 0$ and $\widehat{\lambda}_i g_i(\widehat{\mathbf{x}}) = 0$. Further,

$$d^* \geq L^*(\widehat{\lambda}, \widehat{\mu}) == f(\widehat{\mathbf{x}}) \geq p^*$$

The duality theorem (theorem 81) however states that $p^* \geq d^*$. This implies that

$$d^* = L^*(\widehat{\lambda}, \widehat{\mu}) == f(\widehat{\mathbf{x}}) = p^*$$

This shows that $\widehat{\mathbf{x}}$ and $(\widehat{\lambda}, \widehat{\mu})$ correspond to the primal and dual optimals respectively and the problem therefore has zero duality gap. □

In summary, for any convex optimization problem with differentiable objective and constraint functions, any points that satisfy the KKT conditions are primal and dual optimal, and have zero duality gap.

The KKT conditions play a very important role in optimization. In some rare cases, it is possible to solve the optimization problems by finding a solution to the KKT conditions analytically. Many algorithms for convex optimization are conceived as, or can be interpreted as, methods for solving the KKT conditions.

4.5 Algorithms for Unconstrained Minimization

We will now study some algorithms for solving convex problems. These techniques are relevant for most convex optimization problems that do not yield themselves to closed form solutions. We will start with unconstrained minimization.

Recall that the goal in unconstrained minimization is to solve the convex problem

$$\min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$$

We are not interested in f, whose solution can be obtained in closed form. For example, minimizing a quadratic is very simple and can be solved by linear equations, an example of which was discussed in Section 3.9.2. Let us denote the optimal solution of the minimization problem by p^*. We will assume that f is convex and twice continuously differentiable and that it attains a finite optimal value p^*. Most unconstrained minimization techniques produce a sequence of points $\mathbf{x}^{(k)} \in \mathcal{D}, k = 0, 1, \ldots$ such that $f\left(\mathbf{x}^{(k)}\right) \rightarrow p^*$ as $k \rightarrow \infty$ or, $\nabla f\left(\mathbf{x}^{(k)}\right) \rightarrow \mathbf{0}$ as $k \rightarrow \infty$. Iterative techniques for optimization, further require a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$ and sometimes that $epi(f)$ is closed. The $epi(f)$ can be inferred to be closed either if $\mathcal{D} = \Re^n$ or $f(\mathbf{x}) \rightarrow \infty$ as $\mathbf{x} \rightarrow \partial\mathcal{D}$. The function $f(x) = \frac{1}{x}$ for $x > 0$ is an example of a function whose $epi(f)$ is closed.

While there exist convergence proofs (including guarantees on number of optimization iterations) for many convex optimization algorithms, the proofs assume many conditions,many of which are either not verifiable or involve unknown constants (such as the Lipshitz constant). Thus, most convergence proofs for convex optimization problems are useless in practice, though it is good to know that there are conditions under which the algorithm converges. Since convergence proofs are only of theoretical importance, we will make the strongest possible assumption under which convergence can be proved easily, which is that the function f is strongly convex (*c.f.* Section 4.2.7 for definition of strong convexity) with the strong convexity constant $c > 0$ for which $\nabla^2 f(\mathbf{x}) \succeq cI \; \forall x \in \mathcal{D}$.

Further, it can be proved that for a strongly convex function f, $\nabla^2 f(\mathbf{x}) \preceq DI$ for some constant $D \in \Re$. The ratio $\frac{D}{c}$ is an upper bound on the condition number of the matrix $\nabla^2 f(\mathbf{x})$.

4.5.1 Descent Methods

Descent methods for unconstrained optimization have been in use since the last 70 years or more. The general idea in descent methods is that the next iterate $\mathbf{x}^{(k+1)}$ is the current iterate $\mathbf{x}^{(k)}$ added with a descent or search direction $\Delta\mathbf{x}^{(k)}$ (a unit vector), which is multiplied by a scale factor $t^{(k)}$, called the step length.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$$

The incremental step is determined while ensuring that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. We assume that we are dealing with the extended value extension of the convex function f (*c.f.* definition 36), which returns ∞ for any point outside its domain. However, if we do so, we need to make sure that the initial point indeed lies in the domain $\mathcal{D}$.

A single iteration of the general descent algorithm (shown in Figure 4.45) consists of two main steps, *viz.*, determining a good descent direction $\Delta\mathbf{x}^{(k)}$, which is typically forced to have unit norm and determining the step size using some line search technique. If the function f is convex, and we require that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ then, we must have $\nabla^T f(\mathbf{x}^{(k+1)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) < 0$. This can be seen from the necessary and sufficient condition for convexity stated in equation (4.44) within Section 4.2.9 and restated here for reference.

$$f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)}) + \nabla^T f(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

Since $t^{(k)} > 0$, we must have

$$\nabla^T f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} < 0$$

That is, the descent direction $\Delta\mathbf{x}^{(k)}$ must make an obtuse angle ($\theta \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right)$) with the gradient vector.

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$
repeat
1. Determine $\Delta\mathbf{x}^{(k)}$.
2. Choose a step size $t^{(k)} > 0$ using ray[a] search.
3. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
4. Set $k = k + 1$.

until stopping criterion (such as $||\nabla f(\mathbf{x}^{(k+1)})|| < \epsilon$) is satisfied

[a] Many textbooks refer to this as line search, but we prefer to call it ray search, since the step must be positive.

Figure 4.45: The general descent algorithm.

There are many different empirical techniques for ray search, though it matters much less than the search for the descent direction. These techniques reduce the $n-$dimensional problem to a $1-$dimensional problem, which can be easy to solve by use of plotting and eyeballing or even exact search.

1. **Exact ray search:** The exact ray search seeks a scaling factor t that satisfies

$$t = \underset{t>0}{\operatorname{argmin}} f(\mathbf{x} + t\Delta\mathbf{x}) \tag{4.89}$$

2. **Backtracking ray search:** The exact line search may not be feasible or could be expensive to compute for complex non-linear functions. A relatively simpler ray search iterates over values of step size starting from 1 and scaling it down by a factor of $\beta \in (0, \frac{1}{2})$ after every iteration till the following condition, called the *Armijo condition* is satisfied for some $0 < c_1 < 1$.

$$f(\mathbf{x} + t\Delta\mathbf{x}) < f(\mathbf{x}) + c_1 t \nabla^T f(\mathbf{x})\Delta\mathbf{x} \tag{4.90}$$

Based on equation (4.44), it can be inferred that the Armijo inequality can never hold for $c_1 = 1$; for $c_1 = 1$, the right hand side of the Armijo condition gives a lower bound on the value of $f(\mathbf{x} + t\Delta\mathbf{x})$. The Armijo condition simply ensures that t decreases f sufficiently. Often, another condition is used for inexact line search in conjunction with the Armijo condition.

$$\left|\Delta\mathbf{x}^T \nabla f(\mathbf{x} + t\Delta\mathbf{x})\right| \leq c_2 \left|\Delta\mathbf{x}^T \nabla f(\mathbf{x})\right| \tag{4.91}$$

where $1 > c_1 > c_2 > 0$. This condition ensures that the slope of the function $f(\mathbf{x}+t\Delta\mathbf{x})$ at t is less than c_2 times that at $t = 0$. The conditions in (4.90) and (4.91) are together called the strong Wolfe conditions. These conditions are particularly very important for non-convex problems.

A finding that is borne out of plenty of empirical evidence is that exact ray search does better than empirical ray search in a few cases only. Further, the exact choice of the value of β and α seems to have little effect on the convergence of the overall descent method.

The trend of specific descent methods has been like a parabola - starting with simple steepest descent techniques, then accomodating the curvature hessian matrix through a more sophisticated Newton's method and finally, trying to simplify the Newton's method through approximations to the hessian inverse,

culminating in conjugate gradient techniques, that do away with any curvature matrix whatsoever, and form the internal combustion engine of many sophisticated optimization techniques today. We start the thread by describing the steepest descent methods.

Steepest Descent

Let $\mathbf{v} \in \Re^n$ be a unit vector under some norm. By theorem 75, for convex f,

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{v}) \leq -\nabla^T f(\mathbf{x}^{(k)})\mathbf{v}$$

For small $\mathbf{v}$, the inequality turns into approximate equality. The term $-\nabla^T f(\mathbf{x}^{(k)})\mathbf{v}$ can be thought of as (an upper-bound on) the first order prediction of decrease. The idea in the steepest descent method [**?**] is to choose a norm and then determine a descent direction such that for a unit step in that norm, the first order prediction of decrease is maximized. This choice of the descent direction can be stated as

$$\Delta\mathbf{x} = \text{argmin}\left\{\nabla^T f(\mathbf{x})\mathbf{v} \mid ||\mathbf{v}|| = 1\right\}$$

The algorithm is outlined in Figure 4.46.

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$.
repeat
1. Set $\Delta\mathbf{x}^{(k)} = \text{argmin}\left\{\nabla^T f(\mathbf{x}^{(k)})\mathbf{v} \mid ||\mathbf{v}|| = 1\right\}$.
2. Choose a step size $t^{(k)} > 0$ using exact or backtracking ray search.
3. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
4. Set $k = k + 1$.

until stopping criterion (such as $||\nabla f(\mathbf{x}^{(k+1)})|| \leq \epsilon$) is satisfied

Figure 4.46: The steepest descent algorithm.

The key to understanding the steepest descent method (and in fact many other iterative methods) is that it heavily depends on the choice of the norm. It has been empirically observed that if the norm chosen is aligned with the gross geometry of the sub-level sets[22], the steepest descent method converges faster to the optimal solution. If the norm chosen is not aligned, it often amplifies the effect of oscillations. Two examples of the steepest descent method are the gradient descent method (for the eucledian or L_2 norm) and the coordinate-descent method (for the L_1 norm). One fact however is that no two norms should give exactly opposite steepest descent directions, though they may point in different directions.

Gradient Descent

A classic greedy algorithm for minimization is the gradient descent algorithm. This algorithm uses the negative of the gradient of the function at the current

[22]The alignment can be determined by fitting, for instance, a quadratic to a sample of the points.

point $\mathbf{x}^*$ as the descent direction $\Delta\mathbf{x}^*$. It turns out that this choice of $\Delta\mathbf{x}^*$ corresponds to the direction of steepest descent under the L_2 (eucledian) norm. This can be proved in a straightforward manner using theorem 58. The algorithm is outlined in Figure 4.47. The steepest descent method can be thought

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$
repeat
1. Set $\Delta\mathbf{x}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$.
2. Choose a step size $t^{(k)} > 0$ using exact or backtracking ray search.
3. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
4. Set $k = k + 1$.

until stopping criterion (such as $||\nabla f(\mathbf{x}^{(k+1)})||_2 \leq \epsilon$) is satisfied

Figure 4.47: The gradient descent algorithm.

of as changing the coordinate system in a particular way and then applying the gradient descent method in the changed coordinate system.

Coordinate-Descent Method

The co-ordinate descent method corresponds exactly to the choice of L_1 norm for the steepest descent method. The steepest descent direction using the L_1 norm is given by

$$\Delta\mathbf{x} = -\frac{\partial f(\mathbf{x})}{\partial x_i}\mathbf{u}^i$$

where,

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = ||\nabla f(\mathbf{x})||_\infty$$

and $\mathbf{u}^i$ was defined on page 231 as the unit vector pointing along the i^{th} coordinate axis. Thus each iteration of the coordinate descent method involves optimizing over one component of the vector $\mathbf{x}^{(k)}$ and then updating the vector. The component chosen is the one having the largest absolute value in the gradient vector. The algorithm is outlined in Figure 4.48.

Convergence of Steepest Descent Method

For the gradient method, it can be proved that if f is strongly convex,

$$f(\mathbf{x}^{(k)}) - p^* \leq \rho^k \left(f(\mathbf{x}^{(0)} - p^*\right) \tag{4.92}$$

The value of $\rho \in (0, 1)$ depends on the strong convexity constant c (*c.f.* equation (4.64) on page 277), the value of $\mathbf{x}^{(0)}$ and type of ray search employed. The suboptimality $f(\mathbf{x}^{(k)}) - p^*$ goes down by a factor $\rho < 1$ at every step and this is referred to as *linear convergence*[23]. However, this is only of theoretical

[23] A series $s_1, s_2, \ldots$ is said to have

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$.
Select an appropriate norm $||.||$.
repeat
1. Let $\frac{\partial f(\mathbf{x}^{(k)})}{\partial x_i^{(k)}} = ||\nabla f(\mathbf{x}||_\infty$.
2. Set $\Delta\mathbf{x}^{(k)} = -\frac{\partial f(\mathbf{x}^{(k)})}{\partial x_i^{(k)}}\mathbf{u}^i$.
3. Choose a step size $t^{(k)} > 0$ using exact or backtracking ray search.
4. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
5. Set $k = k + 1$.

until stopping criterion (such as $||\nabla f(\mathbf{x}^{(k+1)})||_\infty \leq \epsilon$) is satisfied

Figure 4.48: The coordinate descent algorithm.

importance, since this method is often very slow, indicated by values of ρ, very close to 1. Use of exact line search in conjunction with gradient descent also has the tendency to overshoot the next best iterate. It is therefore rarely used in practice. The convergence rate depends greatly on the condition number of the Hessian (which is upperbounded by $\frac{D}{c}$). It can be proved that the number of iterations required for the convergence of the gradient descent method is lower-bounded by the condition number of the hessian; large eigenvalues correspond to high curvature directions and small eigenvalues correspond to low curvature directions. Many methods (such as conjugate gradient) try to improve upon the gradient method by making the hessian better conditioned. Convergence can be very slow even for moderately well-conditioned problems, with condition number in the 100s, even though computation of the gradient at each step is only an $O(n)$ operation. The gradient descent method however works very well if the function is isotropic, that is if the level-curves are spherical or nearly spherical.

The convergence of the steepest descent method can be stated in the same form as in 4.92, using the fact that any norm can be bounded in terms of the Euclidean norm, *i.e.*, there exists a constant $\eta \in (0, 1]$ such that

$$||\mathbf{x}|| \geq \eta||\mathbf{x}||_2$$

1. linear convergence to $\overline{s}$ if $\lim_{i\to\infty} \frac{|s_{i+1}-\overline{s}|}{|s_i-\overline{s}|} = \delta \in (0, 1)$. For example, $s_i = (\gamma)^i$ has linear convergence to $\overline{s} = 0$ for any $\gamma < 1$. The rate of decrease is also sometimes called exponential or geometric. This is considered quite slow.

2. superlinear convergence to $\overline{s}$ if $\lim_{i\to\infty} \frac{|s_{i+1}-\overline{s}|}{|s_i-\overline{s}|} = 0$. For example, $s_i = \frac{1}{i!}$ has superlinear convergence. This is the most common.

3. quadratic convergence to $\overline{s}$ if $\lim_{i\to\infty} \frac{|s_{i+1}-\overline{s}|}{|s_i-\overline{s}|^2} = \delta \in (0, \infty)$. For example, $s_i = (\gamma)^{2^i}$ has quadratic convergence to $\overline{s} = 0$ for any $\gamma < 1$. This is considered very fast in practice.

4.5.2 Newton's Method

Newton's method [**?**] is based on approximating a function around the current iterate $\mathbf{x}^{(k)}$ using a second degree Taylor expansion.

$$f(\mathbf{x}) \approx \widetilde{f}(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla^T f(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^T \nabla^2 f(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)})$$

If the function f is convex, the quadratic approximation is also convex. Newton's method is based on solving it exactly by finding its critical point $\mathbf{x}^{(k+1)}$ as a function of $\mathbf{x}^{(k)}$. Setting the gradient of this quadratic approximation (with respect to $\mathbf{x}$) to $\mathbf{0}$ gives

$$\nabla^T f(\mathbf{x}^{(k)}) + \nabla^2 f(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = 0$$

solving which yields the next iterate as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1} \nabla f(\mathbf{x}^{(k)}) \tag{4.93}$$

assuming that the Hessian matrix is invertible. The term $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ can be thought of as an update step. This leads to a simple descent algorithm, outlined in Figure 4.49 and is called the Newton's method. It relies on the invertibility of the hessian, which holds if the hessian is positive definite as in the case of a strictly convex function. In case the hessian is invertible,cholesky factorization (page 207) of the hessian can be used to solve the linear system (4.93). However, the Newton method may not even be properly defined if the hessian is not positive definite. In this case, the hessian could be changed to a nearby positive definite matrix whenever it is not. Or a line search could be added to seek a new point having a positive definite hessian.

This method uses a step size of 1. If instead, the stepsize is chosen using exact or backtracking ray search, the method is called the *damped Newton's method*. Each Newton's step takes $O(n^3)$ time (without using any fast matrix multiplication methods).

The Newton step can also be looked upon as another incarnation of the steepest descent rule, but with the quadratic norm defined by the (local) Hessian $\nabla^2 f(\mathbf{x}^{(k)})$ evaluated at the current iterate $\mathbf{x}^{(k)}$, *i.e.*,

$$||\mathbf{u}||_{\nabla^2 f(\mathbf{x}^{(k)})} = \left(\mathbf{u}\nabla^2 f(\mathbf{x}^{(k)})\mathbf{u}\right)^{\frac{1}{2}}$$

The norm of the Newton step, in the quadratic norm defined by the Hessian at a point $\mathbf{x}$ is called the *Newton decrement* at the point $\mathbf{x}$ and is denoted by $\lambda(\mathbf{x})$. Thus,

$$\lambda(\mathbf{x}) = ||\Delta\mathbf{x}||_{\nabla^2 f(\mathbf{x})} = \nabla^T f(\mathbf{x}) \left(\nabla^2 f(\mathbf{x})\right)^{-1} \nabla f(\mathbf{x})$$

The Newton decrement gives an 'estimate' of the proximity of the current iterate $\mathbf{x}$ to the optimal point $\mathbf{x}^*$ obtained by measuring the proximity of $\mathbf{x}$ to the

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$.
Select an appropriate tolerance $\epsilon > 0$.
repeat
1. Set $\Delta\mathbf{x}^{(k)} = -\left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1} \nabla f(\mathbf{x})$.
2. Let $\lambda^2 = \nabla^T f(\mathbf{x}^{(k)}) \left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1} \nabla f(\mathbf{x}^{(k)})$.
3. If $\frac{\lambda^2}{2} \leq \epsilon$, **quit**.
4. Set step size $t^{(k)} = 1$.
5. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
6. Set $k = k + 1$.

until

Figure 4.49: The Newton's method.

minimum point of the quadratic approximation $\widetilde{f}(\mathbf{x})$. The estimate is $\frac{1}{2}\lambda(\mathbf{x})^2$ and is given as

$$\frac{1}{2}\lambda(\mathbf{x})^2 = f(\mathbf{x}) - \min\ \widetilde{f}(\mathbf{x})$$

Additionally, $\lambda(\mathbf{x})^2$ is also the directional derivative in the Newton direction.

$$\lambda(\mathbf{x})^2 = \nabla^T f(\mathbf{x})\Delta\mathbf{x}$$

The estimate $\frac{1}{2}\lambda(\mathbf{x})^2$ is used to test the convergence of the Newton algorithm in Figure 4.49.

Next, we state an important property of the Newton's update rule.

Theorem 83 *If* $\Delta\mathbf{x}^{(k)} = -\left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1} \nabla f(\mathbf{x}^{(k)})$, $\nabla^2 f(\mathbf{x}^{(k)})$ *is symmetric and positive definite and* $\Delta\mathbf{x}^{(k)} \neq \mathbf{0}$, *then* $\Delta\mathbf{x}^{(k)}$ *is a descent direction at* $\mathbf{x}^{(k)}$, *that is,* $\nabla^T f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} < 0$.

Proof: First of all, if $\nabla^2 f(\mathbf{x}^{-1})$ is symmetric and positive definite, then it is invertible and its inverse is also symmetric and positive definite. Next, we see that

$$\nabla^T f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} = -\nabla^T f(\mathbf{x}) \left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1} \nabla f(\mathbf{x}^{(k)}) < 0$$

because $\left(\nabla^2 f(\mathbf{x})\right)^{-1}$ is symmetric and positive definite. □

The Newton method is independent of affine changes of coordinates. That is, if optimizating a function $f\mathbf{x})$ using the Newton's method with an initial estimate $\mathbf{x}^{(0)}$ involves the series of iterates $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}, \ldots$, then optimizing the same problem using the Newton's method with a change of coordinates given by $\mathbf{x} = A\mathbf{y}$ and the intial estimate $\mathbf{y}^{(0)}$ such that $\mathbf{x}^{(0)} = A\mathbf{y}^{(0)}$ yields the series of iterates $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(k)}, \ldots$, such that $\mathbf{x}^{(k)} = A\mathbf{y}^{(k)}$. This is a great advantage over the gradient method, whose convergence can be very sensitive to affine transformation.

Another well known feature of the Newton's method is that it converges very fast, if at all. The convergence is extremely fast in the vicinity of the point of

optimum. This can be loosely understood as follows. If $\mathbf{x}^*$ is the critical point of a differentiable convex function f, defined on an open domain, the function is approximately equal to its second order taylor approximation in the vicinity of $\mathbf{x}^*$. Further, $\nabla f(\mathbf{x}^*) = \mathbf{0}$. This gives the following approximation at any point $\mathbf{x}$ in the vicinity of $\mathbf{x}^*$.

$$\begin{aligned} f(\mathbf{x}) &\approx f(\mathbf{x}^*) + \nabla^T f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \\ &= f(\mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \end{aligned}$$

Thus, the level curves of a convex function are approximately ellipsoids near the point of minimum $\mathbf{x}^*$. Given this geometry near the minimum, it then makes sense to do steepest descent in the norm induced by the hessian, near the point of minimum (which is equivalent to doing a steepest descent after a rotation of the coordinate system using the hessian). This is exactly the Newton's step. Thus, the Newton's method[24] converges very fast in the vicinity of the solution.

This convergence analysis is formally stated in the following theorem and is due to Leonid Kantorovich.

Theorem 84 *Suppose $f(\mathbf{x}) : \mathcal{D} \to \Re$ is twice continuously differentiable on $\mathcal{D}$ and $\mathbf{x}^*$ is the point corresponding to the optimal value p^* (so that $\nabla f(\mathbf{x}^*) = 0$). Let f be strongly convex on $\mathcal{D}$ with constant $c > 0$. Also, suppose $\nabla^2 f(\mathbf{x}^*)$ is Lipschitz continuous on $\mathcal{D}$ with a constant $L > 0$ (which measures how well f can be approximated by a quadratic function or how fast the second derivative of f changes), that is*

$$||\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})||_2 \leq L||\mathbf{x} - \mathbf{y}||_2$$

Then, there exist constants $\alpha \in (0, \frac{c^2}{L})$ and $\beta > 0$ such that

1. **Damped Newton Phase:** *If $||\nabla^2 f(\mathbf{x})||_2 \geq \alpha$, then $f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \leq -\beta$. That is, at every step of the iteration in the damped Newton phase, the function value decreases by atleast β and the phase ends after at most $\frac{f(\mathbf{x}^{(0)}) - p^*}{\beta}$ iterations, which is a finite number.*

2. **Quadratically Convergent Phase:** *If $||\nabla^2 f(\mathbf{x})||_2 < \alpha$, then $\frac{L}{2c^2}||\nabla f(\mathbf{x}^{(k+1)})||_2 \leq \left(\frac{L}{2c^2}||\nabla f(\mathbf{x}^{(k)})||_2\right)^2$. When applied recursively this inequality yields*

$$\frac{L}{2c^2}||\nabla f(\mathbf{x}^{(k)})||_2 \leq \left(\frac{1}{2}\right)^{2^{k-q}}$$

where q is iteration number, starting at which $||\nabla^2 f(\mathbf{x}^{(q)})||_2 < \alpha$. Using the result for strong convexity in equation (4.50) on page 273, we can derive

[24] Newton originally presented his method for one-dimensional problems. Later on Raphson extended the method to multi-dimensional problems.

$$f(\mathbf{x}^{(k)}) - p^* \leq \frac{1}{2c}||\nabla f(\mathbf{x}^{(k)})||_2^2 \leq \frac{2c^3}{L^2}\left(\frac{1}{2}\right)^{2^{k-q+1}} \tag{4.94}$$

Also, using the result in equation (4.52) on page 273, we get a bound on the distance between the current iterate and the point $\mathbf{x}^*$ *corresponding to the optimum.*

$$||\mathbf{x}^{(k)} - \widehat{\mathbf{x}}^*||_2 \leq \frac{2}{c}||\nabla f(\mathbf{x}^{(k)})||_2 \leq \frac{c}{L}\left(\frac{1}{2}\right)^{2^{k-q}} \tag{4.95}$$

Inequality (4.94) shows that convergence is quadratic once the second condition is satisfied after a finite number of iterations. Roughly speaking, this means that, after a sufficiently large number of iterations, the number of correct digits doubles at each iteration[25]. In practice, once in the quadratic phase, you do not even need to bother about any convergence criterion; it suffices to apply a fixed few number of Newton iterations to get a very accurate solution. Inequality (4.95) states that the sequence of iterates converges quadratically. The Lipschitz continuity condition states that if the second derviative of the function changes relatively slowly, applying Newton's method can be useful. Again, the inequalities are technical junk as far as practical application of Newton's method is concerned, since L, c and α are generally unknown, but it helps to understand the properties of the Newton's method, such as its two phases and identify them in problems. In practice, Newton's method converges very rapidly, if at all.

As an example, consider a one dimensional function $f(x) = 7x - \ln x$. Then $f'(x) = 7 - \frac{1}{x}$ and $f''(x) = \frac{1}{x^2}$. The Newton update rule at a point x is $x^{new} = x - x^2\left(7 - \frac{1}{x}\right)$. Starting with $x^{(0)} = 0$ is really infeasible and useless, since the updates will always be 0. The unique global minimizer of this function is $x^* = \frac{1}{7}$. The range of quadratic convergence for Newton's method on this function is $x \in \left(0, \frac{2}{7}\right)$. However, if you start with an initial infeasible point $x^{(0)} = 0$, the function will quadratically tend to $-\infty$!

There are some classes of functions for which theorem 84 can be applied very constructively. They are

- $-\sum_{i=1}^{m} \ln x_i$
- $-\ln t^2 - \mathbf{x}^T\mathbf{x}$ for $t > 0$
- $-\ln det(X)$

Further, theorem 84 also comes handy for linear combinations of these functions. These three functions are also at the heart of modern interior points method theory.

[25] Linear convergence adds a constant number of digits of accuracy at each iteration.

4.5.3 Variants of Newton's Method

One important aspect of the algorithm in Figure 4.49 is the step (1), which involves solving a linear system $\nabla^2 f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} = \nabla f(\mathbf{x}^{(k)})$. The system can be easy to solve if the Hessian is a 100×100 sparse matrix, but it can get hairy if it is a larger and denser matrix. Thus it can be unfair to claim that the Newton's method is faster than the gradient descent method on the grounds that it takes a fewer number of iterations to converge as compared to the gradient descent, since each iteration of the Newton's method involves inverting the hessian to solve a linear system, which can take time[26] $O(n^3)$ for dense systems. Further, the method assumes that the hessian is positive definite and therefore invertible, which might not always be so. Finally, the Newton's method might make huge-uncontrolled steps, especially when the hessian is positive semi-definite (for example, if the function is flat along the direction corresponding to a 0 or nearly 0 eigenvalue). Due to these disadvantages, most optimization packages do not use Newton's method.

There is a whole suite of methods called *Quasi-Newton methods* that use approximations of the hessian at each iteration in an attempt to either do less work per iteration or to handle singular hessian matrices. These methods fall in between gradient methods and Newton's method and were introduced in the 1960's. Work on quasi-Newton methods sprang from the belief that often, in a large linear system, most variables should not depend on most other variables (that is, the system is generally sparse).

We should however note that in some signal and image processing problems, the hessian has a nice structure, which allows one to solve the linear system $\nabla^2 f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} = \nabla f(\mathbf{x}^{(k)})$ in time much less than $O(n^3)$ (often in time comparble to that required for quasi Newton methods), without having to explicitly store the entire hessian. We next discuss some optimization techniques that use specific approximations to the hessian $\nabla^2 f(\mathbf{x})$ for specific classes of problems, by reducing the time required for computing the second derivatives.

4.5.4 Gauss Newton Approximation

The Gauss Newton method decomposes the objective function (typically for a regression problem) as a composition of two functions[27] $f = l \circ \mathbf{m}$; (i) the vector valued model or regression function $\mathbf{m} : \Re^n \rightarrow \Re^p$ and (ii) the scalar-valued loss (such as the sum squared difference between predicted outputs and target outputs) function l. For example, if m_i is $y_i - r(\mathbf{t}_i, \mathbf{x})$, for parameter vector $\mathbf{x} \in \Re^n$ and input instances $(y_i, \mathbf{t}_i)$ for $i = 1, 2, \ldots, p$, the function f can be written as

$$f(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{p} (y_i - r(\mathbf{t}_i, \mathbf{x}))^2$$

[26] $O(n^{2.7})$ to be precise.
[27] Here, n is the number of weights.

An example of the function r is the linear regression function $r(\mathbf{t}_i, \mathbf{x}) = \mathbf{x}^T\mathbf{t}_i$. Logistic regression poses an example objective function, which involves a cross-entropy loss.

$$f(\mathbf{x}) = -\sum_{i=1}^{p} \left(y_i \log\left(\sigma(\mathbf{x}^T\mathbf{t}_i)\right) + (1-y_i)\log\left(\sigma(-\mathbf{x}^T\mathbf{t}_i)\right)\right)$$

where $\sigma(k) = \frac{1}{1+e^{-k}}$ is the logistic function.

The task of the loss function is typically to make the optimization work well and this gives freedom in choosing l. Many different objective functions share a common loss function. While the sum-squared loss function is used in many regression settings, cross-entropy loss is used in many classification problems. These loss functions arise from the problem of maximizing log-likelihoods in some reasonable way.

The Hessian $\nabla^2 f(\mathbf{x})$ can be expressed using a matrix version of the chain rule, as

$$\nabla^2 f(\mathbf{x}) = \underbrace{J_{\mathbf{m}}(\mathbf{x})^T \nabla^2 l(\mathbf{m}) J_{\mathbf{m}}(\mathbf{x})}_{G_f(\mathbf{x})} + \sum_{i=1}^{p} \nabla^2 m_i(\mathbf{x})(\nabla l(\mathbf{m}))_i$$

where $J_{\mathbf{m}}$ is the jacobian[28] of the vector valued function $\mathbf{m}$. It can be shown that if $\nabla^2 l(\mathbf{m}) \succeq 0$, then $G_f(\mathbf{x}) \succeq 0$. The term $G_f(\mathbf{x})$ is called the Gauss-Newton approximation of the Hessian $\nabla^2 f(\mathbf{x})$. In many situtations, $G_f(\mathbf{x})$ is the dominant part of $\nabla^2 f(\mathbf{x})$ and the approximation is therefore reasonable. For example, at the point of minimum (which will be the critical point for a convex function), $\nabla^2 f(\mathbf{x}) = G_f(\mathbf{x})$. Using the Gauss-Newton approximation to the hessian $\nabla^2 f(\mathbf{x})$, the Newton update rule can be expressed as

$$\Delta\mathbf{x} = -(G_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x}) = -(G_f(\mathbf{x}))^{-1} J_{\mathbf{m}}^T(\mathbf{x}) \nabla l(\mathbf{m})$$

where we use the fact that $(\nabla f(\mathbf{x}))_i = \sum_{k=1}^{p} \frac{\partial l}{\partial m_k}\frac{\partial m_k}{\partial x_i}$, since the gradient of a composite function is a product of the jacobians.

For the cross entropy classification loss or the sum-squared regression loss l, the hessian is known to be positive semi-definite. For example, if the loss function is the sum of squared loss, the objective function is $f = \frac{1}{2}\sum_{i=1}^{p} m_i(\mathbf{x})^2$ and $\nabla^2 l(\mathbf{m}) = I$. The Newton update rule can be expressed as

$$\Delta\mathbf{x} = -(J_{\mathbf{m}}(\mathbf{x})^T J_{\mathbf{m}}(\mathbf{x}))^{-1} J_{\mathbf{m}}(\mathbf{x})^T \mathbf{m}(\mathbf{x})$$

Recall that $(J_{\mathbf{m}}(\mathbf{x})^T J_{\mathbf{m}}(\mathbf{x}))^{-1} J_{\mathbf{m}}(\mathbf{x})^T$ is the Moore-Penrose pseudoinverse $J_{\mathbf{m}}(\mathbf{x})^+$ of $J_{\mathbf{m}}(\mathbf{x})$. The Gauss-Jordan method for the sum-squared loss can be interpreted as multiplying the gradient $\nabla l(\mathbf{m})$ by the pseudo-inverse of the jacobian of $\mathbf{m}$

[28] The Jacobian is a $p \times n$ matrix of the first derivatives of a vector valued function, where p is arity of $\mathbf{m}$. The $(i,j)^{th}$ entry of the Jacobian is the derivative of the i^{th} output with respect to the j^{th} variable, that is $\frac{\partial m_i}{\partial x_j}$. For $m = 1$, the Jacobian is the gradient vector.

instead of its transpose (which is what the gradient descent method would do). Though the Gauss-Newton method has been traditionally used for non-linear least squared problems, recently it has also seen use for the cross entropy loss function. This method is a simple adoption of the Newton's method, with the advantage that second derivatives, which can be computationally expensive and challenging to compute, are not required.

4.5.5 Levenberg-Marquardt

Like the Gauss-Newton method, the Levenberg-Marquardt method has its main application in the least squares curve fitting problem (as also in the minimum cross-entropy problem). The Levenberg-Marquardt method interpolates between the Gauss-Newton algorithm and the method of gradient descent. The Levenberg-Marquardt algorithm is more robust than the Gauss Newton algorithm - it often finds a solution even if it starts very far off the final minimum. On the other hand, for well-behaved functions and reasonable starting parameters, this algorithm tends to be a bit slower than the Gauss Newton algorithm. The Levenberg-Marquardt method aims to reduce the uncontrolled step size often taken by the Newton's method and thus fix the stability issue of the Newton's method. The update rule is given by

$$\Delta\mathbf{x} = -\left(G_f(\mathbf{x}) + \lambda \operatorname{diag}(G_f)\right)^{-1} J_{\mathbf{m}}^T(\mathbf{x}) \nabla l(\mathbf{m})$$

where G_f is the Gauss-Newton approximation to $\nabla^2 f(\mathbf{x})$ and is assumed to be positive semi-definite. This method is one of the work-horses of modern optimization. The parameter $\lambda \geq 0$ adaptively controlled, limits steps to an elliptical model-trust region[29]. This is achieved by adding λ to the smallest eigenvalues of G_f, thus restricting all eigenvalues of the matrix to be above λ so that the elliptical region has diagonals of shorter length that inversely vary as the eigenvalues (*c.f.* page 3.11.3). While this method fixes the stability issues in Newtons method, it still requires the $O(n^3)$ time required for matrix inversion.

4.5.6 BFGS

The Broyden-Fletcher-Goldfarb-Shanno[30] (BFGS) method uses linear algebra to iteratively update an estimate $B^{(k)}$ of $\left(\nabla^2 f(\mathbf{x}^{(k)})\right)^{-1}$ (the inverse of the curvature matrix), while ensuring that the approximation to the hessian inverse is symmetric and positive definite. Let $\Delta\mathbf{x}^{(k)}$ be the direction vector for the k^{th} step obtained as the solution to

$$\Delta\mathbf{x}^{(k)} = -B^{(k)} \nabla f(\mathbf{x}^{(k)})$$

The next point $\mathbf{x}^{(k+1)}$ is obtained as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)} \Delta\mathbf{x}^{(k)}$$

[29] Essentially the algorithm approximates only a certain region (the so-called trust region) of the objective function with a quadratic as opposed to the entire function.

[30] The the 4 authors wrote papers for exactly the same method at exactly at the same time.

where $t^{(k)}$ is the step size obtained by line search. Let $\Delta\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})$. Then the BFGS update rule is derived by imposing the following logical conditions:

1. $\Delta\mathbf{x}^{(k)} = -B^{(k)}\nabla f(\mathbf{x}^{(k)})$ with $B^{(k)} \succ 0$. That is, $\Delta\mathbf{x}^{(k)}$ is the minimizer of the convex quadratic model
$$Q^{(k)}(\mathbf{p}) = f(\mathbf{x}^{(k)}) + \nabla^T f(\mathbf{x}^{(k)})\mathbf{p} + \frac{1}{2}\mathbf{p}^T \left(B^{(k)}\right)^{-1}\mathbf{p}$$
2. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$, where $t^{(k)}$ is obtained by line search.
3. The gradient of the function $Q^{(k+1)} = f(\mathbf{x}^{(k+1)}) + \nabla^T f(\mathbf{x}^{(k+1)})\mathbf{p} + \frac{1}{2}\mathbf{p}^T \left(B^{(k+1)}\right)^{-1}\mathbf{p}$ at $\mathbf{p} = \mathbf{0}$ and $\mathbf{p} = -t^{(k)}\Delta\mathbf{x}^{(k)}$ agrees with gradient of f at $\mathbf{x}^{(k+1)}$ and $\mathbf{x}^{(k)}$ respectively. While the former condition is naturally satisfied, the latter need to be imposed. This quasi-Newton condition yields
$$\left(B^{(k+1)}\right)^{-1}\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right) = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}).$$
This equation is called the *secant equation*.
4. Finally, among all symmetric matrices satisfying the secant equation, $B^{(k+1)}$ is closest to the current matrix $B^{(k)}$ in some norm. Different matrix norms give rise to different quasi-Newton methods. In particular, when the norm chosen is the Frobenius norm, we get the following BGFS update rule
$$B^{(k+1)} = B^{(k)} + R^{(k)} + S^{(k)}$$
where,
$$R^{(k)} = \frac{\Delta\mathbf{x}^{(k)}\left(\Delta\mathbf{x}^{(k)}\right)^T}{\left(\Delta\mathbf{x}^{(k)}\right)^T\Delta\mathbf{g}^{(k)}} - \frac{B^{(k)}\Delta\mathbf{g}^{(k)}\left(\Delta\mathbf{g}^{(k)}\right)^T\left(B^{(k)}\right)^T}{\left(\Delta\mathbf{g}^{(k)}\right)^T B^{(k)}\Delta\mathbf{g}^{(k)}}$$
and
$$S^{(k)} = \mathbf{u}\left(\Delta\mathbf{x}^{(k)}\right)^T B^{(k)}\Delta\mathbf{x}^{(k)}\mathbf{u}^T$$
with
$$\mathbf{u} = \frac{\Delta\mathbf{x}^{(k)}}{\left(\Delta\mathbf{x}^{(k)}\right)^T\Delta\mathbf{g}^{(k)}} - \frac{B^{(k)}\Delta\mathbf{g}^{(k)}}{\left(\Delta\mathbf{g}^{(k)}\right)^T B^{(k)}\Delta\mathbf{g}^{(k)}}$$
We have made use of the Sherman Morrison formula that determines how updates to a matrix relate to the updates to the inverse of the matrix.

The approximation to the Hessian is updated by analyzing successive gradient vectors and thus the Hessian matrix does not need to be computed at any stage. The initial estimate $B^{(0)}$ can be taken to be the identity matrix, so that the first step is equivalent to a gradient descent. The BFGS method has a reduced complexity of $O(n^2)$ time per iteration. The method is summarized

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$ and an approximate $B^{(0)}$ (which could be I).
Select an appropriate tolerance $\epsilon > 0$.
repeat
1. Set $\Delta\mathbf{x}^{(k)} = -B^{(k)}\nabla f(\mathbf{x}^{(k)})$.
2. Let $\lambda^2 = \nabla^T f(\mathbf{x}^{(k)})B^{(k)}\nabla f(\mathbf{x}^{(k)})$.
3. If $\frac{\lambda^2}{2} \leq \epsilon$, **quit**.
4. Set step size $t^{(k)} = 1$.
5. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
6. Compute $\Delta\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})$.
7. Compute $R^{(k)}$ and $S^{(k)}$.
8. Compute $B^{(k+1)} = B^{(k)} + R^{(k)} + S^{(k)}$.
6. Set $k = k + 1$.

until

Figure 4.50: The BFGS method.

in Figure 4.50 The BFGS [**?**] method approaches the Newton's method in behaviour as the iterate approaches the solution. They are much faster than the Newton's method in practice. It has been proved that when BFGS is applied to a convex quadratic function with exact line search, it finds the minimizer within n steps. There is a variety of methods related to BFGS and collectively they are known as Quasi-Newton methods. They are preferred over the Newton's method or the Levenberg-Marquardt when it comes to speed. There is a variant of BFGS, called LBFGS [**?**], which stands for "Limited memory BFGS method". LBFGS employs a limited-memory quasi-Newton approximation that does not require much storage or computation. It limites the rank of the inverse of the hessian to some number $\gamma \in \Re$ so that only $n\gamma$ numbers have to be stored instead of n^2 numbers. For general non-convex problems, LBFGS may fail when the initial geometry (in the form of $B^{(0)}$) has been placed very close to a saddle point. Also, LBFGS is very sensitive to line search.

Recently, L-BFGS has been observed [**?**] to be the most effective parameter estimation method for Maximum Entropy model, much better than improved iterative scaling [**?**] (IIS) and generalized iterative scaling [**?**] (GIS).

4.5.7 Solving Large Sparse Systems

In many convex optimization problems such as least squares, newton's method for optimization, *etc.*, one has to deal with solving linear systems involving large and sparse matrices. Elimination with ordering can be expensive in such cases. A lot of work has gone into solving such problems efficiently[31] using iterative

[31] Packages such as LINPack (which is now renamed to LAPACK), EiSPACK, MINPACK, *etc.*, which can be found under the netlib respository, have focused on efficiently solving large linear systems under general conditions as well as specific conditions such as symmetry or positive definiteness of the coefficient matrix.

methods instead of direct elimination methods. An example iterative method is for solving a system $A\mathbf{x} = \mathbf{b}$ by repeated multiplication of a large and sparse matrix A by vectors to quickly get an answer $\widehat{\mathbf{x}}$ that is sufficiently close to the optimal solution $\mathbf{x}^*$. Multiplication of an $n \times n$ sparse matrix A having k non-zero entries with a vector of dimension n takes $O(kn)$ time only, in contrast to $O(n^3)$ time for Gauss elimination. We will study three types of methods for solving systems with large and sparse matrices:

1. *Iterative Methods.*
2. *Multigrid Methods.*
3. *Krylov Methods.*

The most famous and successful amongst the Krylov methods has been the *conjugate gradient method*, which works for problems with positive definite matrices.

Iterative Methods

The central step in an iteration is

$$P\mathbf{x}_{k+1} = (P - A)\mathbf{x}_k + \mathbf{b}$$

where $\mathbf{x}_k$ is the estimate of the solution at the k^{th} step, for $k = 0, 1, \ldots$. If the iterations converge to the solution, that is, if $\mathbf{x}_{k+1} = \mathbf{x}_k$ one can immediatly see that the solution is reached. The choice of matrix P, which is called the *preconditioner*, determines the rate of convergence of the solution sequence to the actual solution. The initial estimate $\mathbf{x}_0$ can be arbitrary for linear systems, but for non-linear systems, it is important to start with a good approximation. It is desirable to choose the matrix P reasonably close to A, though setting $P = A$ (which is referred to as perfect preconditioning) will entail solving the large system $A\mathbf{x} = \mathbf{b}$, which is undesirable as per our problem definition. If $\mathbf{x}^*$ is the actual solution, the relationship between the errors $\mathbf{e}_k$ and $\mathbf{e}_{k+1}$ at the k^{th} and $(k+1)^{th}$ steps respectively can be expressed as

$$P\mathbf{e}_{k+1} = (P - A)\mathbf{e}_k$$

where $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^*$. This is called the *error equation*. Thus,

$$\mathbf{e}_{k+1} = (I - P^{-1}A)\mathbf{e}_k = M\mathbf{e}_k$$

Whether the solutions are convergent or not is controlled by the matrix M. The iterations are stationary (that is, the update is of the same form at every step). On the other hand, Multigrid and Krylov methods adapt themselves across iterations to enable faster convergence. The error after k steps is given by

$$\mathbf{e}_k = M^k\mathbf{e}_0 \tag{4.96}$$

Using the idea of eigenvector decomposition presented in (3.101), it can be proved that the error vector $\mathbf{e}_k \rightarrow \mathbf{0}$ if the absolute values of all the eigenvalues of M are less than 1. This is the *fundamental theorem of iteration*. In this case, the rate of convergence of $\mathbf{e}_k$ to $\mathbf{0}$ is determined by the maximum absolute eigenvalue of M, called the spectral radius of M and denoted by $\rho(M)$.

Any iterative method should attempt to choose P so that it is easy to compute $\mathbf{x}_{k+1}$ and at the same time, the matrix $M = I - P^{-1}A$ has small eigenvalues. Corresponding to various choices of the preconditioner P, there exist different iterative methods.

1. *Jacobi:* In the simplest setting, P can be chosen to be a diagonal matrix with its diagonal borrowed from A. This choice of A is correponds to the *Jacobi* method. The value of $\rho(M)$ is less than 1 for the Jacobi method, though it is often very close to 1. Thus, the Jacobi method does converge, but the convergence can be very slow in practice. While the residual $\widehat{\mathbf{r}} = A\widehat{\mathbf{x}} - \mathbf{b}$ converges rapidly, the error $\overline{\mathbf{x}} = \widehat{\mathbf{x}} - \mathbf{x}^*$ decreases rapidly in the beginning, but the rate of decrease of $\overline{\mathbf{x}}$ reduces as iterations proceed. This happens because $\overline{\mathbf{x}} = A^{-1}\widehat{\mathbf{r}}$ and A^{-1} happens to have large condition number for sparse matrices. In fact, it can be shown that Jacobi can take upto n^β iterations to reduce the error $\overline{\mathbf{x}}$ by a factor β.

   We will take an example to illustrate the Jacobi method. Consider the following $n \times n$ tridiagonal matrix A.

$$A = \begin{bmatrix} 2 & -1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ . & . & . & \ldots & . & . & . & \ldots & . \\ . & . & . & \ldots & . & . & . & \ldots & . \\ 0 & 0 & 0 & \ldots & -1 & 2 & -1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 & -1 & 2 & \ldots & 0 \\ . & . & . & \ldots & . & . & . & \ldots & . \\ . & . & . & \ldots & . & . & . & \ldots & . \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 2 \end{bmatrix} \tag{4.97}$$

   The absolute value of the i^{th} eigenvalue of M is $\cos\frac{j\pi}{n+1}$ and its spectral radius is $\rho(M) = \cos\frac{\pi}{n+1}$. For extremely large n, the spectral radius is approximately $1 - \frac{1}{2}\left(\frac{\pi}{n+1}\right)^2$, which is very close to 1. Thus, the Jacobi steps converge very slowly.

2. *Gauss-Seidel:* The second possibility is to choose P to be the lower-triangular part of A. The method for this choice is called the *Gauss-Siedel* method. For the example tridiagonal matrix A in (4.97), matrix

$P - A$ will be the strict but negated upper-triangular part of A. For the Gauss-Seidel technique, the components of $\mathbf{x}_{k+1}$ can be determined from $\mathbf{x}_k$ using back-substitution. The Gauss-sidel method provides only a constant factor improvement over the *Jacobi* method.

3. *Successive over-relaxation:* In this method, the preconditioner is obtained as a weighted composition of the preconditioners from the above two methods. It is abbreviated as SOR. In history, this was the first step of progress beyond Jacobi and Gauss-Seidel.

4. *Incomplete LU:* This method involves an incomplete elimination on the sparse matrix A. For a sparse matrix A, many entries in its LU decomposition will comprise of nearly 0 elements; the idea behind this method is to treat such entries as 0's. Thus, the L and U matrices are approximated based on the tolerance threshold; if the tolerance threshold is very high, the factors are exact. Else they are approximate.

Multigrid Methods

Multigrid methods come very handy in solving large sparse systems, especially differential equations using a hierarchy of discretizations. This approach often scales linearly with the number of unknowns n for a pre-specified accuracy threshold. The overall multi-grid algorithm for solving $A_h\mathbf{u}_h = \mathbf{b}_h$ with residual given by $\mathbf{r}_h = \mathbf{b} - A\mathbf{u}_h$ is

1. **Smoothing:** Perform a few (say 2-3) iterations on $A_h\mathbf{u} = \mathbf{b}_h$ using either Jacobi or Gauss-sidel. This will help remove high frequency components of the residual $\mathbf{r} = \mathbf{b} - A_h\mathbf{u}$. This step is really outside the core of the multigrid method. Denote the solution obtained by $\mathbf{u}_h$. Let $\mathbf{r}_h = \mathbf{b} - A_h\mathbf{u}_h$.

2. **Restriction:** Restrict $\mathbf{r}_h$ to coarse grid by setting $\mathbf{r}_{2h} = R\mathbf{r}_h$. That is, $\mathbf{r}_h$ is downsampled to yield $\mathbf{r}_{2h}$ Let $k < n$ characterize the coarse grid. Then, the $k \times n$ matrix R is called the restriction matrix and it takes the residuals from a finer to a coarser grid. It is typically scaled to ensure that a vector of 1's on the fine mesh gets transformed to a vector of 1's on a coarse mesh. Calculations on the coarse grid are way faster than on the finer grid.

3. Solve $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}$ with $A_{2h} = RA_hN$, which is a natural construction for the coarse mesh operation. This could be done by running few iterations of Jacobi, starting with $\mathbf{e}_{2h} = \mathbf{0}$.

4. **Interpolation/Prolongation:** This step involves interpolating the correction computed on a coarses grid to a finer grid. Interpolate back to $\mathbf{e}_h = N\mathbf{e}_{2h}$. Here N is a $k \times n$ interpolation matrix and it takes the residuals from a coarse to a fine grid. It is generally a good idea to connect N to R by setting $N = \alpha R^T$ for some scaling factor α. Add $\mathbf{e}_h$ to $\mathbf{u}_h$. The

analytical expression for $\mathbf{e}_h$ is

$$\mathbf{e}_h = N(A_{2h})^{-1}RA_h(\mathbf{u}-\mathbf{u}_h) = \underbrace{\big(N(RAN)^{-1}RA_h(\mathbf{u}-\mathbf{u}_h)\big)}_{S}(\mathbf{u}-\mathbf{u}_h)$$

A property of the $n\times n$ matrix S is that $S^2 = S$. Thus, the only eigenvalues of S are 0 and 1. Since S is of rank $k < n$, k of its eigenvalues are 1 and $n-k$ are 0. Further, the eigenvectors for the 1 eigenvalues, which are in the null space of $I-S$ form the coarse mesh (and correspond to low frequency vectors) whereas the eigenvectors for the 0 eigenvalues, which are in the null space of S form the fine mesh (and correspond to high frequency vectors). We can easily derive that k eigenvalues of $I-S$ will be 0 and $n-k$ of them will be 1.

5. Finally as a post-smoothing step, iterate $A\mathbf{u}_h = \mathbf{b}_h$ starting from the improved $\mathbf{u}_h + \mathbf{e}_h$, using Jacobi or Gauss-Sidel.

Overall, the error $\mathbf{e}^k$ after k steps will be of the form

$$\mathbf{e}_k = (M^t(I-S)M^t)\mathbf{e}_0 \tag{4.98}$$

where t is the number of Jacobi steps performed in (1) and (5). Typically t is 2 or 3. When you contrast (4.98) against (4.96), we discover that $\rho(M) \geq\geq \rho(M^t(I-S)M^t)$. As t increases, $\rho(M^t(I-S)M^t)$ further decreases by a smaller proportion.

In general, you could have multiple levels of coarse grids corresponding to $2h$, $4h$, $8h$ and so on, in which case, steps (2), (3) and (4) would be repeated as many times with varying specifications of the coarseness. If A is an $n\times n$ matrix, multi-grid methods are known to run in $O(n^2)$ floating point operations (flops). The multi-grid method could be used an iterative method to solve a linear system. Alternatively, it could be used to obtain the preconditioner.

Linear Conjugate Gradient Method

The conjugate gradient method is one of the most popular Krylov methods. The Krylov matrix K_j, for the linear system $A\mathbf{u} = \mathbf{b}$ is given by

$$K_j = \begin{bmatrix}\mathbf{b} & A\mathbf{b} & A^2\mathbf{b} & \dots & A^{j-1}\mathbf{b}\end{bmatrix}$$

The columns of K_j are easy to compute; each column is a result of a matrix multiplication A with the previous column. Assuming we are working with sparse matrices, (often symmetric matrices such as the Hessian) these computations will be inexpensive. The Krylov space $\mathcal{K}_j$ is the column space of K_j. The columns of K_j are computed during the first j steps of an iterative method such as Jacobi. Most Krylov methods opt to choose vectors from $\mathcal{K}_j$ instead of a fixed choice of the j^{th} column of K_j. A method such as *MinRes* chooses a vector

$\mathbf{u}_j \in \mathcal{K}_j$ that minimizes $\mathbf{b} - A\mathbf{u}_j$. One of the well-known Krylov methods is the *Conjugate gradient* method, which assumes that the matrix A is symmetric and positive definite and is faster than MinRes. In this method, the choice of u_j is made so that $\mathbf{b} - A\mathbf{u}_j \perp \mathcal{K}_j$. That is, the choice of $\mathbf{u}_j$ is made so that the residual $\mathbf{r}_j = \mathbf{b} - A\mathbf{u}_j$ is orthogonal to the space $\mathcal{K}_j$. The conjugate gradient method gives an exact solution to the linear system if $j = n$ and that is how they were originally designed to be (and put aside subsequently). But later, they were found to give very good approximations for $j << n$.

The discussions that follow require the computation of a basis for $\mathcal{K}_j$. It is always prefered to have a basis matrix with low condition number[32], and an orthonormal basis is a good choice, since it has a condition number of 1 (the basis consisting of the columns of K_j turns out to be not-so-good in practice). The *Arnoldi* method yields an orthonormal Krylov basis $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j$ to get something that is numerically reasonable to work on. The method is summarized in Figure 4.51. Though the underlying idea is borrowed from Gram-Schmidt at every step, there is a difference; the vector $\mathbf{t}$ is $\mathbf{t} = A\mathbf{Q}_j$ as against simply $\mathbf{t} = \mathbf{q}_j$. Will it be expensive to compute each $\mathbf{t}$? Not if A is symmetric. First we note that by construction, $AQ = QH$, where $\mathbf{q}_j$ is the j^{th} column of Q. Thus, $H = Q^T AQ$. If A is symmetric, then so is H. Further, since H has only one lower diagonal (by construction), it must have only one higher diagonal. Therefore, H must be symmetric and tridiagonal. If A is symmetric, it suffices to subtract only the components of $\mathbf{t}$ in the direction of the last two vectors $\mathbf{q}_{j-1}$ and $\mathbf{q}_j$ from $\mathbf{t}$. Thus, for a symmetric A, the inner 'for' loop needs to iterate only over $i = j - 1$ and $i = j$.

Since A and H are similar matrices, they have exactly the same eigenvalues. Restricting the computation to a smaller number of orthonormal vectors (for some $k << n$), we can save time for computing Q_k and H_k. The k eigenvalues of H_k are good approximations to the first k eigenvalues of H. This is called the *Arnoldi-Lanczos* method for finding the top k eigenvalues of a matrix.

As an example, consider the following matrix A.

$$A = \begin{bmatrix} 0.5344 & 1.0138 & 1.0806 & 1.8325 \\ 1.0138 & 1.4224 & 0.9595 & 0.8234 \\ 1.0806 & 0.9595 & 1.0412 & 1.0240 \\ 1.8325 & 0.8234 & 1.0240 & 0.7622 \end{bmatrix}$$

[32] For any matrix A, the condition number $\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$, where $\sigma_{max}(A)$ and $\sigma_{min}(A)$ are maximal and minimal singular values of A respectively. Recall from Section 3.13 that the i^{th} eigenvalue of $A^T A$ (the gram matrix) is the square of the i^{th} singular value of A. Further, if A is normal, $\kappa(A) = \left|\frac{\lambda_{max}(A)}{\lambda_{min}(A)}\right|$, where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are eigevalues of A with maximal and minimal magnitudes respectively. All orthogonal, symmetric, and skew-symmetric matrices are normal. The condition number measures how much the columns/rows of a matrix are dependent on each other; higher the value of the condition number, more is the linear dependence. Condition number 1 means that the columns/rows of a matrix are linearly independent.

Set $\mathbf{q}_1 = \frac{1}{||\mathbf{b}||}\mathbf{b}$. //The first step in Gram schmidt.
for $j = 1$ to $n-1$ **do**
  $\mathbf{t} = A\mathbf{q}_j$.
  for $i = 1$ to j **do**
    //If A is symmetric, it will be $i = \max(1, j-1)$ to j.
    $H_{i,j} = \mathbf{q}_i^T\mathbf{t}$.
    $\mathbf{t} = \mathbf{t} - H_{i,j}\mathbf{q}_i$.
  end for
  $H_{j+1,j} = ||\mathbf{t}||$.
  $\mathbf{q}_{j+1} = \frac{1}{||\mathbf{t}||}\mathbf{t}$.
end for
$\mathbf{t} = A\mathbf{q}_n$.
for $i = 1$ to n **do**
  //If A is symmetric, it will be $i = n-1$ to n.
  $H_{i,n} = \mathbf{q}_i^T\mathbf{t}$.
  $\mathbf{t} = \mathbf{t} - H_{in}\mathbf{q}_i$.
end for
$H_{j+1,j} = ||\mathbf{t}||$.
$\mathbf{q}_{j+1} = \frac{1}{||\mathbf{t}||}\mathbf{t}$.

Figure 4.51: The Arnoldi algorithm for computing orthonormal basis.

and the vector $\mathbf{b}$

$$\mathbf{b} = \begin{bmatrix} 0.6382 & 0.3656 & 0.1124 & 0.5317 \end{bmatrix}^T$$

The matrix K_4 is

$$K_4 = \begin{bmatrix} 0.6382 & 1.8074 & 8.1892 & 34.6516 \\ 0.3656 & 1.7126 & 7.5403 & 32.7065 \\ 0.1124 & 1.7019 & 7.4070 & 31.9708 \\ 0.5317 & 1.9908 & 7.9822 & 34.8840 \end{bmatrix}$$

Its condition number is 1080.4.

The algorithm in Figure 4.51 computed the following basis for the matrix K_4.

$$Q_4 = \begin{bmatrix} 0.6979 & -0.3493 & 0.5101 & -0.3616 \\ 0.3998 & 0.2688 & 0.2354 & 0.8441 \\ 0.1229 & 0.8965 & 0.1687 & -0.3908 \\ 0.5814 & 0.0449 & -0.8099 & -0.0638 \end{bmatrix}$$

The coefficient matrix H_4 is

$$H_4 = \begin{bmatrix} 3.6226 & 1.5793 & 0 & 0 \\ 1.5793 & 0.6466 & 0.5108 & 0 \\ 0 & 0.5108 & -0.8548 & 0.4869 \\ 0 & 0 & 0.4869 & 0.3459 \end{bmatrix}$$

and its eigenvalues are 4.3125, 0.5677, -1.2035 and 0.0835. On the other hand, the following matrix H_3 (obtained by restricting to K_3) has eigenvalues 4.3124, 0.1760 and -1.0741.

The basic conjugate gradient method selects vectors in $\mathbf{x}_k \in \mathcal{K}_k$ that approach the exact solution to $A\mathbf{x} = \mathbf{b}$. Following are the main ideas in the conjugate gradient method.

1. The rule is to select an $\mathbf{x}_k$ so that the new residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ is orthogonal to all the previous residuals. Since $A\mathbf{x}_k \in \mathcal{K}_{k+1}$, we must have $\mathbf{r}_k \in \mathcal{K}_{k+1}$ and $\mathbf{r}_k$ must be orthogonal to all vectors in $\mathcal{K}_k$. Thus, $\mathbf{r}_k$ must be a multiple of $\mathbf{q}_{k+1}$. This holds for all k and implies that

   $$\mathbf{r}_k^T \mathbf{r}_i = 0$$

   for all $i < k$.

2. Consequently, the difference $\mathbf{r}_k - \mathbf{r}_{k-1}$, which is a linear combination of $\mathbf{q}_{k+1}$ and $\mathbf{q}_k$, is orthogonal to each subspace $\mathcal{K}_i$ for $i < k$.

3. Now, $\mathbf{x}_i - \mathbf{x}_{i-1}$ lies in the subspace $\mathcal{K}_i$. Thus, $\Delta\mathbf{r} = \mathbf{r}_k - \mathbf{r}_{k-1}$ is orthogonal to all the previous $\Delta\mathbf{x} = \mathbf{x}_i - \mathbf{x}_{i-1}$. Since $\mathbf{r}_k - \mathbf{r}_{k-1} = -A(\mathbf{x}_k - \mathbf{x}_{k-1})$, we get the following 'conjugate directions' condition for the updates

   $$(\mathbf{x}_i - \mathbf{x}_{i-1})^T A(\mathbf{x}_k - \mathbf{x}_{k-1}) = 0$$

   for all $i < k$. This is a necessary and sufficient condition for the orthogonality of the new residual to all the previous residuals. Note that while the residual updates are orthogonal in the usual inner product, the variable updates are orthogonal in the inner product with respect to A.

The basic conjugate gradient method consists of 5 steps. Each iteration of the algorithm involves a multiplication of vector $\mathbf{d}_{k-1}$ by A and computation of two inner products. In addition, an iteration also involves around three vector updates. So each iteration should take time upto $(2+\theta)n$, where θ is determined by the sparsity of matrix A. The error $\mathbf{e}_k$ after k iterations is bounded as follows.

$$||\mathbf{e}_k||_A = (\mathbf{x}_k - \mathbf{x})^T A(\mathbf{x}_k - \mathbf{x}) \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k ||\mathbf{e}_0||$$

The 'gradient' part of the name *conjugate gradient* stems from the fact that solving the linear system $A\mathbf{x} = \mathbf{b}$ is corresponds to finding the minimum value

$\mathbf{x}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{d}_0 = \mathbf{r}_0$, $k = 1$.
repeat
1. $\alpha_k = \frac{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}{\mathbf{d}_{k-1}^T A \mathbf{d}_{k-1}}$. //Step length for next update. This corresponds to the entry $H_{k,k}$.
2. $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{d}_{k-1}$.
3. $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k A \mathbf{d}_{k-1}$. //New residual obtained using $\mathbf{r}_k - \mathbf{r}_{k-1} = -A(\mathbf{x}_k - \mathbf{x}_{k-1})$.
4. $\beta_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}$. //Improvement over previous step. This corresponds to the entry $H_{k,k+1}$.
5. $\mathbf{d}_k = \mathbf{r}_k + \beta_k \mathbf{d}_{k-1}$. //The next search direction, which should be orthogonal to the search direction just used.

$k = k + 1$.
until $\beta_k < \theta$.

Figure 4.52: The conjugate gradient algorithm for solving $A\mathbf{x} = \mathbf{b}$ or equivalently, for minimizing $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$.

of the convex (for positive definite A) energy function $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x} = \mathbf{r}$ by setting its gradient $A\mathbf{x} - \mathbf{b}$ to the zero vector. The steepest descent method makes a move along at the direction of the residual $\mathbf{r}$ at every step but it does not have a great convergence; we land up doing a lot of work to make a little progress. In contrast, as reflect in the step $\mathbf{d}_k = \mathbf{r}_k + \beta_k \mathbf{d}_{k-1}$, the conjugate gradient method makes a step in the direction of the residual, but only after removing any component β_k along the direction of the step it just took. Figures 4.53 and 4.54 depict the steps taken by the steepest descent and the conjugate descent techniques respectively, on the level-curves of the function $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$, in two dimensions. It can be seen that while the steepest descent technique requires many iterations for convergence, owing to its oscillations, the conjugate gradient method takes steps that are orthogonal with respect to A (or are orthogonal in the transfomed space obtained by multiplying with A), thus taking into account the geometry of the problem and taking a fewer number of steps. If the matrix A is a hessian, the steps taken by conjugate gradient are orthogonal in the local Mahalonobis metric induced by the curvature matrix A. Note that if $\mathbf{x}^{(0)} = \mathbf{0}$, the first step taken by both methods will be the same.

The conjugate gradient method is guaranteed to reach the minimum of the energy function E in exactly n steps. Further, if A has only r distinct eigenvalues, then the conjugate gradient method will terminate at the solution in at most r iterations.

4.5.8 Conjugate Gradient

We have seen that the Conjugate Gradient method in Figure 4.52 can be viewed as a minimization algorithm for the convex quadratic function $E(\mathbf{x}) =$

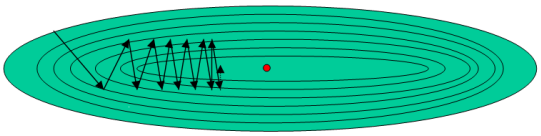

Figure 4.53: Illustration of the steepest descent technique on level curves of the function $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$.

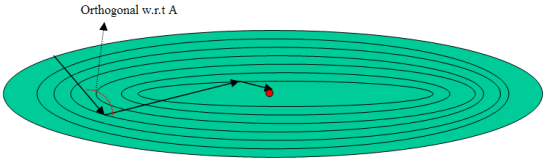

Figure 4.54: Illustration of the conjugate gradient technique on level curves of the function $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$.

$\frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$. Can the approach be adapted to minimize general nonlinear convex functions? Nonlinear variants of the conjugate gradient are well studied [**?**] and have proved to be quite successful in practice. The general conjugate gradient method is essentially an incremental way of doing second order search.

Fletcher and Reeves showed how to extend the conjugate gradient method to nonlinear functions by making two simple changes[33] to the algorithm in Figure 4.52. First, in place of the exact line search formula in step (1) for the step length α_k, we need to perform a line search that identifies an approximate minimum of the nonlinear function f along $\mathbf{d}^{(k-1)}$. Second, the residual $\mathbf{r}^{(k)}$, which is simply the gradient of E (and which points in the direction of decreasing value of E), must be replaced by the gradient of the nonlinear objective f, which serves a similar purpose. These changes give rise to the algorithm for nonlinear optimization outlined in Figure 4.55. The search directions $\mathbf{d}^{(k)}$ are computed by Gram-Schmidt conjugation of the residuals as with linear conjugate gradient. The algorithm is very sensitive to the line minimization step and it generally requires a very good line minimization. Any line search procedure that yields an α_k satisfying the strong Wolfe conditions (see (4.90) and (4.91)) will ensure that all directions $\mathbf{d}^{(k)}$ are descent directions for the function f, otherwise, $\mathbf{d}^{(k)}$ may cease to remian a descent direction as iterations proceed. We note that each iteration of this method costs on $O(n)$, as against the Newton or quasi-newton methods which cost atleast $O(n^2)$ owing to matrix operations. Most often, it yields optimal progress after $h << n$ iterations. Due to this property, the conjugate gradient method drives nearly all large-scale optimization today.

[33] We note that in the algorithm in Figure 4.52, the residuals $\mathbf{r}^{(k)}$ in successive iterations (which are gradients of E) are orthogonal to each other, while the corresponding update directions are orthogonal with respect to A. While the former property is difficult to enforce for general non-linear functions, the latter condition can be enforced.

Select $\mathbf{x}^{(0)}$, Let $f_0 = f(\mathbf{x}^{(0)})$, $\mathbf{g}_0 = \nabla f(\mathbf{x}^{(0)})$, $\mathbf{d}^{(0)} = -\nabla \mathbf{g}_0$, $k = 1$.
repeat
1. Compute α_k by line search.
2. Set $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{d}^{(k-1)}$.
3. Evaluate $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.
4. $\beta_k = \frac{\left(\mathbf{g}^{(k)}\right)^T \mathbf{g}^{(k)}}{\left(\mathbf{g}^{(k-1)}\right)^T \mathbf{g}^{(k-1)}}$.
5. $\mathbf{d}_k = -\mathbf{g}^{(k)} + \beta_k \mathbf{d}^{(k-1)}$.

$k = k + 1$.
until $\frac{||\mathbf{g}^{(k)}||}{||\mathbf{g}^{(0)}||} < \theta$ OR $k > maxIter$.

Figure 4.55: The conjugate gradient algorithm for optimizing nonlinear convex function f.

It revolutionalized optimization ever since it was invented in 1954.

Variants of the Fletcher-Reeves method use different choices of the parameter β_k. An important variant, proposed by Polak and Ribiere, defines β_k as

$$\beta_k^{PR} = \frac{\left(\mathbf{g}^{(k)}\right)^T \left(\mathbf{g}^{(k)} - \mathbf{g}^{(k-1)}\right)}{\left(\mathbf{g}^{(k)}\right)^T \mathbf{g}^{(k)}}$$

The Fletcher-Reeves method converges if the starting point is sufficiently close to the desired minimum. However, convergence of the Polak-Ribiere method can be guaranteed by choosing

$$\beta_k = max\left\{\beta_k^{PR}, 0\right\}$$

Using this value is equivalent to restarting[34] conjugate gradient if $\beta_k^{PR} < 0$. In practice, the Polak-Ribiere method converges much more quickly than the Fletcher-Reeves method. It is generally required to restart the conjugate gradient method after every n iterations, in order to get back conjugacy, *etc.*

If we choose f to be the strongly convex quadratic E and α_k to be the exact minimizer, this algorithm reduces to the linear conjugate gradient method, Unlike the linear conjugate gradient method, whose convergence properties are well understood and which is known to be optimal (see page 321), nonlinear conjugate gradient methods sometimes show bizarre convergence properties. It has been proved by Al-Baali that if the level set $\mathcal{L} = \left\{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x}^{(0)}\right\}$ of a convex function f is bounded and in some open neighborbood of $\mathcal{L}$, f is Lipshitz continuously differentiable and that the algorithm is implemented with a line search that satisfies the strong Wolfe conditions, with $0 < c_1 < c_2 < 1$, then

$$\lim_{k \to \infty} \inf \ ||\mathbf{g}^{(k)}|| = 0$$

[34]Restarting conjugate gradient means forgetting the past search directions, and start it anew in the direction of steepest descent.

In summary, quasi-Newton methods are robust. But, they require $O(n^2)$ memory space to store the approximate Hessian inverse, and so they are not directly suited for large scale problems. Modificationsof these methods called Limited Memory Quasi-Newton methods use $O(n)$ memory and they are suited for large scale problems. Conjugate gradient methods also work well and are well suited for large scale problems. However they need to be implemented carefully, with a carefully set line search. In some situations block coordinate descent methods (optimizing a selected subset of variables at a time) can be very much better suited than the above methods.

4.6 Algorithms for Constrained Minimization

The general form of constrained convex optimization problem was given in (4.20) and is restated below.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array} \tag{4.99}$$

For example, when f is linear and g_i's are polyhedral, the problem is a linear program, which was stated in (4.83) and whose dual was discussed on page 289. Linear programming is a typical example of constraint minimization problem and will form the subject matter for discussion in Section 4.7. As another example, when f is quadratic (of the form $\mathbf{x}^T Q\mathbf{x} + \mathbf{b}^T\mathbf{x}$) and g_i's are polyhedral, the problem is called a quadratic programming problem. A special case of quadratic programming is the least squares problem, which we will take up in details in Section 4.8.

4.6.1 Equality Constrained Minimization

The simpler form of constrained convex optimization is when there is only the equality constrained in problem (4.99) and it turns out to be not much different from the unconstrained case. The equality constrained convex problem can be more explicitly stated as in (4.100).

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array} \tag{4.100}$$

where f is a convex and twice continuously differentiable function and $A \in \Re^{p\times n}$ has rank p. We will assume that the finite primal optimal value p^* is attained by f at some point $\widehat{\mathbf{x}}$. The following fundamental theorem for the equality constrained convex problem (4.100) can be derived using the KKT conditions stated

in Section 4.4.4 that were proved to be necessary and sufficiency conditions for optimality of a convex problem with differentible objective and constraint functions.

Theorem 85 *$\widehat{\mathbf{x}}$ is optimal point for the primal iff there exists a $\widehat{\mu}$ such that the following conditions are satisfied.*

$$\begin{aligned} &\nabla f(\widehat{\mathbf{x}}) + A^T\widehat{\mu} = \mathbf{0} \\ &A\widehat{\mathbf{x}} = \mathbf{b} \end{aligned} \tag{4.101}$$

The term $\nabla f(\widehat{\mathbf{x}}) + A^T\widehat{\mu}$ is sometimes called the dual residual (r_d) while the term $A\widehat{\mathbf{x}} - \mathbf{b}$ is referred to as the primal residual (r_p). The optimality condition basically states that both r_d and r_p should both be 0 and the the success of this test is a certificate of optimality.

As an illustration of this theorem, consider the constrained quadratic problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \\ \text{subject to} \quad & P\mathbf{x} = \mathbf{q} \end{aligned} \tag{4.102}$$

By theorem 85, the necessary and sufficient condition for optimality of a point $(\widehat{\mathbf{x}}, \widehat{\lambda})$ is

$$\underbrace{\begin{bmatrix} A & P^T \\ P & 0 \end{bmatrix}}_{KKT\ matrix} \begin{bmatrix} \widehat{\mathbf{x}} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{b} \\ \mathbf{q} \end{bmatrix}$$

The KKT matrix[35] is nonsingular *iff*, $P + A^T A \succ 0$. In such an event, the system of $n + p$ linear equations in $n + p$ unknowns will have a unique solution corresponding to the point of global minimum of (4.102). The linearly constrained least squared problem is a specific example of this and is discussed in Section 4.8.2.

Eliminating Equality Constraints

Figure 3.3 summarized the number of solutions to the system $A\mathbf{x} = \mathbf{b}$ under different conditions. In particular, when the rank of A is the number of its rows (p) and is less than the number of its columns (n), there are infinitely many solutions. This was logically derived in (3.35), and we restate it here for reference:

$$\mathbf{x}_{complete} = \mathbf{x}_{particular} + \mathbf{x}_{nullspace}$$

where the three vectors are defined with respect to the reduced row echelon form R of A (*c.f.* Section 3.6.2):

[35] This matrix comes up very often in many areas such as optimization, mechanics, *etc.*

1. $\mathbf{x}_{complete}$: specifies any solution to $A\mathbf{x} = \mathbf{b}$

2. $\mathbf{x}_{particular}$: is obtained by setting all free variables (corresponding to columns with no pivots) to 0 and solving $A\mathbf{x} = \mathbf{b}$ for pivot variables.

3. $\mathbf{x}_{nullspace}$: is any vector in the null space of the matrix A, obtained as a linear combination of the basis vectors for $N(A)$.

Using formula (3.27) on page 169 to derive the null basis $N \in \Re^{n \times n-p}$ (that is, $AN = 0$ and the columns of N span $N(A)$), we get the following free parameter expression for the solution set to $A\mathbf{x} = \mathbf{b}$:

$$\{\mathbf{x} \,|A\mathbf{x} = \mathbf{b}\} = \left\{N\mathbf{z} + \mathbf{x}_{particular} \,\middle|\, \mathbf{z} \in \Re^{n-p}\right\}$$

We can express the constrained problem in (4.100) in terms of the variables $\mathbf{z} \in \Re^{n-p}$ (that is through an affine change of coordinates) to get the following equivalent problem:

$$\underset{\mathbf{z} \in \Re^{n-1}}{\text{minimize}} \quad f(N\mathbf{z} + \mathbf{x}_{particular}) \tag{4.103}$$

This problem is equivalent to the original problem in (4.100), has no equality constraints and has p fewer variables. The optimal solutions $\widehat{\mathbf{x}}$ and $\widehat{\mu}$ to the primal and dual of (4.100) respectively can be expressed in terms of the optimal solution $\widehat{\mathbf{z}}$ to (**??**) as:

$$\begin{aligned} \widehat{\mathbf{x}} &= N\widehat{\mathbf{z}} + \mathbf{x}_{particular} \\ \widehat{\mu} &= -(AA^T)^{-1}A\nabla f(\widehat{\mathbf{x}}) \end{aligned} \tag{4.104}$$

Any iterative algorithm that is applied to solve the problem (4.104) will ensure that all intermediate points are feasible, since for any $\mathbf{z} \in \Re^{n-p}$, $\mathbf{x} = F\mathbf{z} + \mathbf{x}_{particular}$ is feasible, that is, $A\mathbf{x} = \mathbf{b}$. However, when the Newton's method is applied, the iterates are independent of the exact affine change of coordinates induced by the choice of the null basis F (*c.f.* page 306). The Newton update rule $\Delta\mathbf{z}^{(k)}$ for (4.103) is given by the solution to:

$$N\nabla^2 f(N\mathbf{z}^{(k)} + \mathbf{x}_{particular})N^T\Delta\mathbf{z}^{(k)} = N\nabla f(N\mathbf{z}^{(k)} + \mathbf{x}_{particular})$$

Due the affine invariance of Newton's method, if $\mathbf{z}^{(0)}$ is the starting iterate and $\mathbf{x}^{(0)} = N\mathbf{z}^{(0)} + \mathbf{x}_{particular}$, the k^{th} iterate $\mathbf{x}^{(k)} = N\mathbf{z}^{(k)} + \mathbf{x}_{particular}$ is independent of the choice of the null basis N. We therefore do not need seperate convergence analysis. The algorithm for the Newton's method was outlined in Figure 4.49. Techniques for handling constrained optimization using Newton's method given an infeasible starting point $\mathbf{x}^{(0)}$ can be found in [**?**].

4.6.2 Inequality Constrained Minimization

The general inequality constrained convex minimization problem is

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq \mathbf{0}, \quad i = 1, \ldots, m \\ & A\mathbf{x} = b \end{array} \tag{4.105}$$

where f as well as the g_i's are convex and twice continuously differentiable. As in the case of equality constrained optimization, we will assume that $A \in \Re^{p \times n}$ and has rank p. Further, we will also assume that the finite primal optimal value p^* is attained by f at some point $\widehat{\mathbf{x}}$. Finally, we will assume that the Slaters constraint qualification (*c.f.* page 292) conditions hold so that strong duality holds and the dual optimum is attained. Linear programs (LP), quadratically constrained quadratic programs (QCQP) (all listed in table 4.4 on page 292) and geometric programs[36] (GP) are some examples of convex optimization problems with inequality constraints. An example geometric program (in its convex form) is

$$\begin{array}{lll} \underset{\mathbf{y} \in \Re^n}{\text{minimize}} & \log\left(\sum_{k=1}^{q} e^{\mathbf{a}_k^T \mathbf{y} + b_k}\right) & \\ \text{subject to} & \log\left(\sum_{k=1}^{r} e^{\mathbf{c}_k^T \mathbf{y} + d_k}\right) \leq 0 & i = 1, 2, \ldots, p \\ & \mathbf{g}_i^T \mathbf{y} + h_i & i = 1, 2, \ldots, m \end{array} \tag{4.106}$$

Semi-definite programs (SDPs) do not satisfy conditions such as zero duality gap, *etc.*, but can be handled by extensions of interior-point methods to problems having generalized inequalities.

Logarithmic Barrier

One idea for solving a minimization problem with inequalities is to replace the inequalities by a so-called barrier term. The barrier term is subtracted from the objective function with a weight μ on it. The solution to (4.105) is approximated by the solution to the following problem.

$$\begin{array}{ll} \text{minimize} & B(\mathbf{x}, \mu) = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln\left(-g_i(\mathbf{x})\right) \\ \text{subject to} & A\mathbf{x} = b \end{array} \tag{4.107}$$

[36] Although geometric programs are not convex in their natural form, they can, however, be transformed to convex optimization problems, by a change of variables and a transformation of the objective and constraint functions.

The objective function $B(\mathbf{x}, \mu)$ is called the *logarithmic barrier function*. This function is convex, which can be proved by invoking the composition rules described in Section 4.2.10. It is also twice continuously differentiable. The barrier term, as a function of $\mathbf{x}$ approaches $+\infty$ as any feasible interior point $\mathbf{x}$ approaches the boundary of the feasible region. Because we are minimizing, this property prevents the feasible iterates from crossing the boundary and becoming infeasible. We will denote the point of optimality $\widehat{\mathbf{x}}(\mu)$ as a function of μ.

However, the optimal solution to the original problem (a typical example being the LP discussed in Section 4.7) is typically a point on the boundary of the feasible region (we will see this in the case of linear programming in Section 4.7). To obtain such a boundary point solution, it is necessary to keep decreasing the parameter μ of the barrier function to 0 in the limit. As a very simple example, consider the following inequality constrained optimization problem.

$$\begin{aligned} &\text{minimize} && x^2 \\ &\text{subject to} && x \geq 1 \end{aligned}$$

The logarithmic barrier formulation of this problem is

$$\text{minimize} \quad x^2 - \mu \ln (x - 1)$$

The unconstrained minimizer for this convex logarithmic barrier function is $\widehat{\mathbf{x}}(\mu) = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 2\mu}$. As $\mu \to 0$, the optimal point of the logarithmic barrier problem approaches the actual point of optimality $\widehat{\mathbf{x}} = 1$ (which, as we can see, lies on the boundary of the feasible region). The generalized idea, that as $\mu \to 0$, $f(\widehat{\mathbf{x}}) \to p^*$ (where p^* is the optimal for (4.105)) will be proved next.

Properties of the estimate $f(\widehat{\mathbf{x}}(\mu))$

The following are necessary and sufficient conditions for $\widehat{\mathbf{x}}(\mu)$ to be a solution to (4.107) for a fixed μ (see KKT conditions in (4.88)):

1. The point $\widehat{\mathbf{x}}(\mu)$ must be strictly feasible. That is,

   $$A\widehat{\mathbf{x}}(\mu) = \mathbf{b}$$

   and

   $$g_i(\widehat{\mathbf{x}}(\mu)) < 0$$

2. There must exist a $\eta \in \Re^p$ such that

$$\nabla f(\widehat{\mathbf{x}}(\mu)) + \sum_{i=1}^{m} \frac{-\mu}{g_i(\widehat{\mathbf{x}}(\mu))} \nabla g_i(\widehat{\mathbf{x}}(\mu)) + A^T \widehat{\eta} = \mathbf{0} \tag{4.108}$$

Define

$$\widehat{\lambda}_i(\mu) = \frac{-\mu}{g_i(\widehat{\mathbf{x}}(\mu))}$$

and

$$\widehat{\eta}(\mu) = \widehat{\eta}$$

We claim that the pair $(\widehat{\lambda}(\mu), \widehat{\eta}(\mu))$ is dual feasible. The following steps prove our claim

1. Since $g_i(\widehat{\mathbf{x}}(\mu)) < 0$ for $i = 1, 2, \ldots, m$, $\widehat{\lambda}(\mu) \succ \mathbf{0}$.
2. Based on the proof of theorem 82, we can infer that $L(\mathbf{x}, \lambda, \eta)$ is convex in $\mathbf{x}$.

   $$L(\mathbf{x}, \lambda, \eta) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \eta^T (A\mathbf{x} - \mathbf{b})$$

   Since the lagrangian is convex in $\mathbf{x}$ and since it is differentiable on its domain, from (4.108), we can conclude that $\widehat{\mathbf{x}}(\mu)$ is a critical point of $L(\mathbf{x}, \lambda, \eta)$ and therefore minimizes it for $(\widehat{\lambda}(\mu), \widehat{\eta}(\mu))$.
3. That is, the dual $L^*(\widehat{\lambda}(\mu), \widehat{\eta}(\mu))$ is defined and therefore, $(\widehat{\lambda}(\mu), \widehat{\eta}(\mu))$ is dual feasible.

$$L^*(\widehat{\lambda}(\mu), \widehat{\eta}(\mu)) = f(\widehat{\mathbf{x}}(\mu)) + \sum_{i=1}^{m} \widehat{\lambda}_i g_i(\widehat{\mathbf{x}}(\mu)) + \widehat{\eta}(\mu)^T (A\widehat{\mathbf{x}}(\mu) - \mathbf{b}) = f(\widehat{\mathbf{x}}(\mu)) - m\mu \tag{4.109}$$

From the weak duality theorem 81, we know that $d^* \leq p^*$, where d^* and p^* are the primal and dual optimals respectively, for (4.105). Since $L^*(\widehat{\lambda}(\mu), \widehat{\eta}(\mu)) \leq d^*$ (by definition), we will have from (4.109), $f(\widehat{\mathbf{x}}(\mu)) - m\mu \leq p^*$. Or equivalently,

$$f(\widehat{\mathbf{x}}(\mu)) - p^* \leq m\mu \tag{4.110}$$

The inequality in (4.110) forms the basis of the barrier method; it confirms the intuitive idea that $\widehat{\mathbf{x}}(\mu)$ converges to an optimal point as $\mu \to 0$. We will next discuss the barrier method.

The Barrier Method

The barrier method is a simple extension of the unconstrained minimization method to inequality constrained minimization. This method is based on the property in (4.110). This method solves a sequence of unconstrained (or linearly constrained) minimization problems, using the last point found as the starting point for the next unconstrained minimization problem. It computes $\widehat{\mathbf{x}}(\mu)$ for a

sequence of decreasing values of μ, until $m\mu \leq \epsilon$, which guarantees that we have an ϵ-suboptimal solution of the original problem. It was originally proposed as the sequential unconstrained minimization technique (SUMT) technique by Fiacco and McCormick in the 1960s. A simple version of the method is outlined in Figure 4.56.

Find a strictly feasible starting point $\widehat{\mathbf{x}}$, $\mu = \mu^{(0)} > 0$, $1 > \alpha > 0$.
Select an appropriate tolerance $\epsilon > 0$.
repeat
1. **Centering Step:** Compute $\widehat{\mathbf{x}}(\mu)$ by minimizing $B(\mathbf{x}, \mu)$ (optionally subject to $A\mathbf{x} = \mathbf{b}$) starting at $\mathbf{x}$.
2. Update $\mathbf{x} = \widehat{\mathbf{x}}(\mu)$.
3. If $m\mu \leq \epsilon$, **quit**.
4. Decrease μ: $\mu = \alpha\mu$.

until

Figure 4.56: The Barrier method.

The centering step (1) can be executed using any of the descent techniques discussed in Section 4.5. It can be proved [**?**] that the duality gap is $m\mu^{(0)}\alpha^k$ after k iterations. Therefore, the desired accuracy ϵ can be achieved by the barrier method after exactly $\left\lceil \frac{\log\left(\frac{m\mu^{(0)}}{\epsilon}\right)}{-log(\alpha)} \right\rceil$ steps.

Successive minima $\widehat{\mathbf{x}}(\mu)$ of the Barrier function $B(\mathbf{x}, \mu)$ can be shown to have the following properties. Let $\overline{\mu} < \mu$ for sufficiently small μ, then

1. $B(\widehat{\mathbf{x}}(\overline{\mu}), \overline{\mu}) < B(\widehat{\mathbf{x}}(\mu), \mu)$
2. $f(\widehat{\mathbf{x}}(\overline{\mu})) \leq f(\widehat{\mathbf{x}}(\mu))$
3. $-\sum_{i=1}^{m} \ln\left(-g_i(\widehat{\mathbf{x}}(\overline{\mu}))\right) \geq -\sum_{i=1}^{m} \ln\left(-g_i(\widehat{\mathbf{x}}(\mu))\right)$

When a strictly feasible point $\widehat{\mathbf{x}}$ is not known, the barrier method is preceded by a preliminary stage, called phase I, in which a strictly feasible point is computed (if it exists). The strictly feasible point found during phase I is then used as the starting point for the barrier method. This is discussed in greater details in [**?**].

4.7 Linear Programming

Linear programming has been widely used in the industry for maximizing profits, minimizing costs, *etc.* The word *linear* implies that the cost function is linear in the form of an inner product.

The inputs to the program are

1. $\mathbf{c}$, a cost vector of size n.

2. An $m \times n$ matrix A.

3. A vector $\mathbf{b}$ of size m.

The unknown is a vector $\mathbf{x}$ of size n, and this is what we will try to determine.

In linear programming (LP), the task is to minimize a linear objective function of the form $\sum_{j=1}^{n} c_j x_j$, subject to linear inequality constraints[37] of the form $\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m$ and $x_i > 0$. The problem can be stated as in (4.111). In contrast to the LP specification on page 289, where the constraint $\mathbf{x} \geq \mathbf{0}$ was absorbed into the more general constraint $-A\mathbf{x} + \mathbf{b} \leq \mathbf{0}$, here we choose to specify it as a seperate constraint.

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \Re^n} & \mathbf{x}^T \mathbf{c} \\ \text{subject to} & -A\mathbf{x} + \mathbf{b} \leq \mathbf{0} \quad \mathbf{x} \geq \mathbf{0} \end{array} \tag{4.111}$$

The flip side of this problem is that it has no analytical formula as a solution. However, that does not make a big difference in practice, because there exist reliable and efficient algorithms and software for linear programming. The computational time is roughly proportional to $n^2 m$, if $m \geq n$. This is basically the cost of one iteration in an interior point method.

Linear programming (LP) problems are harder to recognize in practice and often need reformulations to get into the standard form in (4.111). Minimizing a piecewise linear function of x is not an LP, thought it can be written and solved as an LP. Other problems involving 1 or ∞ norms can also be written as linear programming problems.

The basis for linear programing was mentioned on page 250; linear functions have no critical points and therefore, by theorem 60, the extreme values are always assumed at the boundary of the feasible set. In the case of linear programs, the feasible set is itself defined by linear inequalities: $\{\mathbf{x} | -A\mathbf{x} + \mathbf{b} \leq \mathbf{0}\}$. Applying the argument recursively, it can be proved that the extreme values for a linear program are assumed at some corners (*i.e.*, vertices) of the feasible set. A *corner* is the intersection point of n different planes, each given by a single equation. That is, a corner point is obtained by turning n of the $n+m$ inequalities into equalities and finding their intersection[38] An edge is the intersection of $n-1$ inequalities and connects two corners. Geometrically, it can be observed that when you maximize or minimize some linear function, as your progress in one direction in the search space, the objective will either increase monotonically or decrease monotonically. Therefore, the maximum and minimum will be found at the corners of the allowed region.

[37] It is a rare feature to have linear inequality constraints.

[38] In general, there are $\frac{(n+m)!}{n!m!}$ intersections.

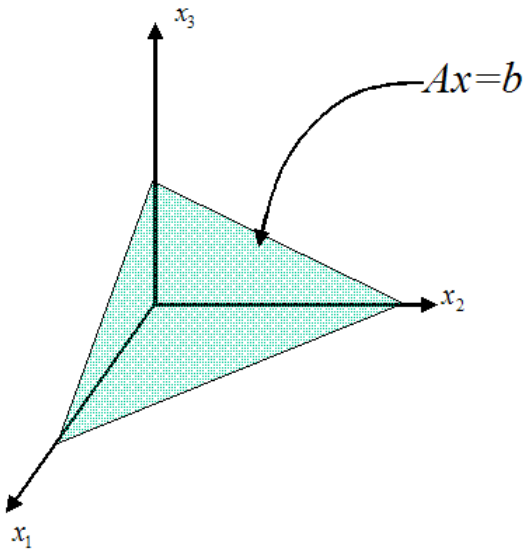

Figure 4.57: Example of the feasible set of a linear program for $n = 3$.

The feasible set is in the form of a finite interval in n dimensions. Figure 4.57 pictorially depicts a typical example of the feasible region for $n = 3$. The constraints $A\mathbf{x} \geq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ would allow a tetrahedron or pyramid in the first (or completely positive) octant. If the constraint was an equality, $A\mathbf{x} = \mathbf{b}$, the feasible set would be the shaded traingle in the figure. In general for any n, the constraint $A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ would yield as the feasible set, a polyhedron. The task of maximizing (or miminizing) the linear objective function $\mathbf{x}^T\mathbf{c} = \sum_{i=1}^{n} c_i x_i$ translates to finding a solution at one of the corners of the feasible region. Corners are points where some of the inequality constraints are tight or active, and others are not. At the corners, some of the inequality constraints translate to equalities. It is just a question of finding the right corner.

Why not just search all corners for the optimal answer? The trouble is that there are lots of corners. In n dimensions, with m constraints, the number of corners grows exponentially and there is no way to check all of them. There is an interesting competition between two quite different approaches for solving linear programs:

1. *The simplex method*

2. *Interior point barrier method*

4.7.1 Simplex Method

The simplex algorithm [**?**] is one of the fundamenetal methods for linear programming, developed in the late 1940s by Dantzig. It is the best established approach for solving linear problems. In the worst case, the algorithm takes a number of steps that is exponential in n; but, in practice it is the most efficient method for solving linear programs.

The simplex method first constructs an admissible solution at a corner (which can be quite a bit of a job) of the polyhedron and then moves along its edges to vertices with successively higher values of the objective function until the optimum is reached. The movement along an edge originating at a vertex is performed by 'loosening' one of the inequalities that were tight at the vertex. The inequality chosen for 'loosening' is the one promising the fastest drop in the objective function $\mathbf{x}^T\mathbf{c}$. The rate of decrease along an edge can be measured using the gradient of the objective. This procedure is carried out iteratively, till the method encounters a vertex which has no edge (constraint) that is a promising descent direction (which means that the cost goes up along all edges incident at that vertex). Since an edge corresponding to decreasing value of the objective cannot correspond to its increasing value, no edge will be traversed twice in this process.

We will first rewrite the constraints $A\mathbf{x} \geq \mathbf{b}$ in the above LP as equations, by introducing a new non-negative "slack" variable s_j for the j^{th} constraint (for all j's) and subtracting it from the left-hand side of each inequality:

$$A\mathbf{x} - \mathbf{s} = \mathbf{b}$$

or equivalently in matrix notation

$$[-A \ +I] \left[\begin{array}{c} \mathbf{x} \ \mathbf{s} \end{array} \right] = -\mathbf{b}$$

We will treat the $m \times n + m$ matrix $M = [-A \ \ I]$ as our new coefficient matrix and $\mathbf{y} = [\mathbf{x} \ \ \mathbf{s}]^T$ as our new variable vector. With this, the above constraint can be rewritten as

$$M\mathbf{y} = -\mathbf{b}$$

The feasible set is now governed by these m equality constraints and the $n + m$ non-negativity constraints $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$. The original cost vector $\mathbf{c}$ is extended to a vector $\mathbf{d}$ by appending m more zero components. This leaves us with the following problem, equivalent to the original LP (4.111).

$$\begin{array}{ll} \min\limits_{\mathbf{y} \in \Re^{n+m}} & \mathbf{y}^T\mathbf{d} \\ \text{subject to} & M\mathbf{y} = -\mathbf{b} \quad \mathbf{y} \geq \mathbf{0} \end{array} \tag{4.112}$$

We will assume that the matrix A (and therefore M) is of full row rank, that is of rank m. In practice, a preprocessing phase is applied to the user-supplied

data to remove some redundancies from the given constraints to get a full row rank matrix.

The following definitions and observations will set the platform for the simplex algorithm, which we will describe subsequently.

1. A vector $\mathbf{y}$ is a *basic feasible point* if it is feasible and if there exists a subset $\mathcal{B}$ of the index set $\{1, 2, ..., n\}$ such that

   (a) $\mathcal{B}$ contains exactly m indices.

   (b) $\mathbf{y} \geq \mathbf{0}$.

   (c) $y_i \geq 0$ can be inactive (that is $y_i > 0$) only if $i \in \mathcal{B}$. In other words, $i \notin \mathcal{B} \Rightarrow y_i = 0$.

   (d) If $\mathbf{m}_i$ is the i^{th} column of M, the $m \times m$ matrix B defined as $B = [\mathbf{m}_i]_{i \in \mathcal{B}}$ is nonsingular.

   A set $\mathcal{B}$ satisfying these properties is called a *basis* for the problem (4.112). The corresponding matrix B is called the basis matrix. Any variable y_i for $i \in \mathcal{B}$ is called a *basic variable*, while any variable y_i for $i \notin \mathcal{B}$ is called a *free variable*.

2. It can be seen that all basic feasible points of (4.112) are corners of the feasible simplex $\mathcal{S} = \{\mathbf{x} | A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ and vice versa. In other words, a corner of $\mathcal{S}$ corresponds to a point $\mathbf{y}$ in the new representation that has n components as zeroes.

3. Any two corners connected by an edge will have exactly $m - 1$ common basic variables. Each corner has n incident edges (corresponding to the addition of any one of n new basic variables and the corresponding drop of a basic variable).

4. Further, it can be proved that

   (a) If (4.112) has a nonempty feasible region, then there is at least one basic feasible point

   (b) If (4.112) has solutions, then at least one such solution is a basic optimal point

   (c) If (4.112) is feasible and bounded, then it has an optimal solution.

   This is known as the fundamental theorem of linear programming.

Using the ideas and notations presented above, the simplex algorithm can be outlined as follows.

1. Each iterate generated by the simplex algorithm is a *basic feasible point* of (4.112).

2. *Entering free variable:* The next iterate is determined by moving along an edge from one basic feasible solution to another. As discussed above, movement along an edge will mean that $m-1$ variables will remain basic while one will become free. On the other hand, a new free variable will become basic. The real decision is which variable should be removed from the basis and which should be added. The idea in the simplex algorithm is to include that free variable y_k, which has the most negative component d_k (something like steepest descent in the L_1 norm).

3. *Leaving basic variable:* The basic variable from the current basis that will leave next is determined using a *pivot rule*. A commonly applied pivot rule is to determine the leaving basic variable through the constraint (say the j^{th} one) that has the smallest non-negative ratio of the right hand side $b_j^{'}$ of the constraint to the coefficient $m_{jk}^{'}$ of the entering variable y_k. If the coefficients of y_k are negative in all the constraints, it implies an unbounded case; the cost can be made $-\infty$ by arbitrarily increasing the value of y_k.

4. In order to facilitate the easy identification of leaving basic variables, we bring the equations into a form such that the basic variables stand by themselves. This is done by treating the new entering variable y_k as a 'pivot' in the j^{th} equation and substituting its value in terms of the other variables in the j^{th} equation into the other equations (as well as the cost function $\mathbf{y}^T\mathbf{d}$). In this form,

   (a) the protocol is that variables corresponding to all columns of M that are in unit form are basic variables, while the rest are free variables.

   (b) the choice of an equality in the step above automatically entails the choice of the leaving variable - the basic variable y_l corresponding to row j will be the next leaving variable.

5. In a large problem, it is possible for a leaving variable to reenter the basis at a later stage. Unless there is *degeneracy*, the costs keep going down and it can never happen that all of the m basic variables are the same as before. Thus, no corner is revisited and the method must end at the optimal corner or conclude that the cost is unbounded below. Degenracy is said to occur if more than the usual n components of $\mathbf{x}$ are 0 (in which case, cycling might occur but extremely rarely).

Since each simplex step involves decisions (choice of entering and leaving basic variables) and row operations (pivoting *etc.*), it is convenient to fit the data into a large matrix or *tableau*. The operations of the simplex method outlined above can be systematically translated to operations on the tableau.

1. The starting tableau is just a bigger $m+1 \times m+n$ matrix

$$T = \begin{bmatrix} M & -\mathbf{b} \\ \mathbf{d} & \mathbf{0} \end{bmatrix}$$

2. Our first step is to get one basic variable alone on each row. Without loss of generality, we will renumber the variables and rearrange the corresponding coefficients of M so that at every iteration, $y_1, y_2, \ldots y_m$ are the basic variables and the rest are free (*i.e.*, 0). The first m columns of A form an $m \times m$ square matrix B and the last n form an $m \times n$ matrix N. The cost vector $\mathbf{d}$ can also be split as $\mathbf{d}^T = [\mathbf{d}_B^T \;\; \mathbf{d}_N^T]$ and the variable vector can be split as $\mathbf{y}^T = [\mathbf{y}_B^T \;\; \mathbf{y}_N^T]$ with $\mathbf{y}_N = \mathbf{0}$. To operate with the tableau, we will split it as

$$\begin{bmatrix} B & N & -\mathbf{b} \\ \mathbf{d}_B^T & \mathbf{d}_N^T & \mathbf{0} \end{bmatrix}$$

Performing Gauss Jordan elimination on the columns corresponding to basic variables, we get the equations into the form that will be preserved across iterations.

$$\begin{bmatrix} I & B^{-1}N & -B^{-1}\mathbf{b} \\ \mathbf{d}_B^T & \mathbf{d}_N^T & \mathbf{0} \end{bmatrix}$$

Further, we will ensure that all the columns corresponding to basic variables are in the unit form.

$$\begin{bmatrix} I & B^{-1}N & -B^{-1}\mathbf{b} \\ \mathbf{d}_B^T - \mathbf{d}_B^T I = \mathbf{0} & \mathbf{d}_N^T - \mathbf{d}_B^T B^{-1}N & \mathbf{d}_B^T B^{-1}\mathbf{b} \end{bmatrix}$$

This corresponds to a solution $\mathbf{y}_B = -B^{-1}\mathbf{b}$ with cost $\mathbf{d}^T\mathbf{y} = -\mathbf{d}_B^T B^{-1}\mathbf{b}$, which is the negative of the expression on the right hand bottom corner.

3. In the above tableau, the components of the expression $\mathbf{r} = \mathbf{d}_N^T - \mathbf{d}_B^T B^{-1}N$ are the *reduced costs* and capture what it costs to use the existing set of free variables; if the direct cost in $\mathbf{d}_N$ is less than the saving due to use of the other basic variables, it will help to try a free variable. This guides us in the choice of the *entering variable*. If $\mathbf{r} = \mathbf{d}_N^T - \mathbf{d}_B^T B^{-1}N$ has any negative component, then the variable corresponding to the most negative component is picked up as the next entering variable and this choice corresponds to moving from a corner of the polytope $\mathcal{S}$ to an adjacent corner with lower cost. Let y_k be the entering variable and d_k the corresponding cost.

4. As the entering component y_k is increased, to maintain $M\mathbf{y} = -\mathbf{b}$, the first component $\mathbf{y}_j$ that decreases to 0 becomes the leaving variable and transforms from a basic to a free variable. The other components of $\mathbf{y}_B$ would have moved around but would remain positive. Thus, the one that drops to zero should satisfy

$$j = \underset{t=1,2,\ldots,m \;\; (B^{-1}N)_{tk}>0}{\operatorname{argmin}} \frac{\left(-B^{-1}\mathbf{b}\right)_t}{(B^{-1}N)_{tk}}$$

Note that the minimum is taken only over the positive components $\left(B^{-1}N\right)_{tk}$. If there are no positive components, the next corner is infinitely far away and the cost can be reduced forever to yield a minimum cost of $-\infty$.

5. With the new choice of basic variables, steps (2)-(4) are repeated till the reduced cost is completely non-negative. The variables corresponding to the unit columns in the final tableau are the basic variables at the optimum.

What we have not discussed so far is how to obtain the initial basic feasible point. If $\mathbf{x} = 0$ satisfies $A\mathbf{x} \geq \mathbf{b}$, we can have an initial basic feasible point with the basic variables comprising of $\mathbf{s}$ and $\mathbf{x}$ constituting the free variables. This is illustrated through the following example. Consider the problem

$$\begin{array}{ll} \min\limits_{x_1,x_2,x_3 \in \Re} & -15x_1 - 18x_2 - 20x_3 \\ \text{subject to} & -\frac{1}{6}x_1 - \frac{1}{4}x_2 - \frac{1}{2}x_3 \geq -60 \\ & -40x_1 - 50x_2 - 60x_3 \geq -2880 \\ & -25x_1 - 30x_2 - 40x_3 \geq -2400 \\ & x_1 \geq 0,\, x_2 \geq 0,\, x_3 \geq 0 \end{array}$$

The initial tableau is

$$\left[\begin{array}{cccccc|c} \frac{1}{6} & \frac{1}{4} & \frac{1}{2} & 1 & 0 & 0 & 60 \\ 40 & 50 & 60 & 0 & 1 & 0 & 2880 \\ 25 & 30 & 40 & 0 & 0 & 1 & 2400 \\ -15 & -18 & -20 & 0 & 0 & 0 & 0 \end{array}\right]$$

The most negative component of the reduced cost vector is for $k = 3$. The pivot row number is $2 = \underset{t=1,2,\ldots,m \quad (B^{-1}N)_{tk}>0}{\operatorname{argmin}} \frac{\left(B^{-1}\mathbf{b}\right)_t}{(B^{-1}N)_{tk}}$. Thus, the leaving basic variable is s_2 (the basic variable corresponding to the second row) while the entering free variable is x_3. Performing Gauss elimination to obtain column $k = 3$ in the unit form, we get

$$\left[\begin{array}{cccccc|c} -\frac{1}{6} & -\frac{1}{6} & 0 & 1 & -\frac{1}{120} & 0 & 36 \\ \frac{2}{3} & \frac{5}{6} & 1 & 0 & \frac{1}{60} & 0 & 48 \\ -\frac{5}{3} & -\frac{10}{3} & 0 & 0 & -\frac{2}{3} & 1 & 480 \\ -\frac{5}{3} & -\frac{4}{3} & 0 & 0 & \frac{1}{3} & 0 & 960 \end{array}\right]$$

This tableau corresponds to the solution $x_1 = 0, x_2 = 0, x_3 = 48, s_1 = 0, s_2 = 36, s_3 = 480$ and cost $\mathbf{c}^T\mathbf{x} = -960$ (negative of the number on the right hand bottom corner). Since the reduced cost vector has still some negative components, it is possible to find a basic feasible solution with lower cost. Using the most negative component of the reduced cost vector, we select the next pivot

element to be $m_{21} = \frac{2}{3}$. Again performing Gaussian elimination, we obtain the tableau corresponding to the next iterate.

$$\left[\begin{array}{cccccc|c} 0 & \frac{1}{24} & \frac{1}{4} & 1 & -\frac{1}{240} & 0 & 48 \\ 1 & \frac{5}{4} & \frac{3}{2} & 0 & \frac{1}{40} & 0 & 72 \\ 0 & -\frac{5}{4} & \frac{5}{2} & 0 & -\frac{5}{8} & 1 & 600 \\ 0 & \frac{3}{4} & \frac{5}{2} & 0 & \frac{3}{8} & 0 & 1080 \end{array}\right]$$

Note that the optimal solution has been found, since the reduced cost vector is non-negative. The optimal solution is $x_1 = 72, x_2 = 0, x_3 = 0, s_1 = 48, s_2 = 0, s_3 = 600$ and cost $\mathbf{c}^T\mathbf{x} = -1080$

What if $\mathbf{x} = 0$ does not satisfy $A\mathbf{x} \geq \mathbf{b}$? The choice of $\mathbf{s}$ as the basic variables and $\mathbf{x}$ as the free variables will not be valid. As an example, consider the problem

$$\begin{array}{ll} \min\limits_{x_1,x_2,x_3 \in \Re} & 30x_1 + 60x_2 + 70x_3 \\ \text{subject to} & x_1 + 3x_2 + 4x_3 \geq 14 \\ & 2x_1 + 2x_2 + 3x_3 \geq 16 \\ & x_1 + 3x_2 + 2x_3 \geq 12 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0 \end{array}$$

The initial tableau is

$$\left[\begin{array}{cccccc|c} -1 & -3 & -4 & 1 & 0 & 0 & -14 \\ -2 & -2 & -3 & 0 & 1 & 0 & -16 \\ -1 & -3 & -2 & 0 & 0 & 1 & -12 \\ 30 & 60 & 70 & 0 & 0 & 0 & 0 \end{array}\right]$$

With the choice of basic and free variables as above, we are not even in the feasible region to start off with. In general, if we have any negative number in the last column of the tableau, $\mathbf{x} = \mathbf{0}$ is not in the feasible region. Further, we have no negative numbers in the bottom row, which does not leave us with any choice of cost reducing free variable. But this is not of primary concern, since we first need to maneuver our way into the feasible region. We do this by moving from one basic point (that is, a point having not more than n zero components) to another till we land in the feasible region, which is indicated by all positive components in the extreme right hand column. This movement from one basic point to another is not driven by negative components in the cost vector, but rather by the negative components in the right hand column. The new rules for moving from one basic point to another are:

1. Pick[39] any negative number in the far right column (excluding the last row). Let this be in the q^{th} row for $q < m + 1$.

[39] Note that there is no priority here.

2. In the q^{th} row, move[40] to left to a column number k where there is another negative number. The variable y_k will be the next entering variable.
3. Choose pivot element m_{jk} which gives the smallest positive ratio of an element in the j^{th} row of the last column to the element m_{jk}. The leaving variable will be y_j.
4. Once the pivot element is chosen, proceed as usual to convert the pivot element to 1 and the other elements in the pivot column to 0.
5. Repeat steps (1)-(4) on the modified tableau until there is no negative element in the right-most column.

Applying this procedure to the tableau above, we pick $m_{2,1} = -2$ as our first pivot element and do row elimination to get the first column in unit form.

$$\left[\begin{array}{cccccc|c} 0 & -2 & -\frac{5}{2} & 1 & -\frac{1}{2} & 0 & -6 \\ 1 & 1 & \frac{3}{2} & 0 & -\frac{1}{2} & 0 & 8 \\ 0 & -2 & -\frac{1}{2} & 0 & -\frac{1}{2} & 1 & -4 \\ 0 & 30 & 25 & 0 & 15 & 0 & -240 \end{array}\right]$$

We pick $m_{35} = -\frac{1}{2}$ as our next pivot element and do similar row elimination operations to obtain

$$\left[\begin{array}{cccccc|c} 0 & 0 & -2 & 1 & 0 & -1 & -2 \\ 1 & 3 & 2 & 0 & 0 & -1 & 12 \\ 0 & 4 & 1 & 0 & 1 & -2 & 8 \\ 0 & -30 & 10 & 0 & 0 & 30 & -360 \end{array}\right]$$

We have still not obtained a feasible basic point. We choose $m_{16} = -1$ as the next pivot and do row eliminations to get the next tableau.

$$\left[\begin{array}{cccccc|c} 0 & 0 & 2 & -1 & 0 & 1 & 2 \\ 1 & 3 & 4 & -1 & 0 & 0 & 14 \\ 0 & 4 & 5 & -2 & 1 & 0 & 12 \\ 0 & -30 & -50 & 30 & 0 & 0 & -420 \end{array}\right]$$

This tableau has not negative numbers in the last column and gives a basic feasible point $x_1 = 14, x_2 = 0, x_3 = 0$. Once we obtain the basic feasible point, we rever to the standard simplex procedure discussed earlier. The most negative component of the reduced cost vector is -50 and this leads to the pivot element $m_{13} = 2$. Row elimination yields

$$\left[\begin{array}{cccccc|c} 0 & 0 & 1 & -\frac{1}{2} & 0 & \frac{1}{2} & 1 \\ 1 & 3 & 0 & 1 & 0 & -2 & 10 \\ 0 & 4 & 0 & \frac{1}{2} & 1 & -\frac{5}{2} & 7 \\ 0 & -30 & 0 & 5 & 0 & 25 & -370 \end{array}\right]$$

[40] Note that there is no priority here either.

Our next pivot element is $m_{32} = 4$. Row elimination yields

$$\left[\begin{array}{cccccc|c} 0 & 0 & 1 & -\frac{1}{2} & 0 & \frac{1}{2} & 1 \\ 1 & 0 & 0 & \frac{5}{8} & -\frac{3}{4} & -\frac{1}{8} & \frac{19}{4} \\ 0 & 1 & 0 & \frac{1}{8} & \frac{1}{4} & -\frac{5}{8} & \frac{7}{4} \\ 0 & 0 & 0 & \frac{35}{4} & \frac{15}{2} & \frac{25}{4} & -\frac{635}{2} \end{array}\right]$$

We are done! The reduced cost vector has no more negative components. The optimal basic feasible point is $x_1 = 4.75, x_2 = 1.75, x_3 = 1$ and the optimal cost is 317.5.

Revised Simplex Method

The simplex method illustrated above serves two purposes:

1. Doing all the eliminations completely makes the idea clear.
2. It easier to follow the process when working out the solution by hand.

For computational purposes however, it is uncommon now to use the method as described earlier. This is because, once $\mathbf{r}$ is computed, none of the columns above $\mathbf{r}$, (except for that corresponding to the leaving variable) are used. Therefore, computing them is a useless effort. Doing the eliminations completely at each step cannot be justified practically. Instead, the more efficient version of the simplex method, as outlined below, is used by software packages. It is called the revised simplex method and is essentially the simplex method itself, boiled down.

Compute the reduced costs $\mathbf{r} = \mathbf{d}_N - (\mathbf{d}_B)B^{-1}N$.
while $\mathbf{r} \not\geq \mathbf{0}$ **do**
1. Let r_k be the most negative component of $\mathbf{r}$.
2. Compute $\mathbf{v} = B^{-1}\mathbf{n}_i$, where $\mathbf{n}_i$ is the i^{th} column of N.
3. Let $j = \underset{t=1,2,\ldots,m \quad (B^{-1}\mathbf{n}_i)_t > 0}{\operatorname{argmin}} \frac{\left(-B^{-1}\mathbf{b}\right)_t}{(B^{-1}\mathbf{n}_i)_t}$.
4. Update B (or B^{-1}) and $\mathbf{x}_B = B^{-1}\mathbf{b}$ to reflect the j^{th} leaving column and the k^{th} entering variable.

Compute the new reduced costs $\mathbf{r} = \mathbf{d}_N - (\mathbf{d}_B)B^{-1}N$.
end while

Figure 4.58: The revised simplex method.

4.7.2 Interior point barrier method

Researchers have dreamt up pathological examples for the simplex method, for which the simplex method takes an exponential amount of time. In practice, however, the simplex method is one of the most efficient methods for a majority

of the LPs. Application of interior point methods to LP have led to a new competitor to the simplex method in the form of interior point methods for linear programming. In contrast to the simplex algorithm, which finds the optimal solution by progressing along points on the boundary of a polyhedral set, interior point methods traverse to the optimal through the interior of the feasible region (polyhedral set in the case of LPs). The first in this league was the iterative Karmarkar's method [**?**], developed by Narendra Karmarkar in 1984. Karmarkar also proved that the algorithm was polynomial time. This line of research was inspired by the *ellipsoid method* for linear programming, outlined by Leonid Khachiyan in 1979; the ellipsoid algorithm itself was introduced by Naum Z. Shor, *et. al.* in 1972 and used by Leonid Khachiyan [**?**] to prove the polynomial-time solvability of linear programs r linear programming , which was the first such algorithm known to have a polynomial running time.

This competitor to the simplex method takes a Newton's method-like approach through the interior of the feasible region. Newton steps are taken till the 'barrier' is encountered. It stirred up the world of optimization and inspired the whole class of barrier methods. Following this, a lot of interest was generated in the application of the erstwhile interior point methods for general non-linear constrainted optimization problems. The Karmarkar's algorithm is now replaced by an improved logarithmic barrier method that makes use of the primal as well as the dual for solving an LP. Shanno and Bagchi [**?**] showed that Karmarkars method is just a special case of the logarithmic barrier function method. We will restrict our discussion to the primal-dual barrier method [**?**].

The dual (4.113) for the linear program (4.111) can be derived in manner similar to the dual on page 289.

$$\begin{array}{ll} \max\limits_{\lambda \in \Re^m} & \lambda^T \mathbf{b} \\ \text{subject to} & A^T \lambda \leq \mathbf{c} \\ & \lambda \geq \mathbf{0} \end{array}$$

The weak duality theorem (theorem 81) states that the objective function value of the dual at any feasible solution is always less than or equal to the objective function value of the primal at any feasible solution. That is, for any primal feasible $\mathbf{x}$ and any dual feasible λ,

$$\mathbf{c}^T \mathbf{x} - \mathbf{b}^T \lambda \geq 0$$

For this specific case, the weak duality is easy to see: $\mathbf{b}^T \lambda \leq \mathbf{x}^T A^T \lambda \leq \mathbf{x}^T \mathbf{c}$.

Further, it can be proved using the *Farkas' lemma* that if the primal has an optimal solution $\mathbf{x}^*$ (which is assumed to be bounded), then the dual also has an optimal solution[41] λ^*, such that $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \lambda^*$.

The following steps will set the platform for the interior point method.

[41] For an LP and its dual D, there are only four possibilities:

1. (LP) is bounded and feasible and (D) is bounded and feasible.

1. As on page 4.7.1, we will rewrite the constraints $A\mathbf{x} \geq \mathbf{b}$ in the above LP as equations, by introducing a new non-negative "slack" variable s_j for the j^{th} constraint (for all j's) and subtracting it from the left-hand side of each inequality. This gives us the constraint $M\mathbf{y} = -\mathbf{b}$, where $M = [-A \quad I]$ and $\mathbf{y} = [\mathbf{x} \quad \mathbf{s}]^T$. As before, the original cost vector $\mathbf{c}$ is extended to a vector $\mathbf{d}$ by appending m more zero components. This gives us the following problem, equivalent to the original LP (**??**).

$$\begin{array}{ll} \min\limits_{\mathbf{y}\in\Re^{n+m}} & \mathbf{y}^T\mathbf{d} \\ \text{subject to} & M\mathbf{y} = -\mathbf{b} \\ & \mathbf{y} \geq \mathbf{0} \end{array} \tag{4.113}$$

   Its dual problem is given by

$$\begin{array}{ll} \max\limits_{\lambda\in\Re^{m}} & -\lambda^T\mathbf{b} \\ \text{subject to} & M^T\lambda \leq \mathbf{d} \end{array} \tag{4.114}$$

2. Next, we set up the barrier method formulation of the dual of the linear program. Letting $\mu > 0$ be a given fixed parameter, which is decreased during the course of the algorithm. We also insert slack variables $\xi = [\xi_1, \xi_2, \ldots, \xi_n]^T \geq \mathbf{0}$. The barrier method formulation of the dual is then given by:

$$\begin{array}{ll} \max\limits_{\lambda\in\Re^{m}} & -\lambda^T\mathbf{b} + \mu\sum_{i=1}^{n}\ln(\xi_i) \\ \text{subject to} & M^T\lambda + \xi = \mathbf{d} \end{array} \tag{4.115}$$

   The conditions $\xi_i \geq 0$ are no longer needed since $\ln(\xi_i) \to \infty$ as $\xi_i \to 0$, if $\xi_i > 0$. This latter property means that $\ln(\xi_i)$ serves as a barrier, discouraging ξ_i from going to 0.

3. To write the first-order necessary conditions for a minimum, we set the partial derivatives of the Lagrangian

$$L(\mathbf{y}, \lambda, \xi) = -\lambda^T\mathbf{b} + \mu\sum_{i=1}^{n}\ln(\xi_i) - \mathbf{y}^T\left(M^T\lambda + \xi - \mathbf{d}\right)$$

---

2. (LP) is infeasible and (D) is unbounded and feasible.
3. (LP) is unbounded and feasible and (D) is infeasible.
4. (LP) is infeasible and (D) is infeasible.

This can be proved using the *Farkas' Lemma*.

with respect to $\mathbf{y}$, λ, ξ to zero. This results in the set of following three equations:

$$\begin{aligned} &M^T\lambda + \xi = \mathbf{d} \\ &M\mathbf{y} = -\mathbf{b} \\ &\text{diag}(\xi)\,\text{diag}(\mathbf{y})\mathbf{1} = \mu\mathbf{1} \end{aligned} \tag{4.116}$$

which include the dual and primal feasibility conditions excluding $\mathbf{y} \geq \mathbf{0}$ and $\xi \geq \mathbf{0}$.

4. We will assume that our current point $\mathbf{y}^{(k)}$ is primal feasible and the current point $(\lambda^{(k)}, \xi^{(k)})$ is dual feasible. We determine a new search direction $\left(\Delta\mathbf{y}^{(k)}, \Delta\lambda^{(k)}, \Delta\xi^{(k)}\right)$ so that the new point $\left(\mathbf{y}^{(k)} + \Delta\mathbf{y}^{(k)}, \lambda^{(k)} + \Delta\lambda^{(k)}, \xi^{(k)} + \Delta\xi)^{(k)}\right)$ satisfies (4.116). This gives us the so-called Newton equations:

$$\begin{aligned} &M^T\Delta\lambda + \Delta\xi = \mathbf{0} \\ &M\Delta\mathbf{y} = \mathbf{0} \\ &(y_i + \Delta y_i)(\xi_i + \Delta\xi_i) = \mu \quad i = 1, 2, \ldots, n \end{aligned} \tag{4.117}$$

Ignoring the second order term $\Delta y_i \Delta\xi_i$ in the third equation, and solving the system of equations in (4.117), we get the following update rules:

$$\begin{aligned} \Delta\lambda^{(k)} &= -\left(M\,\text{diag}(\mathbf{y}^{(k)})\,\text{diag}(\xi^{(k)})^{-1}M^T\right)^{-1} M\,\text{diag}(\xi^{(k)})^{-1}\left(\mu\mathbf{1} - \text{diag}(\mathbf{y}^{(k)})\,\text{diag}(\xi^{(k)})\mathbf{1}\right) \\ \Delta\xi^{(k)} &= -M^T\Delta\lambda^{(k)} \\ \Delta\mathbf{y}^{(k)} &= \text{diag}(\xi^{(k)})^{-1}\left(\mu\mathbf{1} - \text{diag}(\mathbf{y}^{(k)})\,\text{diag}(\xi^{(k)})\mathbf{1}\right) - \text{diag}(\mathbf{y}^{(k)})\,\text{diag}(\xi^{(k)})^{-1}\Delta\xi^{(k)} \end{aligned} \tag{4.118}$$

An affine variant of the algorithm can be developed by setting $\mu = 0$ in the equations (4.118).

5. $\Delta\mathbf{y}^{(k)}$ and $(\Delta\lambda^{(k)}, \Delta\xi^{(k)})$ correspond to partially constrained Newton steps, which might not honour the constraints $\mathbf{y}^{(k)} + \Delta\mathbf{y}^{(k)} \geq \mathbf{0}$ and $\xi^{(k)} + \Delta\xi^{(k)} \geq \mathbf{0}$. Since we have a separate search direction $\Delta\mathbf{y}^{(k)}$ in the primal space and a separate search direction $(\Delta\lambda^{(k)}, \Delta\xi^{(k)})$ in the dual space, we could compute the maximum step length $t^{(k)}_{(max,P)}$ that maintains the pending primal inequality $\mathbf{y}^{(k)} + t^{(k)}_{(max,P)}\Delta\mathbf{y}^{(k)} \geq \mathbf{0}$ and the maximum step length $t^{(k)}_{(max,D)}$ that maintains the pending dual inequality $\xi^{(k)} + t^{(k)}_{(max,D)}\Delta\xi^{(k)} \geq \mathbf{0}$.

6. Now we have a feasible primal solution $\mathbf{y}^{(k+1)}$ and feasible dual solution $(\lambda^{(k+1)}, \xi^{(k+1)})$ given by

$$\begin{aligned}\mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + t^{(k)}_{(max,P)}\Delta\mathbf{y}^{(k)} \\ \lambda^{(k+1)} &= \lambda^{(k)} + \Delta\lambda^{(k)} \\ \xi^{(k+1)} &= \xi^{(k)} + t^{(k)}_{(max,D)}\Delta\xi^{(k)}\end{aligned} \tag{4.119}$$

7. For user specified small thresholds of $\epsilon_1 > 0$ and $\epsilon_2 > 0$, if the duality gap $\mathbf{d}^T\mathbf{y}^{(k+1)} + \mathbf{b}^T\lambda^{(k+1)}$ is not sufficiently close to 0, *i.e.*,

   $$\mathbf{d}^T\mathbf{y}^{(k+1)} + \mathbf{b}^T\lambda^{(k+1)} > \epsilon_1$$

   for a μ not yet sufficiently close to 0, *i.e.*,

   $$\mu > \epsilon_2$$

   we decrease μ by a user specified factor $\rho < 1$ (such as $\rho = 0.1$).

   $$\mu = \mu \times \rho$$

8. Set $k = k + 1$. If μ was not modified in step (7), the duality gap is sufficiently small and the termination condition has been reached. So EXIT. Else, the last condition in (4.116) no longer holds with the modified value of μ. So steps (4)-(7) are re-executed.

In practice, the interior point method for LP gets down the duality gap to within 10^{-8} in just 20-80 steps (which is still slower than the simplex method for many problems), independent of the size of the problem specified through values of m and n.

4.8 Least Squares

Least squares was motivated in Section 3.9.2, based on the idea of projection. Least squares problems appear very frequently in practice. The objective for minimization in the case of least squares is the square of the eucledian norm of $A\mathbf{x} - \mathbf{b}$, where A is a $m \times n$ matrix, $\mathbf{x}$ is a vector of n variables and $\mathbf{b}$ is a vector of m knowns.

$$\min_{\mathbf{x}\in\Re^n} \quad ||A\mathbf{x} - \mathbf{b}||_2^2 \tag{4.120}$$

Very often one has a system of linear constraints on problem (4.120).

$$\begin{aligned}\min_{\mathbf{x}\in\Re^n} \quad & ||A\mathbf{x} - \mathbf{b}||_2^2 \\ \text{subject to} \quad & C^T\mathbf{x} = \mathbf{0}\end{aligned} \tag{4.121}$$

This problem is called the *least squares problem with linear constraints.*

In practice, incorporating the constraints $C^T\mathbf{x} = \mathbf{0}$ properlymakes quite a difference. In lots of regularization problems, the least squares problem often comes with quadratic constraints in the following form.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & ||A\mathbf{x}-\mathbf{b}||_2^2 \\ \text{subject to} & ||\mathbf{x}||_2^2 = \alpha^2 \end{array} \tag{4.122}$$

This problem is termed as the *least squares problem with quadratic constraints.*

The classical statistical model assumes that all the error occurs in the vector $\mathbf{b}$. But sometimes, the data matrix A is itself not very well known, owing to errors in the variables. This is the model we have in the simplest version of the *total least squares problem*, which is stated as follows.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n, E\in\Re^{m\times n}, \mathbf{r}\in\Re^m} & ||E||_F^2 + ||\mathbf{r}||_2^2 \\ \text{subject to} & (A+E)\mathbf{x} = \mathbf{b}+\mathbf{r} \end{array} \tag{4.123}$$

While there is always a solution to the least squares problem (4.120), there is not always a solution to the total least squares problem (4.133). Finally, you can have a combination of linear and quadratic constraints in a least squares problem to yield a *least squares problem with linear and quadratic constraints.*

We will briefly discuss the problem of solving linear least squares problems and total least squares problems with linear or a quadratic constraint (due to regularization) The importance of lagrange multipliers will be introduced in the process. We will discuss stable numerical methods when the data matrix A is singular or near singular. We will also present iterative methods for large and sparse data matrices. There are many applications of least squares problems, which include statistical methods, image processing, data interpolation and surface fitting and finally geometrical problems.

4.8.1 Linear Least Squares

As a user of least squares in practice, one of the most important things to be known is that when A is of full column rank, it has an analytical solution given by $\mathbf{x}^*$ (which was derived in Section 3.9.2 and gives a dual interpretation).

$$\text{Analytical solution:} \quad \mathbf{x}^* = (A^TA)^{-1}A^T\mathbf{b} \tag{4.124}$$

This analytic solution can also be obtained by observing that

1. $||\mathbf{y}||_2^2$ is a convex function for $\mathbf{y} \in \Re^m$.

2. Square of the convex eucledian norm function, applied to an affine transform is also convex. Thus $||A\mathbf{x} - \mathbf{b}||_2^2$ is convex.

3. Every critical point of a convex function defined on an open domain corresponds to its local minimum. The critical point $\mathbf{x}^*$ of $||A\mathbf{x} - \mathbf{b}||_2^2$ should satisfy

$$\nabla(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = 2A^TA\mathbf{x}^* - 2A^T\mathbf{b} = \mathbf{0}$$

Thus,

$$\mathbf{x}^* = (A^TA)^{-1}A^T\mathbf{b}$$

corresponds to a point of local minumum of (4.120) if A^TA is invertible.

This is the classical way statisticians solve least squares problem. It can be solved very efficiently, and there exist many softwares that implement this solution. The computation time is linear in the number of rows of A and quadratic in the number of columns. For extremely large A, it can become important to look at the structure of A to solve it efficiently, but for most problems, it is efficient. In practice least-squares is very easy to recognize as an objective function. There are a few standard tricks to increase the flexibility. For example, constraints can be handled to a certain extent by adding weights. When the matrix A is not full column rank, the solution to (4.120) may not be unique.

We should note that while we get a closed form solution to the problem of minimizing the square of the eucledian norm, it is not so for most other norms such as the infinity norm. However, there exist iterative methods for solving least squares with infinity norm that yield a solution in as much time as is taken in computing the solution using the analytical formula in 4.124. Therefore, having a closed form solution is not always computationally helpful. In general, the method of solution to a least squares problem depends on the sparsity as well as the size of A and the degree of accuracy desired.

In practice, however, it is not recommended to solve least squares problem using the classical equation in 4.124 since the method is numerically unstable. Numerical linear algebra instead recommends the QR decomposition to accurately solve the least squares problem. This method is slower, but more numerically stable than the classical method. In theorem **??**, we state a theory that compares the analytical solution (4.124) and the QR approach to the least squares problem.

Let A be an $m \times n$ matrix of either full row or full column rank. For the case of $n > m$, we saw on page **??** (summarised in Figure 3.3) that the system $A\mathbf{x} = \mathbf{b}$ will have at least one solution which means that minimum value of the objective function will be 0, corresponding to the solution. We are interested in the case $m \geq n$, for which there will either be no solution or a single solution to $A\mathbf{x} = \mathbf{b}$ and we are interested in one that minimizes $||A\mathbf{x} - \mathbf{b}||_2^2$.

1. We first decompose A into the product of an orthonormal $m \times m$ matrix Q with an upper traingular $m \times n$ matrix R, using the gram-schmidt or-

thonormalization process[42] discussed in Section 3.9.4. The decomposition can also be performed using the Householder[43] transformation or Givens rotation. Householder transformation has the added advantage that new rows or columns can be introduced without requiring a complete redo of the decomposition process. The last $m-n$ rows of R will be zero rows. Since $Q^{-1} = Q^T$, the QR decomposition yields the system

$$Q^T A = \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix}$$

2. Applying the same orthogonal matrix to $\mathbf{b}$, we get

$$Q^T \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$

where $\mathbf{d} \in \Re^{m-n}$.

3. The solution to the least squares problem is found by solving $R_1\mathbf{x} = \mathbf{c}$. The solution to this can be found by simple back-substitution.

The next theorem examines how the least squares solution and its residual $||A\mathbf{x} - \mathbf{b}||$ are affected by changes in A and $\mathbf{b}$. Before stating the theorem, we will introduce the concept of the condition number.

Condition Number

The *condition number* associated with a problem is a measure of how numerically well-posed the problem is. A problem with a low condition number is said to be well-conditioned, while a problem with a high condition number is said to be ill-conditioned. For a linear system $A\mathbf{x} = \mathbf{b}$, the condition number is defined as maximum ratio of the relative error in $\mathbf{x}$ (measured using any particular norm) divided by the relative error in $\mathbf{b}$. It can be proved (using the Cauchy Shwarz inequality) that the condition number equals $||A^{-1}A||$ and is independent of $\mathbf{b}$. It is denoted by $\kappa(A)$ and is also called the condition number of the matrix A.

$$\kappa(A) = ||A^{-1}A||$$

If $||.||_2$ is the L_2 norm, then

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} = ||A||_2||(A^TA)^{-1}A^T||_2$$

[42]The classical Gram-Schmidt method is often numerically unstable. Golub [**?**] suggests a modified Gram-Schmidt method that is numerically stable.

[43]Householder was a numerical analyst. However, the first mention of the Householder transformation dates back to the 1930s in a book by Aikins, a statistician and a numerical analyst.

where $\sigma_{max}(A)$ and $\sigma_{min}(A)$ are maximal and minimal singular values of A respectively. For a real square matrix A, the square roots of the eigenvalues of A^TA, are called singular values. Further,

$$\kappa(A)^2 = ||A||_2^2||(A^TA)^{-1}||_2^2$$

Theorem 86 *By $||.||$, we will refer to the L_2 norm. Let*

$$\mathbf{x}^* = \text{argmin } ||A\mathbf{x} - \mathbf{b}||$$

$$\widehat{\mathbf{x}} = \text{argmin } ||(A + \delta A)\mathbf{x} - (\mathbf{b} + \delta\mathbf{b}||$$

where A and δA are in $\Re^{m\times n}$ with $m \geq n$. Let $\mathbf{b}$ and $\delta\mathbf{b}$ be in $\Re^m$ with $\mathbf{b} \neq \mathbf{0}$. Let us set

$$\mathbf{r}^* = \mathbf{b} - A\mathbf{x}^*$$

$$\widehat{\mathbf{r}} = \mathbf{b} - A\widehat{\mathbf{x}}$$

and

$$\rho^* = ||A\mathbf{x}^* - \mathbf{b}||$$

If

$$\epsilon = max\left\{\frac{||\delta A||}{||A||}, \frac{\delta\mathbf{b}}{\mathbf{b}}\right\} < \frac{\sigma_n(A)}{\sigma_1(A)}$$

and

$$\sin\theta = \frac{\rho^*}{||\mathbf{b}||} \neq 1$$

then,

$$\frac{||\widehat{\mathbf{x}} - \mathbf{x}^*||}{||\mathbf{x}^*||} \leq \epsilon\left\{\frac{2\kappa(A)}{\cos\theta} + \tan\theta\kappa(A)^2\right\} + O(\epsilon^2)$$

In this inequality most critical term for our discussion is $\kappa(A)^2$ and this is the term that can kill the analytical solution to least squares. Now matter how accurate an algorithm you use, you still have $\kappa(A)^2$, provided $\tan\theta$ is non-zero. Now $\tan\theta$ does not appear if you are solving a linear system, but if you solve a least squares problem this term appears, bringing along $\kappa(A)^2$. Thus, solving least squares problem is inherently more difficult and sensitive than linear equations. The perturbation theory for the residual vector depends just on the condition number $\kappa(A)$ (and not its square):

$$\frac{||\widehat{\mathbf{r}} - \mathbf{r}^*||}{||\mathbf{b}||} \leq \epsilon\left\{1 + 2\kappa(A)\right\} min\left\{1, m - n\right\} + O(\epsilon^2) + O(\epsilon^2)$$

However, having a small residual does not necessarily imply that you will have a good approximate solution.

The theorem implies that the sensitivity of the analytical solution $\mathbf{x}^*$ for non-zero residual problems is measured by the square of the condition number. Whereas, sensitivity of the residual depends just linearly on $\kappa(A)$. We note that the QR method actually solves a nearby least squares problem.

Linear Least Squares for Singular Systems

To solve the linear least squares problem (4.120) for a matrix A that is of rank $r < min\{m, n\}$, we can compute the pseudo-inverse (*c.f.* page 211) A^+ and obtain the least squares solution[44] as

$$\widehat{\mathbf{x}} = A^+\mathbf{b}$$

A^+ can be computed by first computing a singular orthogonal factorization

$$A = Q \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Z^T$$

where $Q^TQ = I_{m\times m}$ and $Z^TZ = I_{n\times n}$ and R is an $r \times r$ upper traingular matrix. A^+ can be computed in a straightforward manner as

$$A^+ = Z \begin{bmatrix} R^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

The above least squares solution can be justified as follows. Let

$$Q^T\mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$

and

$$Z^T\mathbf{x} = \begin{bmatrix} \mathbf{w} \\ \mathbf{y} \end{bmatrix}$$

Then

$$||A\mathbf{x} - \mathbf{b}||^2 = ||Q^TAZZ^T\mathbf{x} - Q^T\mathbf{b}||^2 = ||R\mathbf{w} - \mathbf{c}||^2 + ||\mathbf{d}||^2$$

The least squares solution is therefore given by

$$\widehat{\mathbf{x}} = Z \begin{bmatrix} R^{-1}\mathbf{c} \\ 0 \end{bmatrix}$$

One particular decomposition that can be used is the singular value decomposition (*c.f.* Section 3.13) of A, with $Q^T \equiv U^T$ and $Z \equiv V$ and $U^TAV = \Sigma$. The pseudo-inverse A^+ has the following expression.

$$A^+ = V\Sigma^{-1}U^T$$

It can be shown that this A^+ is the unique minimal Frobenius norm solution to the following problem.

$$A^+ = \underset{X\in\Re^{n\times m}}{\text{argmin}} \; ||AX - I_{m\times m}||$$

[44] Note that this solution not only minimizes $||A\mathbf{x} - \mathbf{b}||$ but also minimizes $||\mathbf{x}||$. This may or may not be desirable.

This also shows that singular value decomposition can be looked upon as an optimization problem.

A greater problem is with systems that are nearly singular. Numerically and computationally it seldom happens that the rank of matrix is exactly r. A classical example is the following $n \times n$ matrix K, which has a determinant of 1.

$$K = \begin{bmatrix} 1 & -1 & \dots & -1 & -1 & \dots & -1 \\ 0 & 1 & \dots & -1 & -1 & \dots & -1 \\ . & . & \dots & . & . & \dots & . \\ . & . & \dots & . & . & \dots & . \\ 0 & 0 & \dots & 1 & -1 & \dots & -1 \\ 0 & 0 & \dots & 0 & 1 & \dots & -1 \\ . & . & \dots & . & . & \dots & . \\ . & . & \dots & . & . & \dots & . \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix}$$

The eigenvalues of this matrix are also equal to 1, while its rank is n. However, a very small perturbation to this matrix can reduce its rank to $n-1$; the rank of $K - 2^{-(n-1)} I_{n \times n}$ is $n-1$! Such catastrophic problems occur very often when you do large computations. The solution using SVD is applicable for nearly singular systems as well.

4.8.2 Least Squares with Linear Constraints

We first reproduce the least squares problem with linear constraints that was stated earlier in (4.121).

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \Re^n} & ||A\mathbf{x} - \mathbf{b}||^2 \\ \text{subject to} & C^T \mathbf{x} = \mathbf{0} \end{array}$$

Let $C \in \Re^{n \times p}$, $A \in \Re^{m \times n}$ and $\mathbf{b} \in \Re^m$. We note that $||A\mathbf{x} - \mathbf{b}||^2$ is a convex function (since L_2 norm is convex and this function is the L_2 norm applied to an affine transform). We can thus solve this constrained problem by invoking the necessary and sufficient KKT conditions discussed in Section 4.4.4. The conditions can be worked out to yield

$$\begin{bmatrix} A^T A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} A^T \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

We need to now solve not only for the unknowns $\mathbf{x}$, but also for the lagrange multipliers; we have increased the dimensionality of the problem to $n+p$. If $\widehat{\mathbf{x}} =$

$(A^TA)^{-1}A^T\mathbf{b}$ denotes the solution of the unconstrained least squares problem, then, using the first system of equality above, $\mathbf{x}$ can be expressed as

$$\mathbf{x} = \widehat{\mathbf{x}} - (A^TA)^{-1}C\lambda \tag{4.125}$$

In conjunction with the second system, this leads to

$$C^T(A^TA)^{-1}C\lambda = C^T\widehat{\mathbf{x}} \tag{4.126}$$

The unconstrained least squares solution can be obtained using methods in Section 4.8.1. Next, the value of λ can be obtained by solving (4.126). If A is singular or nearly singular, we can use the singular value decomposition (or a similar decomposition) of A to determine $\widehat{\mathbf{x}}$.

$$C^TR^{-1}(R^T)^{-1}C\lambda = C^T\widehat{\mathbf{x}}$$

The QR factorization of $(R^T)^{-1}C$ can be efficiently used to determine λ. Finally, the value of λ can be substituted in (4.125) to solve for $\mathbf{x}$. This technique yields both the solutions, provided that both exist.

Another trick that is often employed when A^TA is singular or nearly singular is to decrease its condition number by augmenting it in (4.125) with the 'harmless' CWC^T and solve

$$\mathbf{x} = \widehat{\mathbf{x}} - (A^TA + CWC^T)^{-1}C\lambda$$

The addition of CWC^T is considered harmless, since $C^T\mathbf{x} = 0$ is to be imposed anyways. Matrix W can be chosen to be an identical or nearly identical matrix that chooses a few columns of C, just to make $A^TA + CWC^T$ non-singular.

If we use the following notation:

$$\mathcal{A}(W) = \begin{bmatrix} A^TA + CWC^T & C \\ C^T & 0 \end{bmatrix}$$

and

$$\mathcal{A} = \mathcal{A}(0) = \begin{bmatrix} A^TA & C \\ C^T & 0 \end{bmatrix}$$

and if $\mathcal{A}$ and $\mathcal{A}(W)$ are invertible for $W \neq 0$, it can be proved that

$$\mathcal{A}^{-1}(W) = \mathcal{A}^{-1} - \begin{bmatrix} 0 & 0 \\ 0 & W \end{bmatrix}$$

Consequently

$$\kappa(\mathcal{A}(W)) \leq \kappa(\mathcal{A}) + ||W||^2||C||^2 + \alpha||W||$$

for some $\alpha > 0$. That is, the condition number of $\mathcal{A}(W)$ is bounded by the condition number of $\mathcal{A}$ and some positive terms.

Another useful technique is to find an approximation to (4.121) by solving the following weighted unconstrained minimization problem.

$$\min_{\mathbf{x}\in\Re^n} \quad ||A\mathbf{x}-\mathbf{b}||^2 + \mu^2||C^T\mathbf{x}||^2$$

For large values of μ, the solution $\widehat{\mathbf{x}}(\mu)$ of the unconstrained problem is a good appproximation to the solution $\widehat{\mathbf{x}}$ of the constrained problem (4.121). We can use the generalized singular value decompositions of matrices A and C^T, that allows us to simultaneously diagonalize A and C^T.

$$U^TAX = \mathbf{d}iag(\alpha_1,\ldots,\alpha_m)$$

$$V^TC^TX = \mathbf{d}iag(\gamma_1,\ldots,\gamma_m)$$

where U and V are orthogonal matrices and X is some general matrix. The solution to the constrained problem can be expressed as

$$\widehat{\mathbf{x}} = \sum_{i=1}^{p} \frac{\mathbf{u}_i^T\mathbf{b}}{\alpha_i}\mathbf{x}_i$$

The analytical solution $\widehat{\mathbf{x}}(\mu)$ is then given as

$$\widehat{\mathbf{x}}(\mu) = \sum_{i=1}^{p} \frac{\alpha_i\mathbf{u}_i^T\mathbf{b}}{\alpha_i^2+\mu^2\gamma_i^2}\mathbf{x}_i + \widehat{\mathbf{x}}$$

It can be easily seen that as $\mu^2 \to \infty$, $\widehat{\mathbf{x}}(\mu) \to \widehat{\mathbf{x}}$.

Generally, if possible, it is better to eliminate the constraints, since this makes the problem better conditioned. We will discuss one final approach to solving the linearly constrained least squares problem (4.121), which reduces the dimensionality of the problem by eliminating the constraints. It is hinged on computing the QR factorization of C.

$$Q^TC = \begin{pmatrix} R \\ 0 \end{pmatrix} \begin{matrix} p \\ n-p \end{matrix} \tag{4.127}$$

This yields

$$AQ^T = \begin{pmatrix} A_1 & A_2 \end{pmatrix} \tag{4.128}$$

and

$$Q^T\mathbf{x} = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \begin{matrix} p \\ n-p \end{matrix} \tag{4.129}$$

The constrained problem then becomes

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & ||\mathbf{b} - A_1\mathbf{y} - A_2\mathbf{z}||^2 \\ \text{subject to} & R^T\mathbf{y} = \mathbf{0} \end{array}$$

Since R is invertible, we must have $\mathbf{y} = \mathbf{0}$. Thus, the solution $\widehat{\mathbf{x}}$ to the constrained least squares problem can be determined as

$$\widehat{\mathbf{x}} = Q^T \begin{pmatrix} \mathbf{0} \\ \widehat{\mathbf{z}} \end{pmatrix} \tag{4.130}$$

where

$$\widehat{\mathbf{z}} = \underset{\mathbf{z}}{\text{argmax}}\ ||\mathbf{b} - A_2\mathbf{z}||^2$$

It can be proved that the matrix A_2 is atleast as well-conditioned as the matrix A. Often, the original problem is singular and imposing the constraints makes it non-singular (and is reflected in a non-singular matrix A_2.

4.8.3 Least Squares with Quadratic Constraints

The quadratically constrained least squares problem is often encountered in regularization problems and can be stated as follows.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n} & ||A\mathbf{x} - \mathbf{b}||_2^2 \\ \text{subject to} & ||\mathbf{x}||_2^2 = \alpha^2 \end{array}$$

Since the objective function as well as the constraint function are convex, the KKT conditions (*c.f.* Section 4.4.4) are necessary and sufficient conditions for the optimality of the problem at the primal-dual variable pair given by $(\widehat{\mathbf{x}}, \widehat{\mu})$. The KKT conditions lead to the following equations

$$(A^TA + \mu I)\mathbf{x} = A^T\mathbf{b} \tag{4.131}$$

$$\mathbf{x}^T\mathbf{x} = \alpha^2 \tag{4.132}$$

The expression in (4.131) is the solution to what the statisticians sometimes refer to as the ridge regression problem. The solution to the problem under consideration has the additional constraint though, that the norm of the solution vector $\widehat{\mathbf{x}}$ should equal $|\alpha|$. The two equations above yield the so-called *secular equation* stated below.

$$\mathbf{b}^TA(A^TA + \mu I)^{-2}A^T\mathbf{b} - \alpha^2 = 0$$

Further, the matrix A can be diagonalized using its singular value decomposition $A = U\Sigma V^T$ to obtain the following equation which is to be solved.

$$\sum_{i=1}^{n} \beta_i^2 \frac{\sigma_i^2}{(\sigma_i^2 + \mu)^2} - \alpha^2 = 0$$

4.8.4 Total Least Squares

The total least squares problem is stated as

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\Re^n, E\in\Re^{m\times n}, \mathbf{r}\in\Re^m} & ||E||_F^2 + ||\mathbf{r}||_2^2 \\ \text{subject to} & (A+E)\mathbf{x} = \mathbf{b} + \mathbf{r} \end{array}$$