CHAPTER 4: NONLINEAR PROGRAMMING

Overview

To this point, we have considered optimization problems where the constraints are linear, and the objective function is linear or quadratic. One may argue that LPs and QPs are too simplistic for real-world problems, and rightfully so in many applications. We now study the general nonlinear programming (NLP) problem

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

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

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

where f, g_i, h_j can all be nonlinear. In this chapter we restrict ourselves to continuous design variables where $x \in \mathbb{R}^n$. Note that LPs and QPs are special cases of NLPs. However, the previous developments have enabled us to perform analysis and use solvers that exploit their special structure. We now cannot rely on that structure and require additional conceptual framework. In particular, 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. In NLP problems, we will discover that several types of minima may occur and therefore definitions to differentiate these minima are necessary. To determine existence and uniqueness of such minima, we require the notions of convex sets and convex functions, discussed as mathematical preliminaries. Next we discuss the gradient algorithm to solve NLPs in the unconstrained case. Interestingly, a problem with constraints can be converted into an approximate unconstrained problem via barrier or penalty functions, thereby enabling the application of gradient descent. Finally, we derive conditions for optimality for constrained NLPs, without approximation. These conditions include the Method of Lagrange Multipliers and the Karush-Kuhn-Tucker (KKT) Conditions. The resulting conditions often render a system of nonlinear equations that can be solved to determine the optimum. We close with a discussion of sensitivity analysis, which examines how the optimum changes with respect to perturbations in the constraints. We will see this provides an intuitive interpretation to the seemingly mathematical construction of Lagrange multipliers.

Chapter Organization

This chapter is organized as follows:

- (Section 1) Mathematical Preliminaries
- (Section 2) Definition of Minimizers
- (Section 3) Gradient Descent
- (Section 4) Barrier & Penalty Functions
- (Section 5) Optimality Conditions
- (Section 6) Sensitivity Analysis

1 Mathematical Preliminaries

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 chapter, we discuss a more general class of nonlinear problems and corresponding tools for analysis and computation. To begin, we start with some useful mathematical concepts. The first two concepts are *convex sets* and *convex functions*.

1.1 Convex Sets

Definition 1.1 (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*

$$\lambda a + (1 - \lambda) b \in D \tag{4}$$

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.

1.1.1 Examples

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

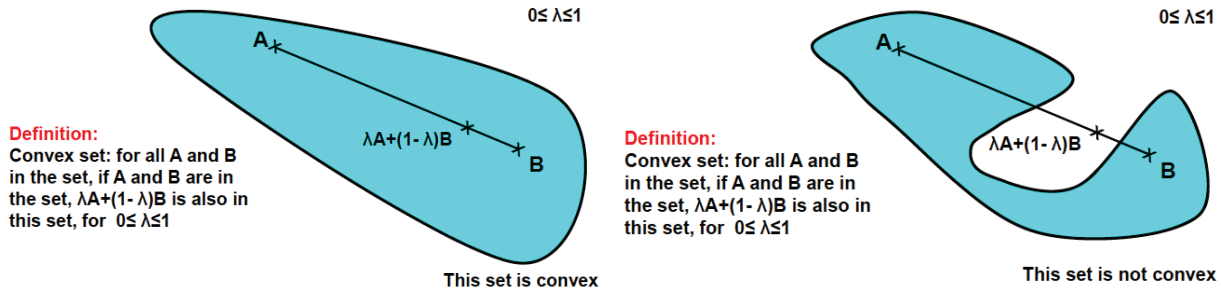

Figure 1: Visualization of convex [left] and non-convex [right] sets.

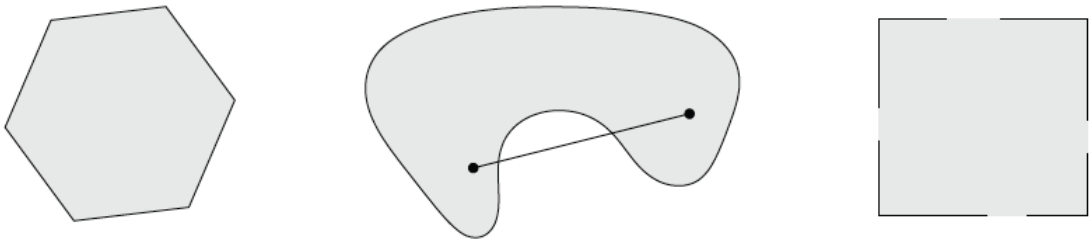

Figure 2: Some simple convex and nonconvex sets. [Left] The hexagon, which includes its boundary (shown darker), is convex. [Middle] The kidney shaped set is not convex, since the line segment between the two points in the set shown as dots is not contained in the set. [Right] The square contains some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

- The empty set , any single point (i.e. a singleton), $\{x_0\}$, and the whole space $\mathbb{R}^n$ are 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 1.1. 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 $\{x \in \mathbb{R}^n \mid \alpha \leq a^T x \leq \beta\}$.

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

(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, [1] 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.*

1.2 Convex Functions

Definition 1.2 (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 convex on* D *if*

$$f(x) = f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda) f(b) \tag{5}$$

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

Definition 1.3 (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 concave on* D *if*

$$f(x) = f(\lambda a + (1 - \lambda)b) \geq \lambda f(a) + (1 - \lambda) f(b) \tag{6}$$

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 above the function. A function is concave if a line segment connecting any two points within domain D is below 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 (5), (6) 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.

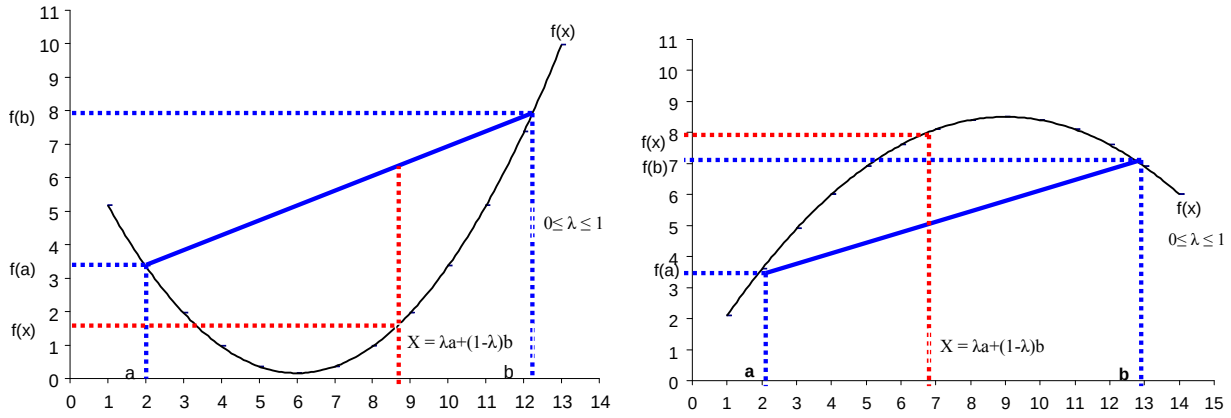

Figure 3: Visualization of convex [left] and concave [right] function definitions.

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

1.2.1 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 (5), (6), 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 $f(x) = \log\left(\exp^{x_1} + \cdots + \exp^{x_n}\right)$ 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 3.
- *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 (5), (6), or by checking that the Hessian is positive semi-definite (or negative semi-definite). These are left as exercises for the reader.

1.2.2 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.

If is easy to verify from (5) 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

$$f(x) = \alpha_1 f_1(x) + \cdots + \alpha_m f_m(x) \tag{7}$$

that is also convex.

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

$$f(x) = \max\{f_1(x), f_2(x)\} \tag{8}$$

is convex on $\mathcal{D}$. This property can be verified via (5) by considering $0 \leq \lambda \leq 1$ and $a, b \in \mathcal{D}$.

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

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

$$f(x) = \max\{f_1(x), \cdots, f_m(x)\} \tag{9}$$

is also convex.

2 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 2.1 (Global Minimizer)**.** *$x^* \in D$ is a global minimizer of $f(x)$ on D if*

$$f(x^*) \leq f(x), \qquad \forall\, x \in D \tag{10}$$

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

Definition 2.2 (Local Minimizer)**.** *$x^* \in D$ is a local minimizer of $f(x)$ on D if*

$$\exists\, \epsilon > 0 \quad \textit{s.t.} \quad f(x^*) \leq f(x), \qquad \forall\, x \in D \cap \{x \in \mathbb{R} \mid \|x - x^*\| < \epsilon\} \tag{11}$$

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.

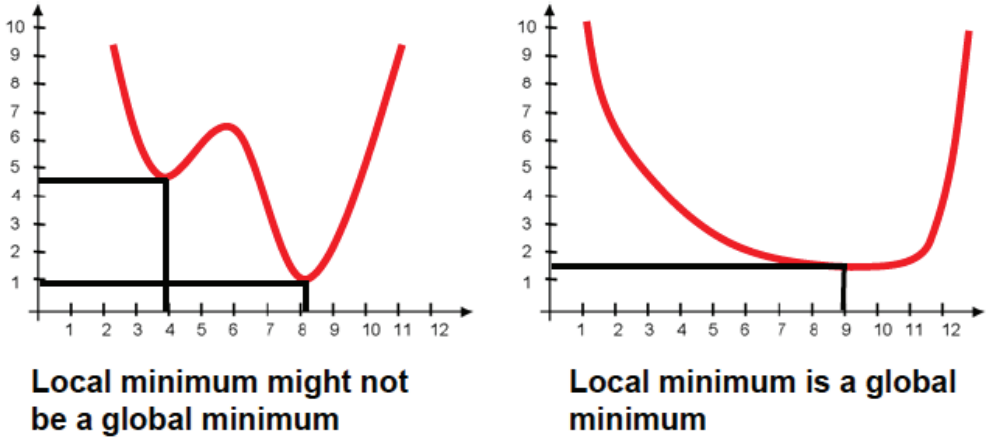

Figure 4: The LEFT figure contains two local minimizers, but only one global minimizer. The RIGHT figure contains a local minimizer, which is also the global minimizer.

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.

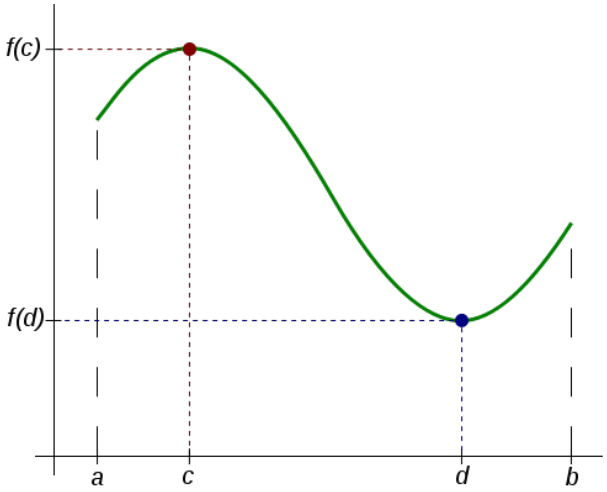

Figure 5: In this graph, $f(x)$ is continuous and bounded. The convex set is $D = [a, b]$. The function f attains a global minimum at $x = d$ and a global maximum at $x = c$.

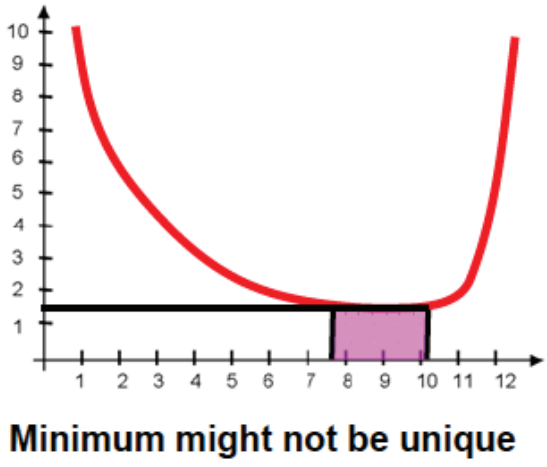

Figure 6: A local or global minimum need not be unique.

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.

2.1 Convex Problems

A *convex optimization problem* has the form

$$\min_{x} \quad f(x) \tag{12}$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \cdots, m \tag{13}$$

$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{14}$$

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 (14) 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 engineering design 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.

3 Gradient Descent

Next we investigate how to find optima in NLPs. 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. 7, 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

$$x_{k+1} = x_k - h \cdot \nabla f(x) \tag{15}$$

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

$$\|\nabla f(x_k)\| \leq \epsilon \tag{16}$$

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

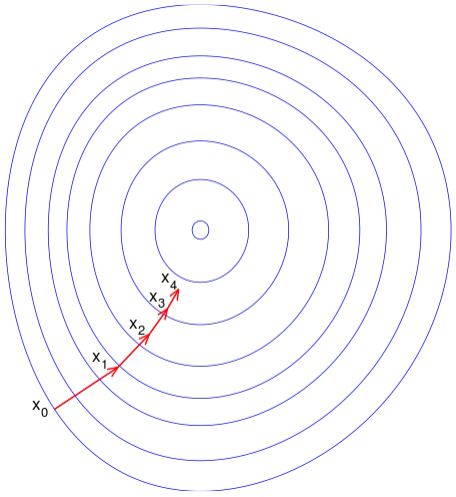

Figure 7: Illustration of gradient descent with step size proportional to the gradient.

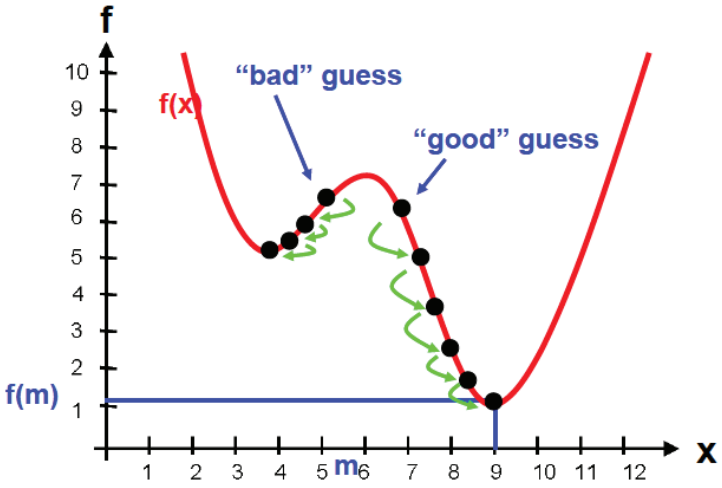

Figure 8: In non-convex functions, gradient descent converges to the local minimum. Consequently, different initial guesses may result in different solutions.

Exercise 4. *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* $\|\nabla f(x_k)\|_2 = \sqrt{x_{1,k}^2 + x_{2,k}^2} \leq \epsilon = 0.01$.

For non-convex problems, such as the one illustrated in Fig. 8, 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 an effective sub-optimal solution in practice.

4 Barrier & 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

$$\min_x \quad f(x) \tag{17}$$
$$\text{subject to} \quad g(x) \leq 0. \tag{18}$$

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

$$\min_x \quad f(x) + \phi(x;\epsilon) \tag{19}$$

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. 9. We seek to find the minimum within the range $[0.5, 1.5]$. Mathematically, this is a one-dimensional problem written as

$$\min_x \quad f(x) \tag{20}$$
$$\text{s. to} \quad x \leq b \tag{21}$$
$$x \geq a \tag{22}$$

4.1 Log Barrier Function

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

$$\phi(x;\varepsilon) = -\varepsilon \log\left(\frac{(x-a)(b-x)}{b-a}\right) \tag{23}$$

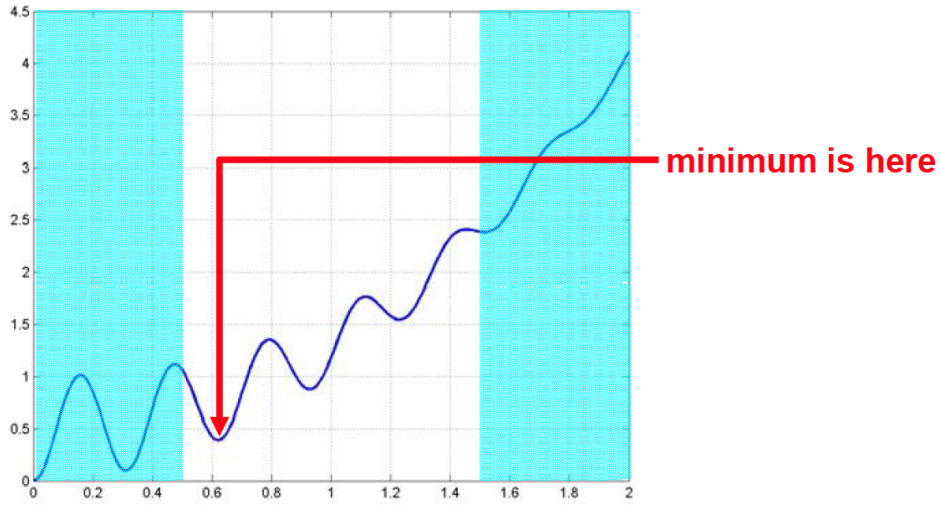

Figure 9: Find the optimum of the function shown above within the range [0.5, 1.5].

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 $\varepsilon \to 0$.

4.2 Quadratic Penalty Function

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

$$\phi(x;\varepsilon) = \begin{cases} 0 & \text{if } a \leq x \leq b \\ \frac{1}{2\varepsilon}(x-a)^2 & \text{if } x < a \\ \frac{1}{2\varepsilon}(x-b)^2 & \text{if } x > b \end{cases} \tag{24}$$

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 Optimality Conditions

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. We discover these conditions provide a set of nonlinear equations that can be solved to determine the optimal solution, under certain assumptions.

5.1 Method of Lagrange Multipliers

Consider the equality constrained optimization problem

$$\begin{aligned} \text{min} \quad & f(x) && (25)\\ \text{s. to} \quad & h_j(x) = 0, \qquad j = 1, \cdots, l && (26) \end{aligned}$$

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

$$\begin{aligned} L(x) &= f(x) + \sum_{j=1}^{l} \lambda_j h_j(x) && (27)\\ &= f(x) + \lambda^T h(x) && (28) \end{aligned}$$

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

$$\begin{aligned} &\frac{\partial L}{\partial x}(x^*) = \frac{\partial f}{\partial x}(x^*) + \lambda^T \frac{\partial h}{\partial x}(x^*) = 0 \qquad \textit{(stationarity)}, && (29)\\ &\frac{\partial L}{\partial \lambda}(x^*) = h(x^*) = 0 \qquad \textit{(feasibility)}. && (30) \end{aligned}$$

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

Remark 5.1. *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 5.2. *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 5.1. Consider the equality constrained QP

$$\begin{aligned} \text{min} \quad & \frac{1}{2} x^T Q x + R^T x && (31)\\ \text{s. to} \quad & Ax = b && (32) \end{aligned}$$

Form the Lagrangian,

$$L(x) = \frac{1}{2} x^T Q x + R^T x + \lambda^T (Ax - b). \tag{33}$$

Then the FONC is

$$\frac{\partial L}{\partial x}(x^*) = Qx^* + R + A^T \lambda = 0. \tag{34}$$

Combining the FONC with the equality constraint yields

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda \end{bmatrix} = \begin{bmatrix} -R \\ b \end{bmatrix} \tag{35}$$

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

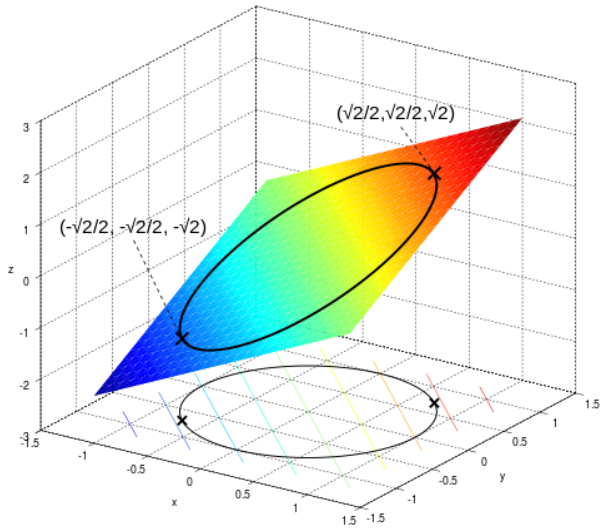

Figure 10: Visualization of circle-plane problem from Example 5.3.

Example 5.2. Consider a circle inscribed on a plane, as shown in Fig. 10. Suppose we wish to find the "lowest" point on the plane while being constrained to the circle. This can be abstracted as the NLP:

$$\begin{aligned} \text{min} \quad & f(x,y) = x + y && (36) \\ \text{s. to} \quad & x^2 + y^2 = 1 && (37) \end{aligned}$$

Form the Lagrangian

$$L(x,y,\lambda) = x + y + \lambda(x^2 + y^2 - 1) \tag{38}$$

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

$$\begin{aligned} \frac{\partial L}{\partial x} &= 1 + 2\lambda x = 0 && (39) \\ \frac{\partial L}{\partial y} &= 1 + 2\lambda y = 0 && (40) \\ \frac{\partial L}{\partial \lambda} &= x^2 + y^2 - 1 = 0 && (41) \end{aligned}$$

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

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

5.2 Karush-Kuhn-Tucker (KKT) Conditions

Now we consider the general constrained optimization problem

$$\text{min} \quad f(x) \tag{42}$$

$$\text{s. to} \quad g_i(x) \leq 0, \qquad i = 1, \cdots, m \tag{43}$$

$$h_j(x) = 0, \qquad j = 1, \cdots, l \tag{44}$$

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

$$L(x) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \tag{45}$$

$$= f(x) + \mu^T g(x) + \lambda^T h(x) \tag{46}$$

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

$$\frac{\partial f}{\partial x}(x^*) + \sum_{i=1}^{m} \mu_i \frac{\partial}{\partial x} g_i(x^*) + \sum_{j=1}^{l} \lambda_j \frac{\partial}{\partial x} h_j(x^*) = 0, \qquad \textit{Stationarity} \tag{47}$$

$$g_i(x^*) \leq 0, \quad i = 1, \cdots, m \qquad \textit{Feasibility} \tag{48}$$

$$h_j(x^*) = 0, \quad j = 1, \cdots, l \qquad \textit{Feasibility} \tag{49}$$

$$\mu_i \geq 0, \quad i = 1, \cdots, m \qquad \textit{Non-negativity} \tag{50}$$

$$\mu_i g_i(x^*) = 0, \quad i = 1, \cdots, m \qquad \textit{Complementary slackness} \tag{51}$$

which can also be written in matrix-vector form as

$$\frac{\partial f}{\partial x}(x^*) + \mu^T \frac{\partial}{\partial x} g(x^*) + \lambda^T \frac{\partial}{\partial x} h(x^*) = 0, \qquad \textit{Stationarity} \tag{52}$$

$$\begin{aligned} g(x^*) &\leq 0, && \textit{Feasibility} && (53)\\ h(x^*) &= 0, && \textit{Feasibility} && (54)\\ \mu &\geq 0, && \textit{Non-negativity} && (55)\\ \mu^T g(x^*) &= 0, && \textit{Complementary slackness} && (56) \end{aligned}$$

Remark 5.3. 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. 11. 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 $\nabla f(x^*) + \mu_1 \nabla g_1(x^*) + \mu_2 \nabla g_2(x^*) = 0$.

Example 5.3. Consider again the circle-plane problem, as shown in Fig. 10. 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:

$$\begin{aligned} \text{min} \quad & f(x,y) = x + y && (57)\\ \text{s. to} \quad & x^2 + y^2 \leq 1 && (58) \end{aligned}$$

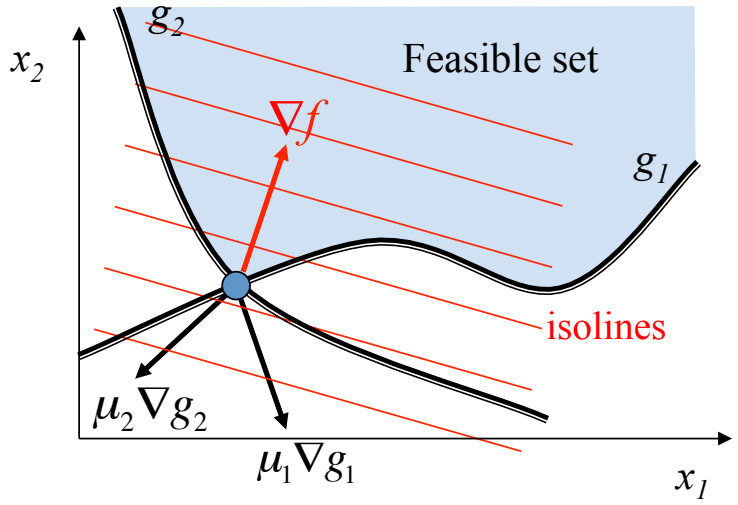

Figure 11: Geometric interpretation of KKT conditions

Form the Lagrangian

$$L(x, y, \mu) = x + y + \mu(x^2 + y^2 - 1) \tag{59}$$

Then the KKT conditions are

$$\begin{aligned}
\frac{\partial L}{\partial x} &= 1 + 2\mu x = 0 && (60)\\
\frac{\partial L}{\partial y} &= 1 + 2\mu y = 0 && (61)\\
\frac{\partial L}{\partial \mu} &= x^2 + y^2 - 1 \leq 0 && (62)\\
&\mu \geq 0 && (63)\\
&\mu(x^2 + y^2 - 1) = 0 && (64)
\end{aligned}$$

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 5.4 (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. 12. Reading left to right, the springs have spring constants k_1, k_2, k_3.

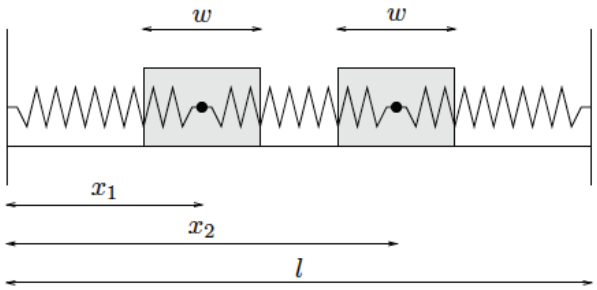

Figure 12: Spring-block system for Example 5.4

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.

$$\begin{aligned} \text{min} \quad & f(x_1, x_2) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2 \left(x_2 - x_1\right)^2 + \frac{1}{2}k_3 \left(l - x_2\right)^2 && (65)\\ \text{s. to} \quad & x_1 - \frac{w}{2} \geq 0, && (66)\\ & x_1 + \frac{w}{2} \leq x_2 - \frac{w}{2}, && (67)\\ & x_2 + \frac{w}{2} \leq l && (68) \end{aligned}$$

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:

$$L(x, \mu) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2 \left(x_2 - x_1\right)^2 + \frac{1}{2}k_3 \left(l - x_2\right)^2 + \mu_1 \left(\frac{w}{2} - x_1\right) + \mu_2 \left(x_1 - x_2 + w\right) + \mu_3 \left(x_2 + \frac{w}{2} - l\right) \quad (69)$$

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,

$$\mu_1 \left(\frac{w}{2} - x_1\right) = 0, \qquad \mu_2 \left(x_1 - x_2 + w\right) = 0, \qquad \mu_3 \left(x_2 + \frac{w}{2} - l\right) = 0 \quad (70)$$

for complementary slackness, and

$$\begin{bmatrix} k_1 x_1 - k_2 \left(x_2 - x_1\right) \\ k_2 \left(x_2 - x_1\right) - k_3 \left(l - x_2\right) \end{bmatrix} + \mu_1 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \mu_3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \quad (71)$$

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. 13, 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.

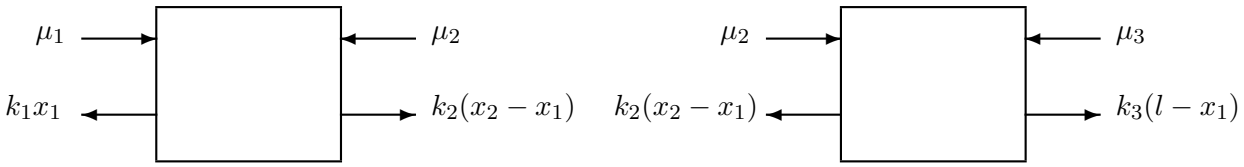

Figure 13: Free-body diagram of spring-block system for Example 5.4

6 Sensitivity Analysis

Until now, the Lagrange multipliers λ, μ seem like purely mathematical bi-products for developing necessary and sufficient conditions for local minima. Interestingly, the relative values of the Lagrange multipliers have an important physical interpretation. Namely, they represent the sensitivity of the minimized cost function with respect to perturbations in the constraints. We explain next.

Consider the perturbed version of the original NLP problem:

$$\begin{aligned} \min_x \quad & f(x), && (72)\\ \text{subject to} \quad & g_i(x) \leq u_i, \quad i = 1, \cdots, m && (73)\\ & h_j(x) = v_j, \quad j = 1, \cdots, l. && (74) \end{aligned}$$

where variables $u_i,\ i = 1, \cdots, m$ and $v_j,\ j = 1, \cdots, l$ represent perturbations in the constraints. Of course, when $u = 0, v = 0$, then this problem degenerates into the original NLP problem. When u_i is positive, then we have relaxed the ith inequality constraint. When u_i is negative, it means that we have tightened the constraint. Similarly, a non-zero v_j perturbs the equality constraint in the positive or negative direction.

Now we seek to study how the minimized cost function, denoted as $J^*(u, v)$, changes as a result of tightening or loosening each constraint. Assuming that $J^*(u, v)$ is differentiable with respect to u and v at $u = 0$ and $v = 0$, we seek to compute $\frac{\partial J^*}{\partial u}(0, 0)$ and $\frac{\partial J^*}{\partial v}(0, 0)$. Let us re-write the perturbed constraints into standard form by defining new constraint functions $\tilde{g}(x)$, $\tilde{h}(x)$.

$$\begin{aligned} \tilde{g}_i(x) &= g_i(x) - u_i = 0, \quad i = 1, \cdots, m && (75)\\ \tilde{h}_j(x) &= h_j(x) - v_j = 0, \quad j = 1, \cdots, l. && (76) \end{aligned}$$

The associated Lagrangian is

$$\begin{aligned} L(x,\lambda,\mu) &= f(x) + \lambda^T \tilde{h}(x) + \mu^T \tilde{g}(x), && (77)\\ &= f(x) + \lambda^T \left[h(x) - v\right] + \mu^T \left[g(x) - u\right]. && (78) \end{aligned}$$

Now, assume the optimal design x^* for given parameter perturbations (u, v) admits the minimized Lagrangian

$$L(x^*,\lambda,\mu) = f(x^*) + \lambda^T \left[h(x^*) - v\right] + \mu^T \left[g(x^*) - u\right]. \quad (79)$$

Then it is easy to see that $\frac{\partial J^*}{\partial u}(0,0) = \frac{\partial L}{\partial u}(x^*,\lambda,\mu)|_{u=0,v=0}$ and $\frac{\partial J^*}{\partial v}(0,0) = \frac{\partial L}{\partial v}(x^*,\lambda,\mu)|_{u=0,v=0}$. This renders the following sensitivities

$$\begin{aligned} \frac{\partial J^*}{\partial v}(0,0) &= -\lambda, && (80)\\ \frac{\partial J^*}{\partial u}(0,0) &= -\mu. && (81) \end{aligned}$$

In other words, the sensitivity of the minimized cost function to perturbations in the equality and inequality constraints is given by $-\lambda$ and $-\mu$ respectively. In other words, λ, μ gives us a quantitative measure of *how active* a constraint is at the optimum x^*. If inequality constraint $g_i(x^*) < 0$, then it follows that the constraint can be tightened or loosened a small amount without affecting the optimal value. By complementary slackness (51), the associated Lagrange multiplier $\mu_i = 0$. Now suppose that $g_i(x^*) = 0$, i.e. the ith constraint is active at the optimum. The ith Lagrange multiplier μ_i tells us how active the constraint is: If μ_i is small, it means that the constraint can be loosed a bit without much effect (but some) on the optimal cost; if μ_i is large, it means that if the constraint is loosed or tightening a bit, the effect on the optimal cost will be great.

Shadow Price Interpretation

We give a simple economics interpretation of the result (80)-(81). Consider a convex problem with no equality constraints. Let the design variable $x \in \mathbb{R}^n$ represent decisions for how a firm operates. Let $f(x)$ represent the firm's total cost, i.e. $-f(x)$ represents the firm's total profit, given decision variables x. Now suppose each constraint $g_i(x) \leq 0$ represents limits on some resource such as labor, steel, or warehouse space. The sensitivity $\frac{\partial J^*}{\partial u}$ tells us how much more or less profit could be made if more or less of each resource were made available to the firm. If is is differentiable, then we have

$$\frac{\partial J^*}{\partial u}(0,0) = -\mu. \quad (82)$$

In other words, $-\mu_i$ tells us how much more cost the firm incurs for a small increase in resource i. The negative of this statement is of course true, and perhaps easier to understand. Namely, μ_i tells us how much more profit the firm could make for a small increase in resource i.

It follows that μ_i would be the natural or equilibrium price for resource i, if it were possible for the firm to buy or sell it. Suppose, for example, that the firm can buy or sell resource i, at a price that is less than Lagrange multiplier μ_i. In this case it would certainly buy some of the resource, which would allow it to operate in a way that increases its profit more than the cost of buying the resource. Conversely, if the price exceeds μ_i, the firm would sell some of its allocation of resource i, and obtain a net gain since its income from selling some of the resource would be larger that its drop in profit due to the reduction in availability of the resource. Under these interpretations, we often call Lagrange multiplier μ_i the "shadow price".

7 Notes

Nonlinear programming (NLP) is an extremely rich field, worthy of its own course or set of courses. Today, the most widely read and referenced textbook in NLP is "Convex Optimization" by Boyd and Vandenverghe [1]. Chapters 1 and 2 of [1] provide further details on convex sets, convex functions with a litany of examples and exercises. More details on gradient descent, barrier functions, and penalty functions can be found in Chapter 7 of [2]. The theory of Lagrange multipliers, KKT conditions, and sensitivity analysis can be found, with further detail, in Ch. 5 of [1] and Ch. 5 of [2].

References

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

[2] P. Y. Papalambros and D. J. Wilde, Principles of Optimal Design: Modeling and Computation. Cambridge University Press, 2000.