CHAPTER 4: OPTIMIZATION

1 Overview

Optimization is the process of maximizing or minimizing an objective function, subject to constraints. Optimization is fundamental to nearly all academic fields, and its utility extends beyond energy systems and control.

As energy systems engineers, we are often faced with *design* tasks. Here are some examples:

- Determine the optimal battery energy storage capacity for a wind farm [1].
- Optimally place x–MW of photovoltaic generation capacity in Nicaragua to minimize economic cost, yet meet consumer demand.
- Optimally place N vehicle sharing stations in an urban environment to optimally meet user demand [2, 3].
- Optimally re-distribute shared vehicles in a vehicle sharing system to minimize maintenance cost, yet serve user demand [4].
- Design an optimal plug-in electric vehicle charging strategy to minimize economic cost, yet meet mobility demands and limit battery degradation [5].
- Optimally manage energy flow in a smart home, containing photovoltaics, battery storage, and a plug-in electric vehicle, to minimize economic cost [6].
- Optimally manage water flow in Barcelona's water network to meet user demand, maintain sufficient quality, and minimize economic cost [7].

The number of interesting optimization examples in energy systems engineering is limited only by your creativity. These problems often involve physical first principles of an energy storage/conversion device, user demand (a human element), and market economics. In addition, they are almost always too complex to solve with pure intuition. Consequently, one desires a systematic process for designing energy systems that optimizes some metric of interest (e.g. economic cost or performance), while satisfying some necessary constraints (e.g. user demand, physical operating limits, regulations).

1.1 Canonical Form

Nearly all static optimization problems can be abstracted into the following canonical form:

$$\min_x \quad f(x), \tag{1}$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \cdots, m \tag{2}$$

$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{3}$$

In this formulation, $x \in \mathbb{R}^n$ is a vector of n decision variables. The function $f(x)$ is known as the "cost function" or "objective function," and maps the decision variables x into a scalar objective function value, mathematically given by $f(x) : \mathbb{R}^n \to \mathbb{R}$. Equation (2) represents m inequality constraints. Each function $g_i(x)$ maps the decision variable to a scalar that must be non-positive for constraint satisfaction, mathematically $g_i(x) : \mathbb{R}^n \to \mathbb{R}, \forall i = 1, \cdots, m$. Similarly, equation (3) represents l equality constraints. Each function $h_i(x)$ maps the decision variable to a scalar that must be zero for constraint satisfaction, mathematically $h_j(x) : \mathbb{R}^n \to \mathbb{R}, \forall j = 1, \cdots, l$. We often vectorize the functions $g_i(x)$ and $h_j(x)$ as $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ and $h(x) : \mathbb{R}^n \to \mathbb{R}^l$, respectively, to more compactly write the canonical form as:

$$\min_x \quad f(x) \tag{4}$$

$$\text{subject to} \quad g(x) \leq 0, \tag{5}$$

$$h(x) = 0. \tag{6}$$

Remark 1.1. Note that one can always re-formulate a maximization problem, e.g. $\max_x f(x)$ into a minimization problem by defining $\overline{f}(x) = -f(x)$ and solving $\min_x \overline{f}(x)$.

Given this problem, several questions or issues arise:

1. What, exactly, is the definition of a minimum?
2. Does a solution even exist?
3. Is the minimum unique?
4. What are the necessary and sufficient conditions to be a minimum?
5. How do we solve the optimization problem?

Throughout this chapter we shall investigate these questions within the context of various optimization problem formats.

1.2 Chapter Organization

The remainder of this chapter is organized as follows:

1. Mathematical Preliminaries
2. Linear Programs (LP)

3. Quadratic Programs (QP)
4. Nonlinear Programming (NLP)
5. Convex Programming (CP)

2 Mathematical Preliminaries

Our exposition of optimization begins with some useful mathematical concepts. This background will provide the necessary foundation to discuss the theory and algorithms underlying various optimization problems. The first two concepts are *convex sets* and *convex functions*.

2.1 Convex Sets

Definition 2.1 (Convex Set)**.** *Let D be a subset of $\mathbb{R}^n$. Also, consider scalar parameter $\lambda \in [0,1]$ and two points $a, b \in D$. The set D is convex if*

$$\lambda a + (1-\lambda)b \in D \tag{7}$$

for all points $a, b \in D$.

Figure 1 provides visualizations of convex and non-convex sets. In words, a set is convex if a line segment connecting any two points within domain D is completely within the set D. Figure 2 provides additional examples of convex and non-convex sets.

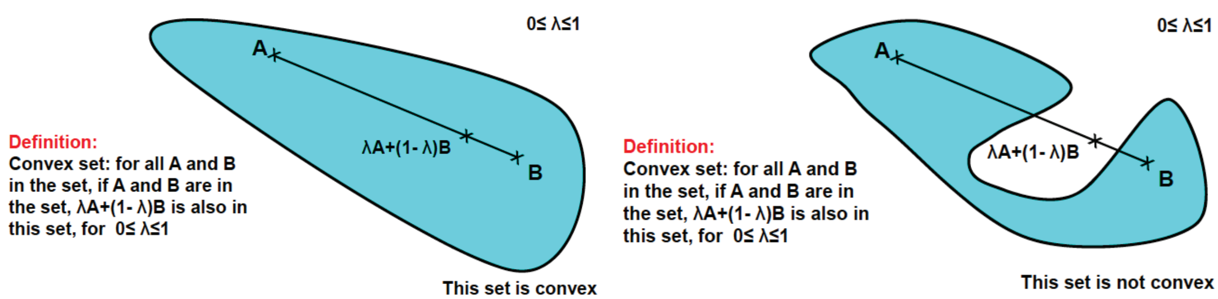


Figure 1: Visualization of convex [left] and non-convex [right] sets.

2.1.1 Examples

The following are some important examples of convex sets you will encounter in design optimization:

- The empty set , any single point (i.e. a singleton), $\{x_0\}$, and the whole space $\mathbb{R}^n$ are convex.

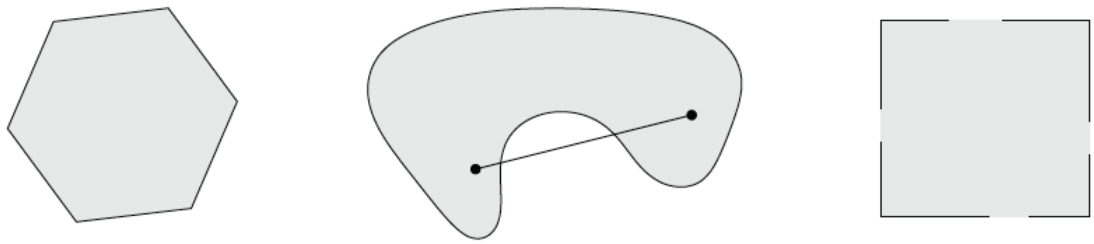


Figure 2: Some simple convex and nonconvex sets. [Left] The hexagon, which includes its boundary (shown darker), is convex. [Middle] The kidney shaped set is not convex, since the line segment between the two points in the set shown as dots is not contained in the set. [Right] The square contains some boundary points but not others, and is not convex.

- Any line in R^n is convex.
- Any line segment in R^n is convex.
- A ray, which has the form $\{x_0 + \theta v \mid \theta \geq 0, v \neq 0\}$ is convex.

Remark 2.1. An interesting property of convex sets is that any convex set can be well-approximated by a linear matrix inequality. That is, any convex set $\mathcal{D}$ can be approximated by a set of linear inequalities, written in compact form as $Ax \leq b$. We call the feasible set given by $Ax \leq b$ a polyhedron, since it represents the intersection of a finite number of half-spaces, as seen in Chapter 1. As the number of linear inequalities goes to infinity, the approximation error for a general convex set goes to zero.

The converse is not true. Any set of linear inequalities, written compactly as $Ax \leq b$, does not necessarily represent a convex set. For example, $x \leq 0$ and $x \geq 1$ produces a non-convex set.

Exercise 1. *Which of the following sets are convex? Draw each set for the two-dimensional case, $n = 2$.*

(a) A box, i.e., a set of the form $\{x \in \mathbb{R}^n \mid \alpha_i \leq x_i \leq \beta_i, i = 1, \cdots, n\}$.

(b) A slab, i.e., a set of the form $\left\{x \in \mathbb{R}^n \mid \alpha \leq a^T x \leq \beta\right\}$.

(c) A wedge, i.e., $\left\{x \in \mathbb{R}^n \mid a_1^T x \leq b_1, a_2^T x \leq b_2\right\}$.

(d) The union of two convex sets, that is $\mathcal{D}_1 \cup \mathcal{D}_2$, where $\mathcal{D}_1, \mathcal{D}_2$ are convex sets.

(e) The intersection of two convex sets, that is $\mathcal{D}_1 \cap \mathcal{D}_2$, where $\mathcal{D}_1, \mathcal{D}_2$ are convex sets.

Exercise 2 (Voronoi description of halfspace, [8] p. 60)**.** *Let a and b be distinct points in $\mathbb{R}^n$. Show that the set of all points that are closer (in Euclidean norm) to a than b, i.e., $\{x \mid \|x - a\|_2 \leq \|x - b\|_2\}$, is a half-space. Describe it explicitly as an inequality of the form $c^T x \leq d$. Draw a picture.*

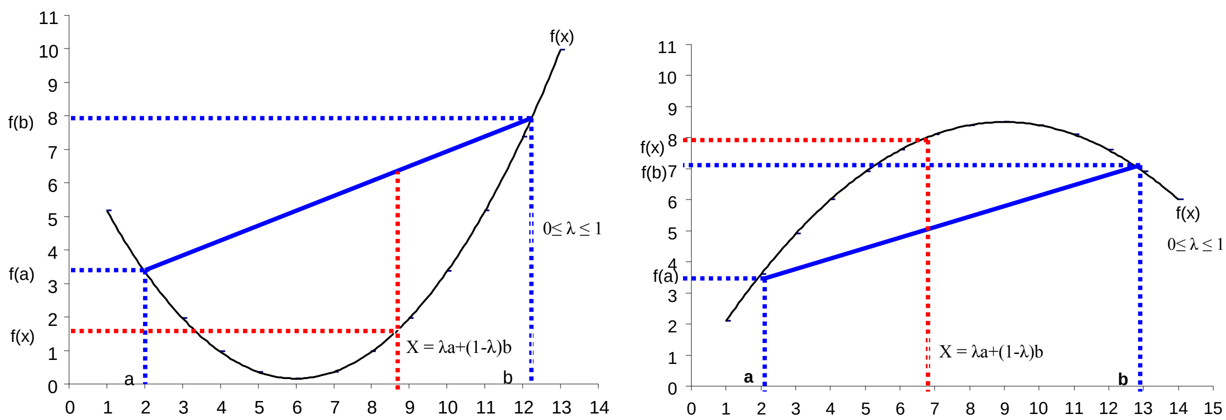


Figure 3: Visualization of convex [left] and concave [right] function definitions.

2.2 Convex Functions

Definition 2.2 (Convex Function)**.** *Let D be a convex set. Also, consider scalar parameter $\lambda \in [0, 1]$ and two points $a, b \in D$. Then the function $f(x)$ is* <ins>*convex*</ins> *on D if*

$$f(x) = f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda)f(b) \tag{8}$$

for all points $a, b \in D$.

Definition 2.3 (Concave Function)**.** *Let D be a convex set. Also, consider scalar parameter $\lambda \in [0, 1]$ and two points $a, b \in D$. Then the function $f(x)$ is* <ins>*concave*</ins> *on D if*

$$f(x) = f(\lambda a + (1 - \lambda)b) \geq \lambda f(a) + (1 - \lambda)f(b) \tag{9}$$

for all points $a, b \in D$.

Figure 3 provides visualizations of the definitions given above. In words, a function is convex if a line segment connecting any two points within domain D is <ins>above</ins> the function. A function is concave if a line segment connecting any two points within domain D is <ins>below</ins> the function.

Exercise 3. *Which of the following functions are convex, concave, neither, or both, over the set $D = [-10, 10]$? You may use graphical arguments or (8), (9) to prove your claim.*

(a) $f(x) = 0$

(b) $f(x) = x$

(c) $f(x) = x^2$

(d) $f(x) = -x^2$

(e) $f(x) = x^3$

(f) $f(x) = \sin(x)$

(g) $f(x) = e^{-x^2}$

(h) $f(x) = |x|$

Convex and concave functions have several useful properties, summarized by the following proposition.

Proposition 1 (Convex/Concave Function Properties)**.** Consider a function $f(x) : \mathbb{R}^n \to \mathbb{R}$ and compact set D.

1. If $f(x)$ is convex on D, then $-f(x)$ is concave on D.
2. If $f(x)$ is concave on D, then $-f(x)$ is convex on D.
3. $f(x)$ is a convex function on $D \iff \frac{d^2 f}{dx^2}(x)$ is positive semi-definite $\forall\ x\ \in D$.
4. $f(x)$ is a concave function on $D \iff \frac{d^2 f}{dx^2}(x)$ is negative semi-definite $\forall\ x\ \in D$.

2.2.1 Examples

It is easy to verify that all linear and affine functions are both convex and concave functions. Here we provide more interesting examples of convex and concave functions. First, we consider functions $f(x)$ where $x \in \mathbb{R}$ is scalar.

- *Quadratic.* $\frac{1}{2}ax^2 + bx + c$ is convex on $\mathbb{R}$, for any $a \geq 0$. It is concave on $\mathbb{R}$ for any $a \leq 0$.
- *Exponential.* e^{ax} is convex on $\mathbb{R}$, for any $a \in \mathbb{R}$.
- *Powers.* x^a is convex on the set of all positive x, when $a \geq 1$ or $a \leq 0$. It is concave for $0 \leq a \leq 1$.
- *Powers of absolute value.* $|x|^p$, for $p \geq 1$ is convex on $\mathbb{R}$.
- *Logarithm.* $\log x$ is concave on the set of all positive x.
- *Negative entropy.* $x \log x$ is convex on the set of all positive x.

Convexity or concavity of these examples can be shown by directly verifying (8), (9), or by checking that the second derivative is non-negative (degenerate of positive semi-definite) or non-positive (degenerate of negative semi-definite). For example, with $f(x) = x \log x$ we have

$$f'(x) = \log x + 1, \qquad f''(x) = 1/x,$$

so that $f''(x) \geq 0$ for $x > 0$. Therefore the negative entropy function is convex for positive x.

We now provide a few commonly used examples in the multivariable case of $f(x)$, where $x \in \mathbb{R}^n$.

- *Norms.* Every norm in $\mathbb{R}^n$ is convex.
- *Max function.* $f(x) = \max\{x_1, \cdots, x_n\}$ is convex on $\mathbb{R}^n$.

- *Quadratic-over-linear function.* The function $f(x, y) = x^2/y$ is convex over all positive x, y.
- *Log-sum-exp.* The function $f(x) = \log(\exp^{x_1} + \cdots + \exp^{x_n})$ is convex on $\mathbb{R}^n$. This function can be interpreted as a differentiable (in fact, analytic) approximation of the max function. Consequently, it is extraordinarily useful for gradient-based algorithms, such as the ones described in Section 5.1.
- *Geometric mean.* The geometric mean $f(x) = (\Pi_{i=1}^n x_i)^{1/n}$ is concave for all elements of x positive, i.e. $\{x \in \mathbb{R}^n \mid x_i > 0 \; \forall \; i = 1, \cdots, n\}$.

Convexity (or concavity) of these examples can be shown by directly verifying (8), (9), or by checking that the Hessian is positive semi-definite (or negative semi-definite). These are left as exercises for the reader.

2.2.2 Operations that conserve convexity

Next we describe operations on convex functions that preserve convexity. These operations include addition, scaling, and point-wise maximum. Often, objective functions in the optimal design of engineering system are a combination of convex functions via these operations. This section helps you analyze when the combination is convex, and how to construct new convex functions.

Linear Combinations

It is easy to verify from (8) that when $f(x)$ is a convex function, and $\alpha \geq 0$, then the function $\alpha f(x)$ is convex. Similarly, if $f_1(x)$ and $f_2(x)$ are convex functions, then their sum $f_1(x) + f_2(x)$ is a convex function. Combining non-negative scaling and addition yields a non-negative weighted sum of convex functions

$$f(x) = \alpha_1 f_1(x) + \cdots + \alpha_m f_m(x) \tag{10}$$

that is also convex.

Pointwise Maximum

If $f_1(x)$ and $f_2(x)$ are convex functions on $\mathcal{D}$, then their *point-wise maximum* f defined by

$$f(x) = \max\{f_1(x), f_2(x)\} \tag{11}$$

is convex on $\mathcal{D}$. This property can be verified via (8) by considering $0 \leq \lambda \leq 1$ and $a, b \in \mathcal{D}$.

$$\begin{aligned} f(\lambda a + (1-\lambda)b) &= \max\{f_1(\lambda a + (1-\lambda)b), f_2(\lambda a + (1-\lambda)b)\} \\ &\leq \max\{\lambda f_1(a) + (1-\lambda)f_1(b), \lambda f_2(a) + (1-\lambda)f_2(b)\} \end{aligned}$$

$$
\begin{aligned}
&\leq \lambda \max\{f_1(a), f_2(a)\} + (1-\lambda)\max\{f_1(b), f_2(b)\} \\
&= \lambda f(a) + (1-\lambda) f(b).
\end{aligned}
$$

which establishes convex of f. It is straight-forward to extend this result to show that if $f_1(x), \cdots, f_m(x)$ are convex, then their point-wise maximum

$$f(x) = \max\{f_1(x), \cdots, f_m(x)\} \tag{12}$$

is also convex.

Exercise 4 (Simple function compositions)**.** *Consider function $g(\cdot) : \mathbb{R} \to \mathbb{R}$ defined over convex set $\mathcal{D}$. Prove each of the following function compositions is convex or concave over $\mathcal{D}$.*

(a) If g is convex then then $\exp g(x)$ is convex.

(b) If g is concave and positive, then $\log g(x)$ is concave.

(c) If g is concave and positive, then $1/g(x)$ is convex.

(d) If g is convex and nonnegative and $p \geq 1$, then $g(x)^p$ is convex.

(e) If g is convex then $-\log(-g(x))$ is convex on $\{x | g(x) < 0\}$.

2.3 Definition of Minimizers

Armed with notions of convex sets and convex/concave functions, we are positioned to provide a precise definition of a minimizer, which we often denote with the "star" notation as x^*. There exist two types of minimizers: global and local minimizers. Their definitions are given as follows.

Definition 2.4 (Global Minimizer)**.** *$x^* \in D$ is a global minimizer of $f(x)$ on D if*

$$f(x^*) \leq f(x), \qquad \forall\, x \in D \tag{13}$$

In words, this means x^* minimizes $f(x)$ everywhere in D. In contrast, we have a local minimizer.

Definition 2.5 (Local Minimizer)**.** *$x^* \in D$ is a local minimizer of $f(x)$ on D if*

$$\exists\, \epsilon > 0 \quad \textit{s.t.} \quad f(x^*) \leq f(x), \qquad \forall\, x \in D \cap \{x \in \mathbb{R} \mid \|x - x^*\| < \epsilon\} \tag{14}$$

In words, this means x^* minimizes $f(x)$ locally in D. That is, there exists some neighborhood whose size is characterized by ϵ where x^* minimizes $f(x)$. Examples of global and local minimizers are provided in Fig. 4

We now have a precise definition for a minimum. However, we now seek to understand when a minimum even exists. The answer to this question leverages the convex set notion, and is called the Weierstrauss Extreme Value Theorem.

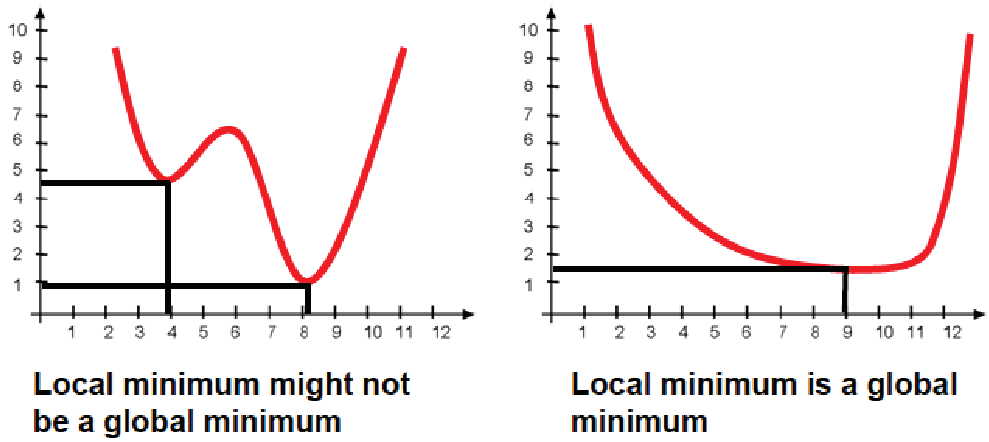


Figure 4: The LEFT figure contains two local minimizers, but only one global minimizer. The RIGHT figure contains a local minimizer, which is also the global minimizer.

Theorem 2.1 (Weierstrass Extreme Value Theorem)**.** *If $f(x)$ is continuous and bounded on a convex set D, then there exists at least one global minimum of f on D.*

A visualization of this theorem is provided in Fig. 5. In practice, the result of the Weierstrauss extreme value theorem seems obvious. However, it emphasizes the importance of having a continuous and bounded objective function $f(x)$ from (1), and constraints (2)-(3) that form a convex set. Consequently, we know a global minimizer exists if we strategically formulate optimization problems where the objective function is continuous and bounded, and the constraint set is convex.

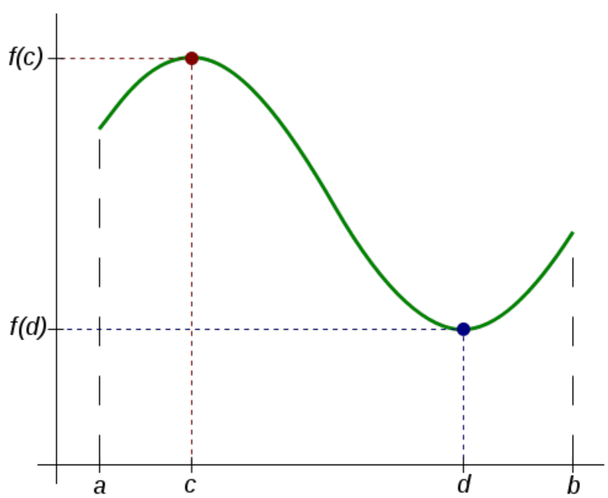


Figure 5: In this graph, $f(x)$ is continuous and bounded. The convex set is $D = [a, b]$. The function f attains a global minimum at $x = d$ and a global maximum at $x = c$.

Figure 6: A local or global minimum need not be unique.

Is the minimum unique? In general, the minimum need not be unique, as illustrated in Fig. 6. There may be two global optima or even infinite global optima. The physical interpretation is that a multitude of designs produce equally good solutions, in terms of the objective function value.

3 Linear Programming (LP)

We begin our exposition of linear programming problems with the following example.

Example 3.1 (Building a Solar Array Farm)**.** You are tasked with designing the parameters of a new photovoltaic array installation. Namely, you must decide on the square footage of the photovoltaic arrays, and the power capacity of the power electronics which interface the generators to the grid. The goal is to minimize installation costs, subject to the following constraints:

1. You cannot select negative PV array area, nor negative power electronics power capacity.
2. The minimum generating capacity for the photovoltaic array is $g_{\min}$.
3. The power capacity of the power electronics must be greater than or equal to the PV array power capacity.
4. The available spatial area for installation is limited by $s_{\max}$.
5. You have a maximum budget of $b_{\max}$.

Table 1: Building a Solar Array Farm

Spatial area of photovoltaic arrays [m^2]	x_1
Power capacity of power electronics [kW]	x_2
Cost of square meter of PV array [USD/m^2]	c_1
Cost of power electronics per kW [USD/kW]	c_2
Min PV array generating capacity [kW]	$g_{\min}$
Power of PV array per area [kW/m^2]	a_1
Max spatial area [m^2]	$s_{\max}$
Maximum budget [USD]	$b_{\max}$

Using the notation in Table 1, we can formulate the following optimization problem:

$$\begin{aligned}
\min_{x_1,x_2} \quad & c_1 x_1 + c_2 x_2 && (15)\\
\text{subject to:} \quad & x_1 \geq 0 && (16)\\
& x_2 \geq 0 && (17)\\
& a_1 x_1 \geq g_{\min} && (18)\\
& a_1 x_1 \leq x_2 && (19)\\
& x_1 \leq s_{\max} && (20)\\
& c_1 x_1 + c_2 x_2 \leq b_{\max} && (21)
\end{aligned}$$

where (16)-(17) encode constraint 1, and (18)-(21) respectively encode constraints 2-5. Rearranging all inequality constraints into less-than-or-equal-to, we arrive at the so-called standard form:

$$\begin{aligned}
\min_{x_1,x_2} \quad & c_1 x_1 + c_2 x_2 && (22)\\
\text{subject to:} \quad & -x_1 \leq 0 && (23)\\
& -x_2 \leq 0 && (24)\\
& -a_1 x_1 \leq -g_{\min} && (25)\\
& a_1 x_1 - x_2 \leq 0 && (26)\\
& x_1 \leq s_{\max} && (27)\\
& c_1 x_1 + c_2 x_2 \leq b_{\max} && (28)
\end{aligned}$$

which can be written into vector-matrix form as

$$\begin{aligned}
\min_{x} \quad & c^T x && (29)\\
\text{subject to:} \quad & Ax \leq b && (30)
\end{aligned}$$

where $x = [x_1, x_2]$ and

$$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \qquad A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -a_1 & 0 \\ a_1 & -1 \\ 1 & 0 \\ c_1 & c_2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ -g_{\min} \\ 0 \\ s_{\max} \\ b_{\max} \end{bmatrix}. \tag{31}$$

Note that the feasible set (30) is a convex set, according to Remark 2.1. Also, the objective function (29) is linear and therefore convex. Consequently, this is a convex optimization problem.

More specifically, this falls within a special subset of convex problems, called linear programs. The optimization problem (29)-(30) is known as a **linear program** (LP), and is fully characterized by matrices c, A, b. The key trait of a LP is that the objective function and constraints are linear functions of the design variables x. Now that we have successively formulated the Solar Array Farm problem as a LP, the problem is essentially solved. In Matlab, one may utilize the `linprog` command to solve the optimization problem, given matrices c, A, b.

Remark 3.1 (Program Reduction)**.** Sometimes constraints take the form of equalities. From Example 3.1, suppose the power capacity of the electronics must be 1.2 times the PV array generating capacity, i.e.

$$x_2 = 1.2\, a_1 x_1 \tag{32}$$

We can replace x_2 by the expression above and reduce the problem size:

$$\begin{aligned}
\min_{x_1} \quad & (c_1 + 1.2\, a_1 c_2)\, x_1 && (33)\\
\text{subject to:} \quad & -x_1 \leq 0 && (34)\\
& -1.2\, a_1 x_1 \leq 0 && (35)\\
& -a_1 x_1 \leq -g_{\min} && (36)\\
& -0.2 a_1 x_1 \leq 0 && (37)\\
& x_1 \leq s_{\max} && (38)\\
& (c_1 + 1.2\, a_1 c_2)\, x_1 \leq b_{\max} && (39)
\end{aligned}$$

Note that some inequalities **dominate** others. For example, constraint (36) **dominates** (34), (35), (37) in the sense that (34), (35), (37) are automatically verified when constraint (36) is true. Thus we can further reduce the problem to

$$\begin{aligned}
\min_{x_1} \quad & (c_1 + 1.2\, a_1 c_2)\, x_1 && (40)\\
\text{subject to:} \quad & -a_1 x_1 \leq -g_{\min} && (41)\\
& x_1 \leq s_{\max} && (42)\\
& (c_1 + 1.2\, a_1 c_2)\, x_1 \leq b_{\max} && (43)
\end{aligned}$$

We now formalize the notion of constraint domination with the following definition.

Definition 3.1 (Constraint Domination)**.** *Inequality constraint i dominates inequality constraint j when satisfaction of constraint j is automatically verified by constraint i.*

Continuing with our example, intuition suggests we can minimize costs by selecting the spatial area x_1 such that the minimum generating capacity is met exactly[1]. That is, at the optimal solution

[1] Naturally, this assumes the problem parameters are such that the feasible set is non-empty, that is, some solution exists.

x_1^*, constraint (36) is true with equality. This produces the minimizer $x_1^* = g_{\min}/a_1$ and the minimum installation cost $(c_1 + 1.2\, a_1\, c_2)\, g_{\min}/a_1$. Note that we denote minimizers with the star notation: x_1^*. Also, when an inequality constraint is true with equality at the optimum, we call the constraint an **active constraint**.

Definition 3.2 (Active Inequalities)**.** *Constraint i is called an active constraint when it is true with equality at the optimum.*

3.1 General Form of LP

Linear programs can be identified as having linear objective functions and linear constraints. Suppose we have N decision variables x_i, M inequality constraints, and L equality constraints. Mathematically, this takes the form

$$\begin{array}{rrcl} \min & c_1x_1 + c_2x_2 + \ldots + c_Nx_N & & \\ & & & \\ \text{subject to:} & a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,N}x_N & \leq & b_1, \\ & a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,N}x_N & \leq & b_2, \\ & & \vdots & \\ & a_{M,1}x_1 + a_{M,2}x_2 + \ldots + a_{M,N}x_N & \leq & b_M, \\ & a_{eq,1,1}x_1 + a_{eq,1,2}x_2 + \ldots + a_{eq,1,N}x_N & = & b_{eq,1}, \\ & a_{eq,2,1}x_1 + a_{eq,2,2}x_2 + \ldots + a_{eq,2,N}x_N & = & b_{eq,2}, \\ & & \vdots & \\ & a_{eq,L,1}x_1 + a_{eq,L,2}x_2 + \ldots + a_{eq,L,N}x_N & = & b_{eq,L}, \end{array}$$

We may equivalently write this problem in “Sigma” notation as follows:

$$\begin{array}{rrcl} \min & \displaystyle\sum_{k=1}^{N} c_k x_k & & \\ \text{subject to:} & \displaystyle\sum_{k=1}^{N} a_{1,k} x_k & \leq & b_1, \\ & \displaystyle\sum_{k=1}^{N} a_{2,k} x_k & \leq & b_2, \\ & \vdots & & \\ & \displaystyle\sum_{k=1}^{N} a_{M,k} x_k & \leq & b_M, \end{array}$$

$$
\begin{aligned}
\sum_{k=1}^{N} a_{eq,1,k} x_k &= b_{eq,1}, \\
\sum_{k=1}^{N} a_{eq,2,k} x_k &= b_{eq,2}, \\
&\vdots \\
\sum_{k=1}^{N} a_{eq,L,k} x_k &= b_{eq,L}
\end{aligned}
$$

The most compact notation uses matrix-vector format, and is given by

$$
\begin{aligned}
\min \quad & c^T x \\
\text{subject to:} \quad & Ax \leq b, \\
& A_{eq} x = b_{eq},
\end{aligned}
$$

where

$$
\begin{aligned}
x &= [x_1, x_2, \ldots, x_N]^T \\
c &= [c_1, c_2, \ldots, c_N]^T \\
[A]_{i,j} &= a_{i,j}, \quad A \in \mathbb{R}^{M \times N} \\
b &= [b_1, b_2, \ldots, b_M]^T, \quad b \in \mathbb{R}^M \\
[A_{eq}]_{i,j} &= a_{eq,i,j}, \quad A_{eq} \in \mathbb{R}^{L \times N} \\
b_{eq} &= [b_{eq,1}, b_{eq,2}, \ldots, b_{eq,L}]^T, \quad b_{eq} \in \mathbb{R}^L
\end{aligned}
$$

3.2 Graphical LP

For problems of one, two, or sometimes three dimensions, we can use graphical methods to visualize the feasible sets and solutions. This visualization provides excellent intuition for the nature of LP solutions. We demonstrate with the following example.

$$
\begin{aligned}
\max \quad & Z = 140x_1 + 160x_2 \\
\\
\text{s. to} \quad & 2x_1 + 4x_2 \leq 28 \\
& 5x_1 + 5x_2 \leq 50 \\
& x_1 \leq 8 \\
& x_2 \leq 6 \\
& x_1 \geq 0
\end{aligned}
$$

$$x_2 \geq 0$$

On a graph, one may successively plot each of the inequality constraints and divide the Cartesian space into feasible "half-spaces." As each half-space is identified, we retain the intersection of the remaining feasible set. This successive construction provides the **feasible set**, as shown in Fig. 7. After constructing the feasible set, we can plot the iso-contours of the objective function. For example, the lower-left-most iso-contour in Fig. 7 corresponds to $Z = 0$. Continuing towards the upper-right, the value of the objective function increases. The intersection of the largest-valued iso-contour and the feasible set occurs when $Z^* = 1480$, at $x_1^* = 6, x_2^* = 4$. Consequently, we have graphically solved the LP.

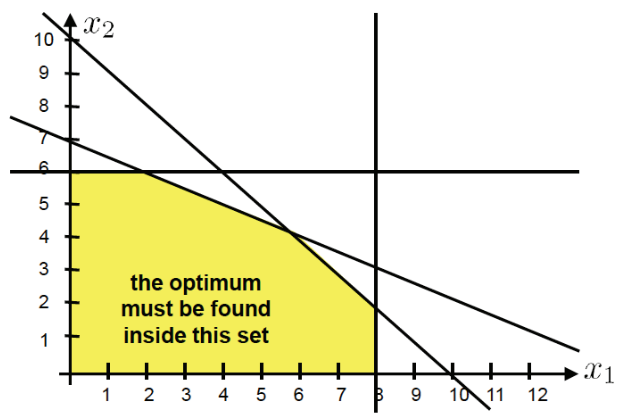


Figure 7: Construction of the feasible set in a linear program [LEFT], and the objective function isolines [RIGHT].

A feasible set can fall within one of the following three categories:

- **[Bounded]** The feasible set is bounded if it forms a closed subset of the Cartesian plane that does not include infinity.
- **[Unbounded]** The feasible set is unbounded if it forms a subset of the Cartesian plane that includes infinity.
- **[Empty]** The feasible set is empty if the intersection of all the inequality constraints forms the empty set. In this case the problem is infeasible. That is, no solution exists.

Exercise 5. *Draw examples of each of the three categories given above.*

Also note that an objective function may be bounded or unbounded. We make these concepts concrete with the following examples. Consider the feasible set defined by inequalities $x_2 \geq 3$, $x_1 + x_2 \geq 6$, and $x_2 \leq 0.5x_1 + 7$, as shown in Fig. 8. On the left, consider the objective $\max Z = 3.24x_1$. The iso-contours continue towards $x_1 = \infty$, without being bounded by the feasible set.

Hence, objective function Z is unbounded. In contrast, consider the objective $\min Z = x_1 + 3x_2$. Although the feasible set is unbounded, the iso-contours are bounded as they decrease in value. In this case, objective function Z is bounded.

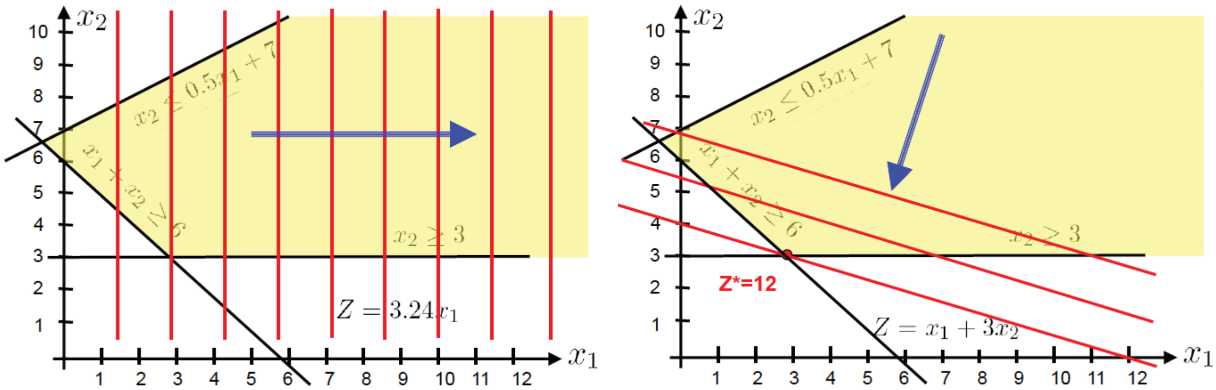


Figure 8: An unbounded [LEFT] and bounded [RIGHT] objective function.

This graphical analysis motivates the following proposition about LP solutions.

Proposition 2 (LP Solutions)**.** The solution to any linear program is characterized by one of the following three categories:

- **[No Solution]** This occurs when the feasible set is empty, or the objective function is unbounded.
- **[One Unique Solution]** There exists a single unique solution at the vertex of the feasible set. That is, two constraints are active and their intersection gives the optimal solution.
- **[A Non-Unique Solution]** There exists an infinite number of solutions, given by one edge of the feasible set. That is, one constraint is active and all solutions along this edge are equally optimal. This can only occur when the objective function gradient is orthogonal to a constraint.

Exercise 6. *Construct graphical examples of each of the three possible LP solutions given above.*

4 Quadratic Programming (QP)

We begin our exposition of quadratic programming problems with the following example.

Example 4.1 (Linear Regression)**.** Suppose you have collected measured data pairs (x_i, y_i), for $i = 1, \cdots, N$ where $N > 6$, as shown in Fig. 9. You seek to fit a fifth-order polynomial to this data, i.e.

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 \tag{44}$$

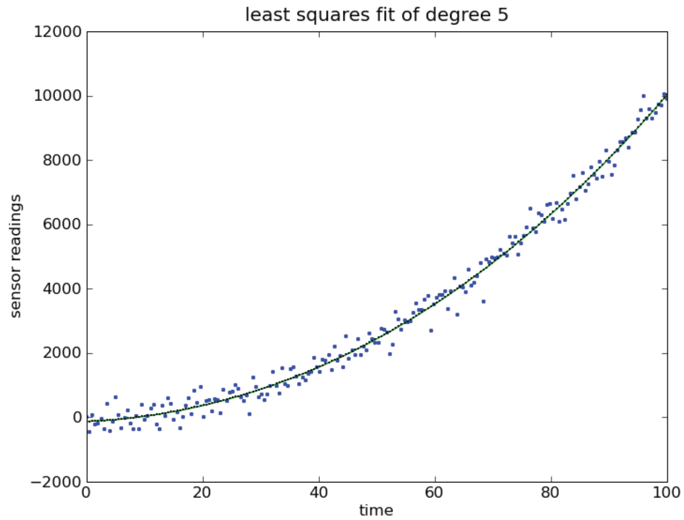


Figure 9: You seek to fit a fifth-order polynomial to the measured data above.

The goal is to determine parameters $c_i,\ i = 0, \cdots, 5$ that "best" fit the data in some sense. To this end, you may compute the residual r for each data pair:

$$\begin{aligned} c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_1^3 + c_4 x_1^4 + c_5 x_1^5 - y_1 &= r_1, \\ c_0 + c_1 x_2 + c_2 x_2^2 + c_3 x_2^3 + c_4 x_2^4 + c_5 x_2^5 - y_2 &= r_2, \\ \vdots &= \vdots \\ c_0 + c_1 x_N + c_2 x_N^2 + c_3 x_N^3 + c_4 x_N^4 + c_5 x_N^5 - y_N &= r_N, \end{aligned} \tag{45}$$

which can be arranged into matrix-vector form $Ac - y = r$, where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 & x_N^4 & x_N^5 \end{bmatrix}, \qquad c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}, \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}. \tag{46}$$

Now we compute an optimal fit for c in the following sense. We seek the value of c which minimizes the squared residual

$$\min_c \quad \frac{1}{2}\|r\|_2^2 = \frac{1}{2} r^T r = \frac{1}{2}(Ac - y)^T (Ac - y) = \frac{1}{2} c^T A^T A c - y^T A c + \frac{1}{2} y^T y. \tag{47}$$

Note that (47) is quadratic in variable c and therefore a convex function of c. This produces another special case of convex problems called **quadratic programs**. In this case the problem is unconstrained. As a result, we can set the gradient with respect to c to zero and directly solve for the minimizer.

$$\begin{aligned}\frac{\partial}{\partial c}\,\frac{1}{2}\|r\|_2^2 = A^T A c - A^T y = 0,& \\ A^T A c = A^T y,& \\ \boxed{c = (A^T A)^{-1} A^T y}& \end{aligned} \tag{48}$$

This provides a direct formula for fitting the polynomial coefficients $c_i,\ i = 0, \cdots, 5$ using the measured data.

Exercise 7 (Tikhonov or L_2 regularization, a.k.a. Ridge Regression)**.** *Consider the fifth-order polynomial regression model in (44). Suppose we seek the value of c which minimizes the squared residual plus a so-called Tikhonov regularization term:*

$$\min_{c} \quad \frac{1}{2}\|r\|_2^2 + \frac{1}{2}\|\Gamma c\|_2^2. \tag{49}$$

for some matrix $\Gamma \in \mathbb{R}^{6\times 6}$. Derive the QP. Solve for the minimizer of this unconstrained QP. Provide a formula for the optimal coefficients c.

Exercise 8. *Consider fitting the coefficients c_1, c_2, c_3 of the following sum of radial basis functions to data pairs (x_i, y_i), $i = 1, \cdots, N$.*

$$y = c_1 e^{-(x-0.25)^2} + c_2 e^{-(x-0.5)^2} + c_3 e^{-(x-0.75)^2} \tag{50}$$

Formulate and solve the corresponding QP problem.

Exercise 9. *Repeat the same exercise for the following Fourier Series:*

$$y = c_1 \sin(\omega x) + c_2 \cos(\omega x) + c_3 \sin(2\omega x) + c_4 \cos(2\omega x) \tag{51}$$

4.1 General Form of QP

Quadratic programs can be identified as having a quadratic objective function and linear constraints. Mathematically, this takes the form

$$\begin{aligned} \min \qquad & \frac{1}{2}x^T Q x + R^T x, \\ \text{subject to:} \qquad & Ax \le b, \\ & A_{eq} x = b_{eq}. \end{aligned}$$

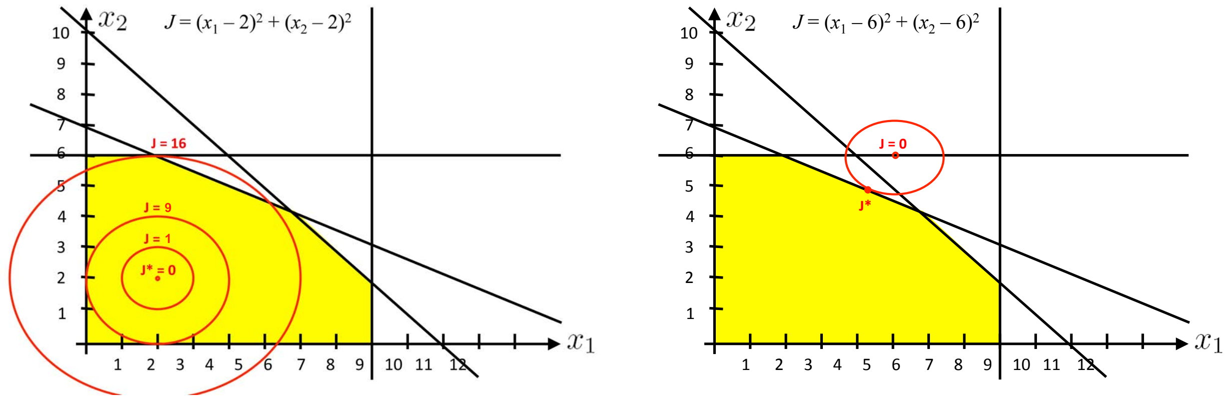


Figure 10: An interior optimum [LEFT] and boundary optimum [RIGHT] for a QP solved graphically.

4.2 Graphical QP

For problems of one, two, or three dimensions, it is possible to solve QPs graphically. Consider the following QP example:

$$\min \qquad J = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\begin{array}{lrcl} \text{s. to} & 2x_1 + 4x_2 & \leq & 28 \\ & 5x_1 + 5x_2 & \leq & 50 \\ & x_1 & \leq & 8 \\ & x_2 & \leq & 6 \\ & x_1 & \geq & 0 \\ & x_2 & \geq & 0 \end{array}$$

The feasible set and corresponding iso-contours are illustrated in the left-hand side of Fig. 10. In this case, the solution is an **interior optimum.** That is, no constraints are active at the minimum. In contrast, consider the objective function $J = (x_1 - 6)^2 + (x_2 - 6)^2$ shown on the right-hand side of Fig. 10. In this case, the minimum occurs at the boundary and is unique.

5 Nonlinear Programming (NLP)

Nonlinear programming problems involve objective functions that are nonlinear in the decision variable x. LP and QP problems are special cases of NLPs. As such, the particular structure of LPs and QPs can be exploited for analysis and computation. In this section, we discuss a more general class of nonlinear problems and corresponding tools for analysis and computation.

A *nonlinear optimization problem* has the form

$$\min_{x} \quad f(x) \tag{52}$$
$$\text{subject to} \quad g_i(x) \le 0, \quad i = 1, \cdots, m \tag{53}$$
$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{54}$$

Note the key difference is the objective function and constraints take a general form that is nonlinear in x. In this general case, we first discuss algorithms for the *unconstrained case*. Then we consider constraints and present general theory on NLPs.

5.1 Gradient Descent

Gradient descent is a first-order iterative algorithm for finding the local minimum of a differentiable function. It is applicable to unconstrained minimization problems. Starting from an initial guess, the main idea is to step in the direction of steepest descent at each iteration. Eventually the algorithm will converge when the gradient is zero, which corresponds to a local minimum.

This concept is illustrated in Fig. 11, which provides iso-contours of a function $f(x)$ that we seek to minimize. In this example, the user provides an initial guess x_0. Then the algorithm proceeds according to

$$x_{k+1} = x_k - h \cdot \nabla f(x) \tag{55}$$

where $h > 0$ is some positive step size. The iteration proceeds until a stopping criterion is satisfied. Typically, we stop when the gradient is sufficiently close to zero

$$\|\nabla f(x_k)\| \le \epsilon \tag{56}$$

where $\epsilon > 0$ is some small user defined stopping criterion parameter.

Exercise 10. *Minimize the function* $f(x_1, x_2) = \frac{1}{2}(x_1^2 + 10x_2^2)$ *with an initial guess of* $(x_{1,0}, x_{2,0}) = (10, 1)$. *Use a step-size of* $h = 1$, *and a stopping criterion of* $\|\nabla f(x_k)\|_2 = \sqrt{x_{1,k}^2 + x_{2,k}^2} \le \epsilon = 0.01$.

For non-convex problems, such as the one illustrated in Fig. 12, the gradient descent algorithm converges to the local minimum. In other words, convergence to a global minimum is not guaranteed unless the function $f(x)$ is convex over the feasible set D. In this case, one may select a variety of initial guesses, x_0, to start the gradient descent algorithm. Then the best of all converged values is used for the proposed solution. This still does not guarantee a global minimum, but is effective at finding a sub-optimal solution in practice.

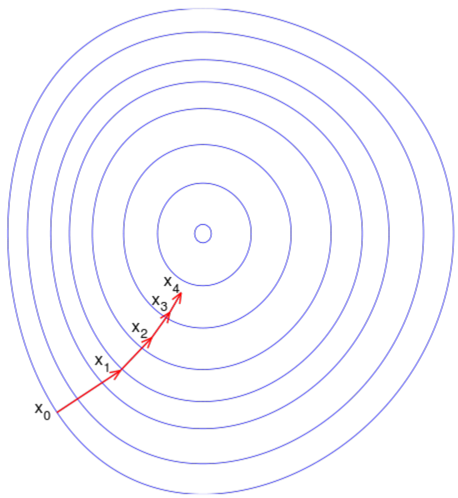


Figure 11: Illustration of gradient descent with step size proportional to the gradient.

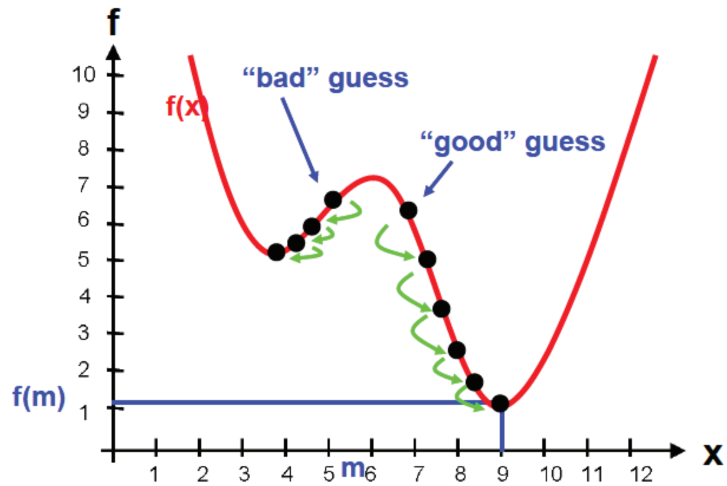


Figure 12: In non-convex functions, gradient descent converges to the local minimum. Consequently, different initial guesses may result in different solutions.

5.2 Barrier and Penalty Functions

A drawback of the gradient descent method is that it does not explicitly account for constraints. Barrier and penalty functions are two methods of augmenting the objective function $f(x)$ to approximately account for the constraints. To illustrate, consider the constrained minimization problem

$$\min_{x} \quad f(x) \tag{57}$$
$$\text{subject to} \quad g(x) \leq 0. \tag{58}$$

We seek to modify the objective function to account for the constraints, in an approximate way. Thus we can write

$$\min_{x} \quad f(x) + \phi(x; \epsilon) \tag{59}$$

where $\phi(x; \varepsilon)$ captures the effect of the constraints and is differentiable, thereby enabling usage of gradient descent. The parameter ε is a user-defined parameter that allows one to more accurately or more coarsely approximate the constraints. Barrier and penalty functions are two methods of defining $\phi(x; \varepsilon)$. The main idea of each is as follows:

- **Barrier Function:** Allow the objective function to increase towards infinity as x approaches the constraint boundary from inside the feasible set. In this case, the constraints are guaranteed to be satisfied, but it is impossible to obtain a boundary optimum.
- **Penalty Function:** Allow the objective function to increase towards infinity as x violates

the constraints $g(x)$. In this case, the constraints can be violated, but it allows boundary optimum.

To motivate these methods, consider the non-convex function shown in Fig. 13. We seek to find the minimum within the range $[0.5, 1.5]$. Mathematically, this is a one-dimensional problem written as

$$\min_{x} \quad f(x) \tag{60}$$
$$\text{s. to} \quad x \leq b \tag{61}$$
$$x \geq a \tag{62}$$

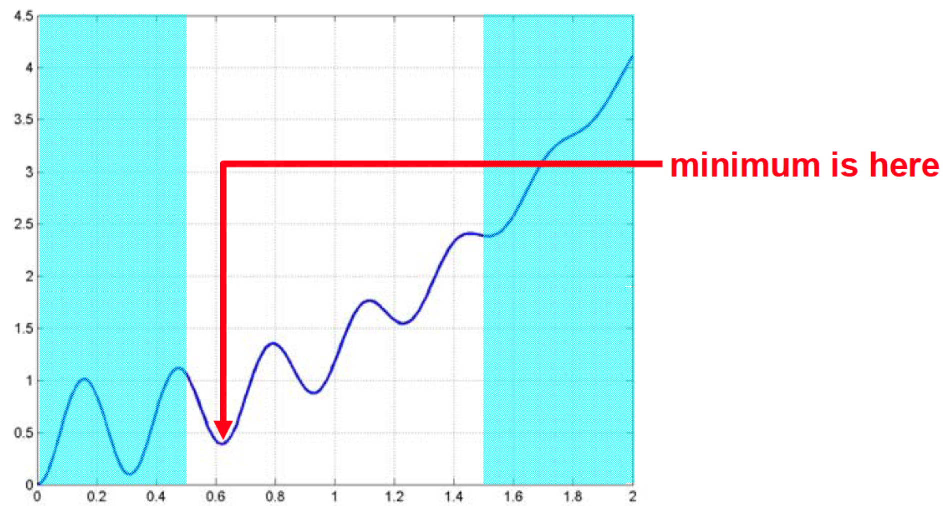


Figure 13: Find the optimum of the function shown above within the range [0.5, 1.5].

5.2.1 Log Barrier Function

Let us define the **log barrier function** as

$$\phi(x; \varepsilon) = -\varepsilon \log \left(\frac{(x-a)(b-x)}{b-a} \right) \tag{63}$$

The critical property of the log barrier function is that $\phi(x; \varepsilon) \to +\infty$ as $x \to a$ from the right side and $x \to b$ from the left side. Ideally, the log barrier function is zero inside the constraint set. This desired property becomes increasingly true as $\varepsilon \to 0$.

5.2.2 Quadratic Penalty Function

Let us define the **quadratic penalty function** as

$$\phi(x;\varepsilon) = \begin{cases} 0 & \text{if } a \leq x \leq b \\ \frac{1}{2\varepsilon}(x-a)^2 & \text{if } x < a \\ \frac{1}{2\varepsilon}(x-b)^2 & \text{if } x > b \end{cases} \tag{64}$$

The critical property of the quadratic penalty function is that $\phi(x;\varepsilon)$ increases towards infinity as x increases beyond b or decreases beyond a. The severity of this increase is parameterized by ε. Also, note that $\phi(x;\varepsilon)$ is defined such that $f(x) + \phi(x;\varepsilon)$ remains differentiable at $x = a$ and $x = b$, thus enabling application of the gradient descent algorithm.

5.3 Sequential Quadratic Programming (SQP)

In our discussion of NLPs so far, we have explained how to solve (i) unconstrained problems via gradient method, and (ii) unconstrained problems augmented with barrier or penalty functions to account for constraints. In this section, we provide a direct method for handling NLPs with constraints, called the Sequential Quadratic Programming (SQP) method. The idea is simple. We solve a single NLP as a sequence QP subproblems. In particular, at each iteration we approximate the objective function and constraints by a QP. Then, within each iteration, we solve the corresponding QP and use the solution as the next iterate. This process continues until an appropriate stopping criterion is satisfied.

SQP is very widely used in engineering problems and often the first "go-to" method for NLPs. For many practical energy system problems, it produces fast convergence thanks to its strong theoretical basis. This method is commonly used under-the-hood of Matlab function `fmincon`.

Consider the general NLP

$$\min_{x} \quad f(x) \tag{65}$$

$$\text{subject to} \quad g(x) \leq 0, \tag{66}$$

$$h(x) = 0, \tag{67}$$

and the k^{th} iterate x_k for the decision variable. We utilize the Taylor series expansion. At each iteration of SQP, we consider the 2nd-order Taylor series expansion of the objective function (65), and 1st-order expansion of the constraints (66)-(67) around $x = x_k$:

$$f(x) \approx f(x_k) + \frac{\partial f}{\partial x}^T (x_k)\,(x - x_k) + \frac{1}{2}\,(x-x_k)^T\,\frac{\partial^2 f}{\partial x^2}(x_k)\,(x - x_k)\,, \tag{68}$$

$$g(x) \approx g(x_k) + \frac{\partial g}{\partial x}^T (x_k)\,(x - x_k) \leq 0, \tag{69}$$

$$h(x) \approx h(x_k) + \frac{\partial h}{\partial x}^T (x_k)\,(x - x_k) = 0 \tag{70}$$

To simplify the notation, define $\tilde{x} = x - x_k$. Then we arrive at the following approximate QP

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x} \tag{71}$$
$$\text{s. to} \quad A\tilde{x} \leq b \tag{72}$$
$$A_{eq}\tilde{x} = b_{eq} \tag{73}$$

where

$$Q = \frac{\partial^2 f}{\partial x^2}(x_k), \qquad R = \frac{\partial f}{\partial x}(x_k) \tag{74}$$
$$A = \frac{\partial g}{\partial x}^T (x_k), \qquad b = -g(x_k) \tag{75}$$
$$A_{eq} = \frac{\partial h}{\partial x}^T (x_k), \qquad b_{eq} = -h(x_k) \tag{76}$$

Suppose (71)-(73) yields the optimal solution $\tilde{x}^*$. Then let $x_{k+1} = x_k + \tilde{x}^*$, and repeat.

Remark 5.1. Note that the iterates in SQP are not guaranteed to be feasible for the original NLP problem. That is, it is possible to obtain a solution to the QP subproblem which satisfies the approximate QP's constraints, but not the original NLP constraints.

Example 5.1. *Consider the NLP*

$$\min_{x_1,x_2} \quad e^{-x_1} + (x_2 - 2)^2 \tag{77}$$
$$\textit{s. to} \quad x_1 x_2 \leq 1. \tag{78}$$

with the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 1]^T$*. By hand, formulate the Q,R,A,b matrices for the first three iterates. Use Matlab command* `quadprog` *to solve each subproblem. What is the solution after three iterations?*

We have $f(x) = e^{-x_1} + (x_2 - 2)^2$ and $g(x) = x_1 x_2 - 1$. The iso-contours for the objective function and constraint are provided in Fig. 14. From visual inspection, it is clear the optimal solution is near $[0.5, 2]^T$. We seek to find the approximate QP subproblem

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x} \tag{79}$$
$$\text{s. to} \quad A\tilde{x} \leq b \tag{80}$$

Taking derivatives of $f(x)$ and $g(x)$, one obtains

$$Q = \begin{bmatrix} e^{-x_1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-x_1} \\ 2(x_2 - 2) \end{bmatrix}, \tag{81}$$

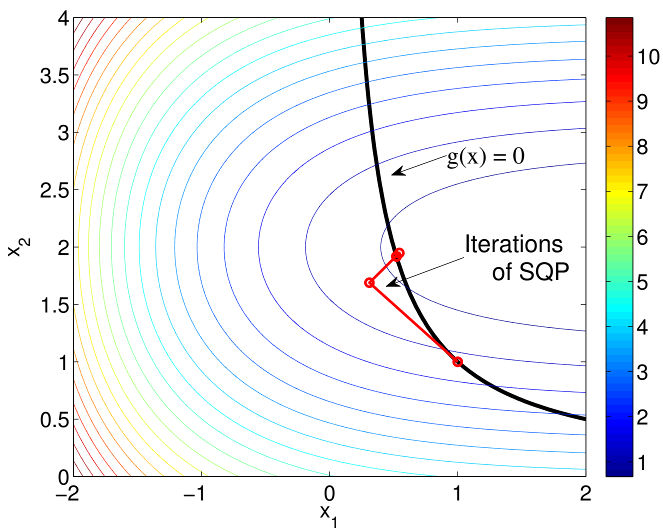
$$A = \begin{bmatrix} x_2 & x_1 \end{bmatrix}, \qquad b = 1 - x_1 x_2 \tag{82}$$

Now consider the initial guess $[x_{1,0}, x_{2,0}]^T = [1,1]^T$. Note that this guess is feasible. We obtain the following matrices for the first QP subproblem

$$Q = \begin{bmatrix} e^{-1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-1} \\ -2 \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad b = 0$$

Solving this QP subproblem results in $\tilde{x}^* = [-0.6893, 0.6893]$. Then the next iterate is given by $[x_{1,1}, x_{2,1}] = [x_{1,0}, x_{2,0}] + \tilde{x}^* = [0.3107, 1.6893]$. Repeating the formulation and solution of the QP subproblem at iteration 1 produces $[x_{1,1}, x_{2,1}] = [0.5443, 1.9483]$. Note that this iterate is infeasible. Continued repetitions will produce iterates that converge toward the true solution.



Iter.	$[x_1, x_2]$	$f(x)$	$g(x)$
0	$[1, 1]$	1.3679	0
1	$[0.311, 1.689]$	0.8295	-0.4751
2	$[0.544, 1.948]$	0.5829	0.0605
3	$[0.522, 1.917]$	0.6002	0.0001
4	$[0.521, 1.919]$	0.6004	-0.0000

Figure 14 & Table 2: [LEFT] Iso-contours of objective function and constraint for Example 5.1. [RIGHT] Numerical results for first three iterations of SQP. Note that some iterates are infeasible.

SQP provides an algorithmic way to solve NLPs in energy system applications. However, it still relies on approximations - namely truncated Taylor series expansions - to solve the optimization problem via a sequence of QP subproblems. Next, we discuss a direct method for solving NLPs, without approximation.

5.4 First-Order Necessary Conditions for Optimality

In calculus, you learned that a necessary condition for minimizers is that the function's slope is zero at the minimum. We extend this notion in this section. Namely, we discuss first-order necessary conditions for optimality for NLPs.

5.4.1 Method of Lagrange Multipliers

Consider the equality constrained optimization problem

$$\min \quad f(x) \tag{83}$$
$$\text{s. to} \quad h_j(x) = 0, \qquad j = 1, \cdots, l \tag{84}$$

Introduce the so-called "Lagrange multipliers" $\lambda_j, j = 1, \cdots, l$. Then we can augment the cost function to form the "Lagrangian" $L(x)$ as follows

$$L(x) = f(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \tag{85}$$
$$= f(x) + \lambda^T h(x) \tag{86}$$

Note that when all constraints are satisfied, that is $h(x) = 0$, then the second term becomes zero. Consequently, the Lagrangian $L(x)$ and cost function $f(x)$ provide identical values for all feasible x. We now state the first-order necessary condition (FONC) for equality constrained problems:

Proposition 3 (FONC for Equality Constrained NLPs)**.** *If a local minimum x^* exists, then it satisfies*

$$\frac{\partial L}{\partial x}(x^*) = \frac{\partial f}{\partial x}(x^*) + \lambda^T \frac{\partial h}{\partial x}(x^*) = 0 \qquad \textit{(stationarity)}, \tag{87}$$
$$\frac{\partial L}{\partial \lambda}(x^*) = h(x^*) = 0 \qquad \textit{(feasibility)}. \tag{88}$$

That is, the gradient of the Lagrangian is zero at the minimum x^*.

Remark 5.2. *This condition is only necessary. That is, if a local minimum x^* exists, then it must satisfy the FONC. However, a design x which satisfies the FONC isn't necessarily a local minimum.*

Remark 5.3. *If the optimization problem is convex, then the FONC is necessary and sufficient. That is, a design x which satisfies the FONC is also a local minimum.*

Example 5.2. Consider the equality constrained QP

$$\min \quad \frac{1}{2} x^T Q x + R^T x \tag{89}$$
$$\text{s. to} \quad Ax = b \tag{90}$$

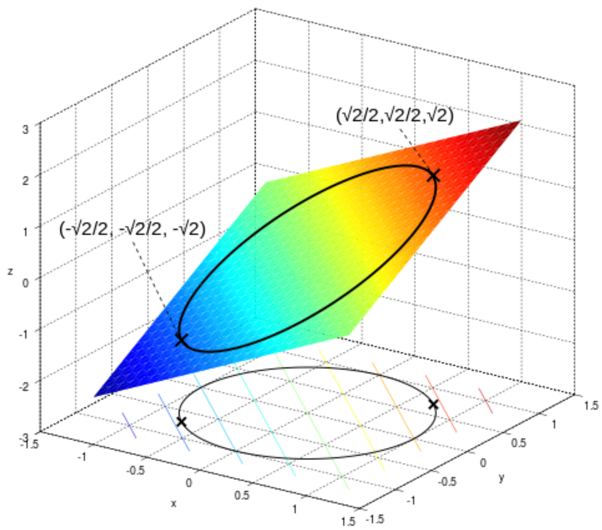


Figure 15: Visualization of circle-plane problem from Example 5.4.

Form the Lagrangian,

$$L(x) = \frac{1}{2}x^T Q x + R^T x + \lambda^T (Ax - b). \tag{91}$$

Then the FONC is

$$\frac{\partial L}{\partial x}(x^*) = Qx^* + R + A^T\lambda = 0. \tag{92}$$

Combining the FONC with the equality constraint yields

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda \end{bmatrix} = \begin{bmatrix} -R \\ b \end{bmatrix} \tag{93}$$

which provides a set of linear equations that can be solved directly.

Example 5.3. Consider a circle inscribed on a plane, as shown in Fig. 15. Suppose we wish to find the “lowest” point on the plane while being constrained to the circle. This can be abstracted as the NLP:

$$\text{min} \quad f(x,y) = x + y \tag{94}$$

$$\text{s. to} \quad x^2 + y^2 = 1 \tag{95}$$

Form the Lagrangian

$$L(x,y,\lambda) = x + y + \lambda(x^2 + y^2 - 1) \tag{96}$$

Then the FONCs and equality constraint can be written as the set of nonlinear equations:

$$\frac{\partial L}{\partial x} = 1 + 2\lambda x = 0 \tag{97}$$

$$\frac{\partial L}{\partial y} = 1 + 2\lambda y = 0 \tag{98}$$
$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 1 = 0 \tag{99}$$

One can solve these three equations for x, y, λ by hand to arrive at the solution

$$\begin{aligned} (x^*, y^*) &= \left(\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2}\right) \\ f(x^*, y^*) &= \pm\sqrt{2} \\ \lambda &= \mp 1/\sqrt{2} \end{aligned}$$

5.4.2 Karush-Kuhn-Tucker (KKT) Conditions

Now we consider the general constrained optimization problem

$$\min \quad f(x) \tag{100}$$
$$\text{s. to} \quad g_i(x) \leq 0, \qquad i = 1, \cdots, m \tag{101}$$
$$h_j(x) = 0, \qquad j = 1, \cdots, l \tag{102}$$

Introduce the so-called "Lagrange multipliers" $\lambda_j, j = 1, \cdots, l$ each associated with equality constraints $h_j(x), j = 1, \cdots, l$ and $\mu_i, i = 1, \cdots, m$ each associated with inequality constraints $g_i(x), i = 1, \cdots, m$. Then we can augment the cost function to form the "Lagrangian" $L(x)$ as follows

$$L(x) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \tag{103}$$
$$= f(x) + \mu^T g(x) + \lambda^T h(x) \tag{104}$$

As before, when the equality constraints are satisfied, $h(x) = 0$, then the third term becomes zero. Elements of the second term become zero in two cases: (i) an inequality constraint is active, that is $g_i(x) = 0$; (ii) the Lagrange multiplier $\mu_i = 0$. Consequently, the Lagrangian $L(x)$ can be constructed to have identical values of the cost function $f(x)$ if the aforementioned conditions are applied. This motivates the first-order necessary conditions (FONC) for the general constrained optimization problem – called the Karush-Kuhn-Tucker (KKT) Conditions.

Proposition 4 (KKT Conditions)**.** *If x^* is a local minimum, then the following necessary conditions hold:*

$$\frac{\partial f}{\partial x}(x^*) + \sum_{i=1}^{m} \mu_i \frac{\partial}{\partial x} g_i(x^*) + \sum_{j=1}^{l} \lambda_j \frac{\partial}{\partial x} h_j(x^*) = 0, \qquad \textit{Stationarity} \tag{105}$$

$$\begin{aligned} g_i(x^*) &\leq 0, \quad i = 1, \cdots, m && \textit{Feasibility} && (106)\\ h_j(x^*) &= 0, \quad j = 1, \cdots, l && \textit{Feasibility} && (107)\\ \mu_i &\geq 0, \quad i = 1, \cdots, m && \textit{Non-negativity} && (108)\\ \mu_i g_i(x^*) &= 0, \quad i = 1, \cdots, m && \textit{Complementary slackness} && (109) \end{aligned}$$

which can also be written in matrix-vector form as

$$\frac{\partial f}{\partial x}(x^*) + \mu^T \frac{\partial}{\partial x} g(x^*) + \lambda^T \frac{\partial}{\partial x} h(x^*) = 0, \qquad \textit{Stationarity} \qquad (110)$$

$$\begin{aligned} g(x^*) &\leq 0, && \textit{Feasibility} && (111)\\ h(x^*) &= 0, && \textit{Feasibility} && (112)\\ \mu &\geq 0, && \textit{Non-negativity} && (113)\\ \mu^T g(x^*) &= 0, && \textit{Complementary slackness} && (114) \end{aligned}$$

Remark 5.4. Note the following properties of the KKT conditions

- Non-zero μ_i indicates $g_i \leq 0$ is active (true with equality). In practice, non-zero μ_i is how we identify active constraints from nonlinear solvers.
- The KKT conditions are necessary, only. That is, if a local minimum x^* exists, then it must satisfy the KKT conditions. However, a design x which satisfies the KKT conditions isn't necessarily a local minimum.
- If problem is convex, then the KKT conditions are necessary and sufficient. That is, one may directly solve the KKT conditions to obtain the minimum.
- Lagrange multipliers λ, μ are sensitivities to perturbations in the constraints
  - In economics, this is called the "shadow price"
  - In control theory, this is called the "co-state"
- The KKT conditions have a geometric interpretation demonstrated in Fig. 16. Consider minimizing the cost function with isolines shown in red, where $f(x)$ is increasing as x_1, x_2 increase, as shown by the gradient vector ∇f. Now consider two inequality constraints $g_1(x) \leq 0, g_2(x) \leq 0$, forming the feasible set colored in light blue. The gradients at the minimum, weighted by the Lagrange multipliers, are such that their sum equals $-\nabla f$. In other words, the vectors balance to zero according to $\nabla f(x^*) + \mu_1 \nabla g_1(x^*) + \mu_2 \nabla g_2(x^*) = 0$.

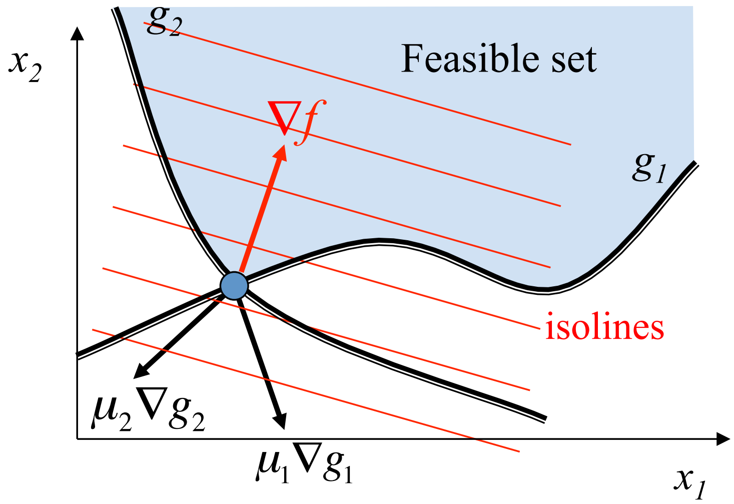


Figure 16: Geometric interpretation of KKT conditions

Example 5.4. Consider again the circle-plane problem, as shown in Fig. 15. Suppose we wish to find the "lowest" point on the plane while being constrained to within or on the circle. This can be abstracted as the NLP:

$$\text{min} \quad f(x,y) = x + y \tag{115}$$

$$\text{s. to} \quad x^2 + y^2 \leq 1 \tag{116}$$

Note this problem is convex, therefore the solution to the KKT conditions provides the minimizer (x^*, y^*). We form the Lagrangian

$$L(x,y,\mu) = x + y + \mu(x^2 + y^2 - 1) \tag{117}$$

Then the KKT conditions are

$$\frac{\partial L}{\partial x} = 1 + 2\mu x^* = 0 \tag{118}$$

$$\frac{\partial L}{\partial y} = 1 + 2\mu y^* = 0 \tag{119}$$

$$\frac{\partial L}{\partial \mu} = (x^*)^2 + (y^*)^2 - 1 \leq 0 \tag{120}$$

$$\mu \geq 0 \tag{121}$$

$$\mu\left((x^*)^2 + (y^*)^2 - 1\right) = 0 \tag{122}$$

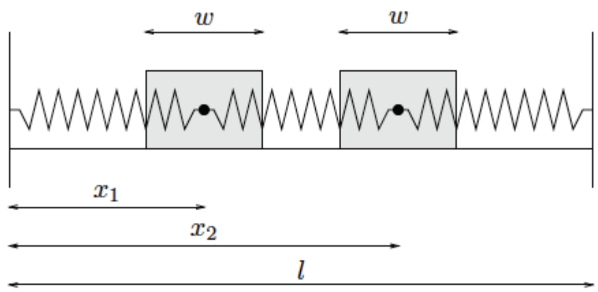


Figure 17: Spring-block system for Example 5.5

One can solve these equations/inequalities for x^*, y^*, μ by hand to arrive at the solution

$$\begin{aligned}(x^*, y^*) &= \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right) \\ f(x^*, y^*) &= -\sqrt{2} \\ \mu &= 1/\sqrt{2}\end{aligned}$$

Example 5.5 (Mechanics Interpretation)**.** Interestingly, the KKT conditions can be used to solve a familiar undergraduate physics example involving the principles of mechanics. Consider two blocks of width w, where each block is connected to each other and the surrounding walls by springs, as shown in Fig. 17. Reading left to right, the springs have spring constants k_1, k_2, k_3. The objective is to determine the equilibrium position of the masses. The principles of mechanics indicate that the equilibrium is achieved when the spring potential energy is minimized. Moreover, we have *kinematic constraints* that restrain the block positions. That is, the blocks cannot overlap with each other or the walls. Consequently, we can formulate the following nonlinear program.

$$\text{min} \quad f(x_1, x_2) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2 (x_2 - x_1)^2 + \frac{1}{2}k_3 (l - x_2)^2 \tag{123}$$

$$\text{s. to} \quad x_1 - \frac{w}{2} \geq 0, \tag{124}$$

$$x_1 + \frac{w}{2} \leq x_2 - \frac{w}{2}, \tag{125}$$

$$x_2 + \frac{w}{2} \leq l \tag{126}$$

It is easy to see this problem is a QP with a convex feasible set. Consequently, we may formulate and solve the KKT conditions directly to find the equilibrium block positions.

Consider Lagrange multipliers μ_1, μ_2, μ_3. Form the Lagrangian:

$$L(x, \mu) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2 (x_2 - x_1)^2 + \frac{1}{2}k_3 (l - x_2)^2 + \mu_1 \left(\frac{w}{2} - x_1\right) + \mu_2 (x_1 - x_2 + w) + \mu_3 \left(x_2 + \frac{w}{2} - l\right) \tag{127}$$

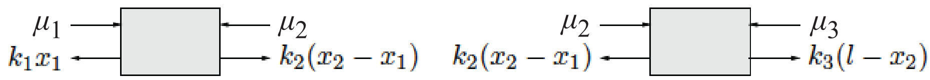


Figure 18: Free-body diagram of spring-block system for Example 5.5

where $x = [x_1, x_2]^T, \mu = [\mu_1, \mu_2, \mu_3]^T$. Now we can formulate the KKT conditions: We have $\mu \geq 0$ for non-negativity,

$$\mu_1 \left(\frac{w}{2} - x_1\right) = 0, \qquad \mu_2 \left(x_1 - x_2 + w\right) = 0, \qquad \mu_3 \left(x_2 + \frac{w}{2} - l\right) = 0 \tag{128}$$

for complementary slackness, and

$$\begin{bmatrix} k_1 x_1 - k_2 \left(x_2 - x_1\right) \\ k_2 \left(x_2 - x_1\right) - k_3 \left(l - x_2\right) \end{bmatrix} + \mu_1 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \mu_3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \tag{129}$$

for stationarity. Interestingly, the μ_i's can be interpreted as contact forces. That is, consider the free-body diagrams for each block shown in Fig. 18, where we denote the contact forces between the left wall–block 1, block 1–block 2, and block 2–right wall for μ_1, μ_2, μ_3, respectively. When no contact exists, then the corresponding contact force is trivially zero, which also indicates the associated inequality constraint is inactive. However, when the contact force μ_i is non-zero, this indicates the corresponding inequality constraint is active.

6 Convex Programming

A *convex optimization problem* has the form

$$\min_{x} \quad f(x) \tag{130}$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \cdots, m \tag{131}$$

$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{132}$$

Comparing this problem with the abstract optimization problem in (1)-(3), the *convex optimization problem* has three additional requirements:

- objective function $f(x)$ must be convex,
- the inequality constraint functions $g_i(x)$ must be convex for all $i = 1, \cdots, m$,
- the equality constraint functions $h_j(x)$ must be affine for all $j = 1, \cdots, l$.

Note that in the convex optimization problem, we can only tolerate affine equality constraints, meaning (132) takes the matrix-vector form of $A_{eq} x = b_{eq}$.

In general, no analytical formula exists for the solution of convex optimization problems. However, there are very effective and reliable methods for solving them. For example, we can easily solve problems with hundreds of variables and thousands of constraints on a current laptop computer, in at most a few tens of seconds. Due to the impressive efficiency of these solvers, many researchers have developed tricks for transforming problems into convex form. As a result, a surprising number of practical energy system problems can be solved via convex optimization. With only a bit of exaggeration, we can say that, if you formulate a practical problem as a convex optimization problem, then you have solved the original problem. Recognizing a convex optimization problem can be difficult, however. The challenge, and art, in using convex optimization is in recognizing and formulating the problem. Once this formulation is done, solving the problem is essentially an off-the-shelf technology.

6.1 Theory

The following statements are true about convex programming problems:

- If a local minimum exists, then it is the global minimum.
- If the objective function is strictly convex, and a local minimum exists, then it is a unique minimum.

6.2 Examples

Convex programming problems arise in a surprising number of practical applications, including

- Linear programming
- Least squares (i.e. QP)
- Convex quadratic minimization with linear constraints
- Quadratically constrained Convex-quadratic minimization with convex quadratic constraints
- Conic optimization
- Geometric programming
- Second order cone programming
- Semidefinite programming
- Entropy maximization with appropriate constraints

6.3 Duality

7 Notes

You can learn more about optimization...

References

[1] P. Denholm, E. Ela, B. Kirby, and M. Milligan, "Role of energy storage with renewable electricity generation," National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep. Technical Report NREL/TP-6A2-47187, 2010.

[2] "Strategic design of public bicycle sharing systems with service level constraints," Transportation Research Part E: Logistics and Transportation Review, vol. 47, no. 2, pp. 284 – 294, 2011.

[3] J. C. Garcia-Palomares, J. Gutierrez, and M. Latorre, "Optimizing the location of stations in bike-sharing programs: A gis approach," Applied Geography, vol. 35, no. 1-2, pp. 235–246, 2012.

[4] R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Busic, "Large-scale vehicle sharing systems: Analysis of velib," International Journal of Sustainable Transportation, vol. 7, no. 1, pp. 85–106, 2013.

[5] S. Bashash, S. J. Moura, J. C. Forman, and H. K. Fathy, "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity," Journal of Power Sources, vol. 196, no. 1, pp. 541 – 549, 2011.

[6] S. J. Tong, A. Same, M. A. Kootstra, and J. W. Park, "Off-grid photovoltaic vehicle charge using second life lithium batteries: An experimental and numerical investigation," Applied Energy, vol. 104, no. 0, pp. 740 – 750, 2013.

[7] P. Trnka, J. Pekar, and V. Havlena, "Application of distributed mpc to barcelona water distribution network," in Proceedings of the 18th IFAC World Congress. Milan, Italy, 2011.

[8] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2009.