1. The Language and Notation of Nonlinear Optimization

In finite dimensional optimization we are interested in locating solutions to the problem

$$\mathcal{P}: \quad \begin{array}{ll} \underset{x \in X}{\text{minimize}} & f_0(x) \\ \text{subject to} & x \in \Omega. \end{array}$$

where X is the variable space (or decision space), $f_0 : X \to \mathbb{R} \cup \{\pm\infty\}$ is called the objective function, and the set $\Omega \subset X$ is called the constraint region. The techniques that one employs in the study of $\mathcal{P}$ are determined by the nature of the space X, the function f_0, and the set Ω. The basic problem categories are as follows:

(1) Variable Type:
  (a) continuous variable: $X = \mathbb{R}^n$
  (b) discrete variable: $X = \mathbb{Z}^n$
  (c) mixed variable: $X = \mathbb{R}^s \times \mathbb{Z}^t$
(2) Constraint Type:
  (a) unconstrained: $\Omega = X$
  (b) constrained: $\Omega \neq X$
(3) Problem Type:
  (a) Convex Programming:
    f_0 is a convex function and Ω is a convex set.

Definition 1.1. *The set $\Omega \subset \mathbb{R}^n$ is said to be convex if for every $x, y \in \Omega$ one has $[x, y] \subset \Omega$ where $[x, y]$ denotes the line segment connecting x and y:*

$$[x, y] = \{\lambda x + (1 - \lambda)y : 0 \leq \lambda \leq 1\}.$$

Definition 1.2. *The function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is said to be convex if the set $epi(f) = \{(x, \mu) : f(x) \leq \mu\}$ is a convex set in $\mathbb{R}^{n+1}$. In particular,*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

for all $0 \leq \lambda \leq 1$ and points x, y for which not both $f(x)$ and $f(y)$ are infinite.

  (b) Linear Programming:
    The minimization or maximization of a linear functional subject to a finite number of linear inequality and/or equality constraints. $f_0(x) := C^T x$ for some $C \in \mathbb{R}^n$ and

$$\Omega := \left\{ x : a_i^T x \begin{array}{ll} \leq b_i & i = 1, \ldots, s \\ = b_i & i = s + 1, \ldots, m \end{array} \right\}.$$

    Linear programming is a special case of convex programming. In this case the constraint region Ω is called a polyhedral convex set. Polyhedra have a very special geometric structure.

(c) Quadratic Programming:

The minimization or maximization of a quadratic objective functions over a convex polyhedron:

$$f_0(x) = \frac{1}{2}x^TQx + b^Tx + \alpha$$

EXERCISE: Show that $\nabla f_0(x) = \frac{1}{2}(Q + Q^T)x + b$ and $\nabla^2 f_0(x) = \frac{1}{2}(Q + Q^T)$.

FACT: f_0 is convex if and only if Q is positive semi-definite.

(d) Mini-Max:

$$f_0(x) = \max\{f_i(x) : i = 1, \ldots, s\}.$$

(e) Nonlinear Programming

$$\Omega := \{x \in X : f_i(x) \leq 0, i = 1, \ldots, s, f_i(x) = 0, i = j + 1, \ldots m\}$$

(i) Differentiable: f_i is smooth $i = 0, \ldots, m$
(ii) Nonsmooth: at least one f_i is not smooth
(iii) Semi-Infinite: $m = +\infty$.

(f) Box Constraints:

$$\begin{aligned} \Omega &:= \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, \ldots, n\} \\ l_i &\in \mathbb{R} \cup \{-\infty\}, u_i \in \mathbb{R} \cup \{+\infty\}, l_i \leq u_i \end{aligned}$$

(g) Parameter Identification:

(i) Data Fitting:

Given data points $\{(x_i, y_i)\}_{i=1}^m \subset \mathbb{R}^t \times \mathbb{R}^s$, find the function $f(x) := m(p, x)$ from a parametrized class of functions

$$\mathcal{M} := \{m(p, \cdot) : p \in Y \subset \mathbb{R}^n\}$$

that "best" fits the data.

(A) Polynomial least-squares: $t = 1 = s$. $m(p, x) = p_0 + p_1x + \cdots + p_nx^n$.
$\mathcal{M} = \mathcal{P}_n =$ set of polynomials of degree less than or equal to n.
Minimize the sum of squares

$$f_0(P) = \sum_{i=1}^{n} (m(p, x_i) - y_i)^2$$

over all choices of $p \in \mathbb{R}^{n+1}$.
Approach:

$$\begin{bmatrix} p_0 + p_1x_1 + p_2x_1^2 + \cdots + p_nx_1^n \\ p_0 + p_1x_2 + p_2x_2^2 + \cdots + p_nx_2^n \\ \\ p_0 + p_1x_m + p_2x_m^2 + \cdots + p_nx_m^n \end{bmatrix} \begin{matrix} = & y_1 \\ = & y_2 \\ \\ = & y_m \end{matrix}$$

$$\begin{array}{c}\text{Vandermonde}\\\text{Matrix}\end{array}\left\{\underbrace{\begin{bmatrix}1 & x_1 & x_1^2 & \cdots & x_1^n\\ 1 & x_2 & x_2^2 & \cdots & x_2^n\\ \vdots & & & & \\ 1 & x_m & x_m^2 & \cdots & x_m^n\end{bmatrix}}_{A}\right. \begin{bmatrix}p_0\\ p_1\\ \vdots\\ p_n\end{bmatrix} = \begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_m\end{bmatrix}$$

$$A \qquad P \quad = \quad Y$$

$$f_0(P) = \|Ap - y\|_2^2$$

$$\text{where } \|z\|_2 = \left(\sum_{i=1}^{m} z_2^2\right)^{\frac{1}{2}}$$

EXERCISE: Show that $\nabla f_0(p) = A^T(Ap - y)$ and $\nabla^2 f_0(p) = A^T A$.

The primary focus of this course is the development of numerical methods for solving the general nonlinear programming problem given above under the assumption that all of the underlying functions are smooth. Nonetheless, on occasion we will also need to consider nondifferentiable functions. Our first objective is to come to some agreement about what we mean by a "solution" to $\mathcal{P}$.

Definition 1.3.

i. *A point $\overline{x} \in \Omega$ is said to be a global solution to the problem*

$$\min\{f_0(x) : x \in \Omega\}$$

if $f_0(\overline{x}) \le f_0(x)$ for all $x \in \Omega$. If in fact $f_0(\overline{x}) < f_0(x)$ for all $x \in \Omega$, then $\overline{x}$ is said to be a strict global solution.

ii. *A point $\overline{x} \in \Omega$ is said to be a local solution to the problem* $\min\{f_0(x) : x \in \Omega\}$ *if there is an $\epsilon > 0$ such that*

$$f_0(\overline{x}) \le f_0(x) \quad \textit{for all } x \in \Omega \textit{ satisfying } \|\overline{x} - x\| \le \epsilon.$$

If $f_0(\overline{x}) < f_0(x)$ for all $x \in \Omega$ with $\|x - \overline{x}\| \le \epsilon$, then $\overline{x}$ is called a strict local solution. The solution $\overline{x}$ is said to be isolated if $\overline{x}$ is the only local solution in the set

$$\{x \in \Omega : \|x - \overline{x}\| \le \epsilon\}.$$

EXAMPLES:

(1) Global: (a) $f_0(x) = x, \quad \Omega = [0,1]$ and (b) $f_0(x) = x^2, \quad \Omega = \mathbb{R}$ } strict

(2) local: (a) $f_0(x) = x(x+1)(x-1)$ (no global)
(b) $f_0(x) = (x-1)^2(x+1)^2$ (not strictly global)

(3) strict global but not isolated

$$\begin{aligned} f_0(x) &= x^2, \Omega := \{x \in [-1,1] : f_1(x) = 0\} \\ f_1(x) &= \begin{cases} x^2 \sin\frac{1}{x}, & x \ne 0 \\ 0, & x = 0 \end{cases} \end{aligned}$$

The above definitions establish what is meant by local and global solutions to $\mathcal{P}$. However, since the definition requires us to test f_0 at potentially infinitely many points in order to verify optimality, the definition in itself is not a constructive test of optimality. From the algorithmic point of view we require a construction test for optimality that can be used to indicate if a point is either optimal or nearly optimal. Such a test is referred to as an "optimality condition". Any point that satisfies an optimality condition is said to be a critical point, or stationary point, for $\mathcal{P}$.

There are many types of optimality conditions. We will concentrate on those involving differential information. Recall the following result from the calculus.

Theorem 1.1. *If f_0 is differentiable at $\overline{x}$ and $\overline{x}$ is a local solution to the problem* $\min\{f_0(x) : x \in \mathbb{R}^n\}$, *then* $\nabla f(\overline{x}) = 0$.

We will recall the proof of this result shortly. But for the moment observe that the condition

$$\nabla f_0(x) = 0$$

is an optimality condition for the unconstrained problem $\min\{f_0(x) : x \in \mathbb{R}^n\}$. This condition is clearly not a sufficient condition for optimality (consider $f_0(x) = x^3$). But it is necessary, indeed, we refer to it as a first-order necessary condition for optimality.

Note that the condition $\nabla f_0(x) = 0$ is testable. Simply evaluate ∇f_0 at $\overline{x}$. Moreover, it provides a ready means for testing "near" optimality, by considering the magnitude of the vector $\nabla f_0(\overline{x})$. Thus, for example, a reasonable stopping criteria for an algorithm would be

$$\|\nabla f(\overline{x})\| \leq \epsilon.$$

Hopefully, this would yield a finitely terminating procedure.

In what follows optimality conditions play a key role in both the design of our algorithms and our tests for termination. Indeed, we design our algorithms to locate points that satisfy some testable, or constructive, optimality conditions and then terminate when the procedure "nearly" solves the conditions. Thus, our first order of business is to derive workable optimality conditions. In order to do this we must first review some facts from multi-variable calculus.