I214 System Optimization
Chapter 5: Nonlinear Programming (Part A)

Brian Kurkoski

Japan Advanced Institute of Science and Technology

2023 January

Recall This Example of Nonlinear Programming

Fig. 1 is a typical curve indicating the time necessary for passing through some road. It is a function of the number of cars x on the road. When x is small, the necessary time can be seen as a constant. However, once x reaches some threshold value, it increases very rapidly.

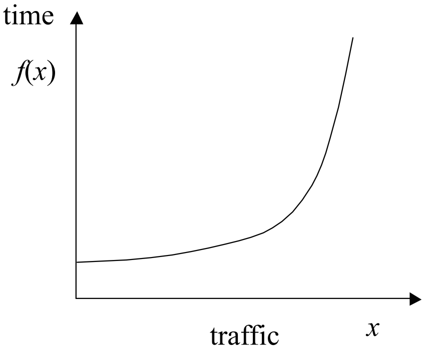


Figure 1: Time required to travel $f(x)$, as a function of the number of cars x.

Outline

5.1 Mathematical Background

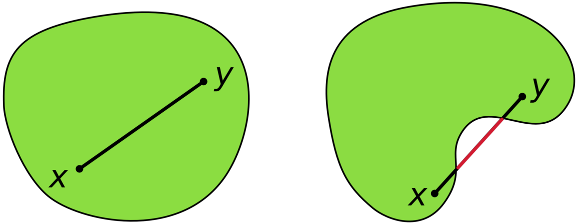
5.1.1 Convex Sets and Convex Functions

Definition

Let $\mathcal{D}$ be a subset of $\mathbb{R}^n$. Then $\mathcal{D}$ is a *convex set* if for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$:

$$\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in \mathcal{D}, \tag{1}$$

for all $0 \leq \lambda \leq 1$. A set which is not convex is called non-convex.



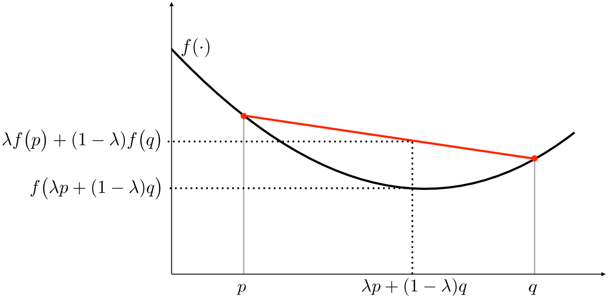
Convex Functions

Definition

Let f be a real-valued function with domain $\mathcal{D} \subseteq \mathbb{R}^n$. The function f is a *convex function* if for any, $\mathbf{p}, \mathbf{q} \in \mathcal{D}$:

$$f\big(\lambda \mathbf{p} + (1-\lambda)\mathbf{q}\big) \leq \lambda f(\mathbf{p}) + (1-\lambda) f(\mathbf{q}), \tag{2}$$

for $0 \leq \lambda \leq 1$.



Strict Convexity

The function is called *strictly convex* if

$$f\big(\lambda\mathbf{p} + (1-\lambda)\mathbf{q}\big) < \lambda f(\mathbf{p}) + (1-\lambda) f(\mathbf{q}), \tag{3}$$

for any $0 \leq \lambda \leq 1$ and $\mathbf{p} \neq \mathbf{q}$.

A function f is said to be (strictly) concave if $-f$ is (strictly) convex.

5.1.2 Differentiability

This chapter generally assumes that functions are first-order and second-order differentiable.

5.1.3 Gradient Vector and Hessian Matrix

Definition

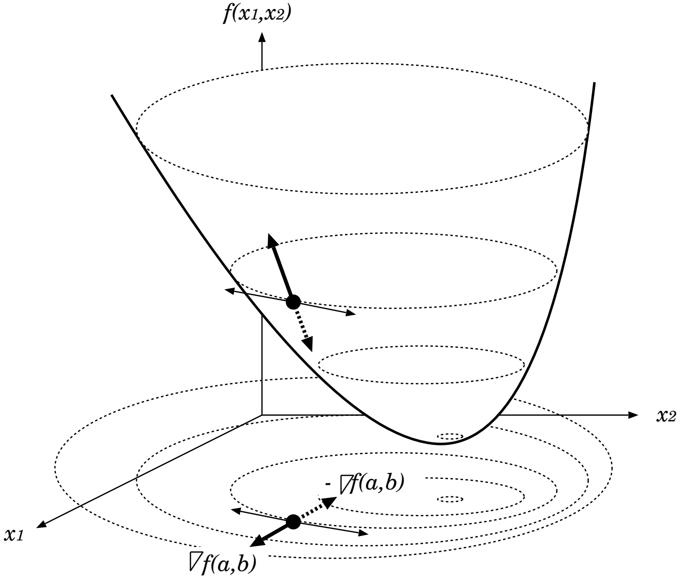
The *gradient* $\nabla f(\mathbf{x})$ of a scalar-valued differentiable function f at the point $\mathbf{x}$ is:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{bmatrix} \tag{4}$$

The gradient is the vector field (or vector-valued function) ∇f.

At a point $\mathbf{x}$, the gradient $\nabla f(\mathbf{x})$ is the direction of fastest increase.

Gradient Vector Illustration



Hessian Matrix

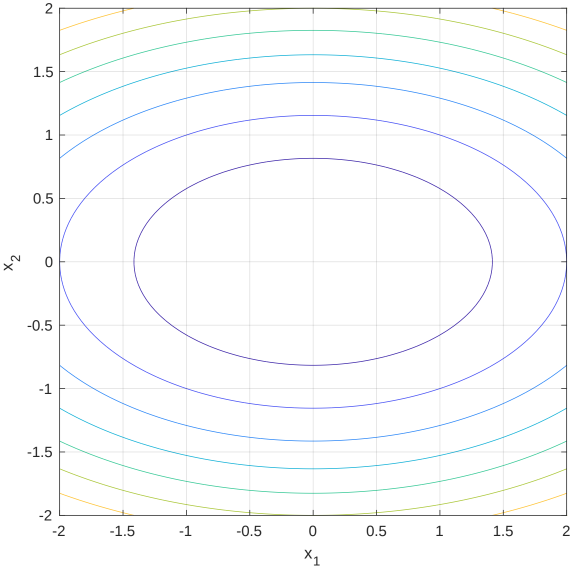
The *Hessian matrix* or Hessian is a square matrix of second-order partial derivatives of a function f. It describes the local curvature of a function of many variables.

The Hessian $\nabla^2 f(\mathbf{x})$ at a point $\mathbf{x}$ is defined as:

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix} \tag{5}$$

The Hessian matrix is a symmetrical matrix. ★1

Level Sets Example



5.1.4 Eigenvalues and Eigenvectors

Definition

Let $\mathbf{A}$ be an $n \times n$ matrix. Then $\mathbf{v}$ is an *eigenvector* of $\mathbf{A}$ if:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{6}$$

where λ is a scalar called the *eigenvalue*.

Proposition

The eigenvalues of a real symmetric matrix are all real.

The eigenvalues can be found by solving the *characteristic equation*. ★2 ★3

5.1.5 Definiteness of Matrices

Definition

For a real symmetric matrix $\mathbf{A}$, the *quadratic form* of $\mathbf{A}$ is:

$$\mathbf{x}^{\mathrm{t}}\mathbf{A}\mathbf{x}. \tag{7}$$

Definition

A symmetric matrix $\mathbf{A}$ is *positive semi-definite* if

$$\mathbf{x}^{\mathrm{t}}\mathbf{A}\mathbf{x} \geq 0$$

for all $\mathbf{x} \in \mathbb{R}^n$. Similarly, we have:

$$\begin{aligned} \text{positive definite: } & \mathbf{x}^{\mathrm{t}}\mathbf{A}\mathbf{x} > 0 \\ \text{negative semi-definite: } & \mathbf{x}^{\mathrm{t}}\mathbf{A}\mathbf{x} \leq 0 \\ \text{negative definite: } & \mathbf{x}^{\mathrm{t}}\mathbf{A}\mathbf{x} < 0 \end{aligned}$$

Eigenvalue Test for Definiteness

Proposition

Let $\mathbf{A}$ be an $n \times n$ real symmetric matrix (more generally, Hermitian matrices). The real eigenvalues of $\mathbf{A}$ are real, and their sign characterize its definiteness:

- $\mathbf{A}$ is positive definite if and only if all of its eigenvalues are positive.

Eigenvalue Test for Definiteness

Proposition

Let $\mathbf{A}$ be an $n \times n$ real symmetric matrix (more generally, Hermitian matrices). The real eigenvalues of $\mathbf{A}$ are real, and their sign characterize its definiteness:

- $\mathbf{A}$ is positive definite if and only if all of its eigenvalues are positive.
- $\mathbf{A}$ is positive semi-definite if and only if all of its eigenvalues are non-negative.
- $\mathbf{A}$ is negative definite if and only if all of its eigenvalues are negative
- $\mathbf{A}$ is negative semi-definite if and only if all of its eigenvalues are non-positive.
- $\mathbf{A}$ is indefinite if and only if it has both positive and negative eigenvalues.

Example

Another Definiteness Test — Sylvester's Criterion

Proposition

A symmetric matrix $\mathbf{A}$ is positive definite if and only if all its leading principal minors are positive.

Proposition

A symmetric matrix $\mathbf{A}$ is positive semi-definite if and only if all its principal minors are nonnegative.

5.2 Nonlinear Programming

5.2.1 Definitions

Linear programming problem is characterized by the linearity of both objective function and constraint function.

If one or both of the objective function and the constraint function are nonlinear with respect to variables, it is called *nonlinear programming problem*.

A general statement of the optimization problem is:

$$\begin{aligned} \textbf{Minimize} \quad & f(\mathbf{x}) \\ \textbf{subject to} \quad & h_i(\mathbf{x}) = 0, \;\; i = 1, 2, \ldots, \ell \\ & g_j(\mathbf{x}) \leq 0, \;\; j = 1, 2, \ldots, m \end{aligned} \tag{8}$$

★4

5.2.2 Locally Optimal Solution

Let $\mathcal{B}_n(\mathbf{x}, r)$ be a sphere with its center at $\mathbf{x}$ and its radius r in n-dimensional Euclidean space, that is,

$$\mathcal{B}_n(\mathbf{x}, r) = \{\mathbf{y} \in \mathbb{R}^n| \ ||\mathbf{x} - \mathbf{y}|| \leq r\}.$$

Definition

A feasible point $\mathbf{x}^* \in \mathcal{S}$ is called a *local optimum* or *local minimum* if there exists $\delta > 0$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in \big(\mathcal{S} \cap \mathcal{B}_n(\mathbf{x}^*, \delta)\big).$$

Locally Optimal Solution

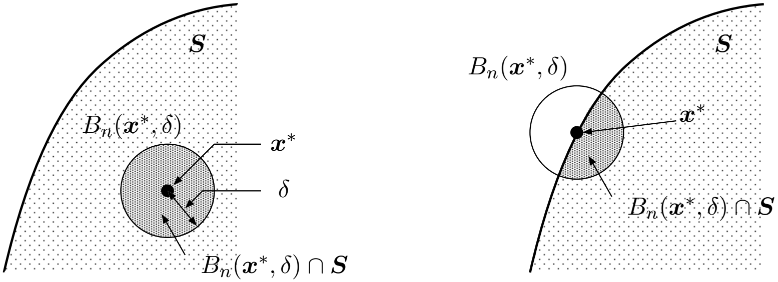


Illustration of $B_n(\mathbf{x}^*, \delta) \cap \mathcal{S}$. The left hand side is for $\mathbf{x}^*$ inside of $\mathcal{S}$, the right hand side for $\mathbf{x}^*$ on the border of $\mathcal{S}$.

Linear Programming Versus Nonlinear Programming

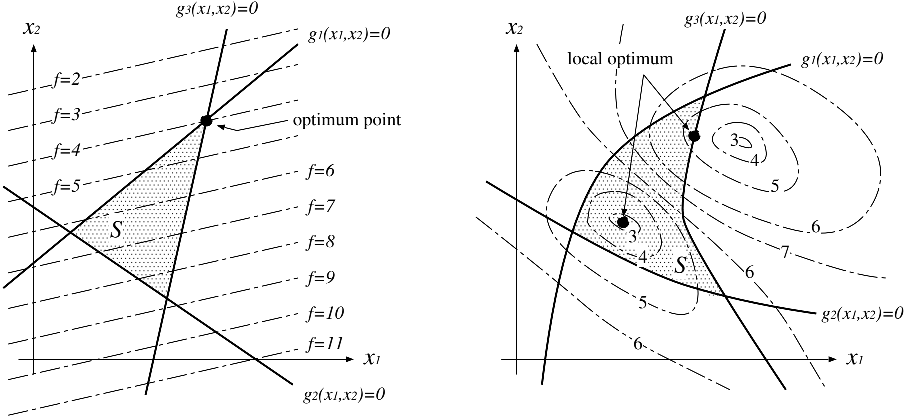


Figure 3: Differences between linear programming and nonlinear programming. (1) boundary: straight vs. curved. (2) feasible region: convex vs. nonconvex (3) objective function: flat surface vs curved surface (4): local optimum: corner of feasible region vs. interior.

5.2.3 Convex Programming

An optimization problem with a convex objective function and its convex feasible region is called a *convex programming problem*.

Recall that the Hessian matrix is a symmetrical matrix.

Proposition

For a convex function, Hessian matrix at any point is positive semi-definite.
Conversely, a function whose Hessian matrix is positive semi-definite at any point is a convex function.

Convex Programming

Proposition

In a convex programming problem, a locally optimal solution is a globally optimal solution.

★5

Class Info

- Homework 5 (Ford-Fulkerson; Exercise 4.5). Do not submit. Solutions will be published before midterm exam.
- Next lecture: Friday, January 20 at 9:00. Continue Nonlinear Programming
- Midterm exam will be Friday, January 20 13:30–15:10. Paper-based in the classroom.