Introduction to Optimization II

Asst. Prof. Dr.-Ing. Sudchai Boonto

Department of Control Systems and Instrumentation Engineering, KMUTT

August 11, 2025

General Form

The General Form of Optimization Problem

$$
\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\
& h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, l \\
& \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U
\end{aligned}
$$

- $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable
- $f(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective or cost function
- $g_i : \mathbb{R}^n \mapsto \mathbb{R}, i = 1, \ldots, m$, are the inequality constraint functions
- $h_i : \mathbb{R}^n \mapsto \mathbb{R}, i = 1, \ldots, l$, are the equality constraint functions.

Standard form: Top Brass Data

Top Brass Trophy Company makes large championship trophies for youth athletic leagues. At the moment, they are planning production for fall sports: US football and football. Each US football trophy has a wood base, an engraved plaque, a large brass US football on top, and returns $12 in profit. Football trophies are similar except that a brass football ball is on top, and the unit profit is only $9. Since the US football has an asymmetric shape, its base requires 4 board feet of wood; the football base requires only 2 board feet. At the moment there are 1000 brass US footballs in stock, 1500 football balls, 1750 plaques, and 4800 board feet of wood. What trophies should be produced from these supplies to maximize total profit assuming that all that are made can be sold?

US football | football | both

Standard form: Top Brass Data

Recipe for building each trophy

	wood	plaques	US footballs	soccer balls	profit
US football	4 ft	1	1	0	$ 12
football	2 ft	1	0	1	$ 9

Quantity of each ingredient in stock

	wood	plaques	US football balls	Football balls
in stock	4800 ft	1750	1000	1500

Top Brass Problem
Example

$$\begin{aligned} \underset{f,s}{\text{maximize}} \quad & 12f + 9s \\ \text{subject to} \quad & 4f + 2s \leq 4800 \\ & f + s \leq 1750 \\ & 0 \leq f \leq 1000 \\ & 0 \leq s \leq 1500 \end{aligned}$$

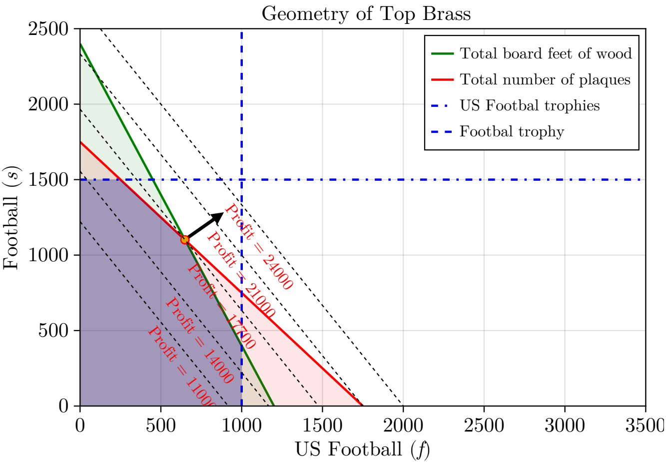
Matrix form

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

This is in matrix form, with:

$$\mathbf{A} = \begin{bmatrix} 4 & 2 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4800 \\ 1750 \\ 1000 \\ 1500 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 12 \\ 9 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} f \\ s \end{bmatrix}$$

Define $z = 12f + 9s$, where $z =$ profit. Here $s = -\frac{12}{9}f + \frac{z}{9}$



Solve Top Brass Problem with MATLAB

```
% Objective function (note: linprog minimizes, so we negate the coefficients)
f_obj = [-12; -9];  % Maximize becomes minimize of negative

% Inequality constraints: A*x ≤ b
A = [4, 2;
     1, 1];
b = [4800;
     1750];

% Variable bounds
lb = [0; 0];          % Lower bounds for f and s
ub = [1000; 1500];    % Upper bounds for f and s

% Solve using linprog
options = optimoptions('linprog','Display','none');  % Suppress output
[x_opt, fval] = linprog(f_obj, A, b, [], [], lb, ub, options);

% Display results
f = x_opt(1);
s = x_opt(2);
max_profit = -fval;  % Negate to get the original maximized value

fprintf('Optimal f: %.2f\n', f);
fprintf('Optimal s: %.2f\n', s);
fprintf('Maximum profit: %.2f\n', max_profit);
```

Solve Top Brass Problem with Julia

```julia
using JuMP, GLPK

m = Model(GLPK.Optimizer)
@variable(m, 0 <= f <= 1000)    # US football trophies
@variable(m, 0 <= s <= 1500)    # football trophies

@constraint(m, 4f + 2s <= 4800) # total board feet of wood
@constraint(m, f + s <= 1750)   # total number of plagues

@objective(m, Max, 12f + 9s)    # maximize profit

# Printing the prepared optimization model
print(m)

# Solving the optimization problem
JuMP.optimize!(m)

# Printing the optimal solutions obtained
println("Build ", JuMP.value(f), " US football trophies.")
println("Build ", JuMP.value(s), " footbal trophies.")
println("Total profit will be \$", JuMP.objective_value(m))
```

Julia code: Top Brass Problem

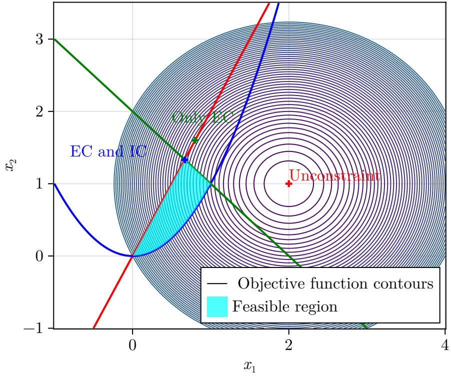
General Form

A Simple Problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \dots, l \end{aligned}$$

Example:

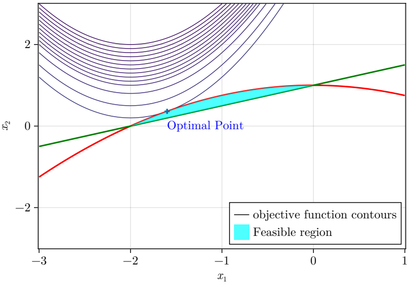
$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{subject to} \quad & x_2 - 2x_1 = 0 \\ & x_1^2 - x_2 \leq 0 \\ & x_1 + x_2 \leq 2 \end{aligned}$$



- Unconstrained Optimal Solution: $\mathbf{x} = \begin{bmatrix} 2 & 1 \end{bmatrix}$
- Solution with only equality con.: $\mathbf{x} = \begin{bmatrix} 0.8 & 1.6 \end{bmatrix}$ (The constraint must be active.)
- Solution with both types of con.: $\mathbf{x} = \begin{bmatrix} 0.67 & 1.33 \end{bmatrix}$

Graphical Solution of One- and Two-Variable Problems

For $n = 1$ or $n = 2$, we could plot the NLP problem to get the solution. The example shows the plotting of the feasible region Ω and objective function contours for the NLP problem with $n = 2$ as follow:



$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & (x_1 + 2)^2 - x_2 \\ \text{subject to} \quad & \frac{x_1^2}{4} + x_2 - 1 \leq 0 \\ & 2 + x_1 - 2x_2 \leq 0 \end{aligned}$$

The real optimal solution is obtained as

$$x_1^* = -1.6, \quad x_2^* = 0.36$$

Julia code: Graphical Method

Matlab code: Graphical Method

Chaâteau Laupt-Himum

The Château Laupt-Humum produces rosé wine and red wine by buying grapes from local producers. This year they can buy up to one ton of Pinot (a read grape) from a winegrower, paying €3 per kilo. They can then vinify the grapes in two ways: either as a white wine to obtain a rosé wine or as a red wine to get Pinot Noir, a full-bodied red wine. The vinification of the rosé wine costs €2 per kilo of grapes, while that of the Pinot Noir costs €3.50 per kilo of grapes.
In order to take into account economies of scale, the Château wants to adjust the price of its wine to the quantity produced. The price for one liter of rosé is €15 minus a rebate of €2 per hundred liters produced Similarly, they sell the Pinot Noir at a price of €23 per liter, minus a rebate of €1 per hundred liters produced.
How should the Château Laupt-Himum be organized in order to maximize its profit, when a kilo of grapes produces 1 liter of wine?

rosé wine red wine Pinot Noir

Chaâteau Laupt-Himum

There are three variables:

- x_1 is the number of liters of rosé wine to produce each year,
- x_2 is the number of liters of Pinot Noir to produce,
- x_3 is the number of kilos of grapes to buy.

The objective is to maximize the profit. We have (terms in red color are rebate.)

- Each liter of rosé wine that is sold gives (in €): $15 - \frac{2}{100}x_1$
- Each liter of Pinot Noir gives (in €): $23 - \frac{1}{100}x_2$
- The revenues corresponding to the production of x_1 liters or rosé wine and x_2 liters of Pinot Noir are equal to $x_1\left(15 - \frac{2}{100}x_1\right) + x_2\left(23 - \frac{1}{100}x_2\right)$
- Grape is €3 per kilo, and a liter of wine need one kilo of vinified grapes, which costs €2 for the rosé and €3.50 for the Pinot Noir: $2x_1 + 3.5x_2 + 3x_3$

The the objective function is

$$f(\mathbf{x}) = x_1\left(15 - \frac{2}{100}x_1\right) + x_2\left(23 - \frac{1}{100}x_2\right) - (2x_1 + 3.5x_2 + 3x_3),$$

and the constraints are $x_1, x_2, x_3 \geq 0$, and $x_1 + x_2 \leq x_3$

Chaâteau Laupt-Himum

We combine the modeling steps to obtain the following optimization problem:

$$
\begin{aligned}
\underset{x_1, x_2, x_3}{\text{maximize}} \quad & f(\mathbf{x}) = x_1 \left(15 - \frac{2}{100} x_1\right) + x_2 \left(23 - \frac{1}{100} x_2\right) - (2x_1 + 3.5x_2 + 3x_3) \\
\text{subject to} \quad & x_1 + x_2 \leq x_3 \\
& x_3 \leq 1000 \\
& x_1 \geq 0 \\
& x_2 \geq 0 \\
& x_3 \geq 0
\end{aligned}
$$

Note: The problem is higher than two-dimension space, so it cannot be solved by using graphical method.

Optimization problems

We consider the following optimization problem with continuous design variables,

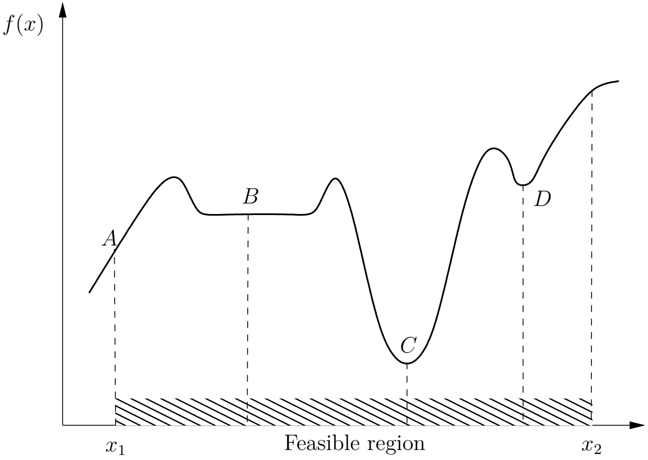
$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}), \\ \text{subject to} \quad & h(\mathbf{x}) = 0 \\ & g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbf{X} \end{aligned}$$

where $\mathbf{x} = [x_1, \dots, x_n]$ is composed of the design variables that the optimization algorithm can change.

Minimum Points:

- the point $\mathbf{x}^*$ is a **weak local minimum** if there exists a $\delta > 0$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ such that $|\mathbf{x} - \mathbf{x}^*| < \delta$, that is $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ in a δ-neighborhood of $\mathbf{x}^*$.
- the point $\mathbf{x}^*$ is a **strong local minimum** if there exists a $\delta > 0$ such that $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x}$ such that $|\mathbf{x} - \mathbf{x}^*| < \delta$.
- $\mathbf{x}^*$ is a **global minimum** if $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x}$

Unconstrained optimization problems



Briefly details of important terms:

- **dom** f here means domain of the function.
- The set $\{x : |x| \leq 1\}$ is an example of a *closed* set while $\{x : |x| < 1\}$ is an *open* set.
- A set is *bounded* if it is contained within some sphere of finite radius, i.e. for any point $\mathbf{a}$ in the set, $\mathbf{a}^T\mathbf{a} < c$, where c is a finite number.
- For example, the set of all positive integers, $\{1, 2, \ldots\}$, is not bounded.
- A set that is both closed and bounded is called a *compact* set.

Let $f(\mathbf{x})$ be a *continuous function* defined over a *closed* and *bounded* set $\Omega \subset$ **dom** f. Then, there exist points $\mathbf{x}^*$ and $\mathbf{x}^{**}$ in Ω where f attains its minimum and maximum, respectively. That is $f(\mathbf{x}^*)$ and $f(\mathbf{x}^{**})$ are the minimum and maximum values of f in the set.

Let $\mathbf{x}^*$ and $\mathbf{x}^{**}$ denote the solutions, if one exists, to the minimization, and maximization problems

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \qquad\qquad \begin{aligned} \text{maximize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned}$$

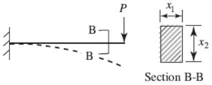
Consider a problem

$$\text{minimize} \quad x \qquad \text{subject to} \qquad 0 < x \leq 1$$

This simple problem does not have a solution. We observe that the constraint set Ω is not closed. Re-writing the constraint as $0 \leq x \leq 1$ results in $x = 0$ being the solution.

Existence of a Minimum and a Maximum: Weierstrass Theorem

Consider the cantilever beam in Fig. with a tip load P and tip deflection δ. The cross-section is rectangular with width and height equal to x_1, x_2, respectively. It is desired to minimize the tip deflection with given amount of material, or



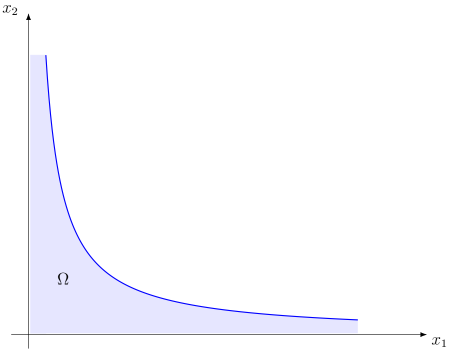
$$\begin{aligned} \text{minimize} \quad & \delta \\ \text{subject to} \quad & A \leq A_0 \end{aligned}$$

This problem does not have a solution. It is *ill-posed*. This can be seen by substituting the beam equation

$$\delta = \frac{PL^2}{3EI} = \frac{c}{x_1 x_2^3},$$

where c is a known scalar. Owing to the fact that x_2 is cubed in the denominator of the δ expression, with a given $A = x_1 x_2$ (the cross section area), the solution will tend to increase x_2 and reduce x_1. That is, the beam will tend to be infinitely slender with $x_1 \mapsto 0$, $x_2 \mapsto \infty$.

Existence of a Minimum and a Maximum: Weierstrass Theorem



Look at the feasible region Ω shown that it is unbounded, violating the simple conditions in Weierstrass theorem.

Consider a function

$$f(x_1, x_2) = x_1^2 - 6x_1x_2 + 9x_2^2$$

We can write $f(x_1, x_2)$ in a matrix notation as:

$$\begin{aligned} f(x_1, x_2) &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \mathbf{x}^T \mathbf{A} \mathbf{x} \end{aligned}$$

The matrix $\mathbf{A}$ is symmetric since

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \left(\frac{\mathbf{A} + \mathbf{A}^T}{2} \right) \mathbf{x}$$

$$\mathbf{B} = \frac{\mathbf{A} + \mathbf{A}^T}{2} \text{ is a symmetric matrix.}$$

positive definiteness

- A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called *positive semidefinite*, denoted by $\mathbf{A} \succeq 0$, if $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \; \forall \mathbf{x} \in \mathbb{R}^n$.
- A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called *positive definite*, denoted by $\mathbf{A} \succ 0$, if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \; \forall \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{x} \neq 0$.

Example Let

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \text{ for any } \mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \in \mathbb{R}^2$$

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2x_1^2 - 2x_1x_2 + x_2^2 = x_1^2 + (x_1 - x_2)^2 \geq 0.$$

The matrix $\mathbf{A}$ is positive semidefinite. Since $x_1^2 + (x_1 - x_2)^2 = 0$ if and only if $x_1 = x_2 = 0$ it follows that $\mathbf{A}$ is positive definite. **Note:** the *negative (semi)definite* is just the opposite sign of the positive (semi)definite.

The matrix, whose components are all positive, is note positive definite since for $\mathbf{x} = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$, we have

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \qquad \mathbf{x}^T \mathbf{A} \mathbf{x} = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = -2$$

eigenvalue characterization (proof is omit.)

Let $\mathbf{A}$ be a symmetric $n \times n$ matrix. Then

- $\mathbf{A}$ is positive definite if and only if all its eigenvalues are positive,
- $\mathbf{A}$ is positive semidefinite if and only if all its eigenvalues are nonnegative,
- $\mathbf{A}$ is negative definite if and only if all its eigenvalues are negative,
- $\mathbf{A}$ is negative semidefinite if and only if all its eigenvalues are nonpositive,
- $\mathbf{A}$ is indefinite if and only if it has at least one positive eigenvalue and at lease one negative eigenvalue.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \qquad \det(\lambda\mathbf{I} - \mathbf{A}) = \begin{vmatrix} \lambda - 1 - 2 \\ -2 \lambda - 1 \end{vmatrix} = 0, \qquad \lambda = -1, 3$$

It is obvious that the matrix $\mathbf{A}$ is not positive definite.

Sylvester's Test for a Positive definite Matrix

Let $\mathbf{A}^i$ denote the submatrix formed by deleting the last $n - i$ rows and columns of $\mathbf{A}$, and let $\det(\mathbf{A}^i)$ is the determinant of $\mathbf{A}^i$. Then, $\mathbf{A}$ is positive definite if and only if $\det(\mathbf{A}^i) > 0$ for $i = 1, 2, \ldots, n$. This is, the determinants of all principal minors are positive.

- $\mathbf{A}$ is positive (semi-)definite if and only if all its leading principal minors are positive, i.e. , $\det \mathbf{A}^i > (\geq) 0$ for all i.
- $\mathbf{A}$ is negative (semi-)definite if and only if all the leading principal minors of $-\mathbf{A}$ are positive, i.e., $\det -\mathbf{A}^i > (\geq) 0$ for all i
- $\mathbf{A}$ is indefinite if neither positive or negative definite.

Consider a matrix $\mathbf{A}$ below:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & -1 \\ 3 & -1 & 2 \end{bmatrix}$$

We have

$$\mathbf{A}^1 = 1 > 0, \quad \mathbf{A}^2 = \begin{vmatrix} 1 & 2 \\ 2 & 5 \end{vmatrix} = 5 - 4 = 1 > 0, \quad \mathbf{A}^3 = \begin{vmatrix} 1 & 2 & 3 \\ 2 & 5 & -1 \\ 3 & -1 & 2 \end{vmatrix} = -56 < 0$$

Thus , $\mathbf{A}$ is not positive definite.

Check the characteristic of the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 4 \\ -2 & 2 & 0 \\ 4 & 0 & -7 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 4 & -2 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & 50 \end{bmatrix}$$

$\mathbf{A}$ is indefinite,and $\mathbf{B}$ is positive definite.

$\mathbb{C}^n$ Continuity of a Function

Continuity

The function f is $\mathbb{C}^0$ *continuous* at a point a if : given any sequence $\{x_k\}$ in **dom**(f) which converts to a, then $f(x_k)$ must converge to $f(a)$. Further, f is continuous over a set $\mathbb{S}$ implies that it is continuous at each pint in $\mathbb{S}$.

$\mathbb{C}^1$ and $\mathbb{C}^2$ Continuity

Let A be an open set of $\mathbb{R}^n$ and $f : \mathbb{R}^n \mapsto \mathbb{R}^1$. if each of the functions $\frac{\partial f}{\partial x_i}, i = 1, \ldots, n$, is continuous on this set then we write $f \in \mathbb{C}^1$ or that f is $\mathbb{C}^1$ continuous or state that f is "smooth". If each of the functions $\frac{\partial f}{\partial x_i, \partial x_j}, 1 \leq i, j \leq n$, is continuous on the set, then we write $f \in \mathbb{C}^2$.

Example: Consider the function f with **dom** $f \in \mathbb{R}^1$:

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

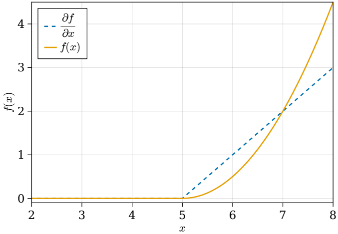
The sequence $x_k = 1/k, k = 1, 2, 3, \ldots$ converges (from the right) to $a = 0$ but the sequence $f(x_k) = 1$ for all k and does not converge to $f(a) = f(0) = 0$. Thus, f is discontinuous at $a = 0$.

Consider

$$f(x) = \frac{1}{2}\max(0, x-5)^2, \qquad x \in \mathbb{R}^1$$

$$\frac{\partial f}{\partial x} = \max(0, x-5)$$

$$\frac{\partial^2 f}{\partial x^2} = \begin{cases} 0 & , \text{ for } x \leq 5 \\ 1 & , \text{ for } x > 5 \end{cases}$$



The first derivative is a continuous function, while the second derivative is not continuous at $x = 5$. Thus f is only $\mathbb{C}^1$ (and not $\mathbb{C}^2$) continuous on $\mathbb{R}^1$. ($a = 5$ and $f''(a) = 5$ but $f''(x_\infty) = 1$)

See ch1/positive_def.jl, $df/dx = \frac{1}{2}(2)\max(0, x-5)d(\max(0, x-5))/dx = \max(0, x-5)$

Derivatives and Gradient

We consider differentiable and continuous functions of n variables $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ denoted by $f(\mathbf{x})$ or $f(x_1, x_2, \ldots, x_3)$.

- **The derivative with respect to the ith parameter** is given by:

$$\frac{\partial f}{\partial x_i} = \lim_{\Delta x_i \to 0} \frac{f(x_1, x_2, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, x_2, \ldots, x_i, \ldots, x_n)}{\Delta x_i}$$
$$\approx \frac{f(x_1, x_2, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, x_2, \ldots, x_i, \ldots, x_n)}{\Delta x_i}$$

- **Gradient Vector:** Given a function $f(\mathbf{x}) \in \mathbb{C}^1$, we introduce the *gradient vector* ∇f, a column vector , as

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right]^T$$

The gradient evaluated at a point $\mathbf{c}$ is denoted by $\nabla f(\mathbf{c})$.

Derivatives and Gradient

Example Consider the function:

$$f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 - x_1 x_2$$

Determine the gradient of the function at the point $\mathbf{x}_0 = \begin{bmatrix} 1.0 & 1.0 \end{bmatrix}^T$

Solution:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2(x_1 - 2) - x_2 \\ 2(x_2 - 1) - x_1 \end{bmatrix}$$

At the point $\mathbf{x}_0$

$$\nabla f = \begin{bmatrix} -3 \\ -1 \end{bmatrix}$$

Jacobian

Sometime we need to consider a number of functions at the same time, like the response of a circuit at a number of frequencies. In this case we have

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & \cdots & f_m \end{bmatrix}^T$$

The Jacobian matrix is the matrix combining all these gradients together and is given by

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \nabla f_1^T \\ \nabla f_2^T \\ \vdots \\ \nabla f_m^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Jacobian

Consider the vector of functions $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 3x_1^2 + x_2^3 & x_1^3 - 2x_3 \end{bmatrix}^T$. Determine the Jacobian at the point $\mathbf{x}_0 = \begin{bmatrix} 2.0 & 1.0 & 1.0 \end{bmatrix}^T$

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \nabla f_1^T \\ \nabla f_2^T \end{bmatrix} = \begin{bmatrix} 6x_1 & 3x_2^2 & 0 \\ 3x_1^2 & 0 & -2 \end{bmatrix}$$

At the point $\mathbf{x}_0$, the Jacobian is given by:

$$\mathbf{J}(\mathbf{x}_0) = \begin{bmatrix} \nabla f_1^T \\ \nabla f_2^T \end{bmatrix} = \begin{bmatrix} 12 & 3 & 0 \\ 12 & 0 & -2 \end{bmatrix}$$

Second-order derivatives, Hessian Matrix

The mixed second-order derivativewith respect to the ith and jth parameters is given by:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \lim_{\Delta x_j \to 0} \frac{\frac{\partial f(x_1, x_2, \ldots, x_j + \Delta x_j, \ldots, x_n}{\partial x_i} - \frac{\partial f(x_1, x_2, \ldots, x_i, \ldots, x_n)}{\partial x_i}}{\Delta x_j}$$

The gradient of a vector of m-function, $\mathbf{H} = \nabla \mathbf{g}$ is defined to be a matrix of dimension $(n \times m)$ as

$$\mathbf{H} = \nabla \mathbf{g} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_2}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \frac{\partial g_1}{\partial x_2} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \frac{\partial g_2}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

The Hessian matrix is a symmetric matrix because the order of differentiation does not make a difference.

Second-order derivatives, Hessian Matrix

Evaluate the gradient and the Hessian of the function
$f(\mathbf{x}) = 3x_1^2 + 2x_1x_2 + x_1x_3 + 2.5x_2^2 + 2x_2x_3 + 2x_3^2 - 8x_1 - 3x_2 - 3x_3$ at the point
$\mathbf{x}_0 = \begin{bmatrix} 1.0 & -1.0 & 1.0 \end{bmatrix}^T$

$$\nabla f(\mathbf{x}) = \mathbf{g}(\mathbf{x}) = \begin{bmatrix} 6x_1 + 2x_2 + x_3 - 8 \\ 2x_1 + 5x_2 + 2x_3 - 3 \\ x_1 + 2x_2 + 4x_3 - 3 \end{bmatrix} \quad \Rightarrow \quad \nabla f(\mathbf{x}_0) = \begin{bmatrix} -3 \\ -4 \\ 0 \end{bmatrix}$$

$$\mathbf{H} = \frac{\partial \mathbf{g}(\mathbf{x})^T}{\partial \mathbf{x}} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

Note:

$$f(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 3 & 1 & 0.5 \\ 1 & 2.5 & 1 \\ 0.5 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} -8 & -3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{x} = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{b}^T \mathbf{x}$$

Gradient and Hessian with Matlab

```
syms x1 x2 x3

f = 3*x1^2 + 2*x1*x2 + x1*x3 + 2.5*x2^2 + 2*x2*x3 ...
    + 2*x3^2 - 8*x1 - 3*x2 - 3*x3

g = gradient(f,[x1, x2, x3])
h = hessian(f,[x1, x2, x3])
g1 = subs(g, [x1, x2, x3], [1, -1, 1])
```

ans = $\begin{pmatrix} 6\,x_1 + 2\,x_2 + x_3 - 8 \\ 2\,x_1 + 5\,x_2 + 2\,x_3 - 3 \\ x_1 + 2\,x_2 + 4\,x_3 - 3 \end{pmatrix}$ ans = $\begin{pmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{pmatrix}$

ans = $\begin{pmatrix} -3 \\ -4 \\ 0 \end{pmatrix}$

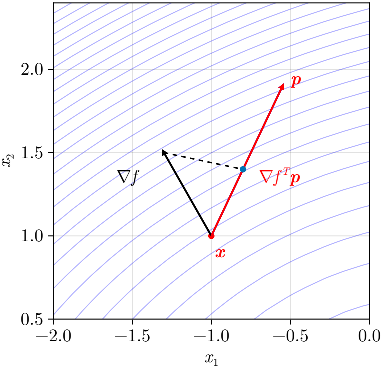
Gradient and Hessian with Julia

```julia
using Symbolics, ForwardDiff

@variables x1 x2 x3

f(x) = 3x[1]^2 + 2x[1]x[2] + x[1]x[3] + 2.5x[2]^2 + 2x[2]x[3] +
                        2x[3]^2 - 8x[1] - 3x[2] - 3x[3]

gf(x) = ForwardDiff.gradient(f,x)
Hf(x) = ForwardDiff.hessian(f,x)

g = gf([x1, x2, x3])
h = Hf([x1, x2, x3])
g1 = gf([1, -1, 1])
```

Directional Derivative

Directional Derivative: The rate of change in a direction $\mathbf{p}$ is quantified by a *directional derivative*, defined as

$$\nabla_{\mathbf{p}} f(\mathbf{x}) = \lim_{\tau \to 0} \frac{f(\mathbf{x} + \tau \mathbf{p}) - f(\mathbf{x})}{\tau} = \nabla f^T \mathbf{p} = \|\nabla f\| \|\mathbf{p}\| \cos\theta$$



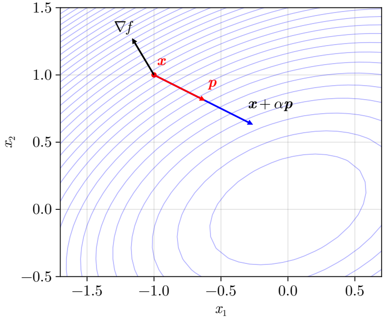
Directional Derivative

Consider the following function of two variables: $f(x_1, x_2) = x_1^2 + 2x_2^2 - x_1 x_2$. The gradient can be obtained using symbolic differentiation, yielding

$$\nabla f(x_1, x_2) = \begin{bmatrix} 2x_1 - x_2 \\ 4x_2 - x_1 \end{bmatrix}, \qquad \nabla f(-1, 1) = \begin{bmatrix} -3 \\ 5 \end{bmatrix}.$$

Taking the derivative in the normalized direction $\mathbf{p} = \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$, we obtain

$$\nabla f^T \mathbf{p} = \begin{bmatrix} -3 & 5 \end{bmatrix} \begin{bmatrix} 2/\sqrt{5} \\ -1/\sqrt{5} \end{bmatrix} = -\frac{11}{\sqrt{5}}$$



Directional Derivative

We wish to compute the directional derivative of $f(\mathbf{x}) = x_1 x_2$ at $x = [1,\ 0]$ in the direction $\mathbf{p} = [-1,\ -1]$:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} x_2 & x_1 \end{bmatrix}^T, \qquad \nabla_{\mathbf{p}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{p} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -1$$

We can also compute the directional derivative as follows:

- Introduce a scalar variable α and denote points along the vector $\mathbf{p}$ emanating from $\mathbf{x}$ as $g(\alpha) = \mathbf{x} + \alpha\mathbf{p}$
- Denote the function $f(\alpha) = f(g(\alpha)) = f(\mathbf{x} + \alpha\mathbf{p})$. We have

$$\nabla_{\mathbf{p}} f(\mathbf{x}) = \left.\frac{df}{d\alpha}\right|_{\alpha=0} = \left.\frac{\partial f}{\partial g}\right|_{\alpha=0} \frac{\partial g}{\partial \alpha} = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \mathbf{p} = \nabla f(\mathbf{x})^T \mathbf{p}$$

- from above example, we have

$$g(\alpha) = f(\mathbf{x} + \alpha\mathbf{p}) = (x_1 + \alpha p_1)(x_2 + \alpha p_2) = (1 - \alpha)(-\alpha) = \alpha^2 - \alpha$$

$$g'(\alpha) = 2\alpha - 1, \quad g'(0) = -1$$

Curvature and Hessian

The rate of change of the gradient–the *curvature*–is also useful information because it tells us if a function's slope is increasing (positive curvature), decreasing (negative curvature), or stationary (zero curvature). Given a function $f(x_1, \ldots, x_n) \in \mathbb{C}^2$, we define the matrix of second partial derivatives

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

The Hessian is a symmetric matrix with $n(n+1)/2$ independent elements.

Curvature and Hessian

We can find the rate of change of the gradient in an arbitrary normalized direction $\mathbf{p}$ by taking the product $\mathbf{H_p}$.

$$\mathbf{H_p} = \nabla_{\mathbf{p}}(\nabla f(\mathbf{x})) = \lim_{\tau \to 0} \frac{\nabla f(\mathbf{x} + \tau \mathbf{p}) - \nabla f(\mathbf{x})}{\tau}$$

To find the curvature of the one-dimensional function along a direction $\mathbf{p}$, we need to project $\mathbf{H_p}$ onto direction $\mathbf{p}$ as

$$\nabla_{\mathbf{p}}(\nabla_{\mathbf{p}} f(\mathbf{x})) = (\mathbf{H}^T \mathbf{p})^T \mathbf{p} = \mathbf{p}^T \mathbf{H} \mathbf{p}$$

which yields a scalar quantity. If we want to get the curvature in the original units of $\mathbf{x}$, $\mathbf{p}$ should be normalized.

Curvature and Hessian

Consider $f(x_1, x_2) = x_1^2 + 2x_2^2 - x_1x_2$. The Hessian of this quadratic is

$$\mathbf{H} = \nabla(\nabla f(x)) = \nabla\left(\begin{bmatrix} 2x_1 - x_2 \\ 4x_2 - x_1 \end{bmatrix}\right) = \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix}$$

To find the curvature in the direction $\mathbf{p} = \begin{bmatrix} -1/2 & -\sqrt{3}/2 \end{bmatrix}^T$, we compute

$$\mathbf{p}^T\mathbf{H}\mathbf{p} = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} = \frac{7 - \sqrt{3}}{2}$$

Taylor's Theorem, Linear and Quadratic Approximations

Suppose that $f(x) \in \mathbb{C}^p$ on an interval $\mathbf{J} = [a, b]$. If x_0, x belong to $\mathbf{J}$, then there exists a number γ between x_0 and x such that

$$f(x) \approx f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(p-1)}(x_0)}{(p-1)!}(x - x_0)^{(p-1)} + \frac{f^{(p)}(x_0)}{p!}(x - x_0)^p$$

- **Linear approximation:** $f_l(x) \approx f(x_0) + f'(x_0)(x - x_0)$
- **Quadratic approximation:** $f_q(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2$

For n-dimension we have

$$f_q(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \text{ H.O.T.}$$

Given $f(\mathbf{x}) = 2x_1 + \frac{x_2}{x_1}$. Construct linear and quadratic approximations to the original function $f(\mathbf{x})$ at $\mathbf{x}_0 = [1\ 0.5]^T$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2 - \frac{x_2}{x_1^2} \\ \\ \frac{1}{x_1} \end{bmatrix}, \qquad \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{x_2}{x_1^3} & -\frac{1}{x_1^2} \\ \\ -\frac{1}{x_1^2} & 0 \end{bmatrix}$$

The linear approximation is

$$f_l(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) = 2.5 + \begin{bmatrix} 1.5 & 1 \end{bmatrix}^T \begin{bmatrix} x_1 - 1 \\ x_2 - 0.5 \end{bmatrix}$$

$$= 0.5 + 1.5x_1 + x_2$$

The quadratic approximation is

$$f_q(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

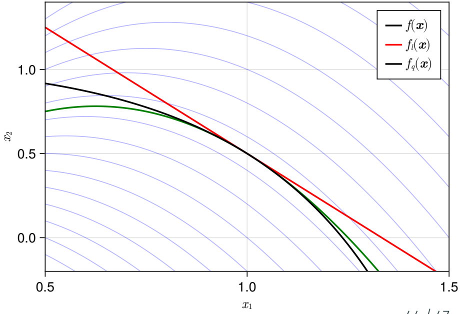
Taylor's Theorem, Linear and Quadratic Approximations

$$f_q(\mathbf{x}) \approx 0.5 + 1.5x_1 + x_2 + \frac{1}{2}\begin{bmatrix}(x_1 - 1) & (x_2 - 0.5)\end{bmatrix}\begin{bmatrix}1 & -1 \\ -1 & 0\end{bmatrix}\begin{bmatrix}x_1 - 1 \\ x_2 - 0.5\end{bmatrix}$$
$$= 0.5 + x_1 + 2x_2 - x_1x_2 + \frac{1}{2}x_1^2$$

Plotting: We need to plot, in variable-space or x-space, the contours

$$f(\mathbf{x}) = c, f_l(\mathbf{x}) = c, f_q(\mathbf{x}) = c,$$

where $c = f(\mathbf{x}_0) = 2.5$



Taylor's Theorem, Linear and Quadratic Approximations using Julia

```julia
using Symbolics, ForwardDiff, LinearAlgebra

@variables x1 x2
f(x) = 2x[1] + x[2]/x[1]
gf = (f, x) -> ForwardDiff.gradient(f,x)
Hf = (f, x) -> ForwardDiff.hessian(f,x)

x0 = [1, 0.5]
fl(x) = f(x0) + gf(f, x0)'*(x - x0)
fq(x) = f(x0) + gf(f, x0)'*(x - x0) + (1/2)*(x - x0)'*H(f, x0)*(x - x0)
println(expand(fl([x1, x2])))
println(expand(fq([x1, x2])))
```

Taylor's Theorem, Linear and Quadratic Approximations using Matlab

```
syms x1 x2 real

f = 2*x1 + x2/x1

x0 = [1 ,0.5];
g = gradient(f, [x1, x2])
H = hessian(f, [x1, x2 ])

f0 = subs(f,[x1, x2], x0)
g0 = subs(g,[x1, x2], x0)
H0 = subs(H, [x1, x2], x0)
dx = [x1; x2] - x0';

fl = f0 + g0'*dx
fq = expand(f0 + g0'*dx + (0.5)*dx'*H0*dx)
```

Reference

1. Joaquim R. R. A. Martins, Andrew Ning, ”*Engineering Design Optimization*,” Cambridge University Press, 2021
2. Michel Bierlair, ”Optimization: Principles and Algorithms,” EPFL Press, 2018
3. Ashok D. Belegundu, Tirupathi R. Chandrupatla, ”Optimization Concepts and Applications in Engineering,” Cambridge University Press, 2019