Contents

1. Linear Algebra

Vector	$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$	
Scalar Vector Multiplication	$c \in \mathbb{R} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ $c\vec{x} = \begin{bmatrix} cx_1 \\ cx_2 \\ \vdots \\ cx_n \end{bmatrix}$	
Dot Product	$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ $\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$ $= x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$ $= \vec{x}^T \vec{y}$	
Cross Product ($\mathbb{R}^3$)	$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$ $\vec{c} = \vec{a} \times \vec{b}$ $\vec{c} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$	Returns a vector orthogonal to the two vectors
Vector Space	Closure under addition $\vec{u}, \vec{v} \in V \quad \Rightarrow \quad \vec{u} + \vec{v} \in V$ Closure under scalar multiplication $\vec{v} \in V \wedge c \in \mathbb{R} \quad \Rightarrow \quad c\vec{v} \in V$ Commutativity of addition $\vec{u} + \vec{v} = \vec{v} + \vec{u}$ Associativity of addition	

	$(\vec{u}+\vec{v})+\vec{w}=\vec{u}+(\vec{v}+\vec{w})$ Additive identity $\exists \mathbf{0} \in V \mid \vec{v}+\mathbf{0}=\vec{v}$ Additive inverse $\forall \vec{v} \in V \exists -\vec{v} \in V \mid \vec{v}+(-\vec{v})=0$ Scalar multiplication (compatibility) $a(b\vec{v})=(ab)\vec{v}$ Distributivity over vector addition $a(\vec{u}+\vec{v})=a\vec{u}+a\vec{v}$ Distributivity over scalar addition $(a+b)\vec{v}=a\vec{v}+b\vec{v}$ Multiplicative identity $1\vec{v}=\vec{v}$	
Subspace	Non-emptiness $\mathbf{0} \in V$ Closure under addition $\text{If } \vec{u}, \vec{v} \in V, \text{then } \vec{u}+\vec{v} \in V$ Closure under scalar multiplication $\text{If } \vec{v} \in V, c \in \mathbb{R}, \text{then } c\vec{v} \in V$	A subspace is a subset of a vector space that is itself a vector space, satisfying the same axioms as the original. If V is a vector space in $\mathbb{R}^n$, then the subspace U is always contained in $\mathbb{R}^n$, meaning $U \subseteq \mathbb{R}^n$
Vector Addition	$\vec{u}=(u_1, u_2, ..., u_n)$ $\vec{v}=(v_1, v_2, ..., v_n)$ $\vec{u}+\vec{v}=(u_1+v_1, u_2+v_2, ..., u_n+v_n)$	
Dot Product	$\vec{u} \cdot \vec{v}=\sum_{i=1}^{n} u_i v_i$	
Orthogonality	$\vec{u} \cdot \vec{v}=\mathbf{0}$	Angle between the two vectors is $90°$
Angle between vectors	$\Theta=\arccos\left(\frac{\vec{u}\cdot\vec{v}}{\lVert\vec{u}\rVert_2 \cdot \lVert\vec{v}\rVert_2}\right)$	
L_1 Norm (Manhattan)	$\lVert\vec{u}\rVert_1=\sum_{i=1}^{n}\lvert u_i\rvert$	
L_2 Norm (Euclidean)	$\lVert\vec{u}\rVert_2=\sqrt{\sum_{i=1}^{n} u_i^2}$	
L_1 Distance (Manhattan)	$d(\vec{u}, \vec{v})=\sum_{i=1}^{n}\lvert u_i-v_i\rvert$	
L_2 Distance (Euclidean)	$d(\vec{u}, \vec{v})=\sqrt{\sum_{i=1}^{n}(u_i-v_i)^2}$	
Projection	$\text{proj}_w(\vec{v})=\frac{\vec{v}\cdot\vec{w}}{\vec{w}\cdot\vec{w}}\vec{w}$	

Linear Independence		A set of vectors is linearly independent if no vector in the set can be written as a linear combination of the others A set of vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ is linearly independent if the only solution to the the equation $$c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_n\vec{v}_n = \mathbf{0}$$ is $c_1 = c_2 = \ldots = c_n = 0$
Transformation	$$T : \mathbb{R}^n \to \mathbb{R}^m$$ $$T(\vec{v}) = A\vec{v}$$ • Additivity $$T(\vec{u} + \vec{v}) = T(\vec{u}) + T(\vec{v})$$ • Homogeneity $$T(c\vec{u}) = cT(\vec{u})$$	• Surjective (onto) Every element in B is the image of at least one element in A. The transformation covers the entire codomain. $\mathrm{Range}(T) = B$ • Injective (one-to-one) Different inputs in A map to different outputs in B.The transformation is information-preserving — doesn't collapse distinct vectors together $T(\vec{x}_1) = T(\vec{x}_2) \Rightarrow x_1 = x_2$ Or equivalently: $\ker(T) = \{\mathbf{0}\}$
Domain	$$T : V \to W$$ $$\mathrm{Domain}(T) = V$$	Set of all **input vectors** V that the **transformation** acts on
Codomain	$$T : V \to W$$ $$\mathrm{Codomain}(T) = W$$	Set of all possible **output vectors** W to which elements of the domain V are mapped under the **transformation**
Transpose	$$\det(A) = \det(A^T)$$ $$(AB)^T = B^T A^T$$ $$(A^T)^{-1} = (A^{-1})^T$$	
Image	$$T : V \to W$$ $$\mathrm{Im}(T) = \{\vec{w} \in W \mid \vec{w} = T(\vec{v}) \text{ for some } \vec{v} \in V\}$$	The image of a **transformation** $T : V \to W$ is the set of all possible outputs $T(v)$ for $v \in V$: • It is a subspace of the codomain W • If T is represented by a matrix A, the image of T is the column space of A
Preimage	$$T : V \to W$$ $$T^{-1}(\vec{w}) = \{\vec{v} \in V \mid T(\vec{v}) = \vec{w}\}$$	The preimage of a transformation refers to the set of all elements in the domain that

		map to a particular element or subset in the codomain
Span	$\text{Span}(\{v_1, v_2, \ldots, v_k\}) = \left\{\sum_{i=1}^{n} c_i \vec{v}_i \mid c_i \in \mathbb{R}\right\}$	The span of a set of vectors is the collection of all possible linear combinations of those vectors
Composition	$T_1 : \mathbb{R}^n \to \mathbb{R}^m \qquad T_2 : \mathbb{R}^m \to \mathbb{R}^p$ $T_1(\vec{v}) = A\vec{v} \qquad T_2(\vec{v}) = B\vec{v}$ $T \circ S(\vec{v}) = T_2(T_1(\vec{v}))$ $T_2 \circ T_1(\vec{v}) = B(A\vec{v}) = (BA)\vec{v}$ • Additivity $(T_3 \circ T_2) \circ T_1 = T_3 \circ (T_2 \circ T_1)$ • Homogeneity $T_2 \circ T_1(c\vec{u}) = c(T_2 \circ T_1)(\vec{u})$ • Identity Transformation $I \circ T = T \qquad T \circ I = T$	
Column Space (Range)	$\text{Col}(A) = \{Ax \mid \vec{x} \in \mathbb{R}^n\}$ Or equivalently $A = [\vec{c}_1 \quad \vec{c}_2 \quad \ldots \quad \vec{c}_3]$ $\text{Col}(A) = \text{span}(\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_3)$	The column space (or range) of a **matrix** A is the set of all linear combinations of its columns

Determinant	$\det(A)$	The determinant of a square matrix A measure of the "scale factor" by which the matrix A transforms a space
		• $\det(A) \neq 0$
		  ‣ A does not collapse the space
		  ‣ A has full rank
		  ‣ A's columns are linearly independent
		  ‣ A is invertable
		• $\det(A) = 0$
		  ‣ A collapses the space into lower dimension
		  ‣ A does not have full rank
		  ‣ A's columns are linearly dependent
		  ‣ A is non-invertable (singular)

Invertibility	$\det(A) \neq 0 \implies \text{Invertible}$ $\det(A) = 0 \implies \text{Non-Invertible}$ $AA^{-1} = A^{-1}A = I_n$	

<table>
<tr><td></td><td>$(AB)^{-1} = B^{-1}A^{-1}$
$(A^T)^{-1} = (A^{-1})^T$</td><td></td></tr>
<tr><td>Basis</td><td>Linear Independence
$c_1\vec{v_1} + c_2\vec{v_2} + \ldots + c_k\vec{v_k} = \mathbf{0}$
$\Rightarrow c_1 = c_2 = \ldots = c_k = 0$
Spanning
$\forall \vec{v} \in V, \exists c_1, \ldots, c_k \in \mathbb{R} \quad s.t.$
$\vec{v} = c_1\vec{v_1} + \ldots + c_k\vec{v_k}$</td><td>• A basis of a vector space V is a set of linearly independent vectors that span the space
• Every vector in V can be uniquely written as a linear combination of the basis vectors
E.g.:</td></tr>
<tr><td>Dimension</td><td>$\dim(V)$</td><td>Number of linearly independent vectors (basis) in a vector space V
$V \subseteq \mathbb{R}^n$
<table>
<tr><td>$\dim(V) = 0$</td><td>$V = \{\mathbf{0}\}$</td></tr>
<tr><td>$\dim(V) = 1$</td><td>V is a line through the origin in $\mathbb{R}^n$</td></tr>
<tr><td>$\dim(V) = 2$</td><td>V is a plane through the origin in $\mathbb{R}^n$</td></tr>
<tr><td>$\dim(V) = k$</td><td>V is a k-dimensional flat subspace of $\mathbb{R}^n$</td></tr>
<tr><td>$\dim(V) = n$</td><td>$V = \mathbb{R}^n$</td></tr>
</table>
• $\mathbb{R}^2$ has dimension 2:
  ‣ A basis: $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$
• $\mathbb{R}^3$ has dimension 3:
  ‣ A basis: $\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$</td></tr>
<tr><td>Rank</td><td>$\text{Rank}(A) = \dim(\text{Col}(A)) = \dim(\text{Row}(A))$</td><td>• The rank of a matrix A is the dimension of its column space (or row space)
• Number of linearly independent columns (or rows)</td></tr>
<tr><td>Eigen</td><td>$Ax = \lambda x, \quad x \neq \mathbf{0}$</td><td>Set of all nonzero vectors $\vec{x}$ such that when the transformation represented by matrix A is applied to $\vec{x}$, the result is a scaled version of $\vec{x}$ itself
These vectors lie along directions that are preserved by the transformation:
• $|\lambda| > 1$: stretched
• $0 < |\lambda| < 1$: shrunk
• $\lambda < 0$: flipped
• $\lambda = 1$: stay the same</td></tr>
</table>

Null Space (kernel)	$\text{Null}(A) = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \mathbf{0}\}$	The null space of a matrix A is the set of all input vectors that get mapped to the zero vector when you multiply them by A
Identity Matrix	$I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$	
Matrix Inverse	$A \cdot A^{-1} = I$	
RREF	1. Row Swapping (Interchange) $R_1 \leftrightarrow R_2$ 2. Row Scaling (Multiplication) $R_1 \rightarrow \frac{1}{3}R_1$ 3. Row Addition (Replacement) $R_1 \rightarrow R_1 - 2R_2$	

1.1. Matrix

$$\begin{array}{cc} & n \\ m & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \end{array})$$

1.1.1. Matrix Vector Product

$$\begin{array}{cc} & n \\ m & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \end{array} \qquad \begin{array}{cc} n & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \end{array}$$

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix}$$

1.1.2. Matrix Multiplication

$$A = \begin{array}{cc} & n \\ m & \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \end{array} \qquad B = \begin{array}{cc} & p \\ n & \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1p} \\ b_{21} & b_{22} & \ldots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{np} \end{bmatrix} \end{array}$$

$$A = \begin{array}{cc} & n \\ m & \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \end{bmatrix} \\ \begin{bmatrix} a_{21} & a_{22} & \ldots & a_{2n} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \end{bmatrix} \end{array} \qquad B = \begin{array}{cc} & p \\ n & \begin{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{bmatrix} & \begin{bmatrix} b_{12} \\ b_{22} \\ \vdots \\ b_{n2} \end{bmatrix} & \ldots & \begin{bmatrix} b_{1p} \\ b_{2p} \\ \vdots \\ b_{np} \end{bmatrix} \end{bmatrix} \end{array}$$

A: Row Representation

$$A = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}$$

$$r_i = \begin{bmatrix} a_{i1} & a_{i2} & \ldots & a_{in} \end{bmatrix}, \quad \text{for } i = 1, 2, \ldots, m$$

B: Column Representation

$$B = \begin{bmatrix} c_1 & c_2 & \ldots & c_p \end{bmatrix}$$

$$c_j = \begin{bmatrix} b_1j \\ b_2j \\ \vdots \\ b_{nj} \end{bmatrix}, \quad \text{for } j = 1, 2, \ldots, p$$

$$C = \begin{array}{cc} & p \\ m & \begin{bmatrix} r_1 \cdot c_1 & r_1 \cdot c_2 & \ldots & r_1 \cdot c_p \\ r_2 \cdot c_1 & r_2 \cdot c_2 & \ldots & r_2 \cdot c_p \\ \vdots & \vdots & \ddots & \vdots \\ r_m \cdot c_1 & r_m \cdot c_2 & \ldots & r_m \cdot c_p \end{bmatrix} \end{array}$$

1.1.3. Transpose

$$A^T$$

m

n

$$A = \begin{array}{cc} m & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \end{array} \qquad A^T = \begin{array}{cc} n & \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \end{array}$$

$$A = \begin{array}{cc} & n \\ m & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ij} & \cdots & a_{mn} \end{bmatrix} \end{array} \qquad A^T = \begin{array}{cc} & m \\ n & \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{i1} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{i2} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{1j} & a_{2j} & \cdots & a_{ij} & \cdots & a_{mj} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{in} & \cdots & a_{mn} \end{bmatrix} \end{array}$$

1.2. Matrix Factorization

1.2.1. LU Decomposition

$$A = LU$$

where:

- L: lower triangular matrix (entries above the diagonal are zero)
- U: upper triangular matrix (entries below the diagonal are zero)

1. $m \times n$ Matrix (with $m \geq n$)

$$L = \begin{array}{cc} & n \\ m & \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mn} \end{bmatrix} \end{array} \qquad U = \begin{array}{cc} & n \\ n & \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{bmatrix} \end{array}$$

2. $m \times n$ Matrix (with $m < n$)

see QR Decomposition

1.3. Singular Value Decomposition (SVD)

$$A = U\Sigma V^T$$

$$U = \begin{array}{cc} & m \\ m & \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mm} \end{bmatrix} \end{array} \qquad \Sigma = \begin{array}{cc} & n \\ m & \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1m} \\ \varepsilon_{21} & \varepsilon_{22} & \cdots & \varepsilon_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} & \cdots & \varepsilon_{nm} \end{bmatrix} \end{array} \qquad V^T = \begin{array}{cc} & n \\ n & \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix} \end{array}$$

	$\left[\begin{array}{ccc\|c} 1 & 2 & -1 & 2 \\ 2 & 3 & 1 & 5 \\ 3 & 4 & -2 & 4 \end{array}\right]$
$R_2 \to R_2 - 2R_1$ $R_3 \to R_3 - 3R_1$	$\left[\begin{array}{ccc\|c} 1 & 2 & -1 & 2 \\ 0 & -1 & 3 & 1 \\ 0 & -2 & 1 & -2 \end{array}\right]$
$R_2 \to -R_2$	$\left[\begin{array}{ccc\|c} 1 & 2 & -1 & 2 \\ 0 & 1 & -3 & -1 \\ 0 & -2 & 1 & -2 \end{array}\right]$
$R_1 \to R_1 - 2R_2$ $R_3 \to R_3 + 2R_2$	$\left[\begin{array}{ccc\|c} 1 & 0 & 5 & 4 \\ 0 & 1 & -3 & -1 \\ 0 & 0 & -5 & -4 \end{array}\right]$
$R_3 \to \frac{1}{-5}R_3$	$\left[\begin{array}{ccc\|c} 1 & 0 & 5 & 4 \\ 0 & 1 & -3 & -1 \\ 0 & 0 & 1 & \frac{4}{5} \end{array}\right]$
$R_1 \to R_1 - 5R_3$ $R_2 \to R_2 + 3R_3$	$\left[\begin{array}{ccc\|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{7}{5} \\ 0 & 0 & 1 & \frac{4}{5} \end{array}\right]$

Example

$$A = \begin{bmatrix} 2 & -1 & -3 \\ -4 & 2 & 6 \end{bmatrix}$$

1. Null Space

The **null-space** of A, denoted $N(A)$, consists of all vectors $\vec{x} \in \mathbb{R}^3$ such that $Ax = 0$. The set of all such vectors is the **pre-image** of the zero vector under the transformation defined by A. In other words, $N(A) = \{x \in \mathbb{R}^3 \mid A\vec{x} = 0\}$, which represents the set of vectors that A maps to zero.

$$N(A) = \{\vec{x} \in \mathbb{R}^3 \mid A\vec{x} = \mathbf{0}\}$$

To find the null space $N(A)$ of the matrix A, we can use the **row-reduced echelon form (RREF)**. By augmenting the matrix A with a zero column and performing row operations, we reduce it to the form:

$$\begin{bmatrix} 2 & -1 & -3 \\ -4 & 2 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

	$\left[\begin{array}{ccc\|c} 2 & -1 & -3 & 0 \\ -4 & 2 & 6 & 0 \end{array}\right]$
$R_1 \to \frac{R_1}{2}$ $R_2 \to \frac{R_2}{4}$	$\left[\begin{array}{ccc\|c} 1 & -\frac{1}{2} & -\frac{3}{2} & 0 \\ -1 & \frac{1}{2} & \frac{3}{2} & 0 \end{array}\right]$
$R_2 \to R_2 - R_1$	$\left[\begin{array}{ccc\|c} 1 & -\frac{1}{2} & -\frac{3}{2} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$

$$\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{3}{2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$x_1 - \frac{1}{2}x_2 - \frac{3}{2}x_2 = 0$$

$$x_1 = \frac{1}{2}x_2 + \frac{3}{2}x_2$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_2 \begin{bmatrix} \frac{1}{2} \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} \frac{3}{2} \\ 0 \\ 1 \end{bmatrix}$$

$$N(a) = \operatorname{span}\left(\left\{ \begin{bmatrix} \frac{1}{2} \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{3}{2} \\ 0 \\ 1 \end{bmatrix} \right\}\right)$$

The dimension of the **null-space** is the number of vectors in this basis, which is 2. This is important because the dimension of the null space gives us insight into how many degrees of freedom exist in the system of equations $Ax = 0$

2. Column Space

$$C(A) = \operatorname{span}\left(\left\{ \begin{bmatrix} 2 \\ -4 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} -3 \\ 6 \end{bmatrix} \right\}\right)$$
$$= \operatorname{span}\left(\left\{ \begin{bmatrix} 2 \\ -4 \end{bmatrix} \right\}\right)$$

3. Basis

$$\begin{bmatrix} 2 \\ -4 \end{bmatrix}$$

4. Rank

Number of vector in the basis of our column space

$$\text{Rank}(A) = 1$$

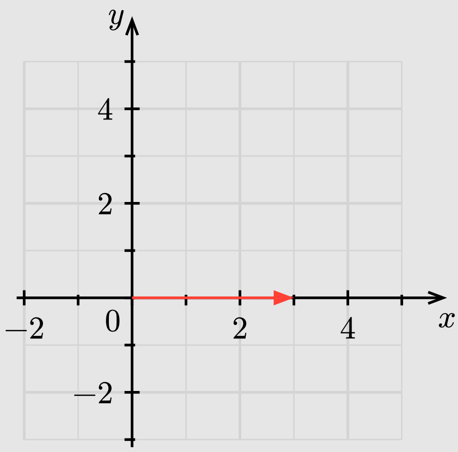
1.4. Vectors

$$\text{Vector} = \text{Magnitude} + \text{Direction}$$

Example

A car is moving:

$$\underbrace{\underbrace{\text{3 MPH}}_{\substack{\text{Magnitude}\\(\text{Speed}\to\text{Scalar})}}\qquad\underbrace{\text{East}}_{\text{Direction}}}_{\text{Velocity (Vector)}}$$



$$\vec{v} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

1.5. Real Coordinate Spaces

N-dimensional Real Coordinate Space

$$\mathbb{R}^n$$

$$\vec{x} \in \mathbb{R}^n$$

All possible real-valued n-tuples

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

1.6. Vector Operations

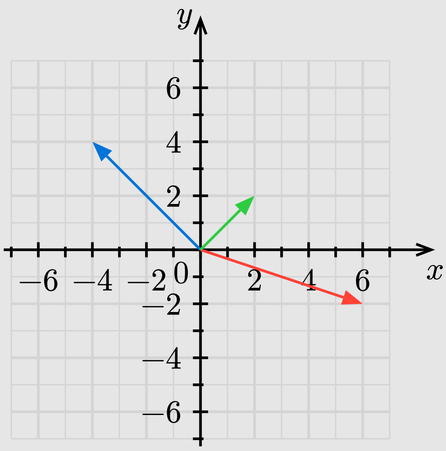
1.6.1. Vector Addition

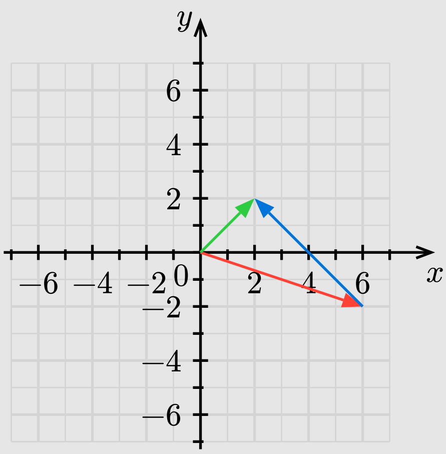
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{bmatrix}$$

Example

$$\vec{a} = \begin{bmatrix} 6 \\ -2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} -4 \\ 4 \end{bmatrix}$$

$$\vec{a} + \vec{b} = \begin{bmatrix} 6 + -4 \\ -2 + 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$





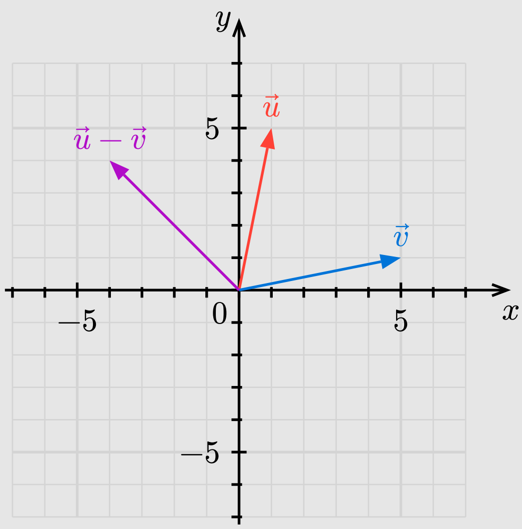
1.6.2. Vector Subtraction

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ x_3 - y_3 \end{bmatrix}$$

Example

$$\vec{u} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

$$\vec{u} - \vec{v} = \begin{bmatrix} 1-5 \\ 5-1 \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \end{bmatrix}$$



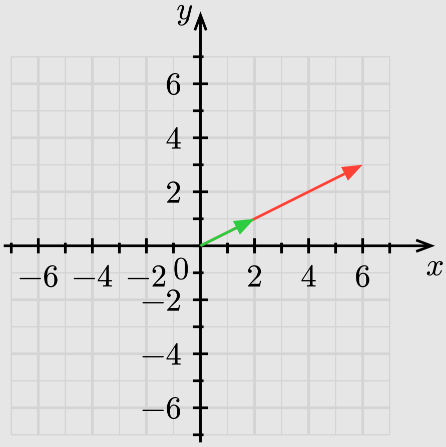
1.6.3. Scalar Multiplication

$$c \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c \times x_1 \\ c \times x_2 \\ c \times x_3 \end{bmatrix}$$

Example

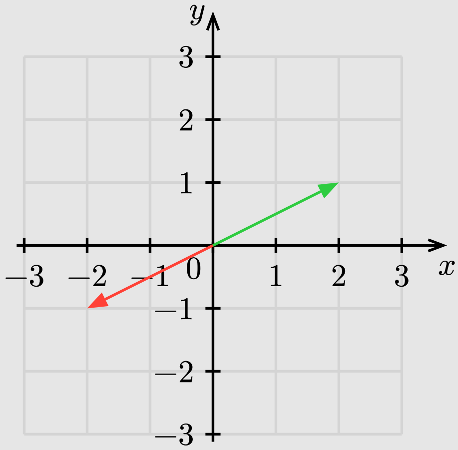
$$\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$3\vec{a} = {\color{red}3} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} {\color{red}3} \cdot 2 \\ {\color{red}3} \cdot 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$



$$\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$-1\vec{a} = {\color{red}-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} {\color{red}-1} \cdot 2 \\ {\color{red}-1} \cdot 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$



1.7. Unit Vector

A vector that has a magnitude (or length) of exactly 1

For a vector $\vec{v}$ in n-dimensional space, a unit vector $\hat{v}$ is defined as:

$$\hat{v} = \frac{\vec{v}}{\parallel \vec{v} \parallel}$$

Where:

- $\|\vec{v}\|$ is the **magnitude** (or **norm**) of the vector $\vec{v}$, computed as:

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$$

Key Properties

- **Magnitude**:

$$\|\hat{v}\| = 1$$

- **Direction**: A unit vector points in the same direction as the original vector $\vec{v}$

Example

Finding unit vector (vector of magnitude 1) with given direction

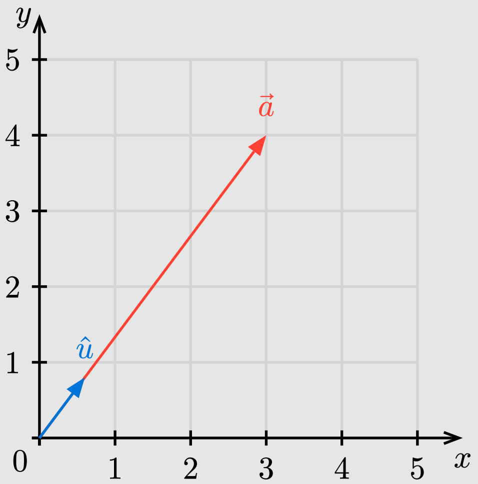
$$\vec{a} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Magnitude

$$\| \vec{a} \| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$$

$$\hat{u} = \left(\frac{3}{\| \vec{a} \|}, \frac{4}{\| \vec{a} \|} \right) = \left(\frac{3}{5}, \frac{4}{5} \right)$$

$$\| \hat{u} \| = \sqrt{\left(\frac{3}{5}\right)^2 + \left(\frac{4}{5}\right)^2} = \sqrt{\frac{9}{25} + \frac{16}{25}} = \sqrt{\frac{25}{25}} = \sqrt{1} = 1$$



1.8. Parametric Representation of line

Set L of all points (i.e., line) equal to the set of all vectors $\vec{x}$ plus some scalar t times the vector $\vec{v}$ such that t can be any real number ($\mathbb{R}$)

$$L = \{\vec{x} + t\vec{v} \mid t \in \mathbb{R}\}$$

Example

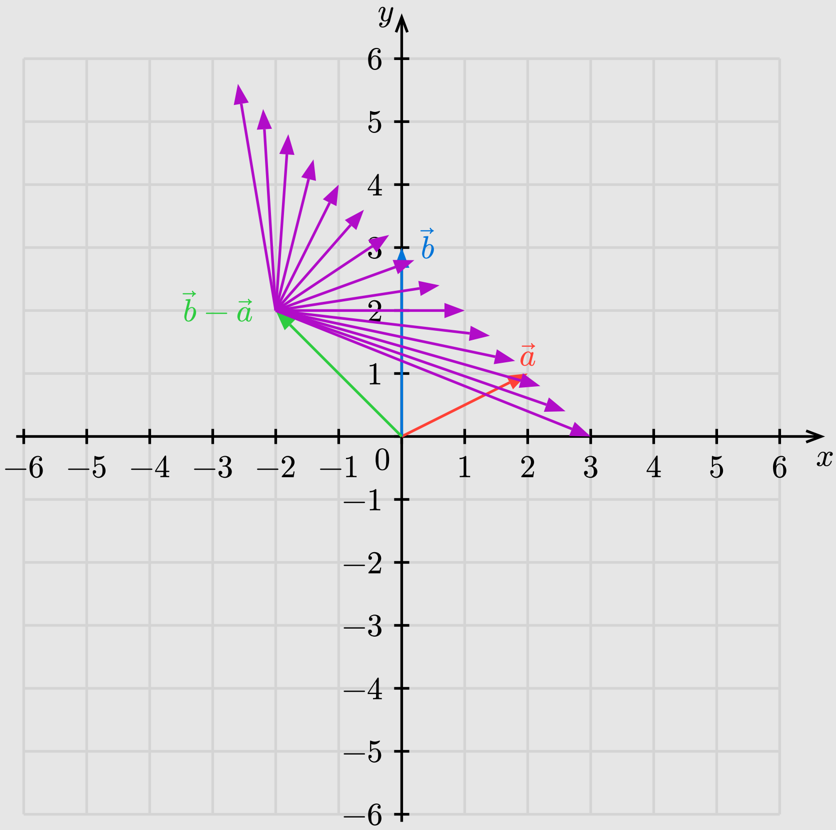
$$\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$t = 1$$

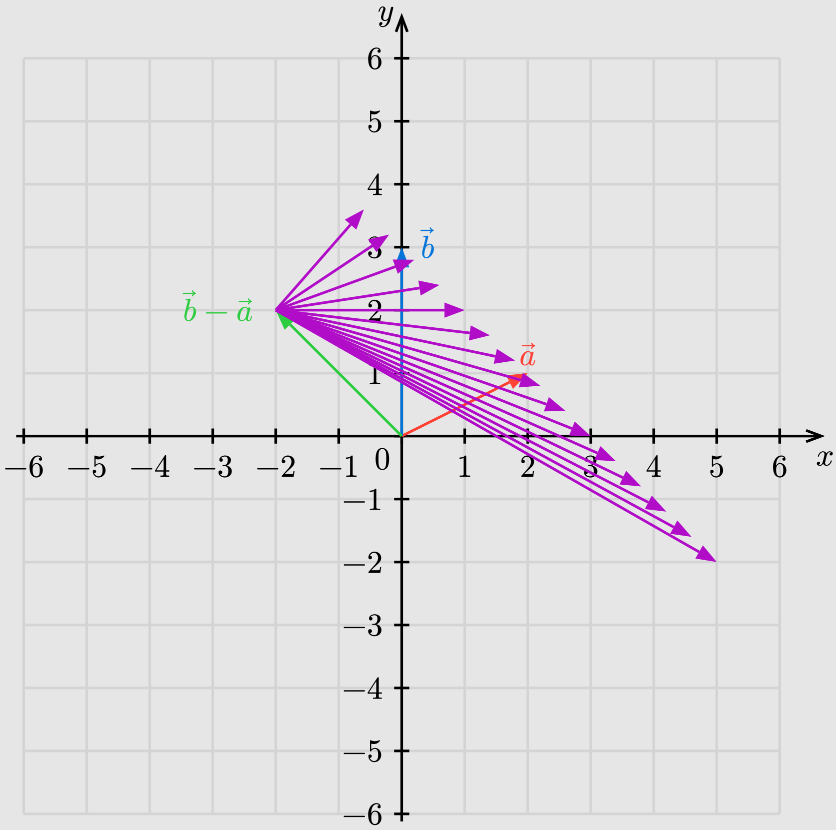
The line L can be defined as:

$$\begin{aligned} \vec{b} + t\left(\vec{b} - \vec{a}\right) &= \begin{bmatrix} 0 \\ 3 \end{bmatrix} + 1\left(\begin{bmatrix} 0 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) \\ &= \begin{bmatrix} 0 \\ 3 \end{bmatrix} + 1\begin{bmatrix} -2 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} -2 \\ 5 \end{bmatrix} \end{aligned}$$



The line L can also be defined as:

$$\vec{a} + t\big(\vec{b} - \vec{a}\big) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + 1\left(\begin{bmatrix} 0 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$
$$= \begin{bmatrix} 2 \\ 1 \end{bmatrix} + 1\begin{bmatrix} -2 \\ 2 \end{bmatrix}$$
$$= \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$



Generalization

$$P_1 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad P_2 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$L = P_1 + t(P_1 - P_2) \mid t \in \mathbb{R}$$

Example

$$\vec{P}_1 = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \qquad \vec{P}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$\vec{P}_1 - \vec{P}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

- L starts at P_1 and moves toward P_2 as t decreases

$$L = P_1 + t(P_1 - P_2) = \begin{bmatrix} -1 \\ 2 \end{bmatrix} + t \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$x = -1 + t(-1)$$
$$y = 2 + t(-1)$$

- L starts at P_2 and moves toward P_1 as t increases

$$L = P_2 + t(P_1 - P_2) = \begin{bmatrix} 0 \\ 3 \end{bmatrix} + t \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$x = 0 + t(-1)$$
$$y = 3 + t(-1)$$

- L starts at P_1 and moves toward P_2 as t increases

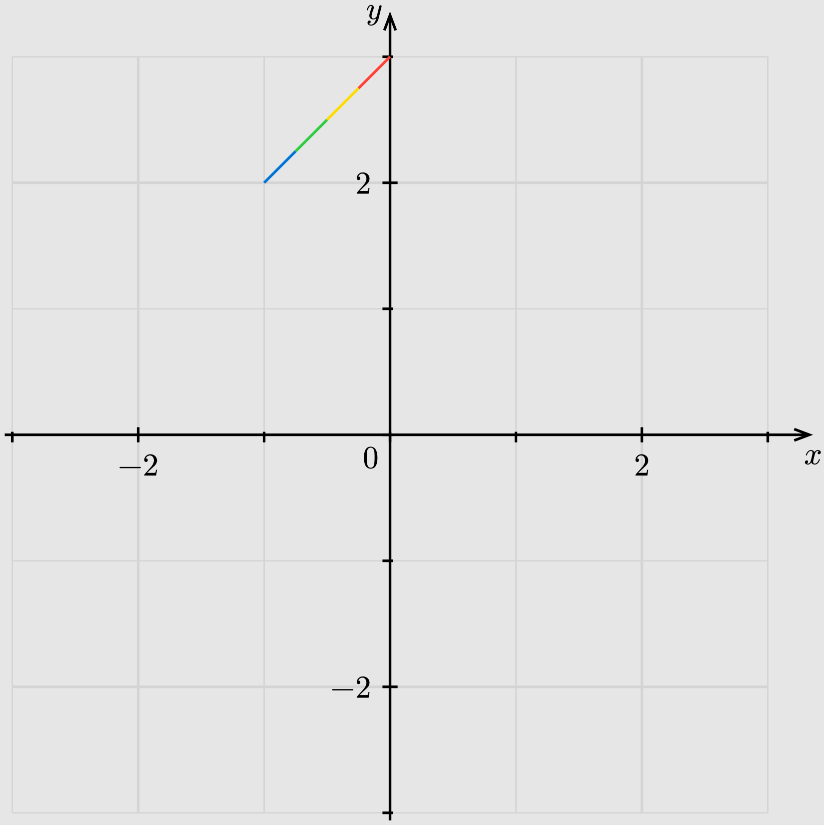
$$L = P_1 + t(P_2 - P_1) = \begin{bmatrix} -1 \\ 2 \end{bmatrix} + t \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$x = -1 + t(1)$$
$$y = 2 + t(1)$$

- L starts at P_2 and moves toward P_1 as t decreases

$$L = P_2 + t(P_2 - P_1) = \begin{bmatrix} 0 \\ 3 \end{bmatrix} + t \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$x = 0 + t(1)$$
$$y = 3 + t(1)$$



1.9. Vector Spaces

Example

Let's say your factory can produce up to 300 units of product 1, 500 units of product 2, and 400 units of product 3. The set of all possible production combinations forms a vector space:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$0 \leq x_1 \leq 300$$
$$0 \leq x_2 \leq 500$$
$$0 \leq x_3 \leq 400$$

2. Matrices

$m \times n$ matrix $\boldsymbol{A}$

- m: rows
- n: columns

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

2.1. Matrix-Vector Products

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boldsymbol{A}\mathbf{x} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

For the dot product to be defined, the number of columns in the matrix $\boldsymbol{A}$ (which is n) must match the number of elements in the vector $\vec{x}$ (also n).

The result of multiplying matrix $\boldsymbol{A}$ and vector $\vec{x}$ will be a column vector with dimensions $m \times 1$, where m is the number of rows in the matrix $\boldsymbol{A}$

$$(m \times n) \cdot (n \times 1) = m \times 1$$

1. As Row vectors

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\mathbf{a}^T = [a_1, a_2, ..., a_n]$$
$$\mathbf{b}^T = [b_1, b_2, ..., b_n]$$

$$\boldsymbol{A} = \begin{bmatrix} [a_1, a_2, ..., a_n] \\ [b_1, b_2, ..., b_n] \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{a}^T \\ \mathbf{b}^T \end{bmatrix} \cdot \mathbf{x} = \begin{bmatrix} \mathbf{a} \cdot \mathbf{x} \\ \mathbf{b} \cdot \mathbf{x} \end{bmatrix}$$

2. As Column Vectors

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$A = \begin{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} & \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \end{bmatrix}$$

$$A = \begin{bmatrix} \vec{a} & \vec{b} \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$A\vec{x} = x_1\vec{a} + x_2\vec{b}$$

2.2. Null Space

The null space (or kernel) of a matrix A is the set of all vectors x that satisfy the equation:

$$A\vec{x} = \mathbf{0}$$

Where:

- A: $m \times n$ matrix
- $\vec{x}$: n-dimansional vector
- $\mathbf{0}$: zero vecor in $\mathbb{R}^m$

$$N(A) = N(\text{rref}(A)) = \text{span}(\vec{v}_1, \vec{v}_2, \vec{v}_3)$$

Example

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

We want to find the null space of A, which consists of all vectors $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ that satisfy:

$$A\vec{x} = \mathbf{0}$$

This expands to the following system of linear equations:

$$\begin{cases} 1x_1 + 1x_2 + 1x_3 + 1x_4 = 0 \\ 1x_1 + 2x_2 + 3x_3 + 4x_4 = 0 \\ 4x_1 + 3x_2 + 2x_3 + 1x_4 = 0 \end{cases}$$

This can be represented as the augmented matrix:

$$
\left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 \\ 4 & 3 & 2 & 1 & 0 \end{array}\right]
$$

2.2.1. Column Space

The **columns space** (or range) of matrix A is span of its columns vectors

If the matrix A has columns $\vec{a}_1, \vec{a}_2, ..., \vec{a}_n$, then the column space of A is defined as:

$$
\mathrm{Col}(A) = \{\vec{y} \in \mathbb{R}^m \mid \vec{y} = A\vec{x} \text{ for some } \vec{x} \in \mathbb{R}^n\}
$$

or equivalently,

$$
\mathrm{Col}(A) = \mathrm{span}(\{\vec{a}_1, \vec{a}_2, ..., \vec{a}_n\})
$$

Example

Consider the simple example of a 2×2 matrix:

$$
A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}
$$

The matrix has two columns:

$$
\vec{a}_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \qquad \text{and} \qquad \vec{a}_2 = \begin{bmatrix} 2 \\ 6 \end{bmatrix}
$$

The column space, denoted $\mathrm{Col}(A)$, is the span of these two vectors:

$$
\mathrm{Col}(A) = \mathrm{span}\left(\left\{\begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix}\right\}\right)
$$

Finding the Column Space

We observe that the two columns $\vec{a}_1$ and $\vec{a}_2$ are **linearly dependent**:

$$
\vec{a}_2 = k\vec{a}_1
$$

This means that $\vec{a}_2$ is a scalar multiple of $\vec{a}_1$, the the two columns are **linearly dependent**. As a result, the column space is spanned by just one vector, $\vec{a}_1$, because any linear combination of $\vec{a}_1$ and $\vec{a}_2$ can be reduced to a multiple of $\vec{a}_1$.

Therefore, the column space of A is:

$$
\mathrm{Col}(A) = \mathrm{span}\left(\left\{\begin{bmatrix} 1 \\ 3 \end{bmatrix}\right\}\right)
$$

which represents all vectors of the form:

$$c \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} c \\ 3c \end{bmatrix} \qquad \text{for any scalar } c$$

In other words, the column space is a line in $\mathbb{R}^2$ through the origin in the direction of

Rank of A

The rank of A, which is the **dimension of its column space**, is 1 because there is only one linearly independent column

This means the column space is the span of the columns of A, or all vectors that can be formed by taking linear combinations of the columns of A.

2.2.2. Dimension of a Subspace

Number of elements in a basis for the subspace

2.2.3. Nullity

Dimension of the **Null Space**

$$\dim(N(\boldsymbol{A}))$$

The nullity of A: number of non-pivot columns (i.e., free variables) in the rref of A

2.2.4. Rank

Dimension of the column **space**

$$\text{rank}(A) = \dim(C(A))$$

2.2.5. Matrix Representation of Systems of Equations

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m = b_n \end{gathered}$$

Coefficient Matrix (A):

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Variable Vector (x):

$$\mathrm{x} \;=\; \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Constant Vector (b):

$$\mathrm{b} \;=\; \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\mathrm{Ax} = b$$

Example

The system of equations:

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 = 100 \\ 4x_1 + 2x_2 + 1x_3 = 80 \\ 1x_1 + 5x_2 + 2x_3 = 60 \end{aligned}$$

Can be represented as a matrix equation:

$$\begin{bmatrix} 2 & 3 & 5 \\ 4 & 2 & 1 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 80 \\ 60 \end{bmatrix}$$

2.3. Matrix Multiplication

$m \times n$ matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$n \times p$ matrix:

$$B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{bmatrix}$$

Compute Each Element of Result Matrix C

$$c_{ij} = \sum_{k=1}^{n} = a_{ik} b_{kj}$$

Example

Let A be an $n \times m$ matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Let B an $p \times n$ matrix:

$$B = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

Calculate Each Element of C

$$\begin{aligned} c_{11} &= (1 \cdot 7) + (2 \cdot 9) + (3 \cdot 11) = 58 \\ c_{12} &= (1 \cdot 8) + (2 \cdot 10) + (3 \cdot 12) = 64 \\ c_{21} &= (4 \cdot 7) + (5 \cdot 9) + (6 \cdot 11) = 138 \\ c_{22} &= (4 \cdot 8) + (5 \cdot 10) + (6 \cdot 12) = 154 \end{aligned}$$

C is a $m \times p$ matrix

$$C = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

2.4. Linear Combinations

Set of vector

$$v_1, v_2, ..., v_n \in \mathbb{R}^n$$

Where

- $v_1, v_2, ..., v_n$: set of vectors
- $\mathbb{R}^n$: set of all ordered tuples of n real numbers

Linear combination of those vector

$$c_1 v_1 + c_2 v_2 + \ldots + c_n v_n$$

$$c_1, c_2, \ldots, c_n \in \mathbb{R}$$

Where:

- $c_1, c_2, \ldots, c_n$: constants or weights

Example

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$0\vec{a} + 0\vec{b}$$

$$3\vec{a} + 2\vec{b}$$

2.5. Span

Represents the subspace of the vector space that is “covered” by these vectors through their linear combinations

If you have a set of vectors $v_1, v_2, \ldots, v_n$, the span of these vectors is the set of all vectors that can be written as:

$$\mathrm{Span}(v_1, v_2, \ldots, v_n) = \{c_1 v_1 + c_2 v_2 + \ldots + c_n v_n \mid c_1, c_2, \ldots, c_n \in \mathbb{R}\}$$

Any vector in $\mathbb{R}^2$ can be represented by a linear combination with some combination of these vectors

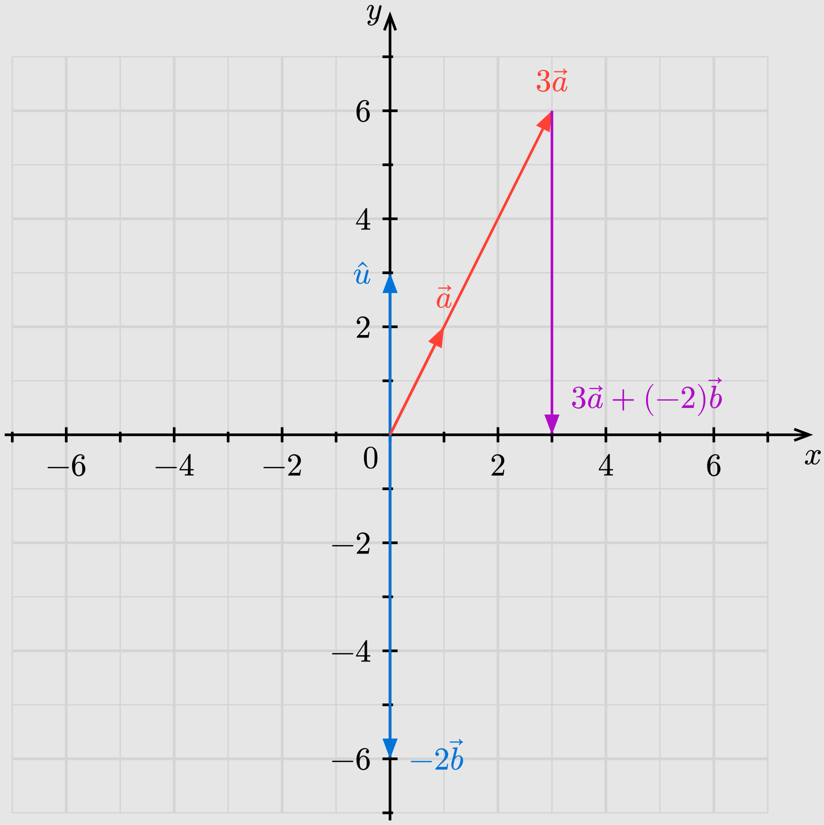
Example

1. Spanning $\mathbb{R}^2$

$$\mathrm{Span}\left(\vec{a}, \vec{b}\right) = \mathbb{R}^2$$

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$3\vec{a} + (-2)\vec{b} = \begin{bmatrix} 3 - 0 \\ 6 - 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$



Any point $\vec{x}$ can be represented as a linear combination of $\vec{a}$ and $\vec{b}$

1. Define the vectors

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2. Express $\vec{x}$ as a linear combinations

$$c_1\vec{a} + c_2\vec{b} = \vec{x}$$

Which expands to

$$c_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

3. Set up the system of equations

$$1c_1 + 0c_2 = x_1 \qquad (1)$$
$$2c_1 + 3c_2 = x_2 \qquad (2)$$

4. Express c_1: From equation (1), we can directly express c_1

$$c_1 = x_1$$

5. Substitute c_1 into equation (2)

$$2x_1 + 3c_2 = x_2$$

Rearranging gives:

$$3c_2 = x_2 - 2x_1$$

6. Solve for c_2 : Dividing both sides by 3 yields

$$c_2 = \frac{x_2 - 2x_1}{3}$$

7. Example with specific values: Let's say we want to find c_1 and c_2 when $\vec{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

   Substitute $x_1 = 2$ and $x_2 = 2$

$$c_1 = x_1 = 2$$
$$c_2 = \frac{2 - 2 \cdot 2}{3} = -\frac{2}{3}$$

8. Final linear combination: Now, substituting c_1 and c_2 back into the linear combination

$$2\vec{a} - \frac{2}{3}\vec{b} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

   Verifying

$$2\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

9. This shows that

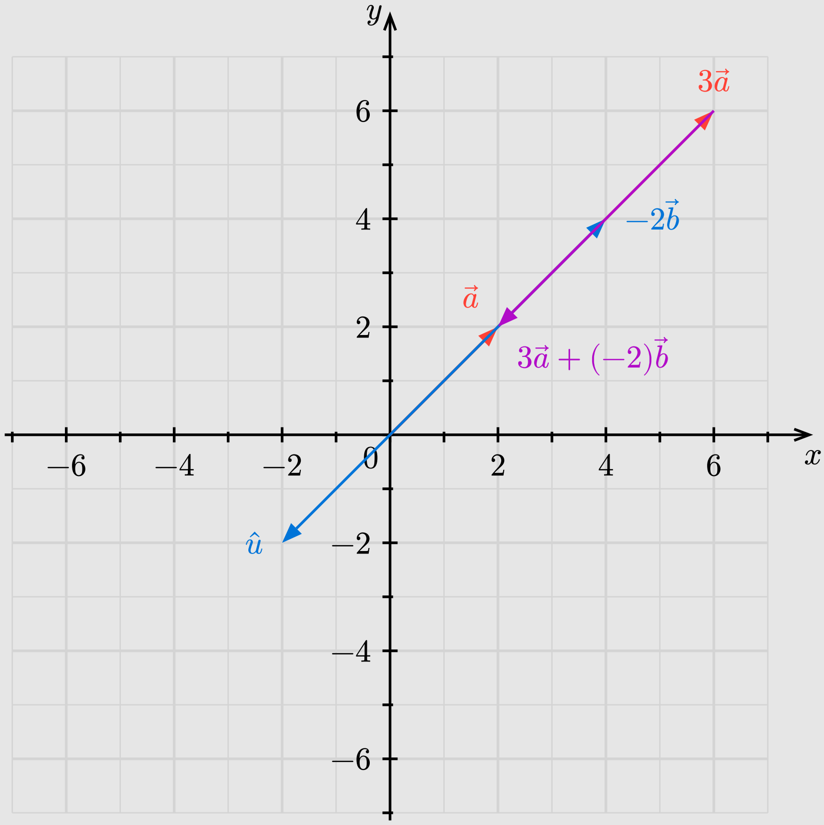
$$2\vec{a} - \frac{2}{3}\vec{b} = \vec{x}$$

2. Spanning Line in $\mathbb{R}^2$

Any linear combination of $\vec{a}$ and $\vec{b}$ will produce vectors that lie along the same line. This is the line through the origin in the direction of $\vec{a}$ (or $\vec{b}$), with all points on the line being scalar multiples of $\vec{a}$

$$\vec{a} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

$$3\vec{a} + (-2)\vec{b} = \begin{bmatrix} 6-4 \\ 6-4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$



2.6. Linear Independence

1. Definition of Linear Independence

The set of vectors

$$S = \{v_1, v_2, \ldots, v_n\}$$

is said to be **linearly independent** if the only solution to the equation

$$c_1 v_1 + c_2 v_2 + \ldots + c_n v_n = \mathbf{0}$$

is $c_1 = c_2 = \ldots = c_n = 0$. In other words, no vector in the set can be written as a linear combination of the others.

If at least one constant c_i is non-zero, the set is linearly dependent.

Example

Example 1: Testing for Linear Independence

Problem: Is the following set of vectors **linearly dependent**?

$$S = \{\vec{v}_1 \vec{v}_2\}$$

Where:

$$\vec{v}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \text{and} \quad \vec{v}_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

For a set of vectors to be **linearly independent**, the only solution to the equation:

$$c_1\vec{v}_1 + c_2\vec{v}_2 = \mathbf{0}$$

must be $c_1 = 0$ and $c_2 = 0$

In this case:

$$c_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \mathbf{0}$$

If not only the zero solution exists (i.e., if c_1 or c_2 can be non-zero), the set is **linearly dependent**.

Step 1. Set up the system of equations:

1. $2c_1 + 3c_2 = 0$
2. $1c_1 + 2c_2 = 0$

2. Eliminate one variable

$$2 \times (1c_1 + 2c_2 = 0) \Rightarrow 2c_1 + 4c_2 = 0$$

Now the system is

$$2c_1 + 3c_2 = 0$$
$$2c_1 + 4c_2 = 0$$

3. Subtract the equations

$$(2c_1 + 3c_2) - (2c_1 + 4c_2) = 0$$

Simplifies to:

$$(2c_1 - 2c_1) + (4c_2 - 3c_2) = 0$$

So:

$$c_2 = 0$$

4. Substitute back to find c_1

Now that we know $c_2 = 0$, substitute this value into one of the original equations. Let's use the second equation:

$$1c_1 + 2c_2 = 0$$

Substitute $c_2 = 0$:

$$1c_1 + 2(0) = 0$$
$$c_1 = 0$$

Conclusion:

Since $c_1 = 0$ $c_2 = 0$, the set of vectors S is **linearly independent**. These vecots span $\mathbb{R}^2$.

Example 2: Testing for Linear Dependence

Problem: Is the following set of vectors **linearly dependent**?

$$S = \{\vec{v}_1 \vec{v}_2\}$$

Where:

$$\vec{v}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \text{and} \quad \vec{v}_2 = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

The span of this set is the collection of all vectors that can be formed by linear combinations of $\vec{v}_1$ and $\vec{v}_2$:

$$c_1 v_1 + c_2 v_2$$

Since $v_2 = 2v_1$, the linear combination becomes:

$$c_1 v_1 + c_2(2v_1) = (c_1 + 2c_2)v_1$$

$$c_1 \begin{bmatrix} 2 \\ 3 \end{bmatrix} + c_2 \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

$$c_1 \begin{bmatrix} 2 \\ 3 \end{bmatrix} + c_2 2 \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$(c_1 + 2c_2) \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$c_3 \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Thus, any linear combination of these vectors is just a scalar multiple of v_1. The span is a single line in $\mathbb{R}^2$, and the vectors are **linearly dependent**.

For any two **colinear** vectors in $\mathbb{R}^2$, their span reduces to a single line.

One vector in the set can be represented by some combination of other vectors in the set

2. General Rule

In $\mathbb{R}^n$, if you have more than n vectors, at least one vector must be linearly dependent on the others, meaning the set cannot be linearly independent.

3. Subspace

V is a linear subspace of $\mathbb{R}^n$:

- **Non-emptiness**: V contains the $\mathbf{0}$ vector

$$\mathbf{0} \in V$$

- **Closure under addition**: If u and v are any vectors in the subspace V, then their sum $u+v$ must also be in V.

$$\text{If } u, v \in V, \text{then } u+v \in V$$

- **Closure under scalar multiplication**: If u is any vector in V and c is any scalar (real number), then the product cu must also be in V.

$$\text{If } u \in V \text{ and } c \in V, \text{then } cu \in V$$

Example

Example 1: Subspace

Problem: Is V a subspace of $\mathbb{R}^2$

$$V = \{\mathbf{0}\} = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

- **Non-emptiness** ✅

$$\mathbf{0} \in V$$

- **Closure under addition** ✅

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- **Closure under scalar multiplication** ✅

$$c \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

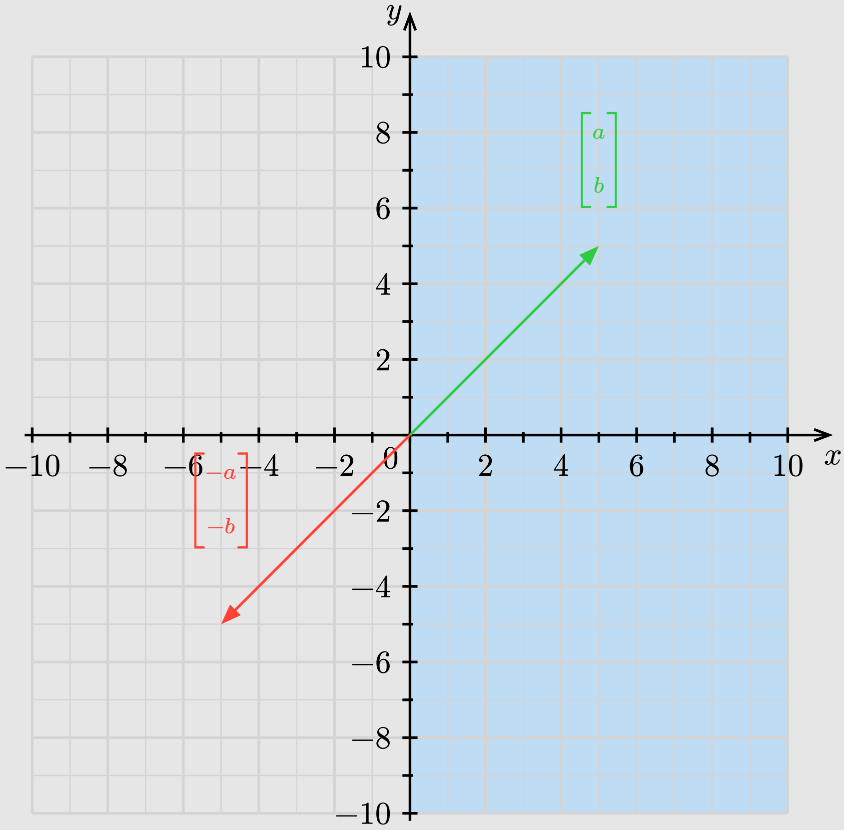
Conclusion

The subset V of $\mathbb{R}^3$ is a **subspace**

Example 2: Not Subspace

Problem: Is S a subspace of $\mathbb{R}^2$ ✅

$$S = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2 \mid x_1 \geq 0 \right\}$$



- **Non-emptiness** ✅

$$\mathbf{0} \in S$$

- **Closure under addition** ✅

$$\begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a+c \\ b+d \end{bmatrix}$$

$$a \geq 0$$
$$b \geq 0$$
$$a + b \geq 0$$

- **Closure under scalar multiplication** ❌

$$-1 \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -a \\ -b \end{bmatrix}$$

Conclusion

Span and Subspace

The span of any set of vectors is a valid subspace

$$U = \text{Span}(v_1, v_2, ..., v_n) = \text{Valid Subspace of } \mathbb{R}^n$$

- **Non-emptiness**

$$0v_1 + 0v_2 + ... + 0v_n = \mathbf{0}$$

- **Closure under addition**

$$\vec{X} = a_1 v_1 + a_2 v_2 + ... + a_n v_n$$
$$\vec{Y} = b_1 v_1 + b_2 v_2 + ... + b_n v_n$$

$$\begin{aligned} \vec{X} + \vec{Y} &= (a_1 + b_1) v_1 + (a_2 + b_2) v_2 + ... + (a_n + b_n) v_n \\ &= c_1 v_1 + c_2 v_2 + ... + c_n v_n \end{aligned}$$

- **Closure under scalar multiplication**

$$\vec{X} = a_1 v_1 + a_2 v_2 + ... + a_n v_n$$

$$\begin{aligned} b\vec{X} &= bc_1 v_1 + bc_2 v_2 + ... + bc_n v_n \\ &= c_1 v_1 + c_2 v_2 + ... + c_n v_n \end{aligned}$$

4. Basis

Non-redundant set of vectors that span $\mathbb{R}^n$

A basis of a vector space is a set of vectors that satisfies two conditions:

1. **Linear Independence**: No vector in the set can be written as a linear combination of the others. This means that **the only way to combine the vectors to get the zero vector is by using all zero coefficients**.

   The vectors $\vec{v}_1, \vec{v}_2, ..., \vec{v}_n$ are linearly independent if the only solution to the equation

$$c_1 \vec{v}_1 + c_2 \vec{v}_2 + ... + c_n \vec{v}_n = \mathbf{0}$$

   is $c_1 = c_2 = ... = c_n = 0$, where c_i are scalar coefficients.

2. **Spanning**: The set of vectors can be linearly combined to form any vector in the vector space. In other words, **every vector in the vector space can be expressed as a linear combination of the basis vectors**.

   The set $\{\vec{v}_1, \vec{v}_2, ..., \vec{v}_n\}$ spans the vector space V if any vector $v \in V$ can be expressed as a linear combination of the basis vectors:

$$\vec{v} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + ... + c_n \vec{v}_n$$

   for some scalars $c_1, c_2, ..., c_n$.

Example

Consider the vector space $\mathbb{R}^2$ (the 2-dimensional Euclidean space). A common basis for $\mathbb{R}^2$ is $\{e_1, e_2\}$, where:

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This set is a basis because:

- Linear Independence: The only solution to $c_1 e_1 + c_2 e_2 = \mathbf{0}$ is $c_1 = c_2 = 0$.
- Spanning: Any vector $\vec{v} = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$ can be written as $\vec{v} = x e_1 + x e_2$

This means $\{e_1, e_2\}$ is a basis for $\mathbb{R}^2$, and the dimension of $\mathbb{R}^2$ is 2.

4.1. Vector Dot Product

$$\vec{a} \cdot \vec{b} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

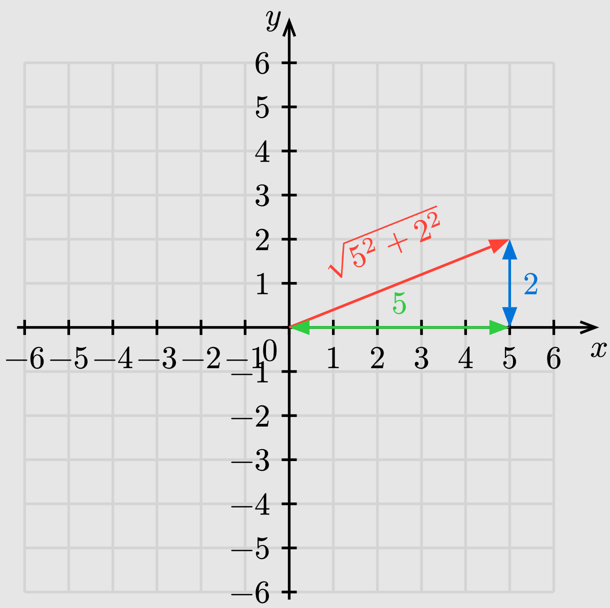
4.1.1. Magnitude (Length)

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$$

Example

$$\vec{a} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

$$\|\vec{a}\| = \sqrt{5^2 + 2^2}$$



$$\vec{a} \cdot \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = a_1^2 + a_2^2 + ... + a_n^2$$

$$\|\vec{a}\| = \sqrt{\vec{a} \cdot \vec{a}}$$

$$\|\vec{a}\|^2 = \vec{a} \cdot \vec{a}$$

4.1.2. Properties

4.1.2.1. Commutative

$$\vec{v} \cdot \vec{w} = \vec{w} \cdot \vec{v}$$

4.1.2.2. Distributive

$$(\vec{v} + \vec{w}) \cdot \vec{x} = \vec{v} \cdot \vec{x} + \vec{w} \cdot \vec{x}$$

4.1.2.3. Associativity

$$(c\vec{v}) \cdot \vec{w} = c(\vec{v} \cdot \vec{w})$$

4.1.3. Cauchy-Schwarz Inequality

$$|\vec{u} \cdot \vec{v}| \leq \|\vec{u}\| \ \|\vec{v}\|$$
$$|\vec{u} \cdot \vec{v}| = \|\vec{u}\| \ \|\vec{v}\| \quad \text{when} \quad \vec{u} = c\vec{v}$$

Where:

- $\vec{u} \cdot \vec{v}$: dot product of vectors $\vec{u}$ and $\vec{v}$
- $\|\vec{u}\|$ and $\|\vec{v}\|$: magnitudes (lengths) of vectors $\vec{u}$ and $\vec{v}$

Step 1: Understand the dot product

The dot product of two vectors $\vec{u} = [u_1, u_2, ..., u_n]$ and $\vec{v} = [v_1, v_2, ..., v_n]$ is calculated as:

$$\vec{u} \cdot \vec{v} = u_1 v_1 + u_2 v_2 + ... + u_n v_n$$

The magnitude (or norm) of a vector $\vec{u}$ is:

$$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + ... + u_n^2}$$

Step 2: Define a new function

We introduce a parameter $t \in \mathbb{R}$ and define a new vector:

$$w(t) = u - tv$$

Now, consider the dot product of this new vector with itself, which is always non-negative because it represents the square of the magnitude of $w(t)$:

$$w(t) \cdot w(t) \geq 0$$

This inequality makes sense because the dot product of any vector with itself is the square of its magnitude, and **a square is always non-negative**.

Step 3: Expand the dot product

Expand $w(t) \cdot w(t)$:

$$w(t) \cdot w(t) = (\vec{u} - t\vec{v}) \cdot (\vec{u} - t\vec{v})$$

Now, we apply the distributive property of the dot product, which behaves similarly to the distributive property of multiplication. We expand each term:

$$(\vec{u} - t\vec{v}) \cdot (\vec{u} - t\vec{v}) = \vec{u} \cdot \vec{u} - t(\vec{u} \cdot \vec{v}) - t(\vec{v} \cdot \vec{u}) + t^2(\vec{v} \cdot \vec{v})$$

Since the dot product is commutative ($\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u}$), we can rewrite this as:

$$\vec{u} \cdot \vec{u} - 2t(\vec{u} \cdot \vec{v}) + t^2(\vec{v} \cdot \vec{v})$$

This simplifies to:

$$\|\vec{u}\|^2 - 2t(\vec{u} \cdot \vec{v}) + t^2 \ \|\vec{v}\|^2$$

We've now expressed the result of expanding the dot product as a quadratic expression in t, where

- $\|\vec{u}\|^2$ is a constant term,
- $-2t(\vec{u} \cdot \vec{v})$ is the linear term in t
- $t^2 \ \|\vec{v}\|^2$ is the quadratic term

Step 4: Treat as a quadratic equation

Now that we have the quadratic expression:

$$\|\vec{v}\|^2 \ t^2 - 2(\vec{u} \cdot \vec{v})t + \|\vec{u}\|^2 \geq 0$$

We recognize this as a standard quadratic inequality of the form $at^2 + bt + c \geq 0$, where:

- $a = \|\vec{v}\|^2$
- $b = -2(\vec{u} \cdot \vec{v})$
- $c = \|\vec{u}\|^2$

For any quadratic expression $at^2 + bt + c$ to always be non-negative, its discriminant must be less than or equal to zero. The discriminant of a quadratic equation $at^2 + bt + c = 0$ is given by:

$$\Delta = b^2 - 4ac$$

Substituting in the values of a, b, and c from our expression:

$$\Delta = (-2(\vec{u} \cdot \vec{v}))^2 - 4 \cdot \|\vec{v}\|^2 \cdot \|\vec{u}\|^2$$

Simplifying:

$$\Delta = 4(\vec{u} \cdot \vec{v})^2 - 4 \ \|\vec{v}\|^2 \ \|\vec{u}\|^2$$

Step 5: Apply the discriminant condition

For the quadratic inequality to hold, the discriminant must be less than or equal to zero:

$$\Delta = 4(\vec{u} \cdot \vec{v})^2 - 4 \ \|\vec{v}\|^2 \ \|\vec{u}\|^2 \leq 0$$

Divide by 4:

$$\Delta = (\vec{u} \cdot \vec{v})^2 \leq \|\vec{v}\|^2 \; \|\vec{u}\|^2$$

Take the square root of both sides:

$$|\vec{u} \cdot \vec{v}| \leq \|\vec{u}\| \; \|\vec{v}\|$$

Example

$$\vec{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Step 1: Compute the dot product $\vec{u} \cdot \vec{v}$

$$\vec{u} \cdot \vec{v} = (1)(3) + (2)(4) = 3 + 8 = 11$$

Step 2: Compute the norms of $\vec{u}$ and $\vec{v}$

- The norm $\|\vec{u}\|$ is:

$$\|\vec{u}\| = \sqrt{1^2 + 2^2} = \sqrt{1 + 4} = \sqrt{5}$$

- The norm $\|\vec{v}\|$ is:

$$\|\vec{v}\| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

Step 3: Verify the Cauchy-Schwarz inequality

The inequality states:

$$|\vec{u} \cdot \vec{v}| \leq \|\vec{u}\| \; \|\vec{v}\|$$

Substitute the values:

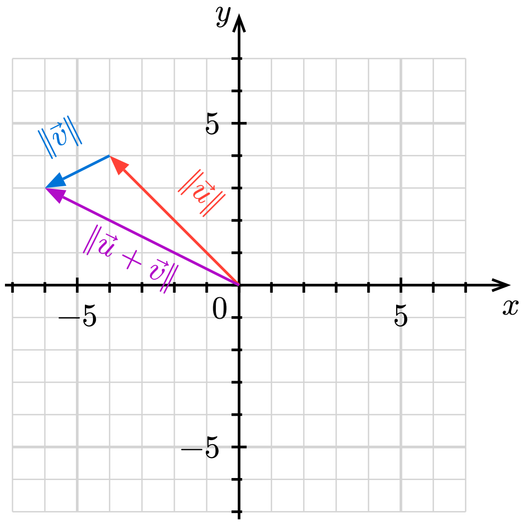
$$|11| \leq 5\sqrt{5} = 11.18$$

Since $11 \leq 11.18$, the inequality holds.

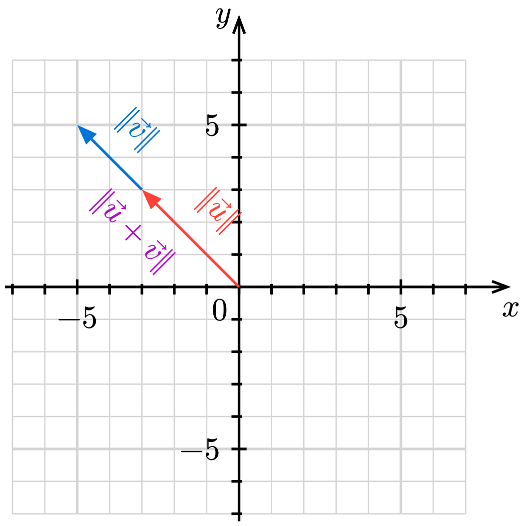
The Cauchy-Schwarz inequality is satisfied for the vectors $\vec{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\vec{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

4.1.4. Vector Triangle Inequality

$$\|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$$



$$\|\vec{u}+\vec{v}\| = \|\vec{u}\| + \|\vec{v}\|$$



$$\vec{u} = c\vec{v} \qquad c > 0$$

4.2. Cartesian & Polar Coordinates

4.2.1. Cartesian

Describe a point’s position using perpendicular axes

The coordinates are (x, y), where:

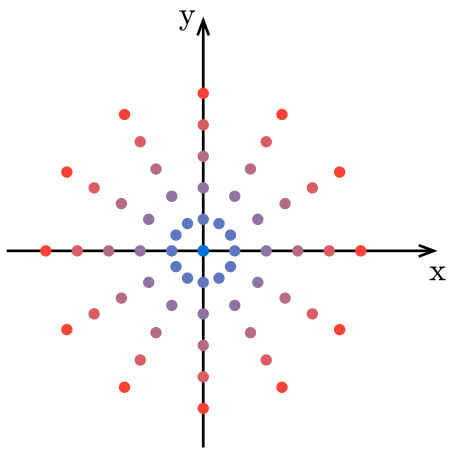
- x: horizontal distance from the origin
- y: vertical distance from the origin

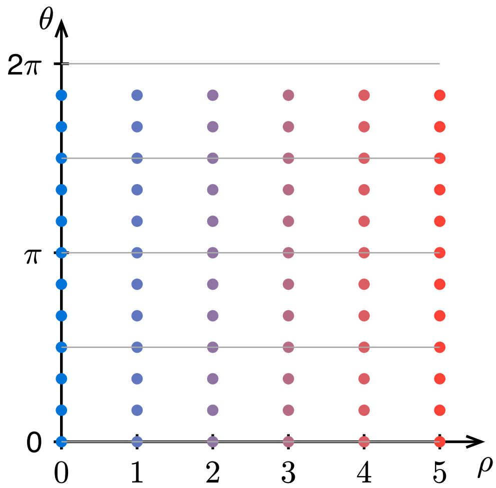
4.2.2. Polar

Describe a point’s position using a distance from the origin and an angle from a reference direction

The coordinates are (r, θ), where:

- r: distance from the origin θ
- θ: angle from the positive x-axis (in radians or degrees)

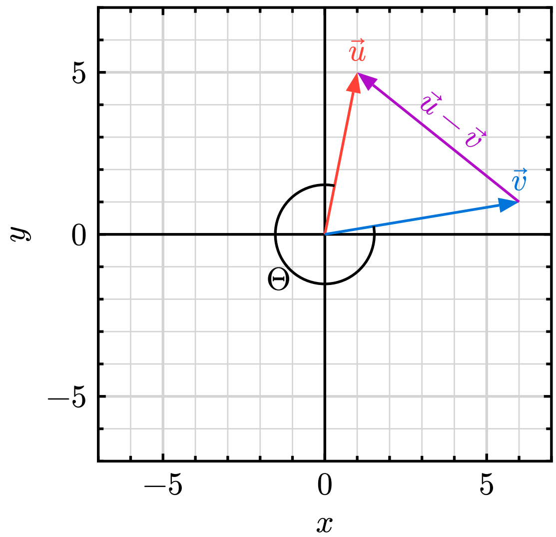


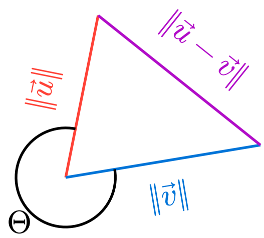


4.3. Angles Between Vectors

The scalar $\|\vec{u}\|$ is the length of the vector $\vec{u}$

Say $\vec{u}, \vec{v} \in \mathbb{R}^n$





Law of Cosines

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(C)$$

Where:

- a, b and c: lengths of the sides of a triangle
- C: angle opposite the side c

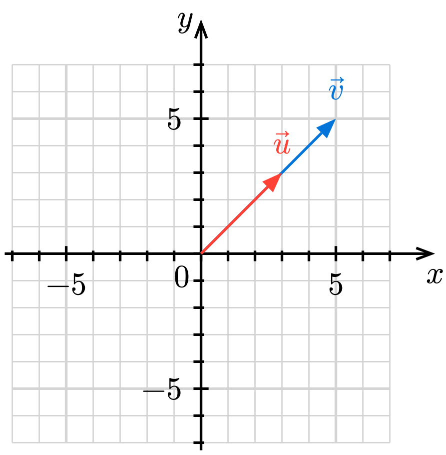
$$\begin{aligned} \|\vec{u}-\vec{v}\|^2 &= \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\,\|\vec{u}\|\,\|\vec{v}\| \cdot \cos(\Theta) \\ (\vec{u}-\vec{v}) \cdot (\vec{u}-\vec{v}) &= \\ \vec{u}\cdot\vec{u} - \vec{u}\cdot\vec{v} - \vec{v}\cdot\vec{u} + \vec{v}\cdot\vec{v} &= \\ \cancel{\|\vec{u}\|^2} \cancel{-2}(\vec{u}\cdot\vec{v}) + \cancel{\|\vec{v}\|} &= \cancel{\|\vec{u}\|^2} + \cancel{\|\vec{v}\|^2} \cancel{-2}\,\|\vec{u}\|\,\|\vec{v}\| \cdot \cos(\Theta) \end{aligned}$$

$$\boxed{\vec{u}\cdot\vec{v} = \|\vec{u}\|\,\|\vec{v}\| \cdot \cos(\Theta)}$$

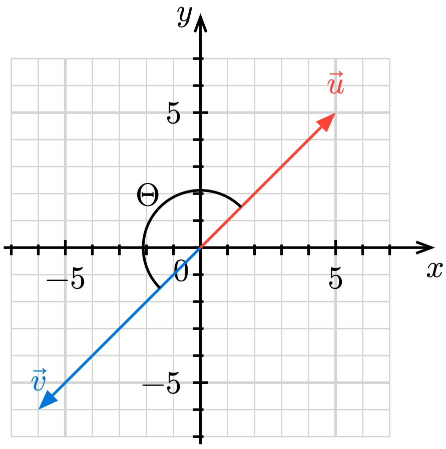
$$\boxed{\frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\,\|\vec{v}\|} = \cos(\Theta)}$$

$$\boxed{\Theta = \arccos\left(\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\,\|\vec{b}\|}\right)}$$

So, if $\vec{u}$ is a scalar multiple of $\vec{v}$ ($\vec{u} = c\vec{v}$) where $c > 0$, then $\Theta = 0°$



And, if $\vec{u}$ is a scalar multiple of $\vec{v}$ ($\vec{u} = c\vec{v}$) where $c < 0$, then $\Theta = 180°$



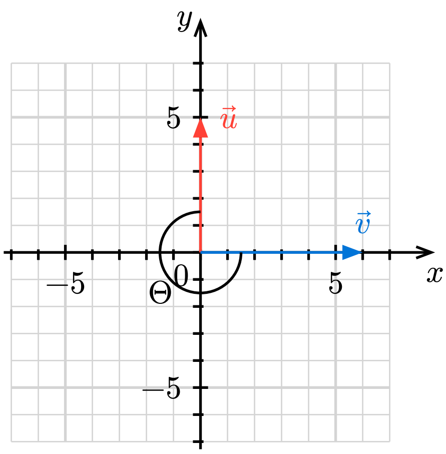
$\vec{u}$ and $\vec{v}$ are perpendicular if the angle Θ between them is $90°$

$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \; \|\vec{v}\| \cdot \cos(90)$$
$$\vec{u} \cdot \vec{v} = 0$$

If $\vec{u}$ and $\vec{v}$ are perpendicular, then $\vec{u} \cdot \vec{v} = 0$

If $\vec{u}$ and $\vec{v}$ are non-zer0 and $\vec{u} \cdot \vec{v} = 0$, then they are perpendicular

If $\vec{u} \cdot \vec{v} = 0$ then $\vec{u}$ and $\vec{v}$ are **orthogonal**.



4.3.1. Plane in $\mathbb{R}^3$

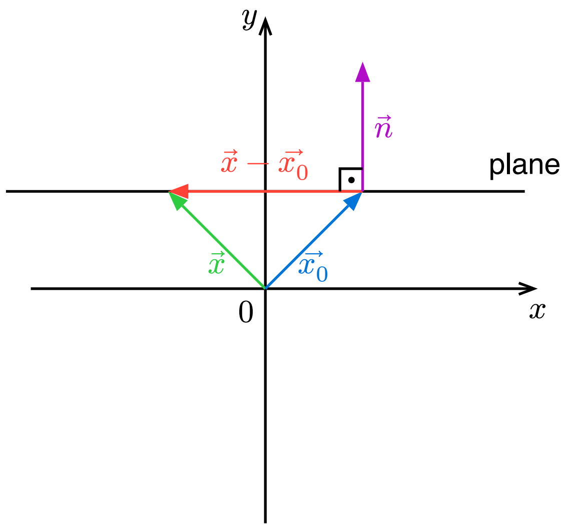
Plane: Each point (x, y, z) on the satifies the equation

$$ax + by + cy = d$$

Normal Vector: vector that is perpendicular (orthogonal) to a plane, line, or curve, at a specific point

1. **Plane**

If a plane is defined by the equation $ax + by + cz = d$, the vector $\vec{n} = \langle a, b, c \rangle$ is a normal vector to the plane because it is perpendicular to any vector that lies in the plane



$$\vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \vec{x}_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \qquad \vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

$$\vec{x} - \vec{x}_0 = \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}$$

$\vec{x} - \vec{x}_0$ is a vector that lies on the plane, then $\vec{n}$ is normal if:

$$\vec{n} \cdot \vec{x} - \vec{x}_0 = 0$$

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = 0$$

$$\boxed{n_1(x - x_0) + n_2(y - y_0) + n_3(z - z_0) = 0}$$

Example

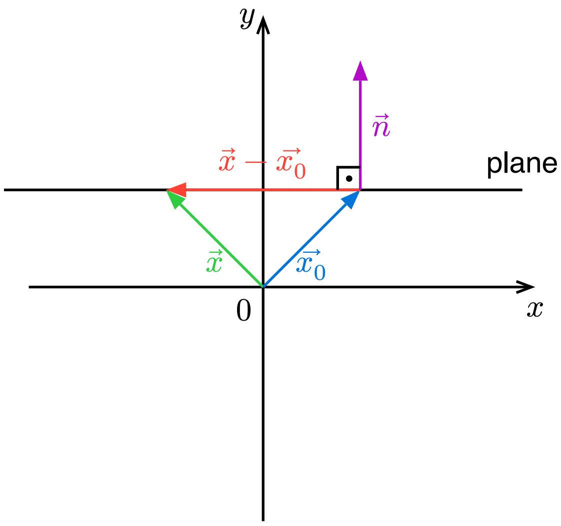
Find the equation of the plane given the point on the plane $\vec{x}_0$ and a the normal vector $\vec{n}$

$$\vec{n} = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} \qquad \vec{x}_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\vec{x} - \vec{x_0} = \begin{bmatrix} x - 1 \\ y - 2 \\ z - 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} x - 1 \\ y - 2 \\ z - 3 \end{bmatrix} = 0$$

$$(x - 1) + 3(y - 2) - 2(z - 3) = 0$$
$$x - 1 + 3y \cancel{- 6} - 2z \cancel{+ 6} = 0$$
$$x + 3y - 2z = 1$$



The normal vector to a plane can be directly obtained from the coefficients of x, y, and z in the plane equation of the form:

$$Ax + By + Cz = D$$

$$\vec{n} = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Example

Find the equation of the normal vector $\vec{n}$ given the equation for the plane:

$$-3x + \sqrt{2}y + 7z = \pi$$

$$\vec{n} = -3\hat{\imath} + \sqrt{2}\hat{\jmath} + 7\hat{k}$$

$$\vec{n} = \begin{bmatrix} -3 \\ \sqrt{2} \\ 7 \end{bmatrix}$$

2. **Curve**

For a curve described by a function $y = f(x)$, the normal vector at a point on the curve is perpendicular to the tangent line at that point. If the tangent vector has slope $f'(x)$, the normal vector will have a slope of $-\frac{1}{f'(x)}$

4.3.2. Point Distance to Plane

$$\boxed{d = \frac{Ax_0 + By_0 + Cz_0 - D}{\sqrt{A^2 + B^2 + C^2}}}$$

Example

Given a the equation of the plane:

$$1x - 2y + 3z = 5$$

And a point **not** on the plane:

$$(2, 3, 1)$$

Find the shortest path (normal vector) from the plane to the point

$$\begin{aligned} d &= \frac{Ax_0 + By_0 + Cz_0 - D}{\sqrt{A^2 + B^2 + C^2}} \\ &= \frac{1 \cdot 2 - 2 \cdot 3 + 3 \cdot 1 - 5}{\sqrt{1^2 + 2^2 + 3^2}} \\ &= \frac{2 - 6 + 3 - 5}{\sqrt{1 + 4 + 9}} \\ &= -\frac{6}{\sqrt{14}} \end{aligned}$$

4.3.3. Distance Between Planes

4.3.4. Cross Product

Only defined in $\mathbb{R}^3$

Returns a vector orthogonal to the two vectors

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\vec{c} = \vec{a} \times \vec{b}$$

$$\vec{c} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Example

$$\vec{a} = \begin{bmatrix} 1 \\ -7 \\ 1 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix}$$

$$\vec{c} = \vec{a} \times \vec{b}$$

$$\vec{c} = \begin{bmatrix} -7 \cdot 4 - 1 \cdot 2 \\ 1 \cdot 5 - 1 \cdot 4 \\ 1 \cdot 2 - (-7) \cdot 5 \end{bmatrix} = \begin{bmatrix} -30 \\ 1 \\ 37 \end{bmatrix}$$

$\vec{c}$ is orthogonal to both $\vec{a}$ and $\vec{b}$

Proof: $\vec{c}$ is orthogonal to $\vec{a}$ and $\vec{b}$

When the dot product of two vectors is equal to 0, it means that the two vectors are perpendicular (or orthogonal) to each other

1. Orthogonal to Vector $\vec{a}$

$$\vec{c} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$a_1 a_2 b_3 - a_1 a_3 b_2 + a_2 a_3 b_1 - a_2 a_1 b_3 + a_3 a_1 b_2 - a_3 a_2 b_1$$

$$\cancel{a_1 a_2 b_3} \cancel{- a_1 a_3 b_2} + \cancel{a_2 a_3 b_1} \cancel{- a_2 a_1 b_3} + \cancel{a_3 a_1 b_2} \cancel{- a_3 a_2 b_1} = 0$$

2. Orthogonal to Vector $\vec{b}$

$$\vec{c} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$b_1a_2b_3 - b_1a_3b_2 + b_2a_3b_1 - b_2a_1b_3 + b_3a_1b_2 - b_3a_2b_1$$

$$\cancel{b_1a_2b_3} - \cancel{b_1a_3b_2} + \cancel{b_2a_3b_1} - \cancel{b_2a_1b_3} + \cancel{b_3a_1b_2} - \cancel{b_3a_2b_1} = 0$$

4.3.5. Proof: Relationship Between Cross Product and Sin of Angle

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\vec{c} = \vec{a} \times \vec{b}$$

$$\vec{c} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix}$$

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \ \|\vec{b}\| \cos(\Theta)$$

$$\|\vec{a} \times \vec{b}\| = \|\vec{a}\| \ \|\vec{b}\| \sin(\Theta)$$

4.3.6. Dot and Cross Products

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \ \|\vec{b}\| \cos(\Theta)$$

$$\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \ \|\vec{b}\|} = \cos(\Theta)$$

$$\Theta = \arccos()$$

4.4. Row Echelon Form (REF)

Visual Structure:

1. Pivot (leading 1): The leading entry of each non-zero row is 1
2. Zeros below pivots: Every pivot has zeros below it in its column
3. Rightward movement of pivots: Each leading 1 in a lower row is further to the right than in the row above it
4. Rows of all zeros (if any) are at the bottom of the matrix

$$\left[\begin{array}{cccc|c} 1 & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 1 & a_{23} & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Elementary row operations:

1. **Row Swapping**: Swap two rows
2. **Row Multiplication**: Multiply a row by a non-zero scalar
3. **Row Addition / Subtraction**: Add or subtract a multiple of one row from another row

Example

Consider the system of linear equations:

$$\begin{aligned} 2x + y + z &= 8 \\ -3x - y + 2z &= -11 \\ -2x + 1y + 2z &= -3 \end{aligned}$$

The augmented matrix for this system is:

$$\left[\begin{array}{ccc|c} 2 & 1 & 1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$$

Step 1: Make the leading entry of the first row a 1

We divide the first row by 2 (row multiplication):

$$R_1 \to \frac{1}{2}R_1 = \left[\begin{array}{ccc|c} 1 & 0.5 & 0.5 & 4 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$$

Step 2: Eliminate the entries below the first pivot We now want the entries below the first pivot (1 in the first column) to become zeros. We use row addition:

1. $R_2 \to R_2 + 3R_1$
2. $R_3 \to R_3 + 2R_1$

This gives:

$$\left[\begin{array}{ccc|c} 1 & 0.5 & 0.5 & 4 \\ 0 & 0.5 & 3.5 & 1 \\ 0 & 2 & 3 & 5 \end{array}\right]$$

Step 3: Make the leading entry of the second row a 1

We divide the second row by 0.5 (row multiplication):

$$R_2 \to \frac{1}{0.5}R_2 = \left[\begin{array}{ccc|c} 1 & 0.5 & 0.5 & 4 \\ 0 & 1 & 7 & 2 \\ 0 & 2 & 3 & 5 \end{array}\right]$$

Step 4: Eliminate the entry below the second pivot

We now want the entry below the second pivot (1 in the second column) to become zero. We use row addition:

$$R_3 \to R_2 - 2R_2 = \left[\begin{array}{ccc|c} 1 & 0.5 & 0.5 & 4 \\ 0 & 1 & 7 & 2 \\ 0 & 0 & -11 & 1 \end{array}\right]$$

Step 5: Make the leading entry of the third row a 1

We divide the third row by −11 (row multiplication):

$$R_3 \to -\frac{1}{11}R_3 = \left[\begin{array}{ccc|c} 1 & 0.5 & 0.5 & 4 \\ 0 & 1 & 7 & 2 \\ 0 & 0 & 1 & -\frac{1}{11} \end{array}\right]$$

Step 6: Back-substitute to solve for the variables

From the third row:

$$z = -\frac{1}{11}$$

Substitute z into the second row:

$$y + 7z = 2$$
$$y + 7\left(-\frac{1}{11}\right) = 2$$
$$y = 2 + \frac{7}{11}$$
$$y = \frac{29}{11}$$

Substitute y and z into the first row:

$$x + 0.5y + 0.5z = 4$$
$$x + 0.5\left(\frac{29}{11}\right) + 0.5\left(-\frac{1}{11}\right) = 4$$
$$x = 4 - \frac{14}{11} + \frac{1}{11}$$
$$x = \frac{31}{11}$$

Final Solution:

$$x = \frac{31}{11} \qquad y = \frac{29}{11} \qquad z = -\frac{1}{11}$$

4.4.1. Solution Types in Linear Systems: Unique, Infinite, or None

1. Unique Solution

$$\left[\begin{array}{cccc|c} 1 & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 1 & a_{23} & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

2. No Solution

$$0 = a$$

3. No Unique Solution (Infinite Number of Solutions)

Column 2 and 4 indicate free variables x_2 and x_4 becuase they have no pivot entries

$$\left[\begin{array}{cccc|c} 1 & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & 0 & 1 & a_{24} & b_2 \\ 0 & 0 & 1 & a_{34} & b_3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

4.4.2. Special Cases

Rows of all zeros appear in row echelon form (REF) in the following situations:

1. Dependent Equations

Some equations are multiples or linear combinations of others

$$2x + 4y = 8$$
$$x + 2y = 4$$

Augmented matrix:

$$\left[\begin{array}{cc|c} 2 & 4 & 8 \\ 1 & 2 & 4 \end{array}\right]$$

After row reduction:

$$\left[\begin{array}{cc|c} 1 & 2 & 4 \\ 0 & 0 & 0 \end{array}\right]$$

2. Underdetermined Systems

Number of variables is greater than the number of independent equations

$$x + y + z = 2$$
$$2x + 3y + z = 5$$

Augmented matrix:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 2 & 3 & 1 & 5 \end{array}\right]$$

After row reduction:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

3. Inconsistent Systems

No solution exists

Rows of zeros on the left side (coefficients of the variables) and a non-zero entry on the right side (augmented column)

$$\begin{aligned} x + y &= 3 \\ 2x + 2y &= 7 \end{aligned}$$

Augmented matrix:

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 2 & 2 & 7 \end{array}\right]$$

After row reduction:

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 0 & 1 \end{array}\right]$$

5. Matrices

$m \times n$ matrix $\boldsymbol{A}$

- m: rows
- n: columns

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

5.1. Matrix-Vector Products

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boldsymbol{A}\mathbf{x} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

For the dot product to be defined, the number of columns in the matrix $\boldsymbol{A}$ (which is n) must match the number of elements in the vector $\vec{x}$ (also n).

The result of multiplying matrix $\boldsymbol{A}$ and vector $\vec{x}$ will be a column vector with dimensions $m \times 1$, where m is the number of rows in the matrix $\boldsymbol{A}$

$$(m \times n) \cdot (n \times 1) = m \times 1$$

1. As Row vectors

$$\color{red}{\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}}$$

$$\color{blue}{\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}$$

$$\color{red}{\mathbf{a}^T = [a_1, a_2, \ldots, a_n]}$$
$$\color{blue}{\mathbf{b}^T = [b_1, b_2, \ldots, b_n]}$$

$$\boldsymbol{A} = \begin{bmatrix} \color{red}{[a_1, a_2, \ldots, a_n]} \\ \color{blue}{[b_1, b_2, \ldots, b_n]} \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \color{red}{\mathbf{a}} \\ \color{blue}{\mathbf{b}} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{a}^T \\ \mathbf{b}^T \end{bmatrix} \cdot \mathbf{x} = \begin{bmatrix} \mathbf{a} \cdot \mathbf{x} \\ \mathbf{b} \cdot \mathbf{x} \end{bmatrix}$$

2. As Column Vectors

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} & \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \vec{a} & \vec{b} \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{A}\vec{x} = x_1\vec{a} + x_2\vec{b}$$

5.2. Null Space

The null space (or kernel) of a matrix $\boldsymbol{A}$ is the set of all vectors $\boldsymbol{x}$ that satisfy the equation:

$$\boldsymbol{A}\vec{x} = \mathbf{0}$$

Where:

- $\boldsymbol{A}$: $m \times n$ matrix
- $\vec{x}$: n-dimansional vector
- $\mathbf{0}$: zero vecor in $\mathbb{R}^m$

$$N(\boldsymbol{A}) = N(\text{rref}(\boldsymbol{A})) = \text{span}(\vec{v}_1, \vec{v}_2, \vec{v}_3)$$

Example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

We want to find the null space of A, which consists of all vectors $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ that satisfy:

$$\boldsymbol{A}\vec{x} = \boldsymbol{0}$$

This expands to the following system of linear equations:

$$\begin{cases} 1x_1 + 1x_2 + 1x_3 + 1x_4 = 0 \\ 1x_1 + 2x_2 + 3x_3 + 4x_4 = 0 \\ 4x_1 + 3x_2 + 2x_3 + 1x_4 = 0 \end{cases}$$

This can be represented as the augmented matrix:

$$\left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 \\ 4 & 3 & 2 & 1 & 0 \end{array}\right]$$

5.2.1. Column Space

The **columns space** (or range) of matrix A is span of its columns vectors

If the matrix A has columns $\vec{a}_1, \vec{a}_2, ..., \vec{a}_n$, then the column space of A is defined as:

$$\text{Col}(A) = \{\vec{y} \in \mathbb{R}^m \mid \vec{y} = A\vec{x} \text{ for some } \vec{x} \in \mathbb{R}^n\}$$

or equivalently,

$$\text{Col}(A) = \text{span}(\{\vec{a}_1, \vec{a}_2, ..., \vec{a}_n\})$$

Example

Consider the simple example of a 2×2 matrix:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$$

The matrix has two columns:

$$\vec{a}_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \text{and} \quad \vec{a}_2 = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

The column space, denoted $\mathrm{Col}(A)$, is the span of these two vectors:

$$\mathrm{Col}(A) = \mathrm{span}\left(\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix} \right\} \right)$$

Finding the Column Space

We observe that the two columns $\vec{a}_1$ and $\vec{a}_2$ are **linearly dependent**:

$$\vec{a}_2 = k\vec{a}_1$$

This means that $\vec{a}_2$ is a scalar multiple of $\vec{a}_1$, the the two columns are **linearly dependent**. As a result, the column space is spanned by just one vector, $\vec{a}_1$, because any linear combination of $\vec{a}_1$ and $\vec{a}_2$ can be reduced to a multiple of $\vec{a}_1$.

Therefore, the column space of A is:

$$\mathrm{Col}(A) = \mathrm{span}\left(\left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right\} \right)$$

which represents all vectors of the form:

$$c \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} c \\ 3c \end{bmatrix} \quad \text{for any scalar } c$$

In other words, the column space is a line in $\mathbb{R}^2$ through the origin in the direction of

Rank of A

The rank of A, which is the **dimension of its column space**, is 1 because there is only one linearly independent column

This means the column space is the span of the columns of A, or all vectors that can be formed by taking linear combinations of the columns of A.

5.2.2. Dimension of a Subspace

Number of elements in a basis for the subspace

5.2.3. Nullity

Dimension of the **Null Space**

$$\dim(N(\boldsymbol{A}))$$

The nullity of A: number of non-pivot columns (i.e., free variables) in the rref of A

5.2.4. Rank

Dimension of the column **space**

$$\text{rank}(A) = \dim(C(A))$$

5.2.5. Matrix Representation of Systems of Equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m &= b_n \end{aligned}$$

Coefficient Matrix (A):

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Variable Vector (x):

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Constant Vector (b):

$$\mathrm{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\mathrm{Ax} = b$$

Example

The system of equations:

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 &= 100 \\ 4x_1 + 2x_2 + 1x_3 &= 80 \\ 1x_1 + 5x_2 + 2x_3 &= 60 \end{aligned}$$

Can be represented as a matrix equation:

$$\begin{bmatrix} 2 & 3 & 5 \\ 4 & 2 & 1 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 80 \\ 60 \end{bmatrix}$$

5.3. Matrix Multiplication

$m \times n$ matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$n \times p$ matrix:

$$B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{bmatrix}$$

Compute Each Element of Result Matrix C

$$c_{ij} = \sum_{k=1}^{n} = a_{ik} b_{kj}$$

Example

Let A be an $n \times m$ matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Let B an $p \times n$ matrix:

$$B = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

Calculate Each Element of C

$$
\begin{aligned}
c_{11} &= (1 \cdot 7) + (2 \cdot 9) + (3 \cdot 11) = 58 \\
c_{12} &= (1 \cdot 8) + (2 \cdot 10) + (3 \cdot 12) = 64 \\
c_{21} &= (4 \cdot 7) + (5 \cdot 9) + (6 \cdot 11) = 138 \\
c_{22} &= (4 \cdot 8) + (5 \cdot 10) + (6 \cdot 12) = 154
\end{aligned}
$$

C is a $m \times p$ matrix

$$C = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

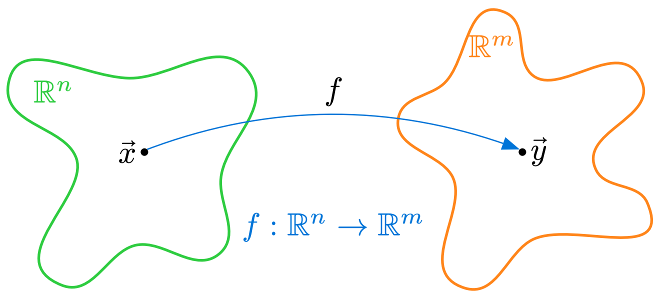
5.4. Linear Transformation

5.4.1. Functions

A function f that maps elements from a set X (the domain) to a set Y (the codomain):

$$f : X \to Y$$

- Domain: The set X contains all possible inputs for the function f
- Codomain: The set Y is the space where all possible outputs of f reside, though not every element in Y must be an output of f



Example

If

$$f : \mathbb{R} \to \mathbb{R}$$

is defined by

$$f(x) = x^2$$
$$f : x \mapsto x^2$$

then:

- **Domain**: $X = \mathbb{R}$, any real number $((\infty, \infty))$
- **Codomain**: $Y = \mathbb{R}$, any real number $((\infty, \infty))$
- **Range**: the subset of the codomain ($\mathbb{R}$), $[0, \infty)$

5.5. Vector Transformation

A function f that maps an n-dimensional vector in $\mathbb{R}^n$ to an m-dimensional vector in $\mathbb{R}^m$:

1. Function Definition

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

This means f takes as input a vector in $\mathbb{R}^n$ (an n-dimensional space of real numbers) and maps it to a vector in $\mathbb{R}^m$ (an m-dimensional space of real numbers).

2. Input Vector $\vec{x}$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{where} \quad x_1, x_2, \ldots, x_n \in \mathbb{R}$$

Here, $\vec{x}$ is an n-dimensional vector, and each component x_i is a real number

3. Input Vector $\vec{y}$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \text{where} \quad y_1, y_2, \ldots, y_m \in \mathbb{R}$$

The output $\vec{y}$ is an m-dimensional vector, and each component y_i is also a real number

Summary

The function f takes an n-dimensional vector of real numbers as input and produces an m-dimensional vector of real numbers as output

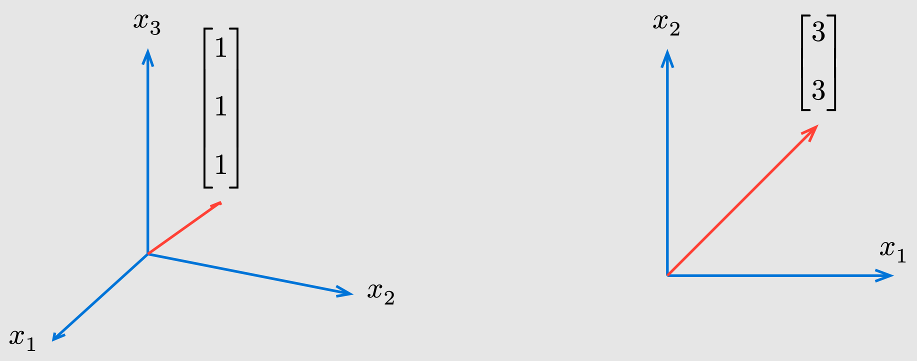
Example

$$f(x_1, x_2, x_3) = (x_1 + 2x_2, 3x_3)$$

$$f : \mathbb{R}^3 \to \mathbb{R}^2$$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_3 \end{bmatrix}$$

$$f\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$



5.6. Linear Transformation

$$T : \mathbb{R}^n \to \mathbb{R}^m$$

$$\vec{a}, \vec{b} \in \mathbb{R}^n$$

For a transformation **linear** it must satisfy two conditions:

1. Additivity (or linearity of addition)

$$\boxed{T\left(\vec{a} + \vec{b}\right) = T(\vec{a}) + T\left(\vec{b}\right)}$$

Example

Let's consider a linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 2x \\ 3y \end{bmatrix}$$

Now let's take two vectors in $\mathbb{R}^2$:

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Then the additivity property can be verified as follows:

1. First, find $T(\vec{a}) + T\left(\vec{b}\right)$ separately:

$$T(\vec{a}) = T\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 2 \cdot 1 \\ 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

$$T\left(\vec{b}\right) = T\left(\begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 2 \cdot 3 \\ 3 \cdot 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

$$T(\vec{a}) + T\left(\vec{b}\right) = \begin{bmatrix} 2 \\ 6 \end{bmatrix} + \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 2+6 \\ 6+3 \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \end{bmatrix}$$

2. Next, find $T\left(\vec{a}+\vec{b}\right)$:

$$\vec{a}+\vec{b}=\begin{bmatrix}1\\2\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}=\begin{bmatrix}4\\3\end{bmatrix}$$

$$T\left(\vec{a}+\vec{b}\right)=T\left(\begin{bmatrix}4\\3\end{bmatrix}\right)=\begin{bmatrix}2\cdot 4\\3\cdot 3\end{bmatrix}=\begin{bmatrix}8\\9\end{bmatrix}$$

Since $T\left(\vec{a}+\vec{b}\right)=T(\vec{a})+T\left(\vec{b}\right)$, this confirms the additivity (linearity of addition) property of the transformation T

2. Homogeneity (or linearity of scalar multiplication):

$$\boxed{T(c\vec{a})=cT(\vec{a})}$$

Example

Let's consider a linear transformation $T:\mathbb{R}^2\to\mathbb{R}^2$

$$T\left(\begin{bmatrix}x\\y\end{bmatrix}\right)=\begin{bmatrix}2x\\3y\end{bmatrix}$$

Now let's take one vectors in $\mathbb{R}^2$:

$$\vec{a}=\begin{bmatrix}1\\2\end{bmatrix}$$

and a scalar $c=3$

Then the homogeneity property can be verified as follows:

1. First, find $cT(\vec{a})$

$$T(\vec{a})=T\left(\begin{bmatrix}1\\2\end{bmatrix}\right)=\begin{bmatrix}2\cdot 1\\3\cdot 2\end{bmatrix}=\begin{bmatrix}2\\6\end{bmatrix}$$

$$cT(\vec{a})=3\cdot\begin{bmatrix}2\\6\end{bmatrix}=\begin{bmatrix}3\cdot 2\\3\cdot 6\end{bmatrix}=\begin{bmatrix}6\\18\end{bmatrix}$$

2. Then, find $T(c\vec{a})$ by c to get $c\vec{a}$

$$c\vec{a}=3\cdot\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}3\cdot 1\\3\cdot 2\end{bmatrix}=\begin{bmatrix}3\\6\end{bmatrix}$$

$$T(c\vec{a})=T\left(\begin{bmatrix}3\\6\end{bmatrix}\right)=\begin{bmatrix}2\cdot 3\\3\cdot 6\end{bmatrix}=\begin{bmatrix}6\\18\end{bmatrix}$$

Since $T(c\vec{a}) = cT(\vec{a})$, this confirms the homogeneity (linearity of scalar multiplication) property of the transformation T

5.7. Matrix Vector Products

Matrix product with vector is always a linear transformation

$$T : \mathbb{R}^n \to \mathbb{R}^m$$
$$T(\vec{x}) = A\vec{x}$$

$$A = \begin{bmatrix} \\ v_1 & v_2 & \dots & v_n \\ \\ \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$A\vec{x} = \begin{bmatrix} \\ v_1 & v_2 & \dots & v_n \\ \\ \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 v_1 + x_2 v_2 + \dots + x_n v_n$$

1. Additivity (or linearity of addition)

$$\boxed{T\left(\vec{a} + \vec{b}\right) = T(\vec{a}) + T\left(\vec{b}\right)}$$

$$\begin{aligned} A \cdot \left(\vec{a} + \vec{b}\right) = A \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{bmatrix} &= (a_1 + b_1)v_1 + (a_2 + b_2)v_2 + \dots + (a_n + b_n)v_n \\ &= a_1 v_1 + b_1 v_1 + a_2 v_2 + b_2 v_2 + \dots + a_n v_n + b_n v_n \\ &= (a_1 v_1 + a_2 v_2 + \dots + a_n v_n) + (b_1 v_1 + b_2 v_2 + \dots + b_n v_n) \\ &= A \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} + A \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \end{aligned}$$

Example

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

1. Calculate $A(\boldsymbol{u} + \boldsymbol{v})$

$$\begin{aligned} \boldsymbol{u} + \boldsymbol{v} &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ &= \begin{bmatrix} 4 \\ 6 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} A(\boldsymbol{u} + \boldsymbol{v}) &= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \end{bmatrix} \\ &= \begin{bmatrix} (2 \cdot 1) + (1 \cdot 6) \\ (0 \cdot 4) + (3 \cdot 6) \end{bmatrix} \\ &= \boxed{\begin{bmatrix} 14 \\ 18 \end{bmatrix}} \end{aligned}$$

2. Calculate $A\boldsymbol{u} + A\boldsymbol{v}$

$$\begin{aligned} A\boldsymbol{u} &= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} (2 \cdot 1) + (1 \cdot 2) \\ (0 \cdot 1) + (3 \cdot 2) \end{bmatrix} \\ &= \begin{bmatrix} 4 \\ 6 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} A\boldsymbol{v} &= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ &= \begin{bmatrix} (2 \cdot 3) + (1 \cdot 4) \\ (0 \cdot 3) + (3 \cdot 4) \end{bmatrix} \\ &= \begin{bmatrix} 10 \\ 12 \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
A\boldsymbol{u} + A\boldsymbol{v} &= \begin{bmatrix} 4 \\ 6 \end{bmatrix} + \begin{bmatrix} 10 \\ 12 \end{bmatrix} \\
&= \boxed{\begin{bmatrix} 14 \\ 18 \end{bmatrix}}
\end{aligned}
$$

2. Homogeneity (or linearity of scalar multiplication):

$$
\boxed{T(c\vec{a}) = cT(\vec{a})}
$$

$$
\begin{aligned}
A \cdot (c\vec{a}) &= \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix} \begin{bmatrix} ca_1 \\ ca_2 \\ \vdots \\ ca_n \end{bmatrix} \\
&= ca_1v_1 + ca_2v_2 + \dots + ca_nv_n \\
&= c\underbrace{(a_1v_1 + a_2v_2 + \dots + a_nv_n)}_{A\vec{a}}
\end{aligned}
$$

Example

$$
A = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \qquad c = 5
$$

1. Calculate $A(c\boldsymbol{v})$

$$
cv = 5 \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 15 \\ 20 \end{bmatrix}
$$

$$
\begin{aligned}
A(c\boldsymbol{v}) &= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 15 \\ 20 \end{bmatrix} \\
&= \begin{bmatrix} (2 \cdot 15) + (1 \cdot 20) \\ (0 \cdot 15) + (3 \cdot 20) \end{bmatrix} \\
&= \boxed{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}
\end{aligned}
$$

2. Calculate $c(A\boldsymbol{v})$

$$
\begin{aligned}
A\boldsymbol{v} &= \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\
&= \begin{bmatrix} (2 \cdot 3) + (1 \cdot 4) \\ (0 \cdot 3) + (3 \cdot 4) \end{bmatrix} \\
&= \begin{bmatrix} 10 \\ 12 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
c(A\boldsymbol{v}) &= 5 \cdot \begin{bmatrix} 10 \\ 12 \end{bmatrix} \\
&= \boxed{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}
\end{aligned}
$$

5.8. Linear transformations as matrix vector products

The $n \times n$ matrix I_n:

$$
I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}
$$

$$
\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
$$

$$
I_n\vec{x} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
$$

Standard Basis

$$I_n\vec{x} = \begin{bmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underbrace{0}_{e_1} & \underbrace{0}_{e_2} & \underbrace{0}_{e_3} & ... & \underbrace{1}_{e_n} \end{bmatrix}$$

$\{e_1, e_2, ..., e_n\}$ is the standard basis for $\mathbb{R}^n$

$$I_n \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = a_1\vec{e}_1 + a_2\vec{e}_2 + ... + a_n\vec{e}_n$$

$$= a_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + a_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + ... + a_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a_2 \\ \vdots \\ 0 \end{bmatrix} + ... + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ a_n \end{bmatrix}$$

5.9. Image of a subset under transformation

$$\vec{x}_0 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \qquad \vec{x}_1 = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \qquad \vec{x}_2 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$

$$L_0 = \{\vec{x}_0 + t(\vec{x}_1 - \vec{x}_0) \mid 0 \leq t \leq 1\}$$
$$L_1 = \{\vec{x}_1 + t(\vec{x}_2 - \vec{x}_1) \mid 0 \leq t \leq 1\}$$
$$L_2 = \{\vec{x}_2 + t(\vec{x}_0 - \vec{x}_2) \mid 0 \leq t \leq 1\}$$

The triangle T can be defines as the set of these points:

$$S = \{L_0, L_1, L_2\}$$

Let's define a transformation

$$T : \mathbb{R}^2 \to \mathbb{R}^2$$

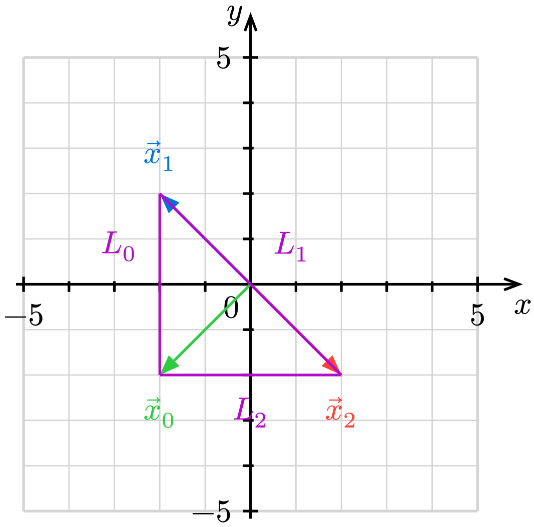
$$T(\vec{x}) = \begin{bmatrix} 1 & -1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

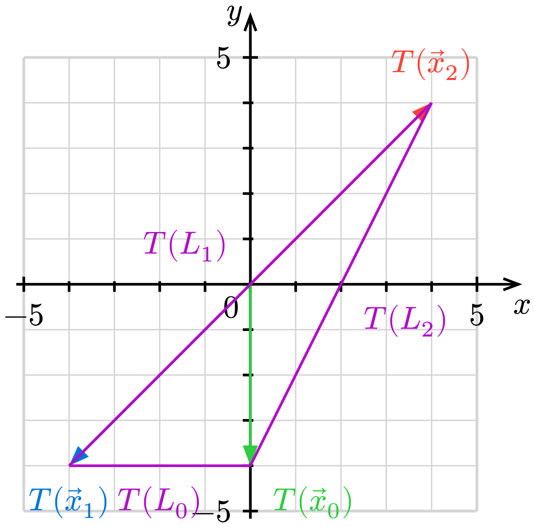
$$\begin{aligned} T(L_0) &= \{T(\vec{x}_0 + t(\vec{x}_1 - \vec{x}_0)) \mid 0 \le t \le 1\} \\ &= \{T(\vec{x}_0) + T(t(\vec{x}_1 - \vec{x}_0)) \mid 0 \le t \le 1\} \\ &= \{T(\vec{x}_0) + tT(\vec{x}_1 - \vec{x}_0) \mid 0 \le t \le 1\} \\ &= \{T(\vec{x}_0) + tT(\vec{x}_1) - T(\vec{x}_0) \mid 0 \le t \le 1\} \end{aligned}$$

$$T(\vec{x}_0) = \begin{bmatrix} 1 & -1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

$$T(\vec{x}_1) = \begin{bmatrix} 1 & -1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ 2 \end{bmatrix} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

$$T(\vec{x}_2) = \begin{bmatrix} 1 & -1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$





$T(L_0)$ is the image of L_0 under T

$T(S)$ is the image of S under T

5.10. Image of a transformation

The **image** of a transformation T is defined as:

$$\operatorname{im}(T) = \{T(\vec{x}) \mid \vec{x} \in \mathbb{R}^n\}$$

or, equivalently,

$$T(\mathbb{R}^n)$$

$T(\mathbb{R}^n)$ is the image of $\mathbb{R}^2$ under T

This is the set of all possible outputs when T is applied to vectors in $\mathbb{R}^n$

Undertanding $T(\mathbb{R}^n)$

1. Whole space transformation

   The image of $\mathbb{R}^n$ under T is the complete set of transformed vectors, often denoted as $\mathrm{im}(T)$

2. Subset Transformation

   For any subset $V \subseteq \mathbb{R}^n$, the image of V under T is the set of transformed vectors from V

Matrix representation of T

If T is represented by a $m \times n$ matrix A, then:

$$T(\vec{x}) = \{A\vec{x} \mid \vec{x} \in \mathbb{R}^n\}$$

where:

- A is the matrix associated with T
- $\vec{x} \in \mathbb{R}^n$ represents a vector in the input space

Transformation in therms of columns of A

If $A = \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & ... & \vec{a}_n \end{bmatrix}$, then for $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$:

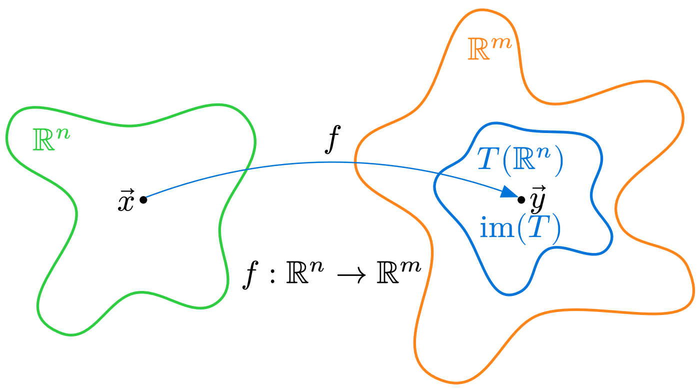
$$A\vec{x} = x_1\vec{a}_1 + x_2\vec{a}_2 + ... + x_n\vec{a}_n$$

Column Space of A

The image of T (or $\mathrm{im}(T)$) is the column space of A:

$$C(A) = \mathrm{span}(\vec{a}_1, \vec{a}_2, ..., \vec{a}_n)$$

This is the set of all possible linear combinations of the columns of A, and thus represents all possible outputs of the transformation T



Example

Supose we have a matrix A:

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

Matrix A defines the transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ such that for any vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$, the image under T is:

$$T(\vec{x}) = A\vec{x} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Calculating Transformation

To see what T does to vectors in $\mathbb{R}^2$, let's compute a few specific examples:

1. for $\vec{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$T\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = A\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

2. for $\vec{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$T\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = A\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Image of T

The image of T, $\mathrm{im}(T)$, is the set of all linear combinations of the vectors $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$:

$$\mathrm{im}(T) = \mathrm{span}\left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}\right)$$

Thus, any vector in the image of T can be written as:

$$y = x_1\begin{bmatrix} 2 \\ 1 \end{bmatrix} + x_2\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 3x_2 \end{bmatrix}$$

where $x_1, x_2 \in \mathbb{R}$.

Column space interpretation

The image of T is all the vectors in $\mathbb{R}^2$ that can be formed as linear combinations of $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$.

5.11. Preimage of a set

The preimage of a set S under a function T, denoted $T^{-1}(S)$, is the set of all elements in the domain of T the domain of T that map to elements in S under the transformation T.

If $T : \mathbb{R}^n \to \mathbb{R}^m$ is a function from a set $\mathbb{R}^n$ to a set $\mathbb{R}^m$, and $S \subseteq \mathbb{R}^m$ is a subset of a target space, the the preimage of S under T is:

$$T(-1)(S) = \{\vec{x} \in \mathbb{R}^n \mid T(\vec{x}) \in S\}$$

This means that $T(-1)(S)$ consists of all elements in $\mathbb{R}^n$ that, when transformed by T, end up in S.

For any subset $S \subseteq \mathbb{R}^m$, the preimage $T^{-1}(S)$ collects all points in the domain that end up in S after applying T. If S is a single point, the preimage will be the set of all points in the domain that map to that specific point (this could be empty, a single point, or even a set of points, depending on the function).

Example

Consider the linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ given by the matrix:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

This transformation T maps any vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ in $\mathbb{R}^2$ to:

$$T(\vec{x}) = A\vec{x} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 3x_2 \end{bmatrix}$$

Now, let's find the preimage of a subset $S \subseteq \mathbb{R}^2$. Suppose we want the preimage of the set $S = \left\{ \begin{bmatrix} 4 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\}$

The primage of S under T, denoted $T^{-1}(S)$, consists of all vectors $\vec{x} \in \mathbb{R}^2$ such that $T(\vec{x}) \in \left\{ \begin{bmatrix} 4 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\}$. To find this, we solve for $\vec{x}$ in both cases: $T(\vec{x}) = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$ and $T(\vec{x}) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

Preimage of $\begin{bmatrix} 4 \\ 6 \end{bmatrix}$

$$A\vec{x} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

or

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Solving each component:

1. $2x_1 = 4 \Rightarrow x_1 = 2$
2. $3x_2 = 6 \Rightarrow x_2 = 2$

Thus, the primage of S is the single point:

$$T^{-1}\left(\begin{bmatrix} 4 \\ 6 \end{bmatrix}\right) = \left\{\begin{bmatrix} 2 \\ 2 \end{bmatrix}\right\}$$

Preimage of $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$$A\vec{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

or

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Solving each component:

1. $2x_1 = 2 \Rightarrow x_1 = 1$
2. $3x_2 = 3 \Rightarrow x_2 = 1$

Thus, the primage of S is the single point:

$$T^{-1}\left(\begin{bmatrix} 2 \\ 3 \end{bmatrix}\right) = \left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\}$$

Preimage of the set S

Since $S = \left\{\begin{bmatrix} 4 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}\right\}$, the prrimage of S is the union of the preimage of each vector in S:

$$T^{-1}(S) = \left\{\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\}$$

5.11.1. Kernel of a Transofrmation

The kernel of a transformation $T : \mathbb{R}^n \to \mathbb{R}^m$, denoted $\ker(T)$, is the set of all vectors in $\mathbb{R}^n$ that T maps to the zero vector in $\mathbb{R}^m$. Formally, we define the kernel as:

$$\ker(T) = \left\{\vec{x} \in \mathbb{R}^n \mid T(\vec{x}) = \vec{0}\right\}$$

The consists of all vectors that are “annihilated” by T, resulting in the zero vector after applying T.

Example

The transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ defined by the matrix

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

so that $T(\vec{x}) = A\vec{x} = \begin{bmatrix} 2x_1 \\ 3x_2 \end{bmatrix}$ for any $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$

To find the kernel of T, we need to find all the vectors $\vec{x} \in \mathbb{R}^2$ that satisfy:

$$T(\vec{x}) = \vec{0} \Rightarrow \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This leads to the system of equations:

1. 3 x_1 = 0 arrow.double x_1 = 0
2. 3 x_2 = 0 arrow.double x_2 = 0

Thus, the only solution is $\vec{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

5.11.2. Kernel and Null Space

The kernel of T

$$\ker(T) = \mathrm{Null}(A)$$

5.12. Sum and Scalar Multiples of Linear Transformation

5.12.1. Sum

$$T : \mathbb{R}^n \to \mathbb{R}^m \qquad S : \mathbb{R}^n \to \mathbb{R}^m$$

$$(T + S) : \mathbb{R}^n \to \mathbb{R}^m$$

$$A = \begin{bmatrix} \\ \vec{a}_1 & \vec{a}_2 & ... & \vec{a}_n \\ \\ \end{bmatrix} \quad B = \begin{bmatrix} \\ \vec{b}_1 & \vec{b}_2 & ... & \vec{b}_n \\ \\ \end{bmatrix}$$

$$
\begin{aligned}
(T+S)(\vec{x}) &= T(\vec{x}) + S(\vec{x}) \\
&= A\vec{x} + B\vec{x} \\
&= x_1\vec{a}_1 + x_2\vec{a}_2 + \ldots + x_n\vec{a}_n + x_1\vec{b}_1 + x_2\vec{b}_2 + \ldots + x_n\vec{b}_n \\
&= x_1\left(\vec{a}_1 + \vec{b}_1\right) + x_2\left(\vec{a}_2 + \vec{b}_2\right) + \ldots + x_n\left(\vec{a}_n + \vec{b}_n\right) \\
&= \begin{bmatrix} \vec{a}_1 + \vec{b}_1 & \vec{a}_2 + \vec{b}_2 & \ldots & \vec{a}_n + \vec{b}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \\
&= \boxed{(A+B)\vec{x}}
\end{aligned}
$$

5.12.2. Scalar Multiplication

$$T : \mathbb{R}^n \to \mathbb{R}^m$$

$$cT : \mathbb{R}^n \to \mathbb{R}^m$$

$$
\begin{aligned}
(cT)(\vec{x}) &= c(T(\vec{x})) \\
&= c(x_1\vec{a}_1 + x_2\vec{a}_2 + \ldots + x_n\vec{a}_n) \\
&= x_1c\vec{a}_1 + x_2c\vec{a}_2 + \ldots + x_nc\vec{a}_n \\
&= \begin{bmatrix} c\vec{a}_1 & c\vec{a}_2 & \ldots & c\vec{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \\
&= \boxed{cA\vec{x}}
\end{aligned}
$$

5.13. Projection

The projection of a vector $\vec{x}$, onto a line L, denoted as $\mathrm{Proj}_L(\vec{x})$, is a vector that lies on the line L, such that the difference between $\vec{x}$ and its projection, $\mathrm{Proj}_L(\vec{x}) - \vec{x}$, is orthogonal to L

$\mathrm{Proj}_L(\vec{x})$ can be seen as the “shadow”cast by $\vec{x}$ onto L when light shines perpendicularly to L.

$$\mathrm{Proj}_L(\vec{x}) = c\vec{v} = \left(\frac{\vec{x} \cdot \vec{v}}{\vec{v} \cdot \vec{v}}\right)\vec{v}$$

Equivalently

$$\mathrm{Proj}_L(\vec{x}) = c\vec{v} = \left(\frac{\vec{x} \cdot \vec{v}}{\|\vec{v}\|^2}\right)\vec{v}$$

Where:

- $\vec{v}$ is a direction vector for the line L
- $c = \frac{\vec{x}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}$ is a scalar

Example

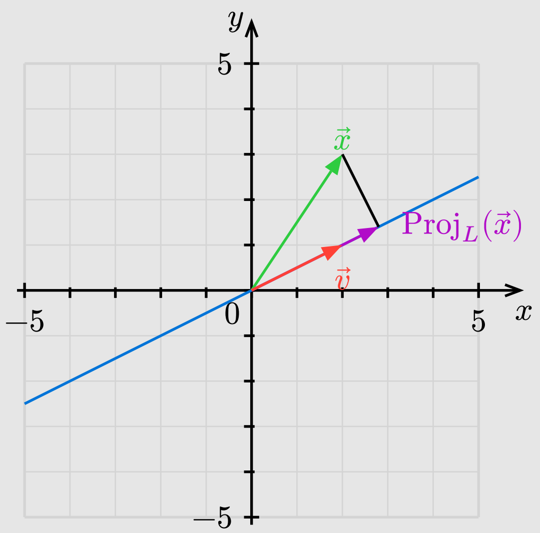
$$\vec{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \qquad \vec{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

1. Define the line L as all vectors of the form $c\vec{v}$, where c is a scalar:

$$\begin{aligned} L &= \{c\vec{v} \mid c \in \mathbb{R}\} \\ &= \left\{ c \begin{bmatrix} 2 \\ 1 \end{bmatrix} \mid c \in \mathbb{R} \right\} \end{aligned}$$

2. Compute the projection using

$$\begin{aligned} \mathrm{Proj}_L(\vec{x}) &= \frac{\vec{x}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}\vec{v} \\ &= \frac{\begin{bmatrix} 2 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix}}{\begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \frac{7}{5} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 2.8 \\ 1.4 \end{bmatrix} \end{aligned}$$



Projection as a Transformation

As a matrix vector product

$$L = \{c\vec{v} \mid c \in \mathbb{R}\}$$

$$\text{Proj}_L : \mathbb{R}^n \to \mathbb{R}^n$$

$$\vec{v} \cdot \vec{v} = \| \vec{v} \|^2$$

If $\vec{v}$ is a unit vector:

$$\|\vec{v}\| = 1$$

Then

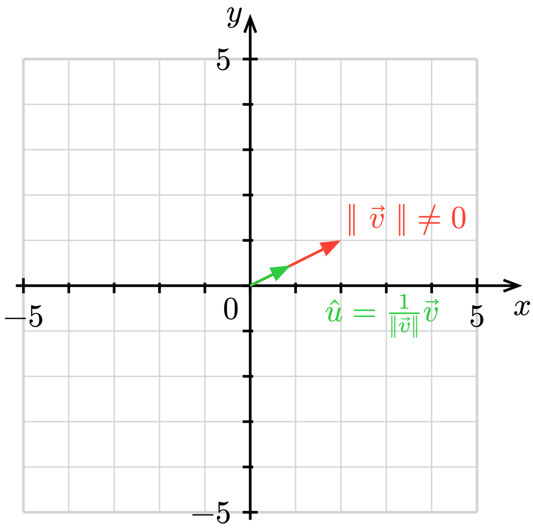
$$\begin{aligned}\text{Proj}_L(\vec{x}) &= \left(\frac{\vec{x} \cdot \vec{v}}{\vec{v} \cdot \vec{v}}\right)\vec{v} \\ &= \left(\frac{\vec{x} \cdot \vec{v}}{\| \vec{v} \|^2}\right)\vec{v}\end{aligned}$$

If we redefine our line L as all the scalar multiples of out unit vector $\hat{u}$:

$$L = \{c\hat{u} \mid c \in \mathbb{R}\}$$

Simplifies to:

$$\boxed{(\vec{x} \cdot \hat{u})\hat{u}}$$



Projection as a Linear Transformation

Let $\hat{u} \in \mathbb{R}^n$ be a unit vector:

$$\hat{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \quad \text{where } \|\hat{u}\| = 1$$

The projection matrix A is:

$$A = \hat{u}\hat{u}^T = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \ldots & u_n \end{bmatrix}$$

Expands to:

$$A = \begin{bmatrix} u_1u_1 & u_1u_2 & \ldots & u_1u_n \\ u_2u_1 & u_2u_2 & \ldots & u_2u_n \\ \vdots & \vdots & \ddots & \vdots \\ u_nu_1 & u_nu_2 & \ldots & u_nu_n \end{bmatrix}$$

For any $\vec{x} \in \mathbb{R}^n$, the projection of $\vec{x}$ onto the line spanned by $\hat{u}$ is:

$$\text{Proj}_L(\vec{x}) = A\vec{x} = (\hat{u} \cdot \vec{x})\hat{u}$$

Example

Consider a vector $\vec{v}$ in $\mathbb{R}^2$:

$$\vec{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

1. Construct the Unit Vector

$$\|\vec{v} = \sqrt{2^2 + 1^2} = \sqrt{5}$$

$$\hat{u} = \frac{\vec{v}}{\| \vec{v} \|} = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix}$$

This unit vector $\hat{u}$ defines the line L, which consists of all the scalar multiples og $\vec{v}$:

$$L = \{c\vec{v} \mid c \in \mathbb{R}\}$$

2. Derive the Projection Matrix

The projection of any vector $\vec{x}$ onto the line L is given by:

$$\text{Proj}_L(\vec{x}) = A\vec{x}$$

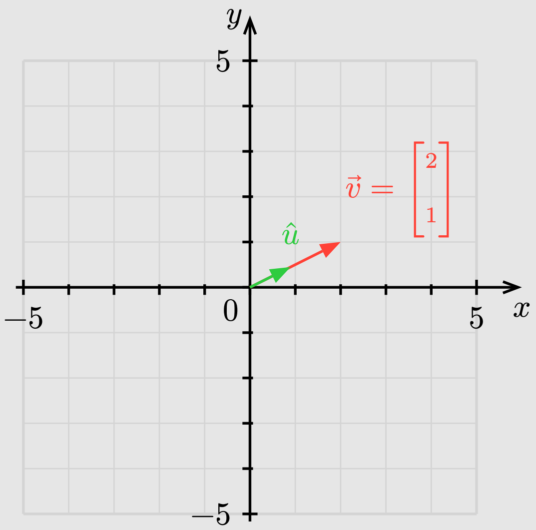
Where A is the projection matrix. To construct A we use the formula:

$$
\begin{aligned}
A &= \hat{u}\hat{u}^T \\
&= \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} u_1 & u_2 \end{bmatrix} \\
&= \begin{bmatrix} u_1 \cdot u_1 & u_1 \cdot u_2 \\ u_2 \cdot u_1 & u_2 \cdot u_2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \\
&= \begin{bmatrix} \left(\frac{2}{\sqrt{5}}\right)^2 & \frac{1}{\sqrt{5}}\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}}\frac{1}{\sqrt{5}} & \left(\frac{1}{\sqrt{5}}\right)^2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{4}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} \end{bmatrix}
\end{aligned}
$$

3. Applying the Projection

To project any vector $\vec{x}$ onto L, we multiply $\vec{x}$ by the matrix A:

$$
\begin{aligned}
\mathrm{Proj}_L(\vec{x}) &= A\vec{x} \\
&= \begin{bmatrix} \frac{4}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} \end{bmatrix} \vec{x}
\end{aligned}
$$



1. Additivity of Projections (Linearity with respect to addition)

$$
\begin{aligned}
\mathrm{Proj}_L\left(\vec{a}+\vec{b}\right) &= \left(\left(\vec{a}+\vec{b}\right)\cdot\hat{u}\right) \\
&= \left(\vec{a}\cdot\hat{u}+\vec{b}\cdot\hat{u}\right)\hat{u} \\
&= (\vec{a}\cdot\hat{u})\hat{u}+\left(\vec{b}\cdot\hat{u}\right)\hat{u} \\
&= \mathrm{Proj}_L(\vec{a})+\mathrm{Proj}_L\left(\vec{b}\right)
\end{aligned}
$$

2. Homogeneity of Projections (Linearity with respect to scalar multiplication)

$$
\begin{aligned}
\mathrm{Proj}_L(c\vec{a}) &= (c\vec{a}\cdot\hat{u})\hat{u} \\
&= c(\vec{a}\cdot\hat{u})\hat{u} \\
&= c\mathrm{Proj}_L(\vec{a})
\end{aligned}
$$

General Properties of A

1. Idempotence

$$A^2 = A$$

2. Symmetry

$$A^T = A$$

3. Rank

$$\mathrm{rank}(A) = 1$$

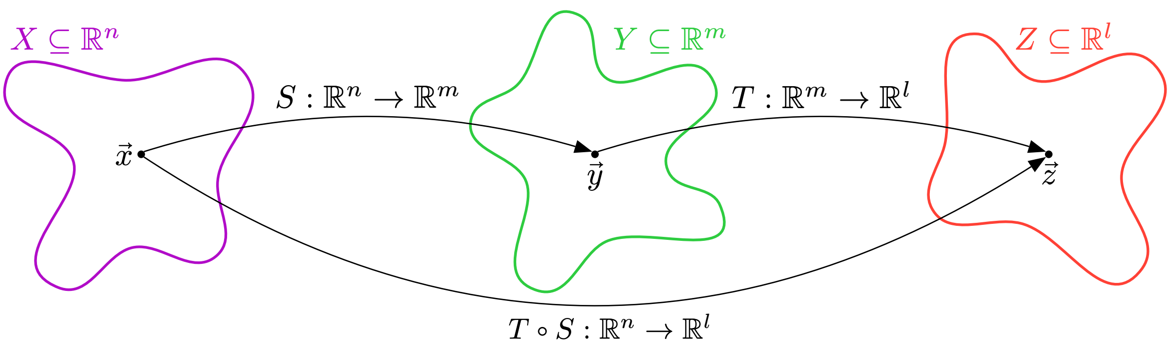
Because $\hat{u}\hat{u}^T$ projects onto a one-dimensional subspace spanned by $\hat{u}$

5.14. Composition of Linear Transformations

$$S : X \to Y \qquad T : Y \to Z$$

$$T \circ S : X \to Z$$

$$S(\vec{x}) = \underbrace{A}_{m\times n}\,\vec{x} \qquad T(\vec{x}) = \underbrace{B}_{l\times m}\,\vec{x}$$



$$
\begin{aligned}
T \circ S(\vec{x}) &= T(S(\vec{x})) \\
&= A(B(\vec{x})) \\
&= AB\vec{x}
\end{aligned}
$$

Consider two linear transformations T and S, where:

- T maps $\mathbb{R}^m \to \mathbb{R}^l$

- S maps $\mathbb{R}^n \to \mathbb{R}^m$

The composition $T \circ S$ is a nre transformation mapping $\mathbb{R}^n \to \mathbb{R}^l$ defined by:

$$T \circ S(\vec{x}) = T(S(\vec{x}))$$

Key Properties of $T \circ S$

1. Additivity

$$\begin{aligned} T \circ S(\vec{x} + \vec{y}) &= T(S(\vec{x} + \vec{y})) \\ &= T(S(\vec{x}) + S(\vec{y})) \\ &= T(S(\vec{x})) + T(S(\vec{y})) \\ &= T \circ S(\vec{x}) + T \circ S(\vec{y}) \end{aligned}$$

2. Homogeneity

$$\begin{aligned} T \circ S(c\vec{x}) &= T(S(c\vec{x})) \\ &= T(cS(\vec{x})) \\ &= cT(S(\vec{x})) \\ &= c(T \circ S)(\vec{x}) \end{aligned}$$

Matrix Representation of $T \circ S$

Let S be represented by the matrix A ($m \times n$), and let T be respesented by the matrix B ($l \times m$)

For a vector $\vec{x} \in \mathbb{R}^n$

$$\begin{aligned} T \circ S(\vec{x}) = T(S(\vec{x})) = T(A\vec{x}) \\ = \underbrace{B}_{l \times m} \left(\underbrace{A}_{m \times n} \vec{x} \right) \\ = \underbrace{C}_{l \times n} \vec{x} \end{aligned}$$

The composition $T \circ S$ is therefor represented by the matrix $C = A \cdot B$, where C is of size $l \times n$

Column-Wise Interpretation

The matrix A can be decomposed comlumn-wise:

$$A = \begin{bmatrix} \\ \vec{a}_1 & \vec{a}_2 & \ldots & \vec{a}_n \\ \\ \end{bmatrix}$$

where $\vec{a}_i$ is the i-th column of A and I_n is the identity matrix in $\mathbb{R}^n$, and its columns are the standard basis vectors $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n$.

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0}_{e_1} & \underbrace{0}_{e_2} & \dots & \underbrace{1}_{e_n} \end{bmatrix}$$

To compute C:

1. For each $\vec{e}_i$ in the basis of $\mathbb{R}^n$, $A\vec{e}_i = \vec{a}_i$, the i-th column of A

$$\begin{aligned} C &= \begin{bmatrix} B(Ae_1) & B(Ae_2) & \dots & B(Ae_n) \end{bmatrix} \\ &= \begin{bmatrix} B\left(A\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) & B\left(A\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}\right) & \dots & B\left(A\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}\right) \end{bmatrix} \\ &= \begin{bmatrix} B\vec{a}_1 & B\vec{a}_2 & \dots & B\vec{a}_n \end{bmatrix} \end{aligned}$$

The composition $T \circ S$ is the linear map respresented by $\boldsymbol{C = B \cdot A}$

Each column of C reflects how T transforms the action of S on a standard basis vector

5.15. Matrix Product

$$\underbrace{A}_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad \underbrace{B}_{n\times p} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{np} \end{bmatrix}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Example

$$\underbrace{A}_{2\times 3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \underbrace{B}_{3\times 2} = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

$$AB = \begin{bmatrix} A \begin{bmatrix} 7 \\ 9 \\ 11 \end{bmatrix} & A \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \\ 11 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix} \\ \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \\ 11 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix} \end{bmatrix}$$

5.16. Matrix Product Associativity

5.17. Eigen

$$Ax = \lambda x$$

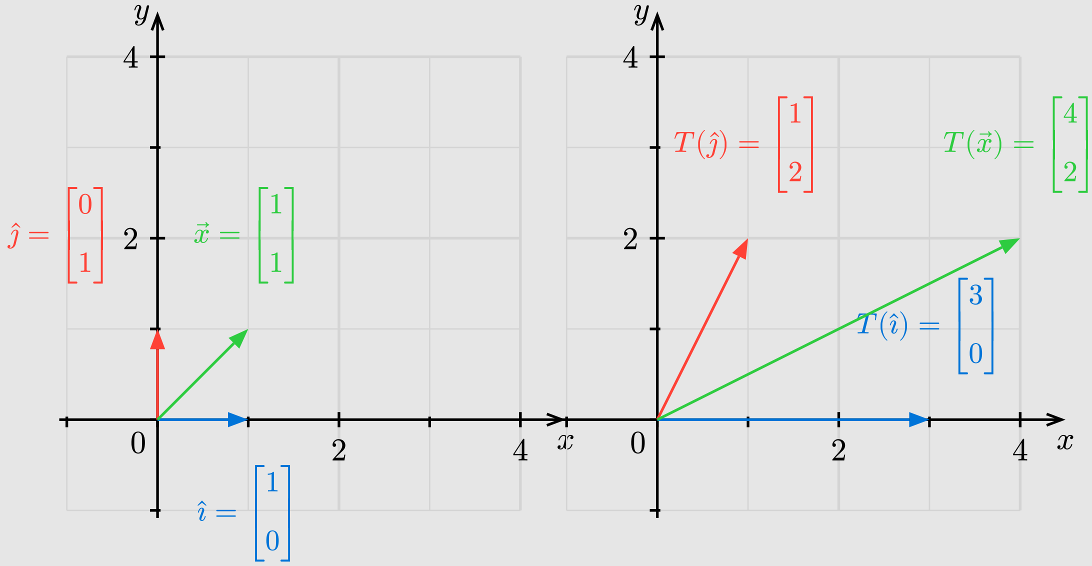
When a transformation A is applied to a vector $\mathbf{x}$, it is equivalent to scaling the vector by a factor of λ. The vector $\mathbf{x}$ is called an **eigenvector** of the transformation A, and the scalar λ is called the corresponding **eigenvalue**.

Example

Transformation

Cectors $\hat{\imath}, \hat{\jmath}, \vec{x}$ under transformation A

$$A = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$



$$\begin{aligned} T(\hat{\imath}) &= A\hat{\imath} \\ &= \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 3 \\ 0 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} T(\hat{\jmath}) &= A\hat{\jmath} \\ &= \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{aligned}$$

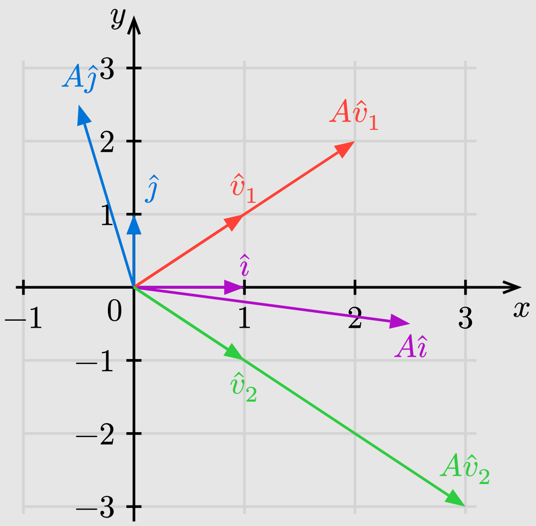
$$\begin{aligned} T(\vec{x}) &= A\vec{x} \\ &= \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 4 \\ 2 \end{bmatrix} \end{aligned}$$

Example

Transformation

Vectors $\hat{\imath}, \hat{\jmath}, \hat{v}_1, \hat{v}_2$ under transformation A

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$



An $n \times n$ matrix A has exactly n eigenvalues because the eigenvalues are the **roots** of its characteristic polynomial, which is degree n

Characteristic polynomial:

$$p(\lambda) = \det(A - \lambda I)$$

Example

2×2 Matrix

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

$$\det(A - \lambda I) = (2 - \lambda)(3 - \lambda) = 0$$

Solve for λ:

$$\lambda_1 = 2, \lambda_2 = 3$$

3×3 Matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\det(A - \lambda I) = (1 - \lambda)(2 - \lambda)(4 - \lambda) = 0$$

Solve for λ:

$$\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 4$$

5.17.1. LU Decomposition

Given a matrix A, LU decomposition aims to express A as:

$$A = LU$$

Where:

- L: Lower triangular matrix (all elements above the diagonal are zero)
- U: Upper triangular matrix (all elements below the diagonal are zero)

1. Solve $Ly = b$ Using Forward Substitution

$$Ly = b$$

Where:

- L: Lower triangular matrix (all elements above the diagonal are zero)
- y: Intermediate vector we are solving for
- b: Right-hand side vector

$$\begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

- First row: $L_{11}y_1 = b_1$, so $y_1 = \frac{b_1}{L_{11}}$
- Second row: $L_{21}y_1 + L_{22}y_2 = b_2$, substitute y_1 into this equation and solve for y_2:

$$y_2 = \frac{b_2 - L_{21}y_1}{L_{22}}$$

- Third row: $L_{31}y_1 + L_{32}y_2 + L_{33}y_2 = b_3$, substitute y_1 and y_2 into this equation, solve for y_3:

$$y_3 = \frac{b_2 - L_{31}y_1 - L_{32}y_2}{L_{33}}$$

2. Solve $Ux = y$ Using Backward Substitution

$$Ux = y$$

Where:

- U: Upper triangular matrix (all elements below the diagonal are zero)
- y: Vector of unknowns (solution)
- b: Vector computed from the forward substitution step

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

- Third row: $U_{33}x_3 = y_3$, so $x_3 = \frac{y_3}{U_{33}}$
- Second row: $U_{22}x_2 + U_{23}x_3 = y_2$, substitute x_3 from the previous step and solve for x_2:

$$x_2 = \frac{y_2 - U_{23}x_3}{U_{22}}$$

- First row: $U_{11}x_1 + U_{12}x_2 + U_{13}x_3 = y_1$, substitute x_2 and x_3 from the previous step and solve for x_1:

$$x_1 = \frac{y_1 - U_{12}x_2 - U_{13}x_3}{U_{11}}$$

Example

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 7 & 3 \\ 6 & 18 & 5 \end{bmatrix} \qquad b = \begin{bmatrix} 5 \\ 12 \\ 31 \end{bmatrix}$$

1. Factor A into L and U:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 6 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

2. Solve $Ax = b$
   - $y_1 = \frac{b_1}{L_{11}} = \frac{5}{1} = 5$
   - $y_2 = \frac{b_2 - L_{21}y_1}{L_{22}} = \frac{12 - 2\times 5}{1} = 2$
   - $y_2 = \frac{b_3 - L_{31}y_1 - L_{32}y_2}{L_{33}} = \frac{31 - 4\times 5 - 3\times 2}{1} = 5$

So,

$$y = \begin{bmatrix} 5 \\ 2 \\ 5 \end{bmatrix}$$

3. Solve $Ux = y$
   - $x_3 = \frac{y_3}{U_{33}} = \frac{5}{-}2 = -2.5$
   - $x_2 = \frac{y_2 - U_{23}x_3}{U_{22}} = \frac{2 - 1\times -2.5}{1} = 4.5$
   - $x_1 = \frac{y_1 - U_{12}x_2 - U_{13}x_3}{U_{11}} = \frac{5 - 3\times 4.5 - 1\times -2.5}{2} = -4$

So,

$$x = \begin{bmatrix} -4 \\ 4.5 \\ -2.5 \end{bmatrix}$$

5.18. Unimodularity

5.18.1. Unimodular Matrix

Integer square matrix whose determinant is +1 or −1:

$$\det(A) = \pm 1$$

- Always invertible, and its inverse is also an integer matrix
- Applying such a matrix (transformation) to an integer vector gives another integer vector

5.18.2. Totally Unimodular Matrix

A totally unimodular (TU) matrix is a (not necessarily square) matrix in which every square submatrix (determinant of any square submatrix of any size) has determinant in:

$$\{-1, 0, +1\}$$

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\right) = 0 \qquad \det\left(\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\right) = 0$$

5.19. Adjugate

$$\operatorname{adj}(A)$$

1. Compute **Minor** M_{ij}

For each entry A_{ij} of A, take the determinant of the submatrix that remains when row i and column j are removed

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\det\left(\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}\right) \qquad \det\left(\begin{bmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{bmatrix}\right) \qquad \det\left(\begin{bmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{bmatrix}\right)$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\det\left(\begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix}\right) \quad \det\left(\begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix}\right) \quad \det\left(\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix}\right)$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\det\left(\begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}\right) \quad \det\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{bmatrix}\right) \quad \det\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\right)$$

2. Compute **Cofactor** C_{ij}

Multiply the minor by a sign depending on position

$$C_{ij} = (-1)^{i+j} M_{ij}$$

3. **Adjugate** (Adjoint)

Take the transpose:

$$\mathrm{adj}(A) = C^T$$

Example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

Step 1. Compute minors and cofactors

- **Entry (1, 1)**: remove row 1 and column 1

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 4 & 5 \\ 0 & 6 \end{bmatrix}\right) = 24$$

Cofactor:

$$\begin{aligned} C_{11} &= (-1)^{1+1} \cdot 24 \\ &= 1 \cdot 24 \\ &= 24 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & & \\ & & \\ & & \end{bmatrix}$$

- **Entry (1, 2)**: remove row 1 and column 2

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 0 & 5 \\ 1 & 6 \end{bmatrix}\right) = -5$$

Cofactor:

$$\begin{aligned} C_{12} &= (-1)^{1+2} \cdot -5 \\ &= -1 \cdot -5 \\ &= 5 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & \\ & & \\ & & \end{bmatrix}$$

- **Entry (1, 3)**: remove row 1 and column 3

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 0 & 4 \\ 1 & 0 \end{bmatrix}\right) = -4$$

Cofactor:

$$\begin{aligned} C_{13} &= (-1)^{1+3} \cdot -4 \\ &= 1 \cdot -4 \\ &= -4 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ & & \\ & & \end{bmatrix}$$

- **Entry (2, 1)**: remove row 2 and column 1

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 2 & 3 \\ 0 & 6 \end{bmatrix}\right) = 12$$

Cofactor:

$$\begin{aligned} C_{21} &= (-1)^{2+1} \cdot 12 \\ &= -1 \cdot 12 \\ &= -12 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & & \\ & & \end{bmatrix}$$

- **Entry (2, 2)**: remove row 2 and column 2

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 1 & 3 \\ 1 & 6 \end{bmatrix}\right) = 3$$

Cofactor:

$$\begin{aligned} C_{22} &= (-1)^{2+2} \cdot 3 \\ &= 1 \cdot 3 \\ &= 3 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & \\ & & \end{bmatrix}$$

- **Entry (2, 3)**: remove row 2 and column 3

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}\right) = -2$$

Cofactor:

$$\begin{aligned} C_{22} &= (-1)^{2+3} \cdot -2 \\ &= -1 \cdot -2 \\ &= 2 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & 2 \\ & & \end{bmatrix}$$

- **Entry (3, 1)**: remove row 3 and column 1

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}\right) = -2$$

Cofactor:

$$\begin{aligned} C_{31} &= (-1)^{3+1} \cdot -2 \\ &= 1 \cdot -2 \\ &= -2 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & 2 \\ -2 & & \end{bmatrix}$$

- **Entry (3, 2)**: remove row 3 and column 2

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 1 & 3 \\ 0 & 5 \end{bmatrix}\right) = 5$$

Cofactor:

$$\begin{aligned} C_{32} &= (-1)^{3+2} \cdot -5 \\ &= -1 \cdot 5 \\ &= -5 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & 2 \\ -2 & -5 & \end{bmatrix}$$

- **Entry (3, 3)**: remove row 3 and column 3

Minor:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\det\left(\begin{bmatrix} 1 & 2 \\ 0 & 4 \end{bmatrix}\right) = 4$$

Cofactor:

$$\begin{aligned} C_{33} &= (-1)^{3+3} \cdot 4 \\ &= 1 \cdot 4 \\ &= 4 \end{aligned}$$

Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & 2 \\ -2 & -5 & 4 \end{bmatrix}$$

Step 2. Cofactor matrix

$$C = \begin{bmatrix} 24 & 5 & -4 \\ -12 & 3 & 2 \\ -2 & -5 & 4 \end{bmatrix}$$

Step 3. Transpose to get adjugate

$$\operatorname{adj}(A) = C^T = \begin{bmatrix} 24 & -12 & -2 \\ 5 & 3 & -5 \\ -4 & 2 & 4 \end{bmatrix}$$

Relation to **inverse**:

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A) \qquad \text{if} \det(A) \neq 0$$

Relation to **identity**:

$$A \cdot \operatorname{adj}(A) = \operatorname{adj}(A) \cdot A = \det(A) I_n$$

5.20. Minor

Determinant of some smaller square matrix obtained by deleting certain rows and columns from the original matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\begin{vmatrix} a_{12} & a_{14} \\ a_{32} & a_{24} \end{vmatrix}$$

5.20.1. Principal Minor

Determinant of some smaller square matrix obtained by deleting certain rows and columns from the original matrix

The diagonal of the principal minor is a subset of the diagonal of A

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{23} \end{vmatrix}$$

5.20.2. Levels

A's level-k principal minors is the determinant of a $k \times k$ submatrix whose diagonal is a subset of A's diagonal

For an $n \times n$

- There are n levels of minors
- The number of principle minors at level k is $\binom{n}{k}$

5.20.3. Leading Principal Minor

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

5.21. Solving Systems of Linear Equations

Linear Equation

$$y = a_1 x_1 + a_1 x_2 + \ldots + a_n x_n$$

1. Consistency

Whether a system of linear equations has at least one solution

Example

Consistent System

$$
\begin{aligned}
x + y &= 3 \\
x - y &= 1
\end{aligned}
$$

This system has a unique solution

$$(x, y) = (2, 1)$$

Inconsistent System

$$
\begin{aligned}
x + y &= 3 \\
x + y &= 5
\end{aligned}
$$

This system is inconsistent (equations contradict each other, no solution can satisfy both)

2. Independence

Whether the equations in the system provide unique and non-redundant information about the variables

Example

Independent Equations

$$
\begin{aligned}
x + y &= 3 \\
x - y &= 1
\end{aligned}
$$

Neither equation can be derived from the other (they provide unique information and intersect at a single point)

Dependent Equations

$$
\begin{aligned}
x + y &= 3 \\
2x + 2y &= 6
\end{aligned}
$$

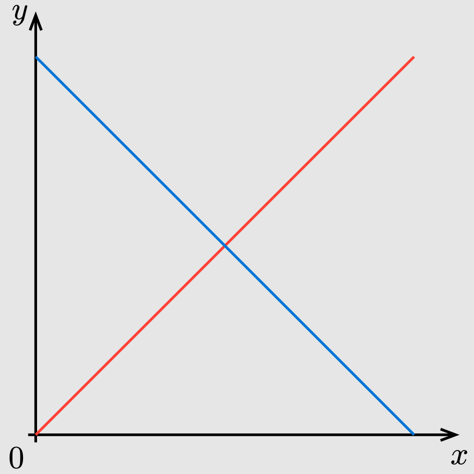
Second equation is just a multiple of the first equation (they describe the same line)

3. Recognizing Systems with No Solution or Infinite Solutions

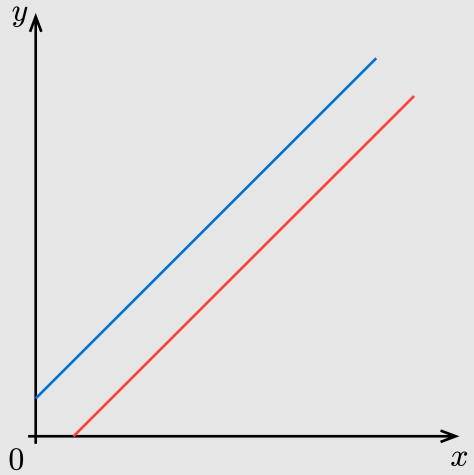
Example

$$
\begin{aligned}
3x + 2y &= 6 \qquad \text{(Equation 1)} \\
6x + 4y &= 12 \qquad \text{(Equation 2)}
\end{aligned}
$$

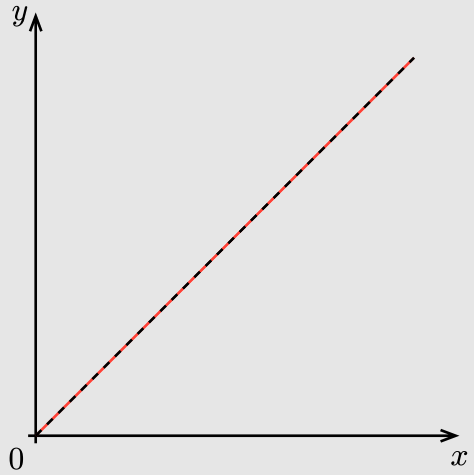
Unique Solution (Consistent and Independent):



No Solution (Inconsistent):



Infinitely Many Solutions (Consistent and Dependent):



2. Matrix Respresentation

System of Equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m
\end{aligned}
$$

Matrix Respresentation

Coefficient vector (A)

$$
A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}
$$

Variable vector (x)

$$
x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
$$

Constant vector (b)

$$
b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
$$

Matrix equation

$$
Ax = b
$$

linear_eq.py

```
from scipy.linalg import solve

X = np.array([
  [1, 1, 1],
  [2, -1, 3],
  [3, 4, -1]
])

Y = np.array([6, 14, 1])

intersection_point = solve(X, Y)
```

5.21.1. Gaussian Elimination

Convert a matrix into its row echelon form (REF) or reduced row echelon form (RREF)

$$
\begin{gathered}
a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m
\end{gathered}
$$

1. Create an augmented matrix

$$
A = \left[\begin{array}{cccc|c}
a_{11} & a_{12} & ... & a_{1n} & b_1 \\
a_{21} & a_{22} & ... & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & b_3 \\
a_{2m1} & a_{m2} & ... & a_{mn} & b_m
\end{array}\right]
$$

2. Forward Elimination

Eliminate the element in the i-th of the k-th column ($k > i$)

$$R_k \leftarrow R_k - \frac{a_{ki}}{a_{ii}} R_i$$

Where

- a_{ii}: Pivot element
- R_k: k-th row
- R_i: i-th row

3. Back Substitution
4. Reduced Row Echelon Form (RREF)

Example

$$
\begin{gathered}
2x_1 + 3x_2 = 5 \\
4x_1 + 5x_2 = 5
\end{gathered}
$$

1. Create an augmented matrix

$$
\left[\begin{array}{cc|c}
2 & 3 & 5 \\
4 & 5 & 6
\end{array}\right]
$$

2. Forward Elimination

$R_k \leftarrow R_k - \frac{a_{ki}}{a_{ii}} R_i$

$R_k \leftarrow R_k - \frac{a_{21}}{a_{11}} R_i$

$R_2 \leftarrow R_2 - \frac{4}{2} R_1$

$R_2 \leftarrow R_2 - 2 \times R_1$

$$\left[\begin{array}{cc|c} 2 & 3 & 5 \\ 4-2\times 2 & 5-2\times 3 & 6-2\times 5 \end{array}\right]$$

Simplifies to:

$$\left[\begin{array}{cc|c} 2 & 3 & 5 \\ 0 & -1 & -4 \end{array}\right]$$

System is now:

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ -1x_2 &= -4 \end{aligned}$$

3. Back Substitution

$$\begin{aligned} -1x_2 &= -4 \\ x_2 &= 4 \end{aligned}$$

Substitute:

$$\begin{aligned} 2x_1 + 3(4) &= 5 \\ 2x_1 + 12 &= 5 \\ x_1 &= -3.5 \end{aligned}$$

Solution:

$$\begin{aligned} x_1 &= -3.5 \\ x_2 &= 4 \end{aligned}$$

5.21.2. Substitution

Example

$$\begin{aligned} x + y &= 10 \qquad \text{(Equation 1)} \\ 2x - y &= 5 \qquad \text{(Equation 2)} \end{aligned}$$

1. Solve Equation 1 for y

$$y = 10 - x$$

2. Substitute into Equation 2

$$2x - (10 - x) = 5$$

3. Solve for x:

$$
\begin{gathered}
2x - 10 + x = 5 \\
3x - 10 = 5 \\
3x = 15 \\
x = 5
\end{gathered}
$$

4. Find y using value of x

$$
\begin{gathered}
y = 10 - x \\
y = 10 - 5 \\
y = 5
\end{gathered}
$$

5.21.3. Addition or Subtraction Method

Example

$$
\begin{aligned}
3x + 2y &= 12 \qquad \text{(Equation 1)} \\
2x - 2y &= 4 \qquad \text{(Equation 2)}
\end{aligned}
$$

1. Add the equations

$$
\begin{gathered}
(3x + 2y) + (2x - 2y) = 12 + 4 \\
5x = 16 \\
x = \frac{16}{5} \\
x = 3.2
\end{gathered}
$$

2. Substitute

$$
\begin{gathered}
3(3.2) + 2y = 12 \\
9.6 + 2y = 12 \\
2y = 12 - 9.6 y = \frac{2.4}{2} \\
y = 1.2
\end{gathered}
$$

6. Calculus I

7. Cheatsheet

7.1. Limits

Epsilon-Delta

For every distance ε around L, there's a δ-range around a that keeps $f(x)$ within ε of L.

$$\lim_{x \to x_0} f(x) = L \iff \forall \varepsilon > 0, \exists \delta > 0 \text{ s.t. } 0 < |x - x_0| < \delta \Rightarrow |f(x) - L| < \varepsilon$$

The limit of $f(x)$ as x approaches x_0 equals L if and only if, for every $\varepsilon > 0$, there exists a $\delta > 0$ such that, whenever $0 < |x - x_0| < \delta$, implies that $|f(x) - L| < \varepsilon$

Limit Type	**Name**	**Quantifiers**
$x \to x_0, f(x) \to L$	Epsilon-Delta	$\forall \varepsilon, \exists \delta$
$x \to x_0, f(x) \to \infty$	M-Delta	$\forall M, \exists \delta$
$x \to \infty, f(x) \to L$	epsilon-N	$\forall \varepsilon, \exists N$
$x \to \infty, f(x) \to \infty$	M-N	$\forall M, \exists N$

Finite $\to$ Finite (**Epsilon-Delta**)

$$\lim_{x \to x_0} f(x) = L \iff \forall \varepsilon > 0, \exists \delta > 0 \text{ s.t. } 0 < |x - x_0| < \delta \Rightarrow |f(x) - L| < \varepsilon$$

Finite $\to$ Infinity (**M-Delta**)

$+\infty$

$$\lim_{x \to x_0} f(x) = +\infty \iff \forall M > 0, \exists \delta > 0 \text{ s.t. } 0 < |x - x_0| < \delta \Rightarrow f(x) > M$$

$-\infty$

$$\lim_{x \to x_0} f(x) = -\infty \iff \forall M > 0, \exists \delta > 0 \text{ s.t. } 0 < |x - x_0| < \delta \Rightarrow f(x) < -M$$

Infinity $\to$ Finite (**Epsilon-N**)

$+\infty$

$$\lim_{x \to +\infty} f(x) = L \iff \forall \varepsilon > 0, \exists N > 0 \text{ s.t. } x > N \Rightarrow |f(x) - L| < \varepsilon$$

$-\infty$

$$\lim_{x \to +\infty} f(x) = L \iff \forall \varepsilon > 0, \exists N > 0 \text{ s.t. } x < -N \Rightarrow |f(x) - L| < \varepsilon$$

Infinity $\to$ Infinity (**M-N**)

$$\lim_{x \to \pm\infty} f(x) = \pm\infty \iff \forall M > 0, \exists N > 0 \text{ s.t. } x < N \Rightarrow f(x) > M$$

7.2. Derivatives

$$\frac{df}{dx}(x) = f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Example

Let $f(x) = x^2$

a. Find $f'(x)$

$$f'(x) = \boxed{\text{2x}}$$

b. Prove a

$$\begin{aligned}
f'(x) &= \lim_{h \to 0} \frac{f(x+h) - f(x)}{(\cancel{x} + h) - \cancel{x}} \\
&= \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \\
&= \lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} \\
&= \lim_{h \to 0} \frac{\cancel{x^2} + 2xh + h^2 - \cancel{x^2}}{h} \\
&= \lim_{h \to 0} \frac{\cancel{h}(2x+h)}{\cancel{h}} \\
&= \lim_{h \to 0} (2x + h) \\
&= \lim_{h \to 0} (2x + 0) \\
&= \boxed{\text{2x}}
\end{aligned}$$

c. Prove b

$$\lim_{x \to x_0} f(x) = L \iff \forall \varepsilon > 0, \exists \delta > 0 \text{ s.t. } 0 < |x - x_0| < \delta \Rightarrow |f(x) - L| < \varepsilon$$

p.f.:

Let $\varepsilon > 0$

Choose $\delta = \varepsilon$

Suppose $0 < |h - 0| < \delta$

Check

$$\left| \frac{(x+h)^2 - x^2}{h} - 2x \right|$$

$$
\begin{aligned}
&= \left| \frac{\cancel{x^2} + 2xh + h^2 - \cancel{x^2}}{h} - 2x \right| \\
&= \left| \frac{\cancel{h}(2x + h)}{\cancel{h}} - 2x \right| \\
&= |\cancel{2x} + h - \cancel{2x}| \\
&= \boxed{|h| < \delta = \varepsilon}
\end{aligned}
$$

7.3. Integrals

7.4. Differential

If you have a function:

$$y = f(x)$$

Then the differential $\mathrm{d}u$ is defined as:

$$\mathrm{d}y = f'(x) \cdot \mathrm{d}x$$

This means:

- $f'(x) = \frac{\mathrm{d}y}{\mathrm{d}x}$ is the derivative
- $\mathrm{d}x$ is a small change in x
- $\mathrm{d}y$ is the corresponding small change in y

So:

$$\mathrm{d}y = \frac{\mathrm{d}y}{\mathrm{d}x} \cdot \mathrm{d}x$$

Example

Let’s say“

$$y = x^2$$

Then:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = 2x$$

$$\frac{d}{dx}\left(\int f(x)\,dx\right) = f(x)$$

$$\int\left(\frac{d}{dx}f(x)\right)dx = f(x)$$

Operation	Notation	Input	Output	Meaning
Derivative	$\frac{df}{dx}$	Function $f(x)$	Function $f'(x)$	Slope/Rate of change at each x
Indefinite Integral	$\int f(x)dx$	Function $f(x)$	Family of functions $F(x)+C$	Function whose slope is $f(x)$
Definite Integral	$\int_a^b f(x)dx$	Function $f(x)$ Bounds $[a,b]$	Number	Total signed area between a and b

Rule	$\frac{d}{dx}$ Rule	$\frac{d}{dx}$ Example	$\int$ Rule	$\int$ Example
Constant	$\frac{d}{dx}[c]=0$	$\frac{d}{dx}[7]=0$	$\int c\,dx = cx + C$	$\int 7\,dx = 7x + C$
Power	$\frac{d}{dx}[x^n] = nx^{n-1}$	$\frac{d}{dx}[x^4] = 4x^3$	$\int x^n\,dx = \frac{x^{n+1}}{n+1} + C$ $(n \neq -1)$	$\int x^3\,dx = \frac{x^4}{4} + C$
Constant Multiple	$\frac{d}{dx}[cf] = cf'$	$\frac{d}{dx}[3x^2] = 3\cdot 2x$ $= 6x$	$\int cf\,dx = c\int f\,dx$	$\int 3x^2\,dx = 3\cdot\frac{x^3}{3}$ $= x^3 + C$
Sum	$\frac{d}{dx}[f+g] = f' + g'$	$\frac{d}{dx}(x^2+x)$ $= \frac{d}{dx}(x^2) + \frac{d}{dx}(x)$ $= 2x+1$	$\int (f+g)\,dx$ $= \int f\,dx + \int g\,dx$	$\int (x^2+x)\,dx$ $= \int x^2\,dx + \int x\,dx$ $= \frac{x^3}{3} + \frac{3^2}{2} + C$
Difference	$\frac{d}{dx}[f-g] = f' - g'$	$\frac{d}{dx}(x^2-x)$ $= \frac{d}{dx}(x^2) - \frac{d}{dx}(x)$ $= 2x-1$	$\int (f-g)\,dx$ $= \int f\,dx - \int g\,dx$	$\int (x^2-x)\,dx$ $= \int x^2\,dx - \int x\,dx$ $= \frac{x^3}{3} - \frac{x^2}{2} + C$
Product	$\frac{d}{dx}[fg] = f'g + fg'$	$\frac{d}{dx}[x\sin x]$ $= 1\cdot\sin x + x\cdot\cos x$	Integration by Parts	

Rule	$\frac{d}{dx}$ Rule	$\frac{d}{dx}$ Example	$\int$ Rule	$\int$ Example
			$\int u\,dv = uv - \int v\,du$	
Quotient	$\frac{d}{dx}\left[\frac{f}{g}\right] = \frac{f'g - fg'}{g^2}$	$\frac{d}{dx}\left[\frac{x^2}{x+1}\right] = \frac{2x(x+1) - x^2(1)}{(x+1)^2} = \frac{x^2+2x}{(x+1)^2}$	Algebraic Manipulation / Substitution	
Chain	$\frac{d}{dx}[f(g(x))] = f'(g(x)) \cdot g'(x)$	$\frac{d}{dx}[\sin(x^2)] = \cos(x^2) \cdot 2x$	Integration by Subsitution $\int f(g(x))g'(x)\,dx = \int f(u)\,du$	
Exponential	$\frac{d}{dx}[e^x] = e^x$	$\frac{d}{dx}[e^x] = e^x$	$\int e^x\,dx = e^x + C$	$\int e^x\,dx = e^x + C$
	$\frac{d}{dx}[a^x] = a^x \ln(a)$	$\frac{d}{dx}[2^x] = 2^x \ln 2$	$\int a^x dx = \frac{a^x}{\ln a} + C$	$\int 2^x\,dx = \frac{2^x}{\ln 2} + C$
Logarithmic	$\frac{d}{dx}[\ln(x)] = \frac{1}{x}$	$\frac{d}{dx}[\ln(x)] = \frac{1}{x}$	$\int \ln(x)\,dx = x\ln x - x + C$	$\int \ln(x)\,dx = x\ln x - x + C$
	$\frac{d}{dx}[\log_a(x)] = \frac{1}{x\ln(a)}$	$\frac{d}{dx}\log_2(x) = \frac{1}{x\ln(2)}$	$\int \log_a(x)\,dx = \frac{x\ln x}{\ln a} - \frac{x}{\ln a} + C$	$\int \log_{10}(x)\,dx = \frac{x\ln x}{\ln 10} - \frac{x}{\ln 10} + C$
Sin	$\frac{d}{dx}[\sin(x)] = \cos(x)$	$\frac{d}{dx}[\sin(x)] = \cos(x)$	$\int \sin(x)\,dx = -\cos x + C$	$\int \sin(x)\,dx = -\cos x + C$
Cos	$\frac{d}{dx}[\cos(x)] = -\sin(x)$	$\frac{d}{dx}[\cos(x)] = -\sin(x)$	$\int \cos(x)\,dx = \sin x + C$	$\int \cos(x)\,dx = \sin x + C$
Tan	$\frac{d}{dx}[\tan(x)] = \sec^2(x)$	$\frac{d}{dx}[\tan(x)] = \sec^2(x)$	$\int \tan(x)\,dx = -\ln\lvert\cos x\rvert + C$	$\int \tan(x)\,dx = -\ln\lvert\cos x\rvert + C$

7.5. Product Rule $\longrightarrow$ Integration by Parts

$$\int u \, \mathrm{d}v = uv - \int v \, \mathrm{d}u$$

Example

Given a function:

$$f(x) = x \cdot e^x$$

Integrate (by parts):

$$\int x \cdot e^x \, \mathrm{d}x$$

Step 1: Choose u and $\mathrm{d}v$

We choose:

- $u = x$ (easy to **differentiate**)
- $\mathrm{d}v = e^x$ (easy to **integrate**)

Step 2: Compute $\mathrm{d}u$ and v

- $u = x \longrightarrow \mathrm{d}u = \mathrm{d}x$
- $\mathrm{d}v = e^x \, \mathrm{d}x \longrightarrow v = \int e^x \, \mathrm{d}x = e^x$

Step 3: Plug into formula

$$\begin{aligned} \int x \cdot e^x \, \mathrm{d}x &= u \cdot v - \int v \cdot \mathrm{d}u \\ &= x \cdot e^x \, \mathrm{d}x - \int e^x \cdot \mathrm{d}x \end{aligned}$$

Step 4: Compute the remaining integral:

$$\int e^x \, \mathrm{d}x = e^x$$

Step 5: Finish the expression:

$$\begin{aligned} \int x \cdot e^x &= x \cdot e^x - e^x + C \\ &= e^x(x-1) + C \end{aligned}$$

7.6. Chain Rule $\longrightarrow$ u-Substitution

$$\int f(g(x)) g'(x) \, \mathrm{d}x = \int f(u) \, \mathrm{d}u$$

7.7. Quotient Rule $\longrightarrow$ Algebraic Manipulation / Substitution

8. Limits & Continuity

9. Properties of Limits

9.1. Continuous

9.1.1. Addition, Substraction, Multiplication, Division

$$\lim_{x \to c}(f(x) * g(x)) = \lim_{x \to c} f(x) * \lim_{x \to c} g(x)$$
$$* \in \{+, -, \times, \div\}$$

9.1.2. Constant

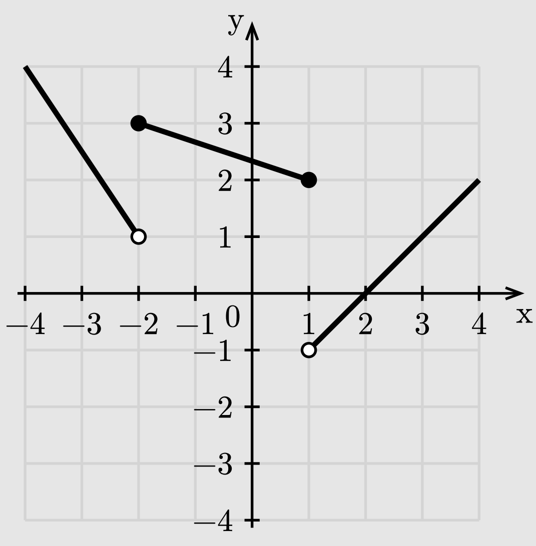
$$\lim_{x \to c} kf(x) = k \lim_{x \to c} f(x)$$

9.2. Non-continuous

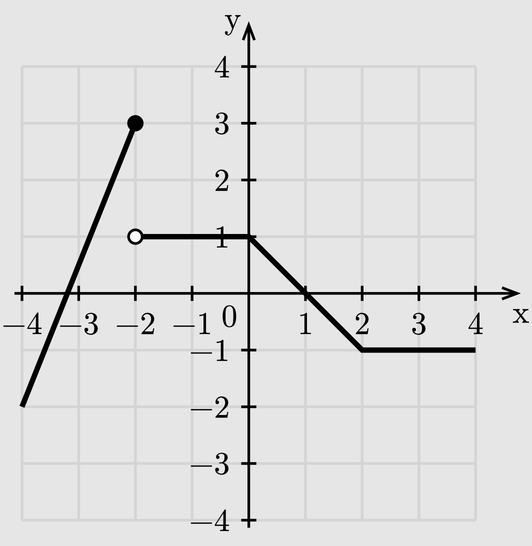
Even though the limit for either function may not exist, their $*$ can exist as long as

$$\lim_{x \to c^-}(f(x) * g(x)) \quad = \quad \lim_{x \to c^+}(f(x) * g(x))$$
$$* \in \{+, -, \times, \div\}$$

Example



$f(x)$



$g(x)$

Problem 1: Limit Exists

$$\lim_{x \to -2}(f(x) + g(x))$$

1. Left-hand limit ($x \to -2^-$)

$$\lim_{x \to -2^-} f(x) = 1 \qquad \lim_{x \to -2^-} g(x) = 3$$

Adding these

$$\begin{aligned}\lim_{x \to -2^-} (f(x) + g(x)) &= \lim_{x \to -2^-} f(x) + \lim_{x \to -2^-} g(x) \\ &= 1 + 3 \\ &= 4\end{aligned}$$

2. Right-hand limit ($x \to -2^+$)

$$\lim_{x \to -2^+} f(x) = 3 \qquad \lim_{x \to -2^+} g(x) = 1$$

Adding these

$$\begin{aligned}\lim_{x \to -2^+} (f(x) + g(x)) &= \lim_{x \to -2^+} f(x) + \lim_{x \to -2^+} g(x) \\ &= 3 + 1 \\ &= 4\end{aligned}$$

3. Since both the left-hand and right-hand limits agree

$$\lim_{x \to -2} (f(x) + g(x)) = 4$$

Problem 2: Limit Does Not Exist

$$\lim_{x \to 1} (f(x) + g(x))$$

1. Left-hand limit ($x \to 1^-$)

$$\lim_{x \to 1^-} f(x) = 2 \qquad \lim_{x \to 1^-} g(x) = 0$$

Adding these

$$\begin{aligned}\lim_{x \to 1^-} (f(x) + g(x)) &= \lim_{x \to 1^-} f(x) + \lim_{x \to 1^-} g(x) \\ &= 2 + 0 \\ &= 2\end{aligned}$$

2. Right-hand limit ($x \to 1^+$)

$$\lim_{x \to 1^+} f(x) = -1 \qquad \lim_{x \to 1^+} g(x) = 0$$

Adding these

$$\begin{aligned}\lim_{x \to 1^+} (f(x) + g(x)) &= \lim_{x \to 1^+} f(x) + \lim_{x \to 1^+} g(x) \\ &= -1 + 0 \\ &= -1\end{aligned}$$

3. Since both the left-hand and right-hand limits do not agree, the limit $\lim_{x \to c} (f(x) + g(x))$ does not exist

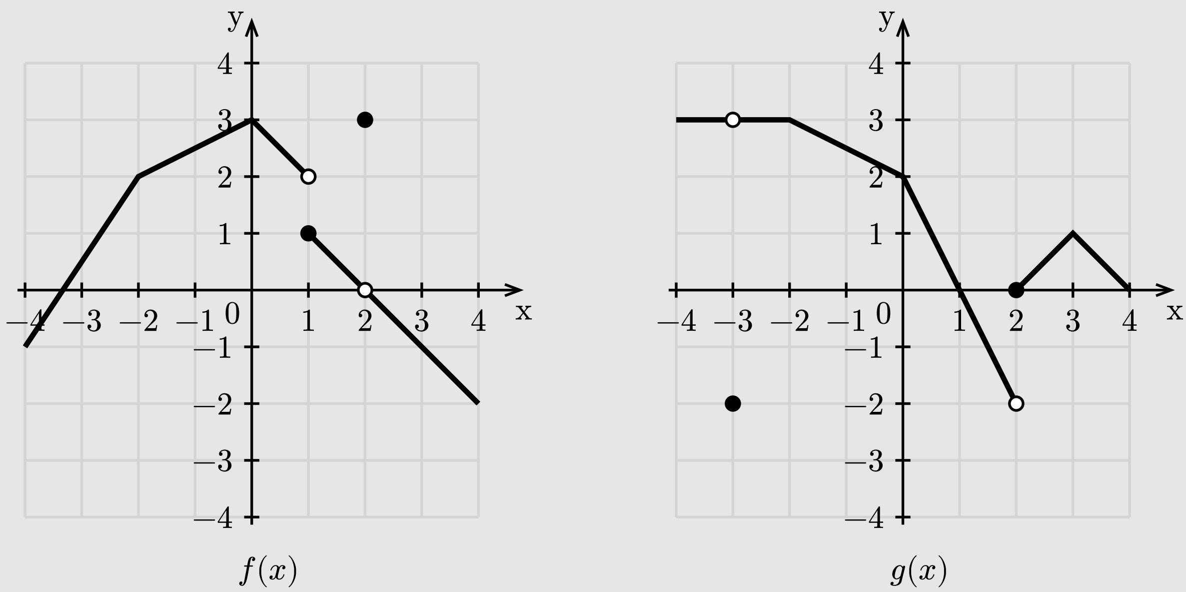
9.3. Composite Functions

$$\lim_{x \to c} f(g(x)) = f\left(\lim_{x \to c} g(x)\right)$$

For this to hold true, two important conditions must be satisfied:

- **Inner limit exists**: The limit $\lim_{x \to c}$ must exist and equal some value L. That is, as x gets arbitrarily close to c, $g(x)$ approaches a well-defined number L
- **Continuity of the outer function**: The function f must be continuous at the point L. Continuity ensures that f behaves predictably near L, without any jumps, gaps, or undefined points

Example



Problem 1: Inner Limit & Continuity Exist

$$\lim_{x \to -3} f(g(x))$$

1. Inner limit $\lim_{x \to -3} g(x)$

Observing $g(x)$, as $x \to -3$, $g(x) \to 3$. The inner limit $L = 3$ exists

$$\lim_{x \to -3} g(x) = 3$$

2. Continuity of $f(x)$ at $x = 3$

Observing $f(x)$, $f(3) = -1$. Since $f(x)$ is continuous at $x = 3$, the composite limit holds

$$\begin{aligned} f\left(\lim_{x \to -3} g(x)\right) &= f(3) \\ &= -1 \end{aligned}$$

Problem 2: Inner Limit Does Not Exist

$$\lim_{x \to 2} f(g(x))$$

1. Inner limit $\lim_{x \to 2} g(x)$

Observing $g(x)$, as $x \to 2$, $g(x) \to 2$. The inner limit does not exist

Problem 3: Continuity Does Not Exist

$$\lim_{x \to 0.5} f(g(x))$$

1. Inner limit $\lim_{x \to 0.5} g(x)$

Observing $g(x)$, as $x \to 0.5$, $g(x) \to 1$. The inner limit $L = 1$ exists

$$\lim_{x \to 0.5} g(x) = 1$$

2. Continuity of $f(x)$ at $x = 1$

Observing $f(x)$, $f(1)$ is not continuous. Since $f(x)$ is not continuous at $x = 1$, the composite limit does not hold

9.4. Limits by Direct Substitution

Example

Limit exists

$$\begin{aligned} \lim_{x \to -1} (6x^2 + 5x - 1) &= 6(-1)^2 + 5(-1) - 1 \\ &= 6 - 5 - 1 \\ &= 0 \end{aligned}$$

Limit does not exist (Undefined)

$$\begin{aligned} \lim_{x \to 1} \frac{x}{\ln(x)} &= \frac{1}{\ln(1)} \\ &= \frac{1}{0} \end{aligned}$$

9.4.1. Limits of Piecewise Functions

9.4.2. Absolute Value

9.5. Limits by Factoring

9.6. Limits by Rationalizing

9.7. Continuity & Differentiability at a Point

Example

Piecewise function:

$$f(x) = \begin{cases} x^2 & \text{if } x < 3 \\ 6x - 9 & \text{if } x \geq 3 \end{cases}$$

1. Check for **Continuity**
   - Value of $f(3)$

$$f(3) = 6(3) - 9 = \boxed{9}$$

   - Left-Hand Limit (LHL)

$$\lim_{x \to 3^-} f(x) = 3^2 = \boxed{9}$$

   - Right-Hand Limit (RHL)

$$\lim_{x \to 3^+} f(x) = 6(3) - 9 = \boxed{9}$$

Since $f(3) = \lim_{x \to 3^-} f(x) = \lim_{x \to 3^+} f(x)$, $f(x)$ is **continuous** at $x = 3$

2. Check for **Differentiability**
   - Left-Hand Derivative (LHD)

$$\begin{aligned} \lim_{x \to 3^-} \frac{f(x) - f(3)}{x - 3} &= \frac{x^2 - 3^2}{x - 3} \\ &= \frac{x^2 - 9}{x - 3} \\ &= \frac{(x+3)(x-3)}{x - 3} \\ &= x + 3 \\ &= \boxed{6} \end{aligned}$$

   - Right-Hand Derivative

$$\begin{aligned} \lim_{x \to 3^+} \frac{f(x) - f(3)}{x - 3} &= \frac{(6x - 9) - 3^2}{x - 3} \\ &= \frac{6x - 9 - 9}{x - 3} \\ &= \frac{6x - 18}{x - 3} \\ &= \frac{6(x - 3)}{x - 3} \\ &= \boxed{6} \end{aligned}$$

Since the left-hand and right-hand derivatives are equal, $f(x)$ is **differentiable** at $x = 3$

Conclusion: $f(x)$ is both continuous & differentiable at $x = 3$

Example

9.8. Power Rule

$$f(x) = x^n, \qquad n \neq 0$$
$$f'(x) = nx^{n-1}$$

Example

$$f(x) = x^3$$
$$f'(x) = 3x^2$$

Example

$$\begin{aligned}\frac{d}{dx}\left(\sqrt[3]{x^2}\right) &= \frac{d}{dx}\left((x^2)^{\frac{1}{3}}\right) \\ &= \frac{d}{dx}\left(x^{2\times\frac{1}{3}}\right) \\ &= \frac{d}{dx}\left(x^{\frac{2}{3}}\right) \\ &= \frac{d}{dx}\left(\frac{2}{3}x^{-\frac{1}{3}}\right)\end{aligned}$$

9.9. Constant Rule

$$\frac{d}{dx}[k] = 0$$

Example

$$\frac{d}{dx}[-3] = 0$$

9.10. Constant Multiple Rule

$$\begin{aligned}\frac{d}{dx}[kf(x)] &= k\frac{d}{dx}[f(x)] \\ &= kf'(x)\end{aligned}$$

Example

$$\begin{aligned}\frac{d}{dx}[2x^5] &= 2\frac{d}{dx}[x^5] \\ &= 2 \cdot 5x^4 \\ &= 10x^4\end{aligned}$$

9.11. Sum Rule

$$\begin{aligned}\frac{d}{dx}[f(x) + g(x)] &= \frac{d}{dx}[f(x)] + \frac{d}{dx}[g(x)] \\ &= f'(x) + g'(x)\end{aligned}$$

Example

$$\begin{aligned}\frac{d}{dx}[x^3 + x^{-4}] &= \frac{d}{dx}[x^3] + \frac{d}{dx}[x^{-4}] \\ &= 3x^2 + (-4x^{-5}) \\ &= 3x^2 - 4x^{-5}\end{aligned}$$

9.12. Difference Rule

$$\begin{aligned}\frac{d}{dx}[f(x) - g(x)] &= \frac{d}{dx}[f(x)] - \frac{d}{dx}[g(x)] \\ &= f'(x) - g'(x)\end{aligned}$$

Example

$$\begin{aligned}\frac{d}{dx}[x^4 - x^3] &= \frac{d}{dx}[x^4] - \frac{d}{dx}[x^3] \\ &= 4x^3 - 3x^2\end{aligned}$$

9.13. Square Root

Example

$$
\begin{aligned}
\frac{d}{dx}\sqrt[4]{x} &= \frac{d}{dx}x^{\frac{1}{4}} \\
&= \frac{1}{4}x^{\frac{1}{4}-1} \\
&= \frac{1}{4}\cdot x^{-\frac{3}{4}} \\
&= \frac{1}{4}\cdot\frac{1}{x^{3/4}} \\
&= \frac{1}{4x^{3/4}}
\end{aligned}
$$

9.14. Derivative of a Polynomial

Example

$$
\begin{aligned}
f(x) &= 2x^3 - 7x^2 + 3x - 100 \\
f'(x) &= 2\cdot 3x^2 - 7\cdot 2x + 3 + 0
\end{aligned}
$$

Example

$$h(x) = 3f(x) + 2g(x)$$

Evaluate $\frac{d}{dx}h(x)$ at $x = 9$

$$
\begin{aligned}
\frac{d}{dx}(h(x)) &= \frac{d}{dx}(3f(x) + 2g(x)) \\
&= \frac{d}{dx}3f(x) + \frac{d}{dx}2g(x) \\
&= 3\frac{d}{dx}f(x) + 2\frac{d}{dx}g(x)
\end{aligned}
$$

Evaluate $h'(9)$

$$h'(9) = 3f'(9) + 2g'(9)$$

Example

$$
\begin{aligned}
g(x) &= \frac{2}{x^3} - \frac{1}{x^2} \\
\frac{d}{dx}(g(x)) &= \frac{d}{dx}(2x^{-3} - 1x^{-2}) \\
g'(x) &= 2\cdot(-3)x^{-4} - (-2)x^{-3} \\
&= -6x^{-4} + 2x^{-3}
\end{aligned}
$$

$$
\begin{aligned}
g'(2) &= -6 \cdot 2^{-4} + 2 \cdot 2^{-3} \\
&= -\frac{6}{2^4} + \frac{2}{2^3} \\
&= -\frac{3}{8} + \frac{2}{8} \\
&= -\frac{1}{8}
\end{aligned}
$$

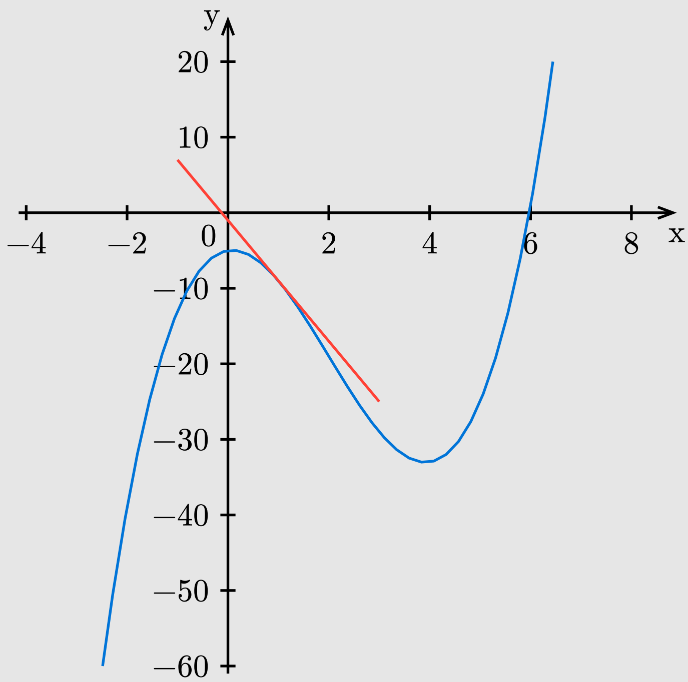
Example

–9 –8

$$
\begin{gathered}
f(x) = x^3 - 6x^2 + x - 5 \\
y = mx + b \\
f'(1) = -8 \; y = -8 \; x + b \\
-9 = -8 \cdot 1 + b \\
-9 = -8 + b \\
-9 + 8 = -8 + 8 + b \\
b = -1
\end{gathered}
$$

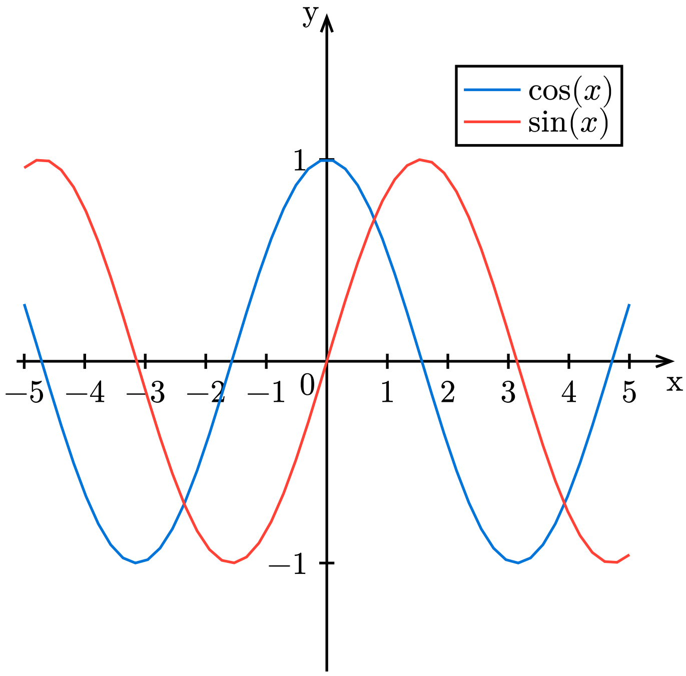
Tangent line to $f(x)$ at $x = 1$

$$
y = -8x - 1
$$



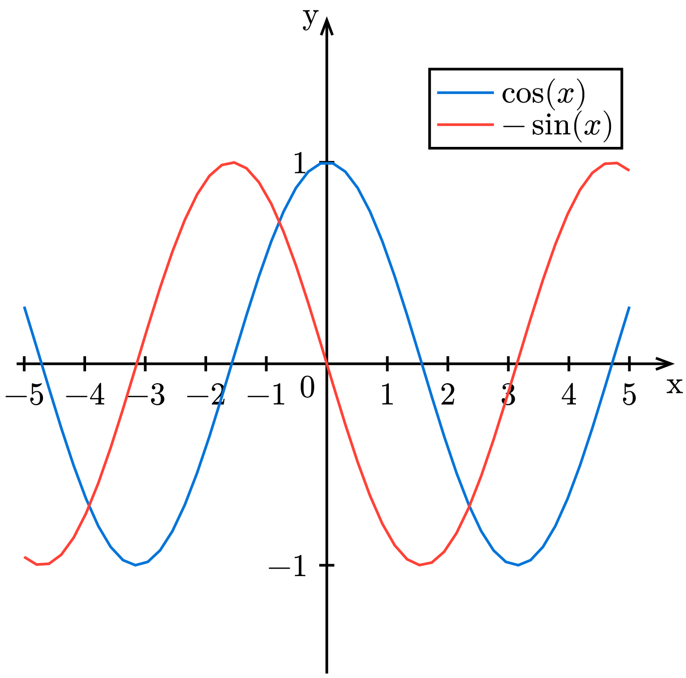
9.15. Sin

$$
\frac{d}{dx} \sin(x) = \cos(x)
$$



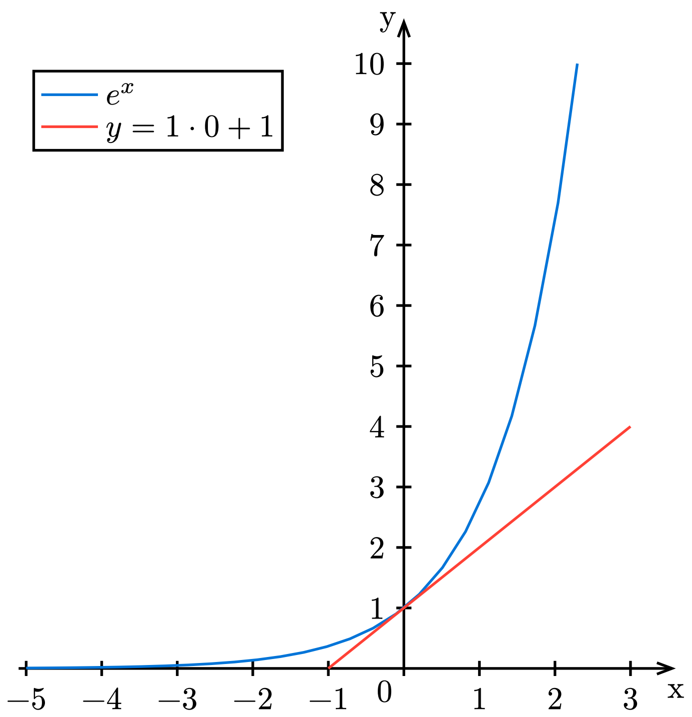
9.16. Cos

$$\frac{d}{dx}\cos(x) = -\sin(x)$$



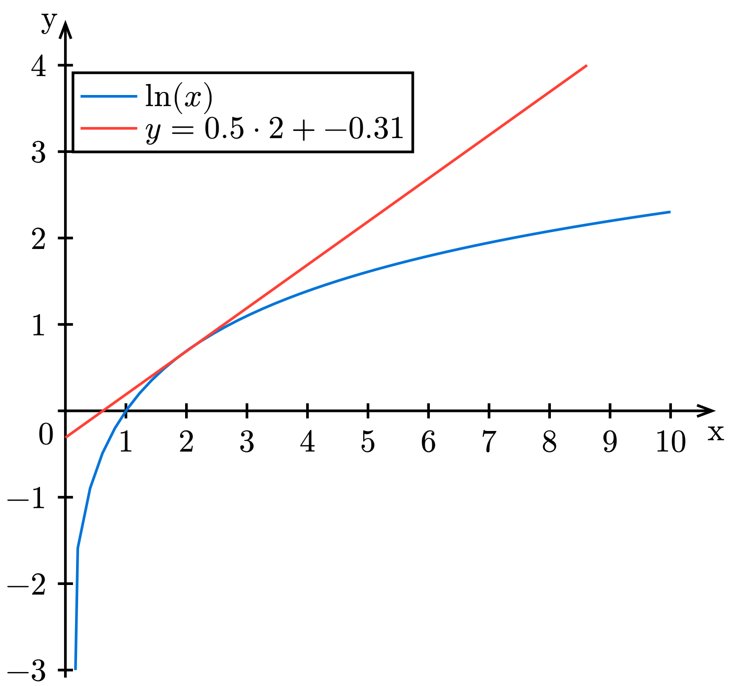
9.17. e^x

$$\frac{d}{dx}e^x = e^x$$



9.18. $\ln(x)$

$$\ln(x) = \frac{1}{x}$$



9.19. Product Rule

$$\frac{d}{dx}[f(x)g(x)] = f'(x)g(x) + f(x)g'(x)$$

Example

$$\frac{d}{dx}[x^2 \sin(x)]$$

$$f(x) = x^2 \qquad g(x) = \sin(x)$$
$$f'(x) = 2x \qquad g'(x) = \cos(x)$$

$$\frac{d}{dx}[x^2 \sin(x)] = 2x \sin(x) + x^2 \cos(x)$$

9.20. Quotient Rule

$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}$$

Example

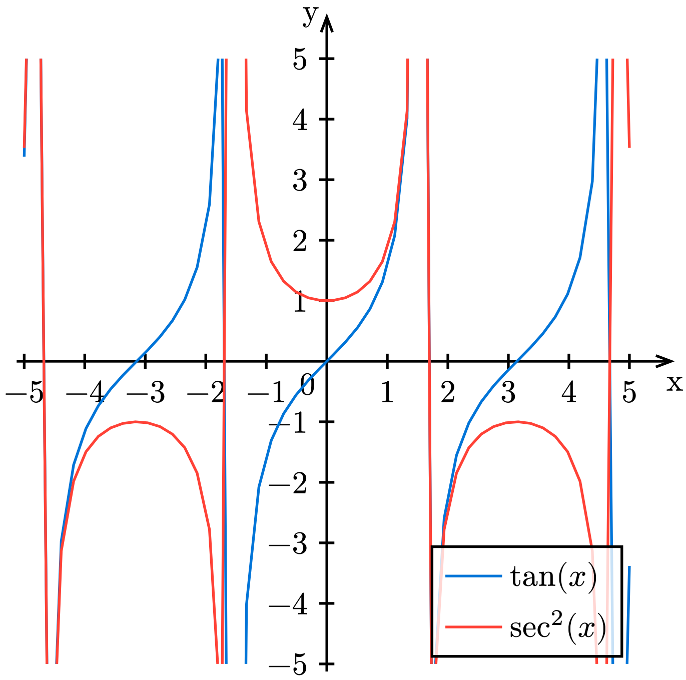
$$\frac{d}{dx}\left[\frac{x^2}{\cos(x)}\right]$$

$$f(x) = x^2 \qquad g(x) = \cos(x)$$
$$f'(x) = 2x \qquad g'(x) = -\sin(x)$$

$$\frac{d}{dx}\left[\frac{x^2}{\cos(x)}\right] = \frac{2x\cos(x) - x^2(-\sin(x))}{[\cos(x)]^2}$$

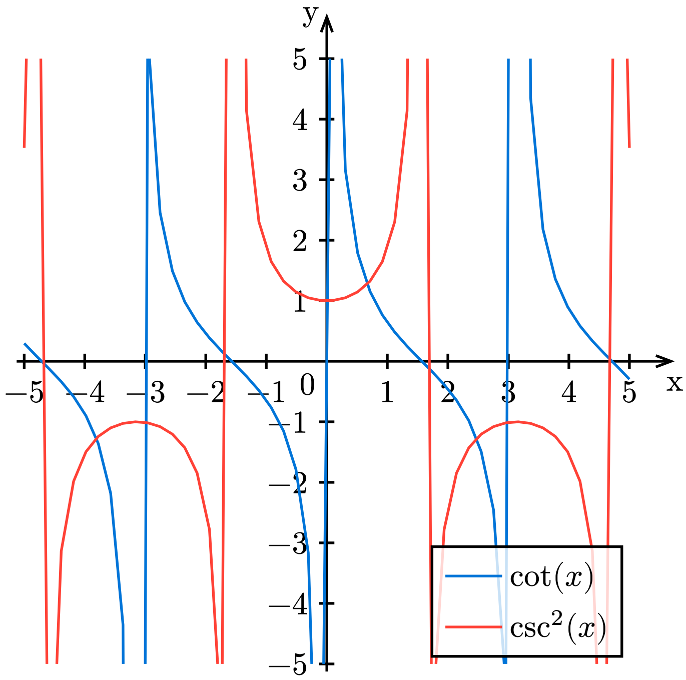
9.20.1. $\tan(x)$

$$\begin{aligned}
\frac{d}{dx}[\tan(x)] &= \frac{d}{dx}\left[\frac{\sin(x)}{\cos(x)}\right] \\
&= \frac{\cos(x)\cdot\cos(x) - \sin(x)\cdot -\sin(x)}{\cos^2(x)} \\
&= \frac{\cos^2(x) + \sin^2(x)}{\cos^2(x)} \\
&= \boxed{\frac{1}{\cos^2(x)}} \\
&= \boxed{\sec^2(x)}
\end{aligned}$$



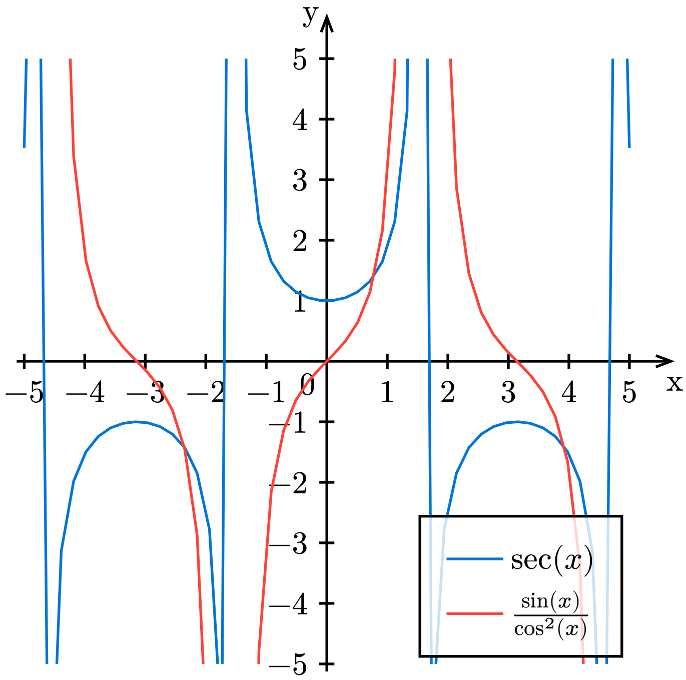
9.20.2. $\cot(x)$

$$\begin{aligned}
\frac{d}{dx}[\cot(x)] &= \frac{d}{dx}\left[\frac{\cos(x)}{\sin(x)}\right] \\
&= \frac{\cos(x)\cdot\cos(x) - \sin(x)\cdot - \sin(x)}{\cos^2(x)} \\
&= \frac{-\sin^2(x) - \cos^2(x)}{} \\
&= \boxed{-\frac{1}{\sin^2}(x)} \\
&= \boxed{-\csc^2(x)}
\end{aligned}$$



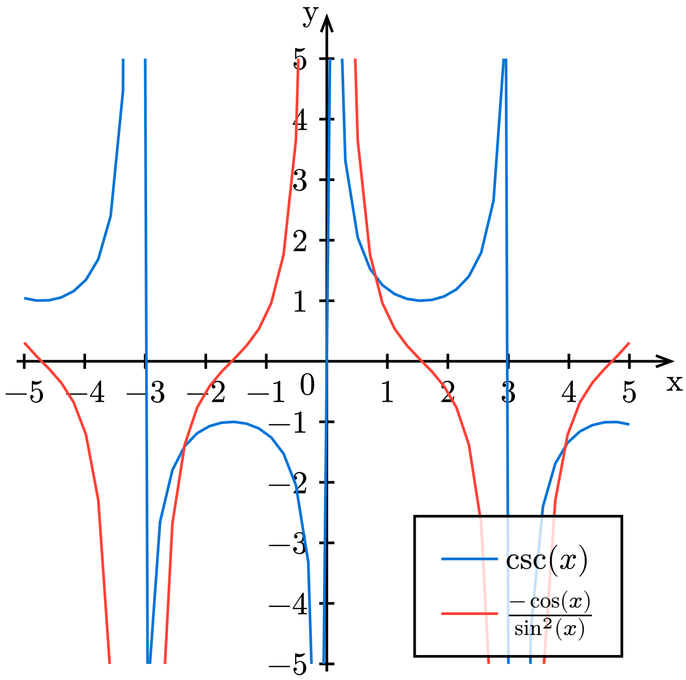
9.20.3. $\sec(x)$

$$\begin{aligned}
\frac{d}{dx}[\sec(x)] &= \frac{d}{dx}\left[\frac{1}{\cos(x)}\right] \\
&= \frac{0\cdot\cos(x) - 1\cdot -\sin(x)}{\cos^2(x)} \\
&= \frac{0 + 1\cdot\sin(x)}{\cos^2(x)} \\
&= \boxed{\frac{\sin(x)}{\cos^2(x)}} \\
&= \frac{\sin(x)}{\cos(x)}\cdot\frac{1}{\cos(x)} \\
&= \boxed{\tan(x)\cdot\sec(x)}
\end{aligned}$$



9.20.4. $\csc(x)$

$$\begin{aligned}
\frac{d}{dx}[\csc(x)] &= \frac{d}{dx}\left[\frac{1}{\sin(x)}\right] \\
&= \frac{0 \cdot \sin(x) - 1 \cdot \cos(x)}{\sin^2(x)} \\
&= \frac{0 - 1 \cdot \cos(x)}{\sin^2(x)} \\
&= \boxed{\tfrac{-\cos(x)}{\sin^2(x)}} \\
&= -\frac{\cos(x)}{\sin(x)} \cdot \frac{1}{\sin(x)} \\
&= \boxed{\cot(x) \cdot \csc(x)}
\end{aligned}$$



9.21. Chain Rule

Compute the derivative of a composite function

$$\begin{aligned} h'(x) &= \frac{d}{dx}[f(g(x))] \\ &= f'(g(x)) \cdot g'(x) \end{aligned}$$

More generally:

$$y = f_1(f_2(f_3(...f_n(x)...)))$$

$$\frac{dy}{dx} = f_1'(f_2(f_3(...f_n(x)...))) \cdot f_2'(f_3(...f_n(x)...)) \cdot f_3'(...f_n(x)...) \cdot ... \cdot f_n'(x)$$

Example

$$\frac{d}{dx}[f(g(x))] = f'(g(x)) \cdot g'(x)$$

$$\frac{d}{dx}\left[e^{\sin(x)}\right]$$

$$\frac{d}{dx}[e^x] = \frac{1}{x} \qquad \frac{d}{dx}[\sin(x)] = \cos(x)$$

$$f'(g(x)) = \frac{1}{\sin(x)} \qquad g'(x) = \cos(x)$$

$$\frac{d}{dx}\left[\underbrace{\ln\left(\overbrace{\sin(x)}^{g(x)}\right)}_{f(g(x))}\right] = \frac{1}{\sin(x)} \cdot \cos(x)$$

$$f(x) = \cos^3(x) = (\cos(x))^3$$

$$f(x) = v(u(x))$$

$$f'(x) = v'(u(x)) \cdot u'(x)$$

$$\begin{aligned} f'(x) &= \frac{dv}{du} \cdot \frac{du}{dx} \\ &= \frac{d(\cos(x))^3}{d\cos(x)} \cdot \frac{d\cos(x)}{x} \\ &= 3(\cos(x))^2 \cdot -\sin(x) \\ &= -3(\cos(x))^2 \sin(x) \end{aligned}$$

9.22. Implicit Differentiation

When a function is not explicitly solved for one variable in terms of another. E.g.:

$$x^2 + y^2 = 1$$

Instead of solving for y explicitly in terms of x, implicit differentiation allows you to differentiate both sides of an equation directly, treating y as an implicit function of x.

Steps for Implicit Differentiation

1. Differentiate both sides of the equation with respect to x, treating y as a function of x
2. Apply the chain rule whenever differentiating y, since $y = y(x)$
2. Solve for $\frac{dx}{dy}$

Example

$$x^2 + y^2 = 1$$

1. Differentiate both sides

$$\frac{d}{dx}[x^2 + y^2] = \frac{d}{dx}[1]$$

$$\frac{d}{dx}[x^2] + \frac{d}{dx}[y^2] = \frac{d}{dx}[1]$$

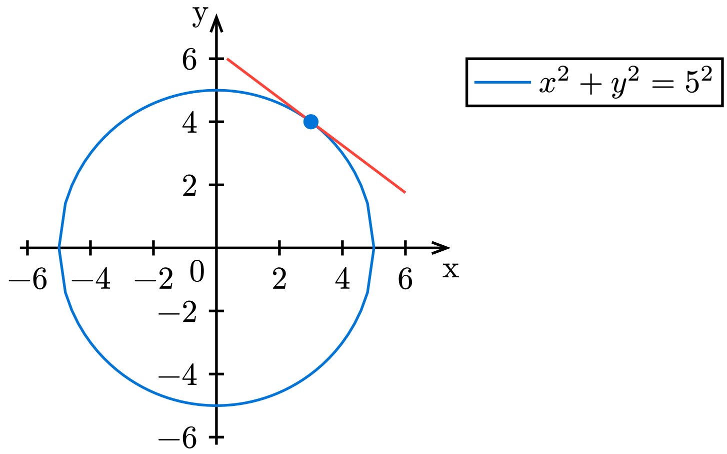
$$\frac{d}{dx}[x^2] + \frac{d}{dx}[y^2] = 0$$

2. Apply the chain rule to y^2

$$2x + 2y\frac{dy}{dx} = 0$$

3. Slove for $\frac{dy}{dx}$

$$\begin{aligned} 2y \cdot \frac{dy}{dx} &= -2x \\ \frac{dy}{dx} &= \frac{-2x}{2y} \\ \frac{dy}{dx} &= -\frac{x}{y} \end{aligned}$$



$$\begin{aligned} x^2 + y^2 &= 5^2 \\ 2xdx + 2ydy &= 0 \\ \frac{dy}{dx} &= -\frac{x}{y} \end{aligned}$$

9.23. Derivatives of Inverse Functions

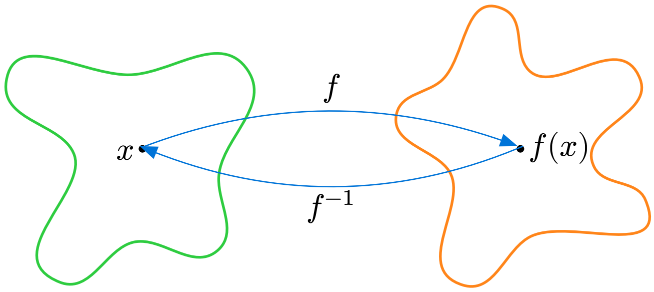
Finding the derivative of the inverse function $f^{-1}(x)$ at a given point directly from the function $f(x)$. Instead of explicitly computing the inverse function $f^{-1}(x)$, we use the inverse function derivative formula:

$$\frac{d}{dx}[f^{-1}(x)] = \frac{1}{f'(f^{-1}(x))}$$

This approach allows us to determine the derivative of the inverse function without needing to express $f^{-1}(x)$ explicitly. Instead, we find the value of x that satisfies $f(x) = a$ (where a a is the given point), evaluate $f'(x)$, and apply the formula.

1. Definition of Inverse Function

$$\begin{aligned} f(f^{-1}(x)) &= x \\ f^{-1}(f(x)) &= x \end{aligned}$$



2. Differentiate Both Sides

Differenatiate both sides with respect to x

$$\frac{d}{dx}[f(f^{-1}(x))] = \frac{d}{dx}[x]$$

The right-hand side simplifies to:

$$\frac{d}{dx}[f(f^{-1}(x))] = 1$$

Using chain-rule:

$$\frac{d}{dx}[f(g(x))] = f'(g(x)) \cdot g'(x)$$

The left side expands as:

$$\frac{d}{dx}[f(f^{-1}(x))] = f'(f^{-1}(x)) \cdot \frac{d}{dx}[f^{-1}(x)]$$

This we get:

$$f'(f^{-1}(x)) \cdot \frac{d}{dx}[f^{-1}(x)] = 1$$

3. Solve for $\frac{d}{dx}[f^{-1}(x)]$

Rearrange to isolate $\frac{d}{dx}[f^{-1}(x)]$:

$$\frac{d}{dx}[f^{-1}(x)] = \frac{1}{f'(f^{-1}(x))}$$

Example

Given the function:

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

We want to find $\frac{d}{dx}f^{-1}(x)$ at $x = 0.5$ using the inverse function derivative formula:

$$\frac{d}{dx}f^{-1}(x) = \frac{1}{f'(f^{-1}(x))}$$

---

1. Compute $f'(x)$

Differenatiate $f(x)$:

$$f'(x) = 3x^2$$

---

2. Solve for x such that $f(x) = 0.5$

We need to find x such that:

$$x^3 = 0.5$$

Solving for x:

$$x = \sqrt[3]{0.5}$$

---

3. Compute $f'\left(\sqrt[3]{0.5}\right)$

Evaluate the derivative at $x = \sqrt[3]{0.5}$:

$$f'\left(\sqrt[3]{0.5}\right) = 3\left(\sqrt[3]{0.5}\right)^2$$

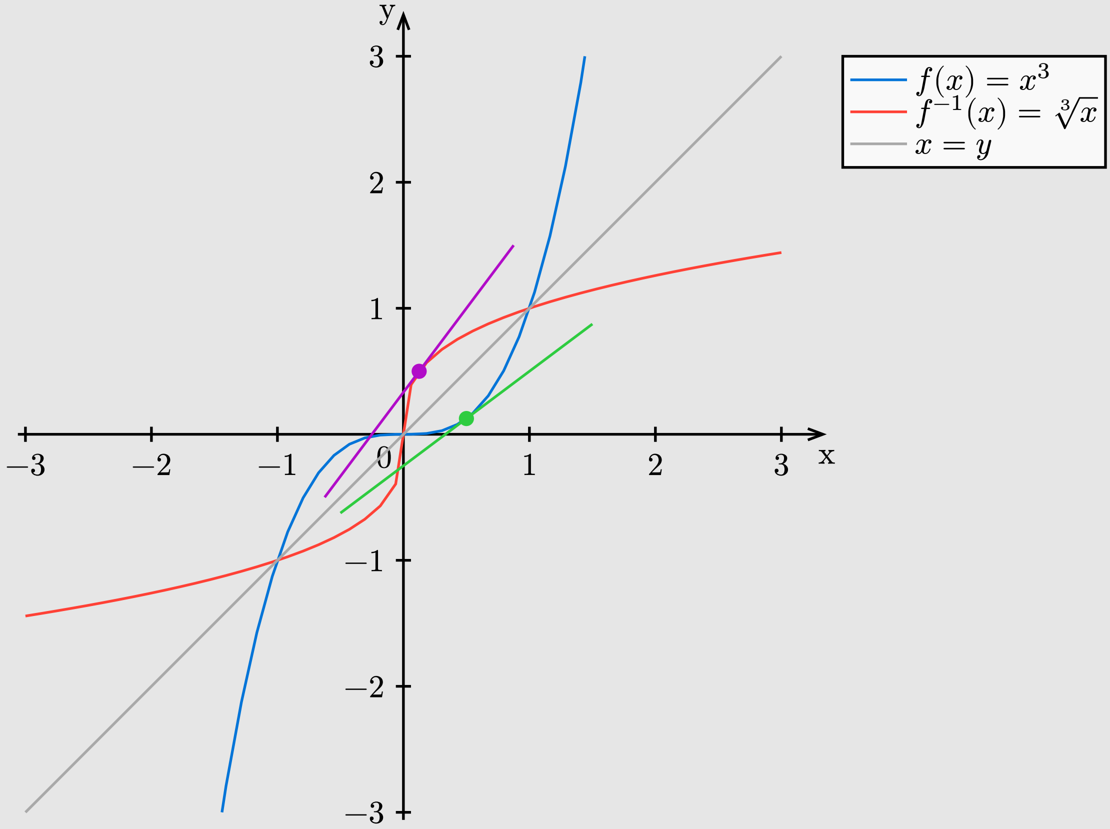
---

4. Use the formula

$$\frac{d}{dx}f^{-1}(0.5) = \frac{1}{3\left(\sqrt[3]{0.5}\right)^2}$$

---

5. Interpretation



The expression:

$$\frac{d}{dx}f^{-1}(0.5) = \frac{1}{3\left(\sqrt[3]{0.5}\right)^2}$$

represents the derivative of the inverse function $f^{-1}(x)$ evaluated at $x = 0.5$. This means it gives the slope of the tangent line to the inverse funcion at $x = 0.5$.

inverse.py

```
from sympy import symbols, solve

# y = x**3 + x

x, y = symbols('x y')

f = x**3 + x - y

inverse = solve(f, x)

print(inverse)
```

9.23.1. Derivative Inverse Sin

$$\frac{d}{dx}\left[\sin^{-1}(x)\right] = \frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}[\arcsin(x)] = \frac{1}{\sqrt{1-x^2}}$$

9.23.2. Derivative Inverse Cos

$$\frac{d}{dx}[\cos^{-1}(x)] = -\frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}[\arccos(x)] = -\frac{1}{\sqrt{1-x^2}}$$

9.23.3. Derivative Inverse Tan

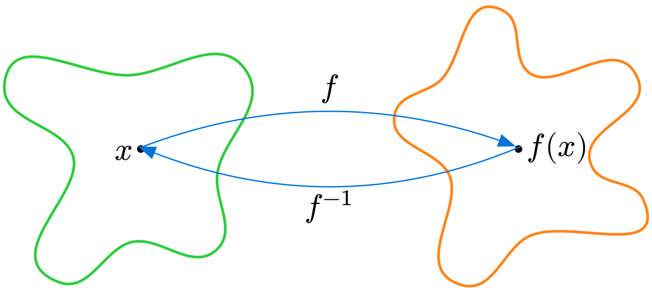
$$\frac{d}{dx}[\tan^{-1}(x)] = -\frac{1}{1+x^2}$$

$$\frac{d}{dx}[\arctan(x)] = -\frac{1}{1+x^2}$$

9.24. Inverse Functions

1. Definition

$$f(f^{-1}(x)) = x$$
$$f^{-1}(f(x)) = x$$



A function $f : A \to B$ has an inverse function f^{-1} if and only if f is **bijective** (i.e., both one-to-one and onto):

- **Injective (One-to-One)**:

  $f(x_1) = f(x_2)$ implies $x_1 = x_2$

  No two inputs map to the same output

- **Surjective (Onto)**:

  Every element in B is mapped from some element in A

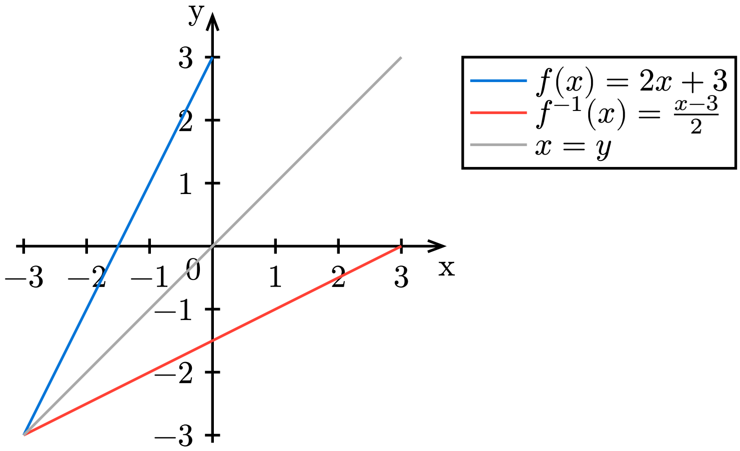
2. Finding Inverse Function

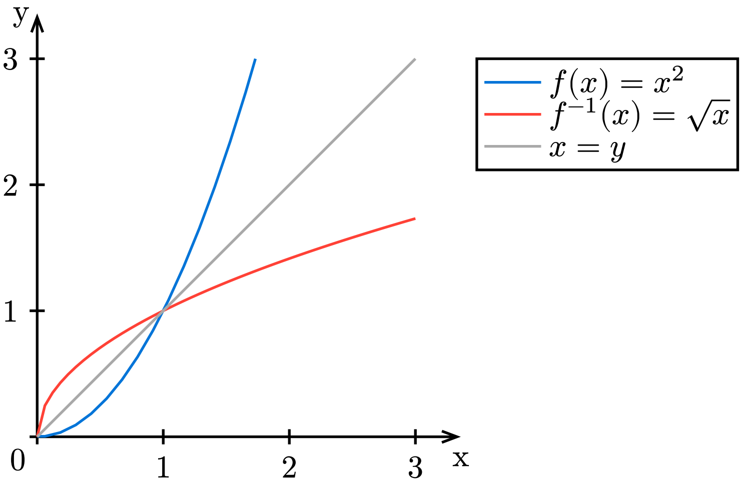
To determine f^{-1}:

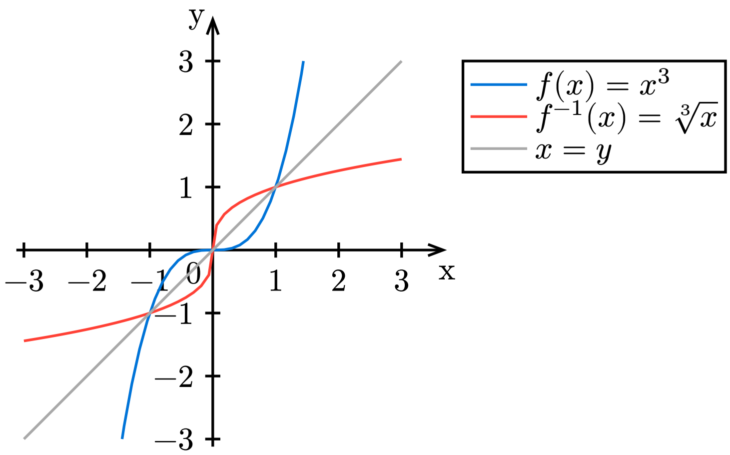
- Express y in terms of x: $y = f(x)$
- Solve for x in terms of y
- Swap x and y, remaining y as $f^{-1}(x)$

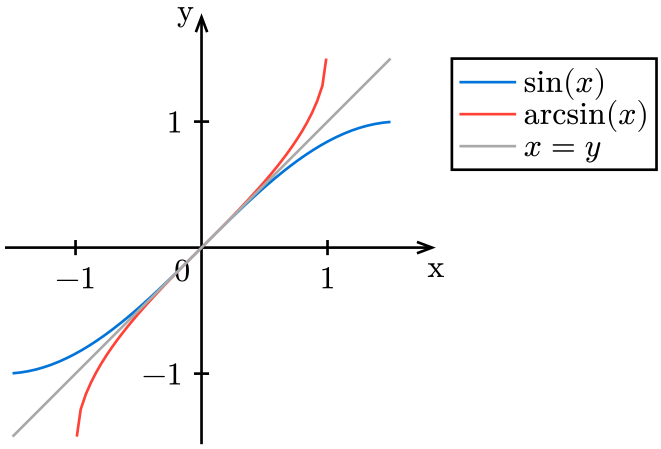
3. Graphical Representation

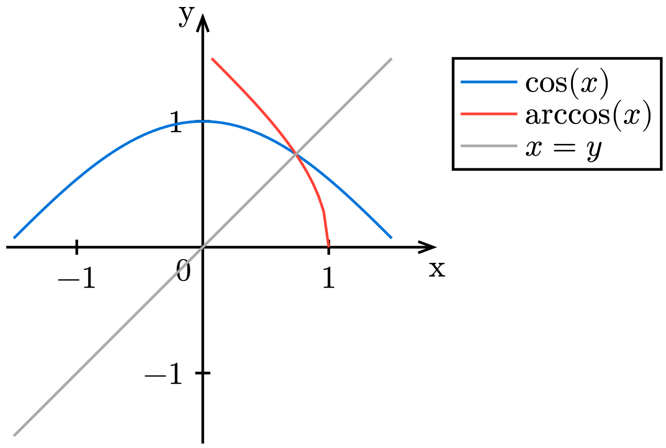
The graph of f^{-1} is a reflection of the graph of f accress the line $x = y$

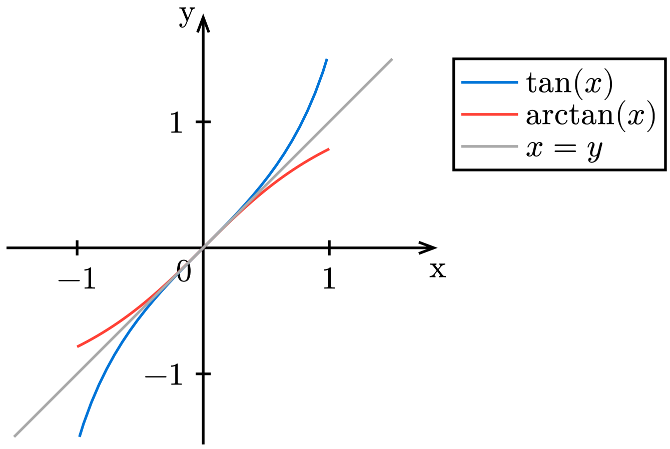




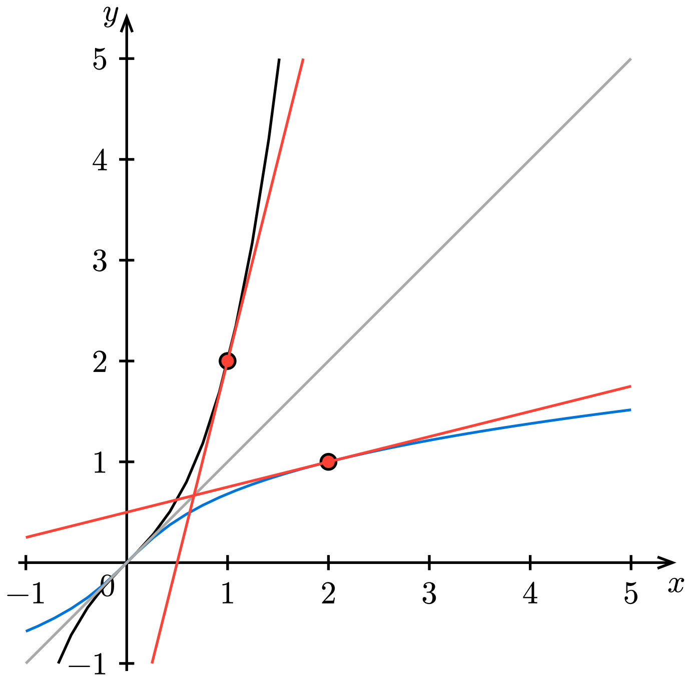








2. Derivative of Inverse Functions



9.25. L'Hôpital's Rule

Evaluating limits that result in a indeterminite form like $\frac{0}{0}$ or $\frac{\infty}{\infty}$

If $\lim_{x\to a} f(x) = 0$ and $\lim_{x\to a} g(x) = 0$ (or both go to $\pm\infty$), and $f(x)$ and $g(x)$ are differentiable near a, then:

$$\lim_{x\to a} \frac{f(x)}{g(x)} = \lim_{x\to a} \frac{f'(x)}{g'(x)}$$

Example

Consider:

$$\lim_{x\to 0} \frac{1-\cos(x)}{x^2}$$

Step 1: Direct Substitution

Substituting $x = 0$:

$$\frac{1-\cos(0)}{0^2} = \frac{1-1}{0} = \frac{0}{0}$$

Since this is an indeterminate form, we apply L'Hôpital's Rule.

Step 2: First Application of L'Hôpital's Rule

Differentiate the numerator and denominator:

- Numerator: $f(x) = 1 - \cos(x) \Rightarrow f'(x) = \sin(x)$
- Denominator: $g(x) = x^2 \Rightarrow g'(x) = 2x$

Thus, applying L'Hôpital's Rule:

$$\lim_{x \to 0} \frac{1 - \cos(x)}{x^2} = \lim_{x \to 0} \frac{\sin(x)}{2x}$$

Step 3: Second Check for Indeterminate Form

Substituting $x = 0$:

$$\frac{\sin(0)}{2(0)} = \frac{0}{0}$$

Since this is still an indeterminate form, we apply L'Hôpital's Rule again.

Step 4: Second Application of L'Hôpital's Rule

Differentiate again:

- Numerator: $f'(x) = \sin(x) \Rightarrow f''(x) = \cos(x)$
- Denominator: $g'(x) = 2x \Rightarrow g''(x) = 2$

Applying L'Hôpital's Rule again:

$$\lim_{x \to 0} \frac{\sin(x)}{2x} = \lim_{x \to 0} \frac{\cos(x)}{2}$$

Step 5: Evaluate the Limit

Now, substituting $x = 0$:

$$\frac{\cos(0)}{2} = \frac{1}{2}$$

Final Answer:

$$\boxed{\lim_{x \to 0} \frac{1 - \cos(x)}{x^2} = \frac{1}{2}}$$

9.26. Mean Value Theorem

Let $f : [a, b] \to \mathbb{R}$ be a function tha satifies the following conditions:

1. f is **continuous** on the closed interval $[a, b]$
2. f is **differentiable** on the open interval (a, b)

Then there exists at least one point $c \in (a, b)$ such that

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

This means that the instantaneous rate of change (derivative) at some point c is equal to the average rate of change over the entire interval

Example

Consider $f(x) = x^2$ on $[1, 3]$

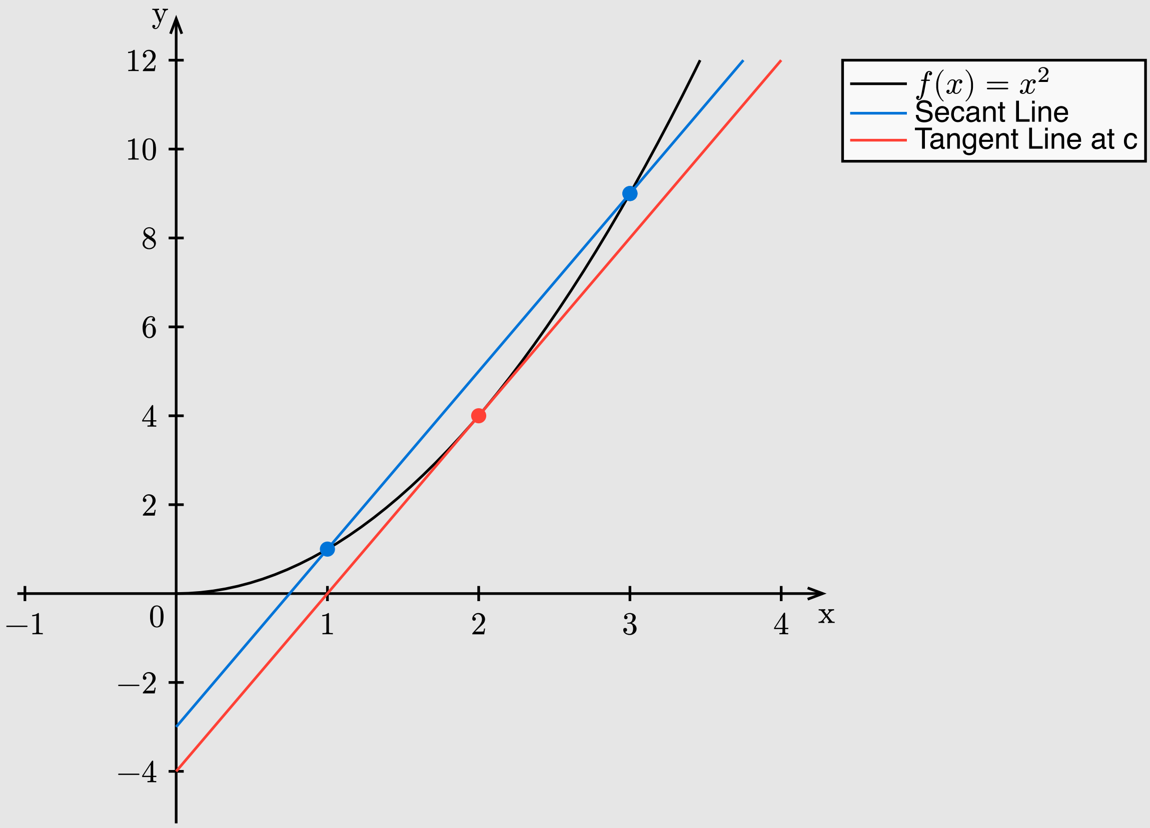
- The average rate of change is:

$$\frac{f(3) - f(1)}{3 - 1} = \frac{9 - 1}{2} = 4$$

- The derivative for $f(x)$ is $f'(x) = 2x$
- Setting $f'(c) = 4$, we solve:

$$2c = 4 \Rightarrow c = 2$$

Thus, at $c = 2$, the instantaneous rate of change matches the average rate of change



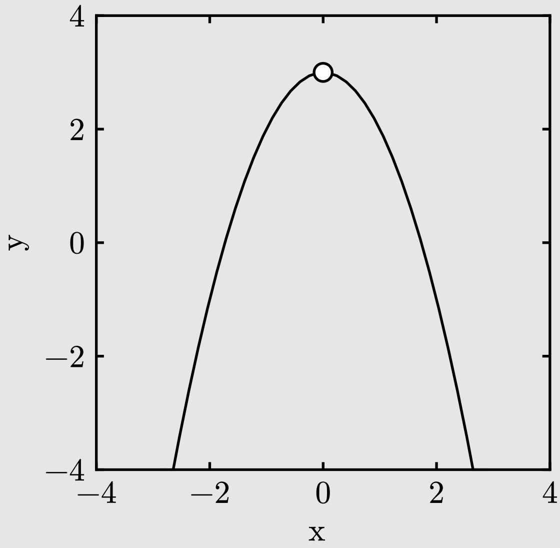
9.27. Extreme Value Theorem

The Extreme Value Theorem states that if a function $f(x)$ is continuous on a closed interval $[a, b]$, then $f(x)$ must attain both a maximum and a minimum value within that interval. This means there exist points $c, d \in [a, b]$ such that:

$$f(c) \geq f(x) \quad \text{and} \quad f(d) \leq f(x) \quad \text{for all} \quad x \in [a, b]$$

1. **Continuity**: The function must be continuous on $[a, b]$. Discontinuities (jumps, asymptotes, holes) can prevent the function from attaining an extreme value

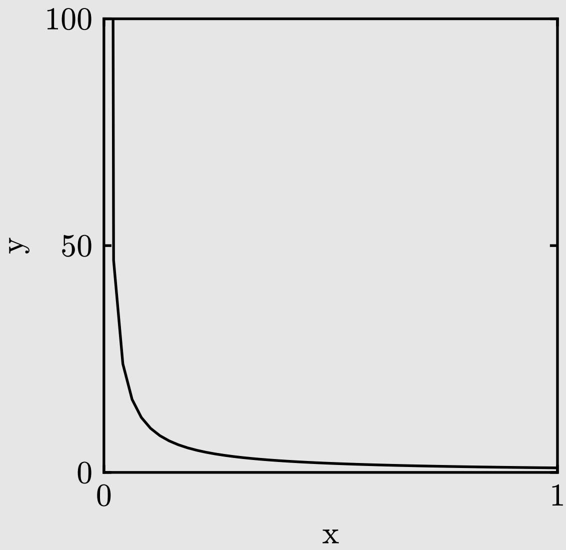
Example



2. **Closed Interval**: If the function is defined on an open interval (a, b), an extremum may not exist

Example

$f(x) = \frac{1}{x}$on $(0, 1]$ has no maximum because it keeps increasing as $x \to 0$.



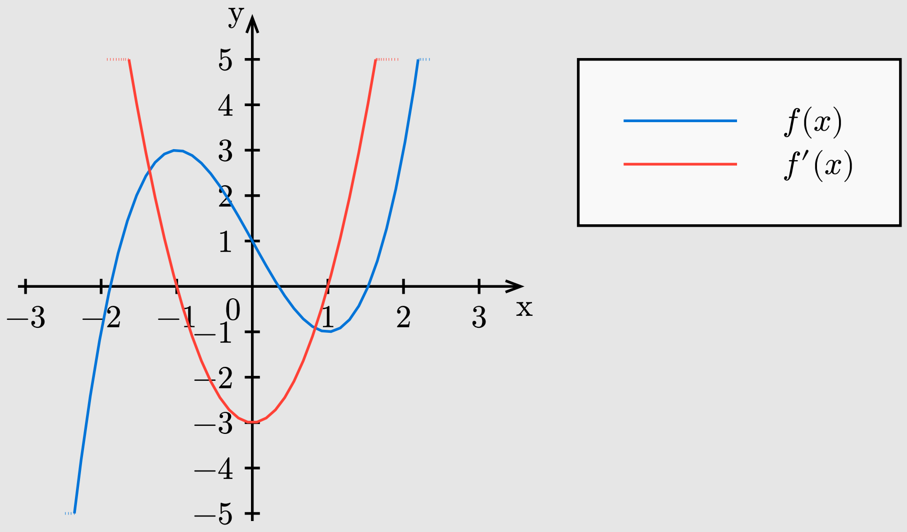
9.27.1. Critical points

A critical point of a function $f(x)$ is a point in the domain where either:

- $f'(x) = 0$
- $f'(x)$ is undefined

Example

$f(x) = \frac{1}{x}$ on $(0, 1]$ has no maximum because it keeps increasing as $x \to 0$.



9.27.2. Global vs. Local Extrema

- $f(c)$ is a **relative maximum** if $f(c) \geq f(x)$ for all $x \in (c - h, c + h)$ for $h > 0$
- $f(d)$ is a **relative minimum** if $f(d) \leq f(x)$ for all $x \in (d - h, d + h)$ for $h > 0$

9.27.3. First and Second Derivative Tests

First Derivative Test and the Second Derivative Test are used to classify critical points, and both aim to determine whether the point is a local maximum, local minimum, or neither

9.28. Derivative Tests

Find critical points by solving $f'(x) = 0$

1^{st} Derivative	$+ \longrightarrow -$	Local Max	Concave Down
	$- \longrightarrow +$	Local Min	Concave Up
2^{nd} Derivative	$f''(c) < 0$	Local Max	Concave Down
	$f''(c) > 0$	Local Min	Concave Up

9.28.1. First Derivative Test

If $f'(x)$ changes sign around a critical point c, we can determine if $f(c)$ is a local maximum or minimum:

- If $f'(x)$ changes from **positive to negative** at c, then $f(c)$ is a **local maximum**
- If $f'(x)$ changes from **negative to positive** at c, then $f(c)$ is a **local minimum**
- If $f'(x)$does **not** change sign, $f(c)$ is **not** a local extremum

Example

If $f'(x)$ changes from **negative to positive** at c, then $f(c)$ is a **local minimum**

Suppose:

$$f(x) = x^2, \qquad f'(x) = 2x$$

Step 1. Find critical point

Solve $f'(x) = 0$:

$$2x = 0 \implies x = 0$$

So, $x = 0$ is the only critical point

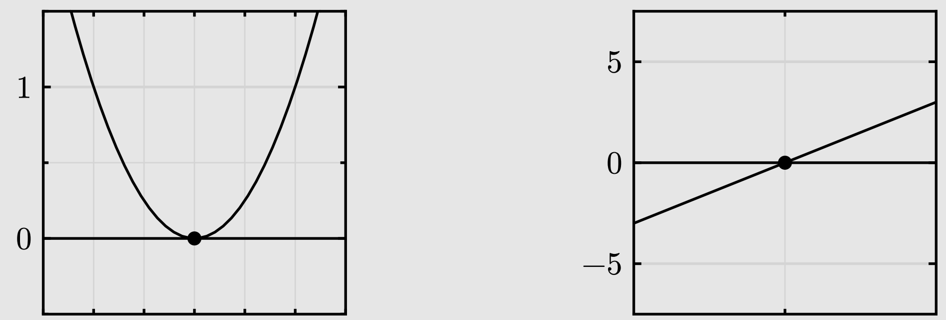
Step 2. Test sign of $f'(x)$ around $x = 0$

- **Left**: $x = -1$

$$f'(-1) = 2(-1) = -2 < 0 \implies f \text{ is } \mathbf{decreasing}$$

- **Right**: $x = 1$:

$$f'(1) = 2(1) = 2 > 0 \implies f \text{ is } \mathbf{increasing}$$



Step 3. Interpretation

Since $f'(x)$ changes from **negative** to **positive** at $x = 0$,

$$f(0) = 0 \implies \mathbf{local\ minimum}$$

Example

If $f'(x)$ changes from **positive to negative** at c, then $f(c)$ is a **local maximum**

Suppose:

$$f(x) = -x^2, \qquad f'(x) = -2x$$

Step 1. Find critical point

Solve $f'(x) = 0$

$$-2x = 0 \implies x = 0$$

So, $x = 0$ is the only critical point

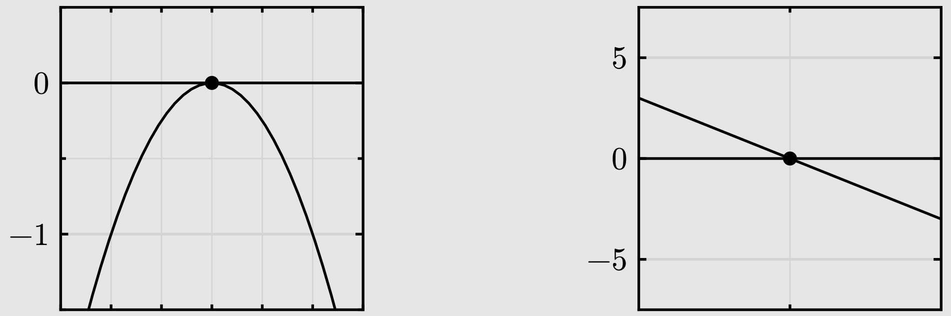
Step 2. Test sign of $f'(x)$ around $x = 0$

- **Left**: $x = -1$

$$f'(-1) = -2(-1) = 2 > 0 \quad \Longrightarrow \quad f \text{ is } \textbf{increasing}$$

- **Right**: $x = 1$

$$f'(1) = -2(1) = -2 < 0 \quad \Longrightarrow \quad f \text{ is } \textbf{decreasing}$$



Step 3. Interpretation

Since $f'(x)$ changes from **positive** to **negative** at $x = 0$,

$$f(0) = 0 \quad \Longrightarrow \textbf{local maximum}$$

Example

If $f'(x)$does **not** change sign, $f(c)$ is **not** a local extremum

Suppose

$$f(x) = x^3, \qquad f'(x) = 3x^2$$

Step 1. Find critical point

Solve $f'(x) = 0$

$$3x^2 = 0 \quad \Longrightarrow \quad x = 0$$

So, $x = 0$ is the only critical point

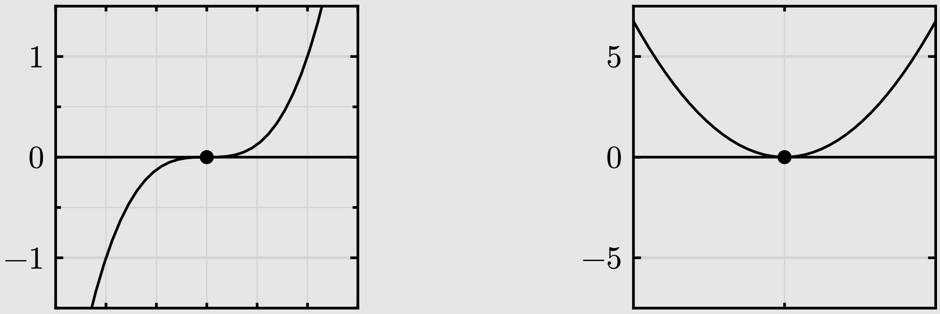
Step 2. Test sign of $f'(x)$ around $x = 0$

- **Left**: $x = -1$

$$f'(-1) = 3(-1)^2 = 3 > 0 \quad \Longrightarrow \quad f \text{ is } \textbf{increasing}$$

- **Right**: $x = 1$

$$f'(1) = 3(1)^2 = 3 > 0 \quad \Longrightarrow \quad f \text{ is } \textbf{increasing}$$



Step 3. Interpretation

Since $f'(x)$ **does not change sign** at $x = 0$,

$$f(0) = 0 \implies \textbf{not a local extremum}$$

but rather a point of inflection

9.28.2. Second Derivative Test

If $f''(x)$ is continuous near a critical point c, and $f'(c) = 0$, then:

- If $f''(c) > 0$, then $f(c)$ is a **local minimum** (concave up)
- If $f''(c) < 0$, then $f(c)$ is a **local maximum** (concave down)
- If $f''(c) = 0$, the test is **inconclusive** — use the first derivative test or other methods

Example

If $f''(c) > 0$, then $f(c)$ is a **local minimum** (concave up)

Suppose

$$f(x) = x^2, \qquad f'(x) = 2x, \qquad f''(x) = 2$$

Step 1. Find critical points

Solve $f'(x) = 0$

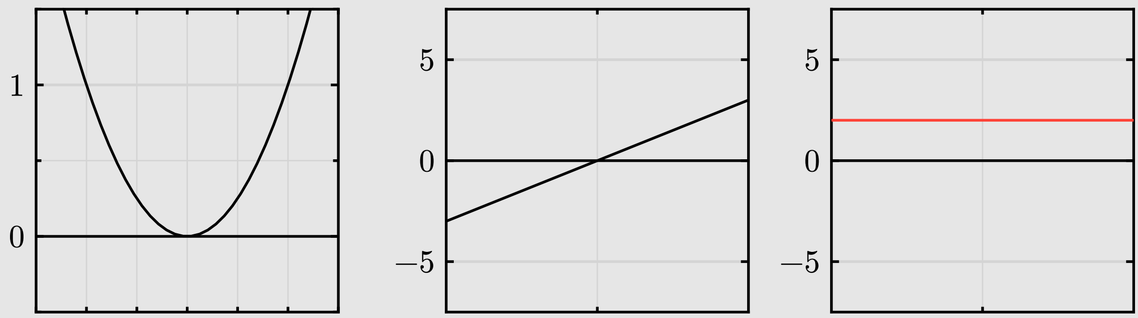
$$2x = 0 \implies x = 0$$

So, $x = 0$ is the only critical point

Step 2. Evaluate the second derivative at $x = 0$

$$f''(0) = 2 > 0$$

Since the second derivative is positive, the function is **concave up** near $x = 0$



Step 3. Interpretation

$$f(0) = 0 \implies \textbf{local minimum}$$

Example

If $f''(c) < 0$, then $f(c)$ is a **local maximum** (concave down)

Suppose

$$f(x) = -x^2, \qquad f'(x) = -2x, \qquad f''(x) = -2$$

Step 1. Find critical points

Solve $f'(x) = 0$

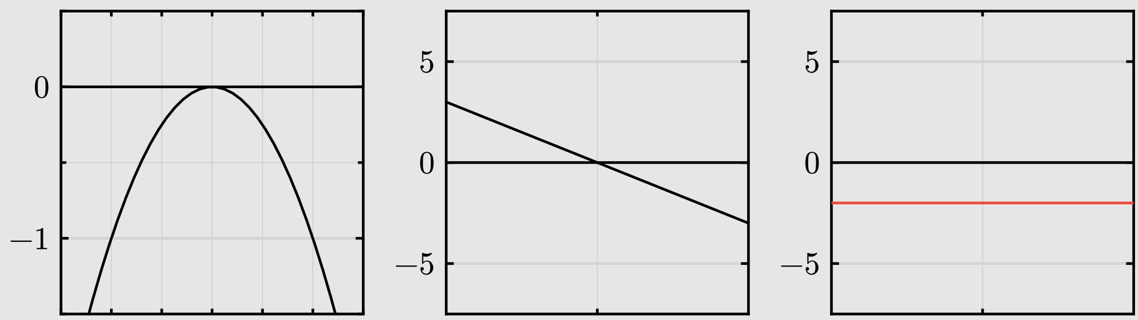
$$-2x = 0 \implies x = 0$$

So, $x = 0$ is the only critical point

Step 2. Evaluate the second derivative at $x = 0$

$$f''(0) = -2 < 0$$

Since the second derivative is negative, the function is **concave down** near $x = 0$



Step 3. Interpretation

$$f(0) = 0 \implies \textbf{local maximum}$$

Example

If $f''(c) = 0$, the test is **inconclusive** — use the first derivative test or other methods

Suppose

$$f(x) = x^3, \qquad f'(x) = 3x^2, \qquad f''(x) = 6x$$

Step 1. Find critical points

Solve $f'(x) = 0$

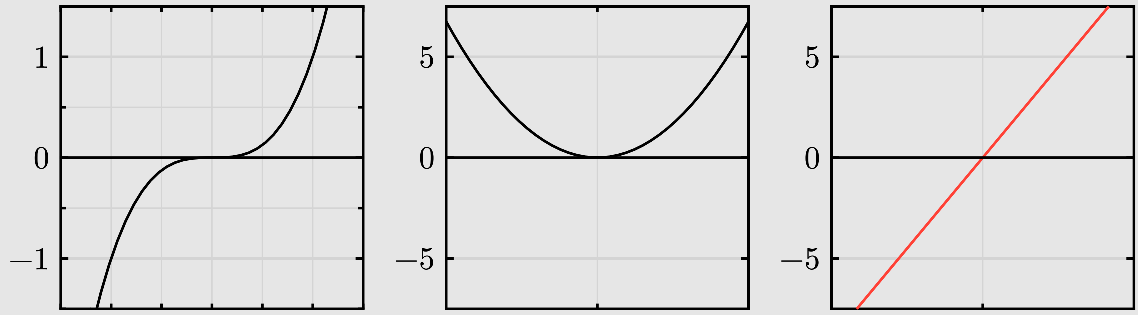
$$3x^2 = 0 \quad \Longrightarrow \quad x = 0$$

So, $x = 0$ is the only critical point

Step 2. Evaluate the second derivative at $x = 0$

$$f''(0) = 6(0) = 0$$

Since the second derivative is 0, the test in **inconclusive**



Step 3. Use another method

Test sign of $f'(x)$ around $x = 0$

- **Left**: $x = -1$

$$f'(-1) = 3(-1)^2 = 3 > 0 \quad \Longrightarrow \quad f \text{ is } \textbf{increasing}$$

- **Right**: $x = 1$

$$f'(1) = 3(1)^2 = 3 > 0 \quad \Longrightarrow \quad f \text{ is } \textbf{increasing}$$

Step 4. Interpretation

Since $f'(x)$ **does not change sign** at $x = 0$,

$$f(0) = 0 \quad \Longrightarrow \quad \textbf{not a local extremum}$$

but rather a point of inflection

10. Calculus II

10.1. Derivative Objects

Gradient	$f : \mathbb{R}^n \to \mathbb{R}$	$n \times 1$	Vector of first derivatives; special case of Jacobian (scalar output)
Jacobian	$F : \mathbb{R}^n \to \mathbb{R}^m$	$m \times n$	Matrix of first derivatives; general case for vector-valued functions
Hessian	$f : \mathbb{R}^n \to \mathbb{R}$	$n \times n$	Matrix of second derivatives; Jacobian of the gradient

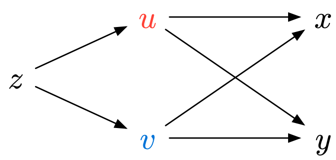
	Input/Output	**Derivatives**	**Notation**
Gradient (Derivative) (1×1)	$f : \mathbb{R} \to \mathbb{R}$ One Input **One Output** Scalar Function	**First** Order	$\nabla f(x) = f'(x) = \left[\frac{\partial f}{\partial x}\right]$
Jacobian $(m \times 1)$	$f : \mathbb{R} \to \mathbb{R}^m$ One Input **Multiple Output** Vector Function	**First** Order	$J_f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} \\ \vdots \\ \frac{\partial f_m}{\partial x} \end{bmatrix}$
Gradient $(n \times 1)$	$f : \mathbb{R}^n \to \mathbb{R}$ Multiple Input **One Output** Scalar Function	**First** Order	$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$
Jacobian $(m \times n)$	$f : \mathbb{R}^n \to \mathbb{R}^m$ Multiple Input **Multiple Output** Vector Function	**First** Order	$J_f(\mathbf{x}) = \begin{bmatrix} \nabla g_1(x)^T \\ \nabla g_2(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$
Hessian $(n \times n)$	$f : \mathbb{R}^n \to \mathbb{R}$ Multiple Input One Output Scalar Function	**Second** Order	$\nabla^2 f(\mathbf{x}) = H_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$

Laplacian	Trace of the Hessian (sum of diagonal entries)		$\mathrm{tr}(H_f(\mathbf{x})) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}(x)$ $H_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$

Gradient	1^{st} derivative	$n \times 1$
Hessian	2^{nd} derivative	$n \times n$
3^{rd} Order Tensor	3^{rd} derivative	$n \times n \times n$
	$\vdots$	
k-th Order Tensor	k^{th} derivative	$n \times n \times \ldots \times n \quad (k \text{ times})$

$$z = f(u, v)$$

$$u = f(x, y) \qquad v = f(x, y)$$



The derivative of z with respect to x is the sum of all possible paths from z to x:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial z}{\partial v}\frac{\partial v}{\partial x}$$

Example

Let:

- $u = x + y$
- $v = xy$
- $z = u^2 + \sin(v)$

We want to compute:

$$\frac{\partial z}{\partial x}$$

Step 1: Apply the multivariable chain rule

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial z}{\partial v}\frac{\partial v}{\partial x}$$

Step 2: Compute partial derivatives

- $\frac{\partial u}{\partial x} = \frac{\partial}{\partial x}(x+y) = 1 + 0 = \boxed{1}$
- $\frac{\partial v}{\partial x} = \frac{\partial}{\partial x}(xy) = 1y = \boxed{\text{y}}$
- $\frac{\partial z}{\partial u} = \frac{\partial}{\partial u}(u^2 + \sin(v)) = 2u + 0 = \boxed{\text{2u}}$
- $\frac{\partial z}{\partial v} = \frac{\partial}{\partial v}(u^2 + \sin(v)) = 0 + \cos(v) = \boxed{\cos(v)}$

Step 3: Plug in:

$$\begin{aligned}\frac{\partial z}{\partial x} &= 2u \cdot 1 + \cos(v) \cdot y \\ &= 2(x+y) + y\cos(xy)\end{aligned}$$

10.2. Double Integrals

10.2.1. 1. Double Indefinite Integral

$$\int\int f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

First Integration:

Integrate with respect to x (treat y as constant):

$$\int f(x,y)\,\mathrm{d}x = F(x,y) + C_1(y)$$

- $F(x,y)$: antiderivative of f w.r.t. x
- $C_1(y)$: Constant of integration (can depend on y)

Second Integration:

Integrate with respect to y (treat x as constant):

$$\int [F(x,y) + C_1(y)]\,\mathrm{d}y = G(x,y) + C_2(x)$$

- $G(x,y)$: antiderivative of $F(x,y)$ with respect to y

Final Output:

A family of functions $G(x,y) + C_2$, representing all functions whose mixed partial derivative $\frac{\partial^2}{\partial y \partial x}$ is $f(x,y)$

10.3. Gradient

The gradient $\nabla f(x,y)$ is a vector of expressions (equations)

If

$$f : \mathbb{R}^n \to \mathbb{R}$$

The gradient is always **a vector in the same space as the input variables**

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n$$

Example

$$f(x, y) = x^2 + xy + y^2$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x + y \\ x + 2y \end{bmatrix}$$

Evaluating at a point

If you plug in numbers for x and y, each component becomes a scalar:

$$(x, y) = (1, 2) \quad \Rightarrow \quad \nabla f(1, 2) = \begin{bmatrix} 2 \cdot 1 + 2 \\ 1 + 2 \cdot 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

10.4. Hessian Matrix

Square matrix of second-order partial derivatives

Describes the local curvature of a multivariable function

$$f : \mathbb{R}^n \to \mathbb{R}$$

Input: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

Output: Scalar

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

Diagonal entries: $\frac{\partial^2 f}{\partial x_i^2}$

Second derivative of f with respect to a single variable twice

How the slope of f changes as you move along the x_1 direction

Off-diagonal entries: $\frac{\partial^2 f}{\partial x_i \partial x_j}$ where $i \neq j$

How the slope in one direction changes when you move in a different direction

$\frac{\partial^2 f}{\partial x_1 \partial x_2}$: How the rate of change of f along x_1 changes as you move in the x_2 direction

Symmetric (Clairaut-Schwarz theorem):

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$$

$$H_f(x) = H_f(x)^T$$

Example

$$f(x, y) = x^3 y + y^2$$

- Symbolic Hessian

$$\nabla^2 f(x, y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 6xy & 3x^2 \\ 3x^2 & 2 \end{bmatrix}$$

- Evaluate at $(x, y) = (1, 2)$

$$\nabla^2 f(1, 2) = \begin{bmatrix} 6 \cdot 1 \cdot 2 & 3 \cdot 1^2 \\ 3 \cdot 1^2 & 2 \end{bmatrix} = \begin{bmatrix} 12 & 3 \\ 3 & 2 \end{bmatrix}$$

Each entry is now a scalar, and the Hessian is numeric

10.5. Positive/Negative Definite/Indefinite/ Semidefinite Matrices

Let $\mathbf{x} \in \mathbb{R}^n$be any nonzero vector. Look at the quadratic form:

$$\mathbf{x}^T H \mathbf{x}$$

<table>
<tr><td colspan="4">$$q(x) = \mathbf{x}^T H \mathbf{x}$$ $$H\mathbf{x} = \lambda \mathbf{x}$$</td></tr>
<tr><td></td><td>Definition</td><td>Eigen Values</td><td>Geometry</td></tr>
</table>

Positive Definite	$q(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0}$	$\lambda_i > 0$ $H \succ 0$	**Min** (bowl-shaped)
Negative Definite	$q(\mathbf{x}) < 0 \quad \forall \mathbf{x} \neq \mathbf{0}$	$\lambda_i < 0$ $H \prec 0$	**Max** (inverted bowl)
Positive Semidefinite	$q(\mathbf{x}) \geq 0 \quad \forall \mathbf{x}$	$\lambda_i \geq 0$ $H \succeq 0$	**Min** but **flat** in some directions
Negative Semidefinite	$q(\mathbf{x}) \leq 0 \quad \forall \mathbf{x}$	$\lambda_i \leq 0$ $H \preceq 0$	**Max** but **flat** in some directions
Indefinite	some $q(\mathbf{x_1}) > 0$ some $q(\mathbf{x_2}) < 0$	some $\lambda_i > 0$ some $\lambda_j < 0$	**Saddle** Up in some directions Down in others

11. Calculus III

11.1. Differential Equations

A differential equation specifies a relationship between an unknown function and its derivatives

Solving for (an) unknow function(s) that satisfy this relationship

Initial or boundary conditions to select a unique solution

11.1.1. Linearity

The **dependent** variable and its derivatives each appear only to the first power, are not multiplied together, and are not inside any other function

Example

Linear

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y = 0$$

Non-Linear

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y^2 = 0$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + \sin(y) = 0$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + e^y = 0$$

11.1.2. Homogeneity

- **Homogeneous**: No terms with just the **independent** varible of **constant**
- **Nonhomogeneous**: Has term(s) with just the **independent** variable or **constants**

Example

Homogeneous

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y = 0$$

Non-Homogeneous

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y = 1$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y = \sin(t)$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y = e^t$$

11.1.3. Order & Degree

- Order: highest derivative
- Degree: exponent on highest derivative

Example

Order

$$\frac{\mathrm{d}y}{\mathrm{d}x} + y = x \qquad \text{1st order}$$
$$\frac{\mathrm{d}^{2}y}{\mathrm{d}x^{2}} + y = x \qquad \text{2st order}$$

Degree

$$\left(\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)^{3} + y = x \qquad \text{3rd degree}$$

11.1.4. Autonomous

The **independent** variable does not appear explicitly in the equation (rate of change depends only on the dependent variable)

Example

- Autonomous

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y$$

- Non-Autonomous

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y + t$$

11.1.5. Ordinary Differential Equation (ODE)

- **One** independent variable
- Involves **ordinary derivatives** with respect to that variable

11.1.6. Partial Differential Equation (PDE)

- **Multiple** independent variables
- Involves **partial derivatives** with respect to those variables

11.1.7. Summary

- Inputs
  - Equation itself
  - Initial/boundary conditions (optional)

- Outputs
  - Function(s) that satisfy it
- Defined for **continuous variables**

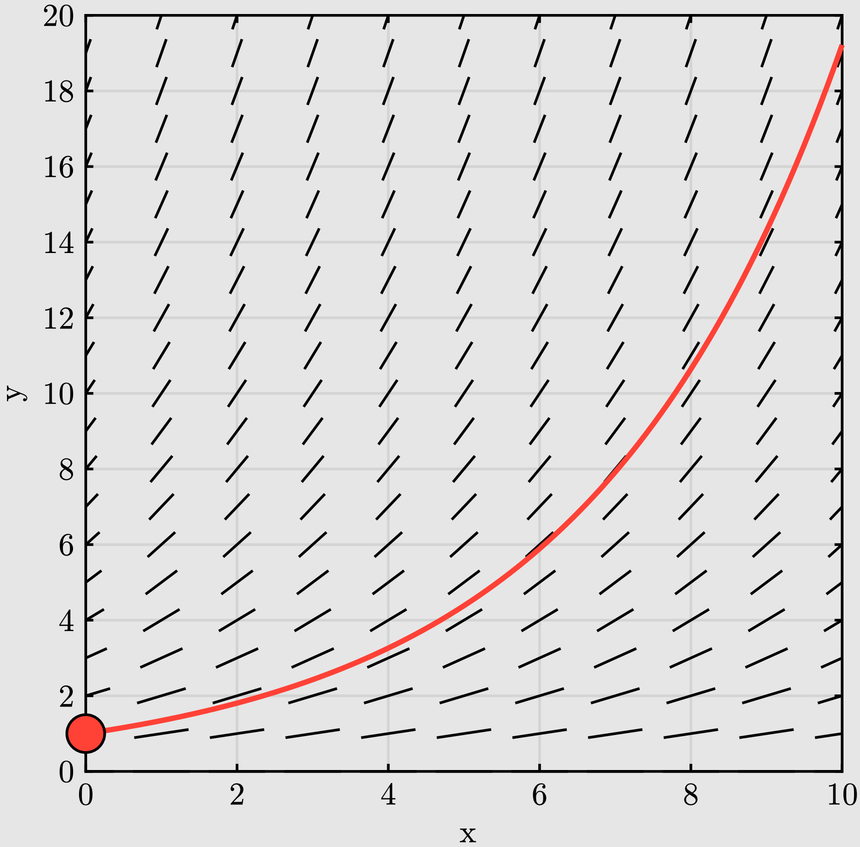
Example

Non-coupled ODE

Exponential Population Growth

$$\frac{\mathrm{d}}{\mathrm{d}t} P(t) = r \cdot P(t)$$

- $P(t)$: population at time t
- $\frac{\mathrm{d}}{\mathrm{d}t} P(t)$: rate of change of the population at time t
- r: growth rate constant
- $r \cdot P(t)$: growth contribution proportional to the current population



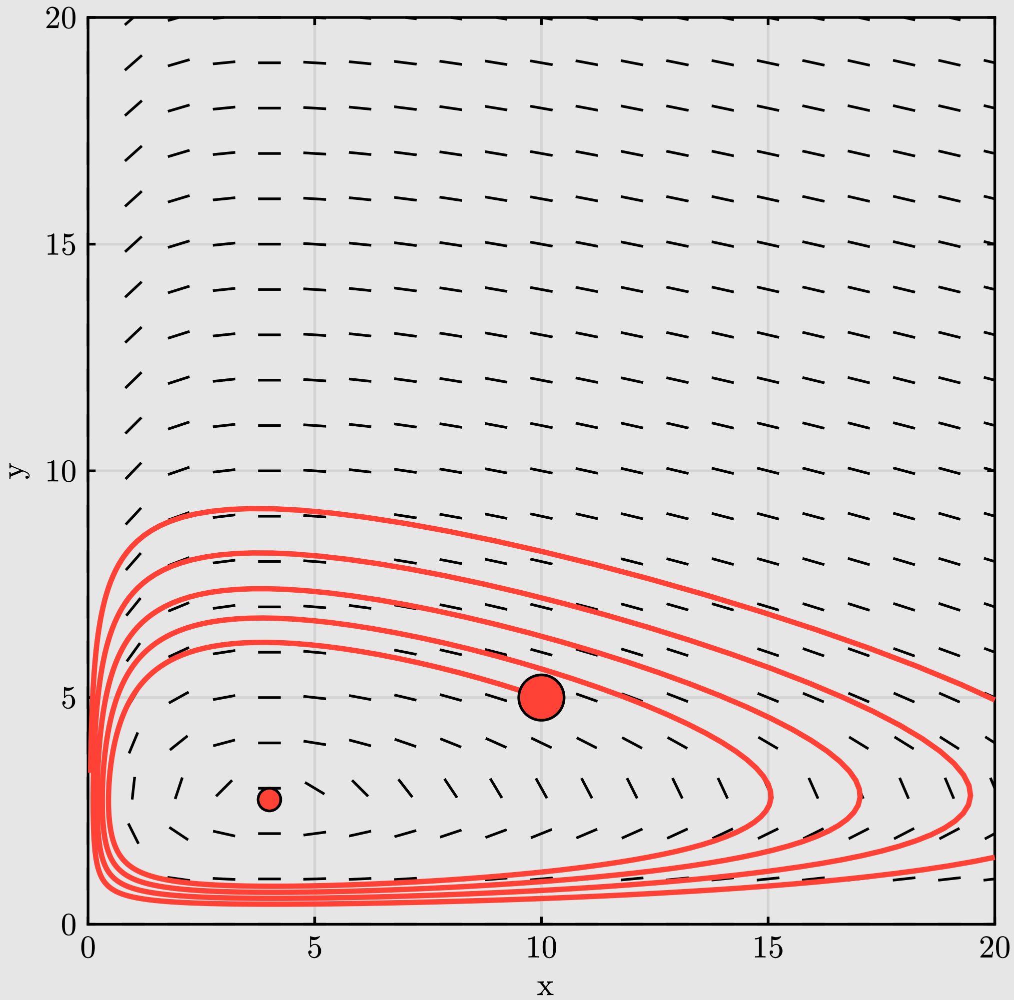
Example

Coupled ODE

Lotka-Volterra ODEs (predator-prey)

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \alpha x(t) - \beta x(t)y(t)$$
$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \delta x(t)y(t) - \gamma y(t)$$

- $x(t)$ = prey population
- $y(t)$ = predator population
- $\frac{\mathrm{d}}{\mathrm{d}t}x(t)$: The rate of change of the prey population at time t
- $\frac{\mathrm{d}}{\mathrm{d}t}y(t)$: The rate of change of the predator population at time t
- α: Prey growth rate — how fast the prey multiply in the absence of predators
- β: Predation rate — how effectively predators consume prey
- δ: Predator growth rate — how much predator population increases when consuming prey
- γ: Predator death rate — how fast predators die without enough food
- $\alpha x(t)$: prey naturally reproduce at a rate proportional to how many exist now
- $-\beta x(t)y(t)$: prey lost to predation, proportional to encounters with predators
- $\delta x(t)y(t)$: predators grow in number based on how much prey they consume
- $-\gamma y(t)$: predators die naturally when there isn't enough food

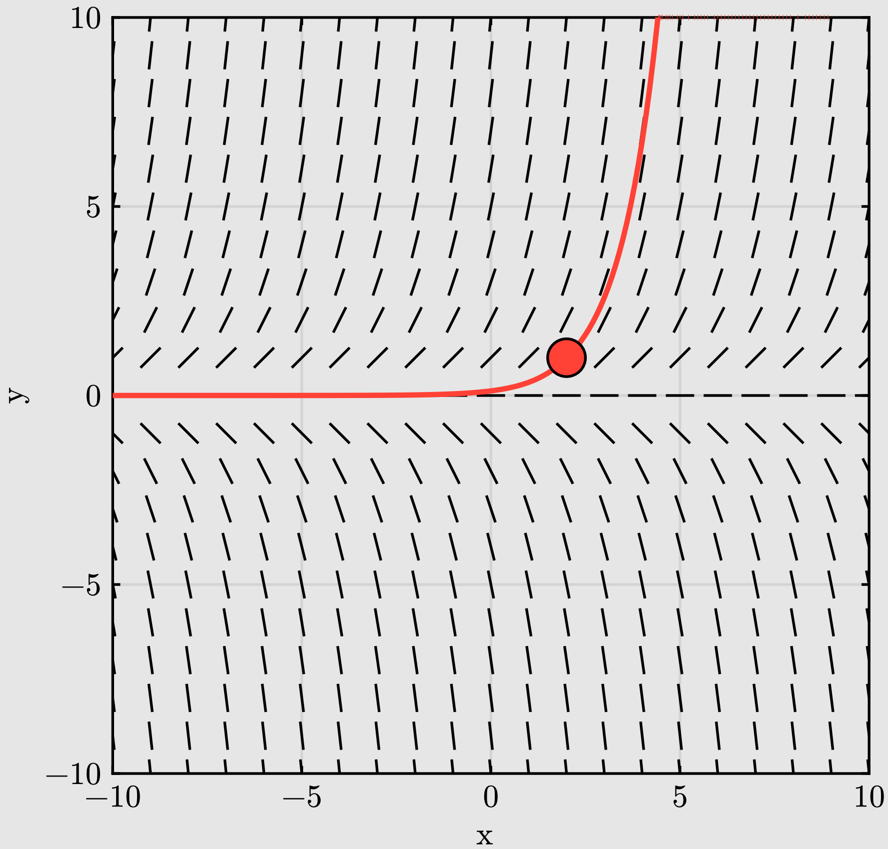


11.2. Slope Field (Direction Field)

Example

IVP (Initial Value Problem)

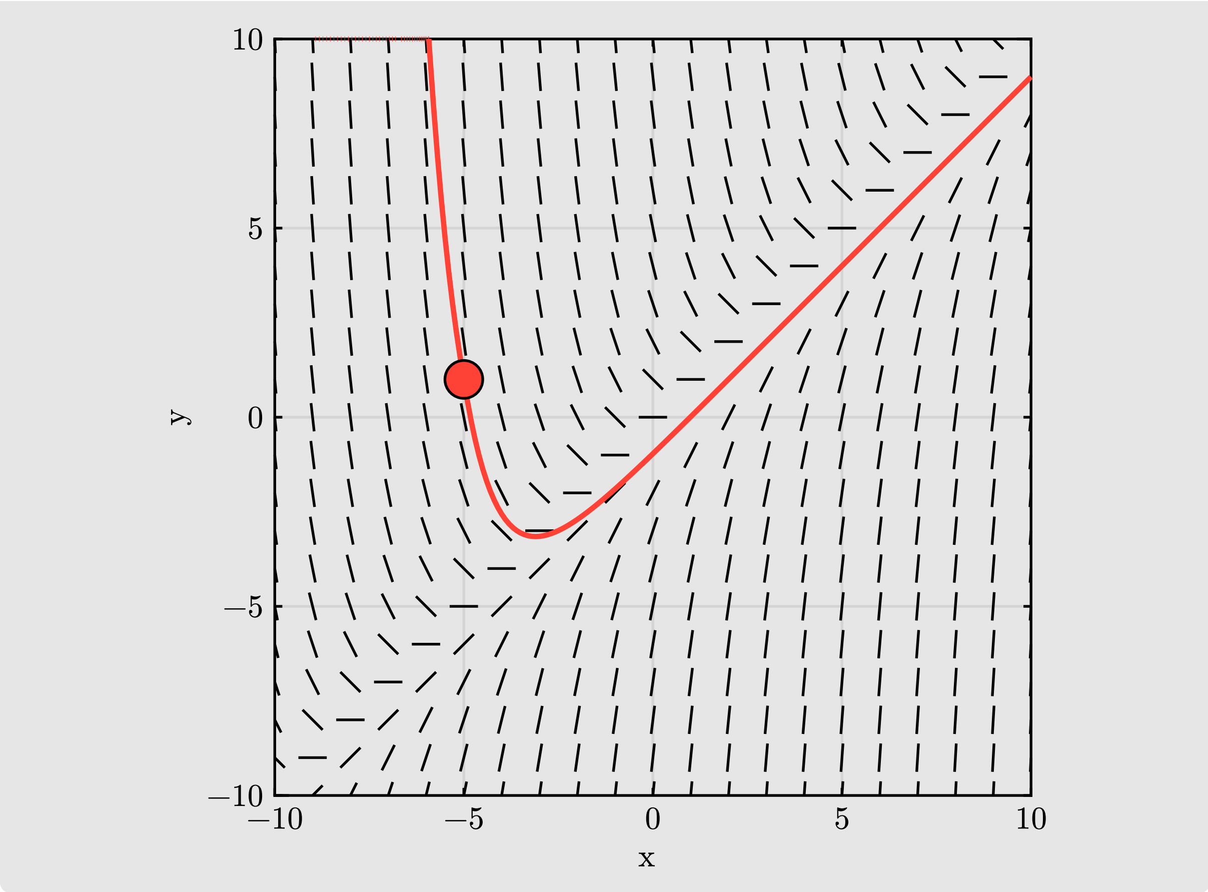
$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}x} = y \\ y(2) = 1 \end{cases}$$



Example

IVP (Initial Value Problem)

$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}x} = x - y \\ y(-5) = 1 \end{cases}$$



10
5
0
−5
−10
y
−10
−5
0
5
10
x

12. Statistics

13. Descriptive Statistics

13.1. Central Tendency

13.1.1. Mean

Sum of all the values divided by the number of values

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

Example

$$[1, 2, 3]$$

$$\bar{x} = \frac{1+2+3}{3} = 150$$

13.1.2. Median

Middle value in a set of values when they are arranged in ascending or descending order

Example

1. **Odd** Number of Values

$$[1, 2, 3]$$

   - **Step 1**: Arrange the Data in Ascending Order

$$[1, 2, 3]$$

   - **Step 2**: Identify the Median

$$\text{Median} = 2$$

2. **Even** Number of Values

$$[1, 2, 3, 4]$$

   - **Step 1**: Arrange the Data in Ascending Order

$$[1, 2, 3, 4]$$

   - **Step 2**: Identify the Median

$$\text{Median} = \frac{2+3}{2} = \frac{5}{2} = 2.5$$

13.1.3. Mode

Value that appears most frequently

Example

$$[1, 1, 2, 3]$$

- **Step 1**: Identify the Most Frequent Number

$$\begin{aligned} 1 &: 2 \\ 2 &: 1 \\ 3 &: 1 \end{aligned}$$

- **Step 2**: Determine the Mode

$$\text{Mode} = 1$$

13.2. Dispersion

13.2.1. Range

$$\text{Range} = \max - \min$$

Example

$$[1, 2, 3, 4, 5]$$

- **Step 1**: Identify the Maximum and Minimum Values

$$\begin{aligned} \max &= 5 \\ \min &= 1 \end{aligned}$$

- **Step 2**: Calculate the Range

$$\text{Range} = 5 - 1 = 4$$

13.2.2. Variance

Quantifies the spread or dispersion of a set of data points in a dataset

- Population

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{n} (x_i - \mu)^2$$

- Sample

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Example

$$[10,\ 12,\ 13,\ 17,\ 20,\ 24]$$

Step 1: Find mean

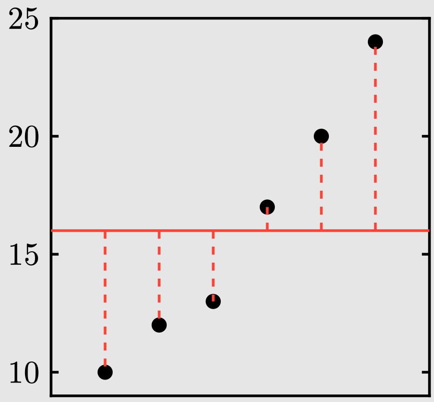
$$\bar{x} = \frac{10 + 12 + 13 + 17 + 20 + 24}{6} = \frac{96}{6} = 16$$

Step 2: Subtract the Mean and Square the result

$$(10 - 16)^2 = -6^2 = 36$$
$$(12 - 16)^2 = -4^2 = 16$$
$$(13 - 16)^2 = -3^2 = 9$$
$$(17 - 16)^2 = 1^2 = 1$$
$$(20 - 16)^2 = 4^2 = 16$$
$$(24 - 16)^2 = 8^2 = 64$$

Step 3: Calculate variance

$$s^2 = \frac{36 + 16 + 9 + 1 + 16 + 64}{6 - 1} = \frac{142}{5} = 28.4$$



13.2.3. Standard deviation

Quantifies the amount of variation or dispersion in a set of data points

- Population

$$\sigma = \sqrt{\sigma^2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

- Sample

$$s = \sqrt{s^2}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

13.2.4. Interquartile Range (IQR)

$$\mathrm{IQR} = \mathrm{Q3} - \mathrm{Q1}$$

Example

$$[1, 2, 3, 4, 5, 6, 7]$$

Step 1: Arrange the Data in Ascending Order

$$[1, 2, 3, 4, 5, 6, 7]$$

Step 2: Find the Quartiles

1. Calculate the Median (Q2)

$$\text{Median (Q2)} = 4$$

2. Find First Quartile (Q1)

Q1 is the median of the first half of the dataset

$$\mathrm{Q1} = 2$$

3. Find Third Quartile (Q3)

Q3 is the median of the second half of the dataset

$$\mathrm{Q3} = 5$$

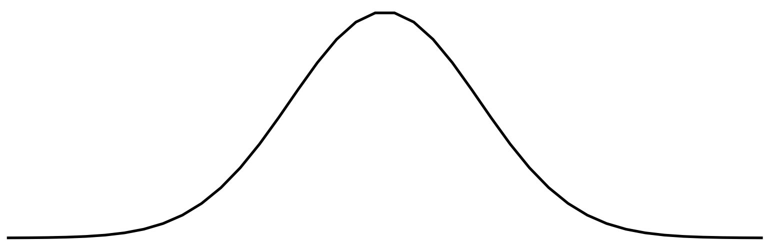
Step 3: Calculate the Interquartile Range (IQR)

$$\mathrm{IQR} = 5 - 2 = 3$$

14. Probability Distributions

14.1. Gaussian (Normal) distribution

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

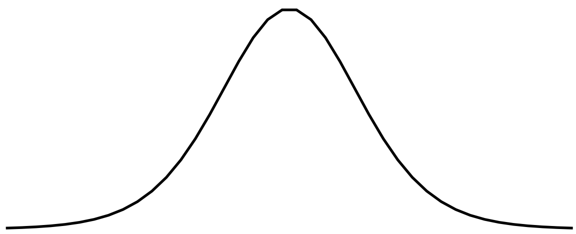


14.2. t-Distribution

$$f(t \mid \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Where:

- t: t-statistic
- ν (or df): degrees of freedom
- Γ: Gamma function (generalizes the factorial function)



Continuous probability distribution for estimating the mean of a normally distributed population in situations where:

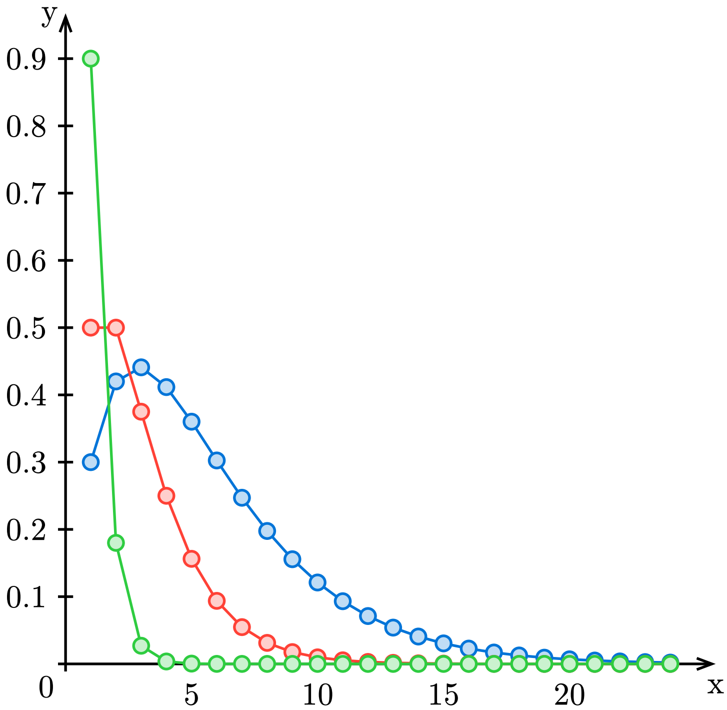
- Sample size is small
- Population standard deviation is unknown

Similar in shape to the normal distribution but has heavier tails, which means it gives more probability to values further from the mean

14.3. Binomial distribution

Discrete probability distribution that describes the number of successes in a fixed number of independent Bernoulli trials, each with the same probability of success

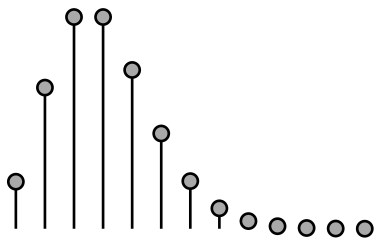
$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$



14.4. Poisson distribution

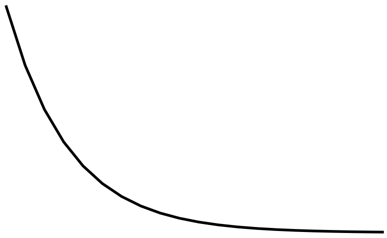
Used to model the number of events that occur within a fixed interval of time or space, given a constant mean rate and assuming that these events occur independently of each other

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$



14.5. Exponential distribution

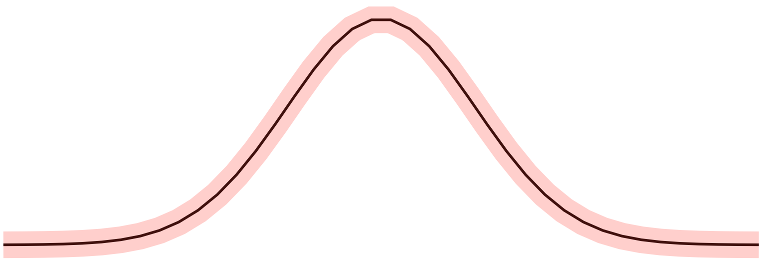
$$f(x \mid \lambda) = \lambda e^{-\lambda x}$$



15. Functions

15.1. PDF (Probability Density Function)

Function that describes the likelihood of a continuous random variable taking on a particular value



Properties:

- The area under the curve of a PDF over the entire range of possible values equals 1
- The PDF itself is non-negative everywhere
- The probability that the variable falls within a certain range is given by the integral of the PDF over that range

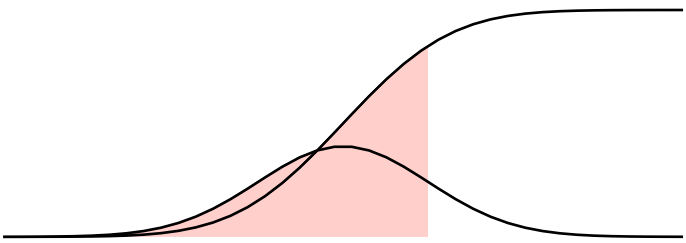
15.2. PMF (Probability Mass Function)

Frequency distribution that provides the probability that a categorical variable takes on each of its possible values

15.3. CDF (Cumulative Distribution Function)

Gives the probability that X will take a value less than or equal to x

$$F(x) = P(X \leq x)$$



1. Categorical

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

2. Continuous

$$F(x) = \sum_{t \leq x} P(X = t)$$

cdf.py

```
from scipy.stats import norm

x = 1
mu = 0
sigma = 1

norm.cdf(x, loc=mu, scale=sigma)
```

15.4. PPF (Percent-Point Function)

Gives the value x such that the probability of a random variable being less than or equal to x is equal to a given probability p

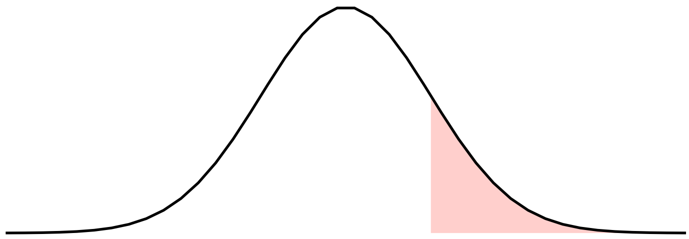
ppf.py

```
from scipy.stats import norm

p = 0.95
mu = 0
sigma = 1

x_value = norm.ppf(p, loc=mu, scale=sigma)
```

15.5. SF (Survival Function)

Probability that a certain event has not occurred by a certain time

$$S(t) = P(T > t)$$



Relationship to PDF:

$$S(t) = 1 - F(t)$$

sf.py

```
from scipy.stats import norm

z = 3.4
mu = 0
sigma = 1

norm.sf(z, loc=mu, scale=sigma)
```

16. Error Metrics

16.1. MAE (Mean Absolute Error)

Average of squared differences between predicted ($\hat{y}_i$) and actual values (y_i)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

16.2. MSE (Mean Squared Error)

Average of squared differences between predicted ($\hat{y}_i$) and actual (y_i) values

$$\mathrm{MSE} = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

16.3. RMSE (Root Mean Squared Error)

square root of the average squared differences between predicted ($\hat{y}_i$) and actual (y_i) values

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

16.4. MAPE (Mean Absolute Percentage Error)

Average percentage difference between predicted ($\hat{y}_i$) and actual (y_i) values

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

16.5. R-squared

Proportion of variance explained by the model

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

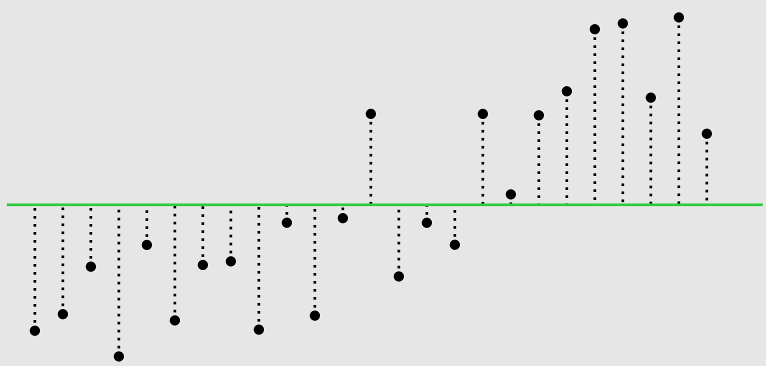
- 1: model explains all the variance in the dependent variable
- 0: model explains none of the variance in the dependent variable

Example

Step 1: Fit the Regression Model

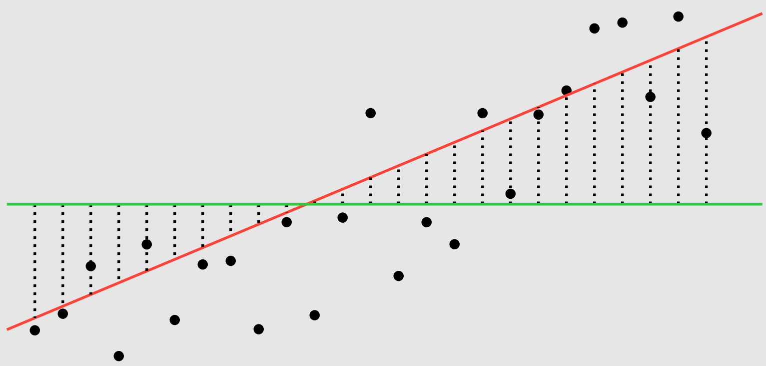
Step 2: Compute **Total Sum of Squares** (SST)

$$\mathrm{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$



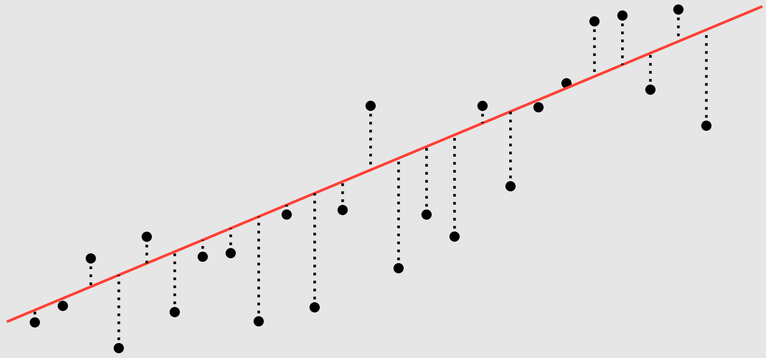
Step 3: Compute **Regression Sum of Squares** (SSR)

$$\mathrm{SSR} = \sum_{i=1}^{n} = (\hat{y}_i - \bar{y})^2$$



Step 4: Compute **Residual Sum of Squares** (SSE)

$$\mathrm{SSE} = \sum_{i=1}^{n} = (y_i - \hat{y}_i)^2$$



Step 5: Calculate R^2

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}}$$

16.6. Adj R-squared

Adjusts the R^2 value based on the number of predictors (penalty for adding non-informative variables)

$$\text{Adj } R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

16.7. MSLE (Mean Squared Logarithmic Error)

When predictions and actual values span several orders of magnitude

$$\text{MSLE} = \frac{1}{n}\sum_{i=1}^{n}(\log(1+y_i) - \log(1+\hat{y}_i))^2$$

16.8. Cross-Entropy Loss (Log Loss)

Binary and multi-class classification

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

17. Hypothesis Testing

17.1. Hypotheses

17.1.1. Null (H_0)

No effect or difference (observed effect is due to sampling variability)

$$H_0 : \mu = \mu_0$$

17.1.2. Alternative (H_1 or H_a)

Presence of an effect or a difference

- Two-Tailed

$$H_1 : \mu \neq \mu_0$$

- Right-Tailed

$$H_1 : \mu > \mu_0$$

- Left-Tailed

$$H_1 : \mu < \mu_0$$

17.2. Error Types

17.2.1. Type I (α)

$$\alpha = P(\text{Reject } H_0 \mid H_0 \text{ is true})$$

17.2.2. Type II (β)

$$\beta = P(\text{Fail to Reject } H_0 \mid H_1 \text{ is true})$$

17.3. t-Tests

17.3.1. One-sample

Tests if the mean of a single sample differs from a known or hypothesized population mean.

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}$$

- $\bar{x}$: sample mean
- μ_0: hypothesized population mean
- s: sample standard deviation
- n: sample size

Example

$$[78, 82, 89]$$

Step 1: State Hypotheses

$$H_0 : \mu = 85 \text{ cm}$$

$$H_1 : \mu \neq 85 \text{ cm}$$

Step 2: Summarize Data

- Sample values:

$$x_1 = 78, x_2 = 82, x_3 = 89$$

- Sample size:

$$n = 3$$

Step 3: Calculate Sample Mean ($\bar{x}$)

$$\bar{x} = \frac{x_1 + x_2 + x_3}{n} = \frac{78 + 82 + 89}{3} = 83$$

Step 4: Calculate Sample Standard Deviation:

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}}$$

- Find the deviations from the mean and square them

$$(x_1 - \bar{x})^2 = (78 - 83)^2 = (-5)^2 = 25$$

$$(x_2 - \bar{x})^2 = (82 - 83)^2 = (-1)^2 = 1$$

$$(x_3 - \bar{x})^2 = (89 - 83)^2 = (6)^2 = 36$$

- Sum of squared deviations

$$\text{SSD} = 25 + 1 + 36$$

- Calculate variance of sample

$$s^2 = \frac{\text{SSD}}{n - 1} = \frac{62}{3 - 1} = \frac{62}{2} = 31$$

- Calculate Sample Standard Deviation

$$s = \sqrt{s^2} = \sqrt{32} = 5.57$$

Step 5: Calculate the Test Statistic

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} = \frac{83 - 85}{\frac{5.57}{\sqrt{3}}} = -\frac{3}{3.22} = -0.62$$

Step 6: Determine the Degrees of Freedom

$$df = n - 1 = 15 - 1 = 14$$

Step 7: Find Critical t-value

- For a two-tailed test at a significance level (α) of 0.05 and 2 degrees of freedom (df)

$$4.303$$

Step 8: Compare the t-Value to the Critical t-Value

- If the absolute value of the test statistic is greater than the critical t-value, reject the null hypothesis.

- If the absolute value of the test statistic is less than the critical t-value, fail to reject the null hypothesis.

Step 8: Find the p-Value

$$t = 4.303 \text{ and } df = 3$$

$$\text{p-value} = 0.58$$

t_test_one_sample.py

```
from scipy import stats
rvs = stats.uniform.rvs(size=50)
stats.ttest_1samp(rvs, popmean=0.5)
```

17.3.2. Independent

Compares the means of two independent samples.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(\frac{s_1^2}{n_1}\right) + \left(\frac{s_2^2}{n_2}\right)}}$$

- $\bar{x}_1$ and $\bar{x}_2$: sample means
- s_1^2 and s_2^2: sample variances
- s_1 and n_2: sample sizes

Example

t_test_independent

```
from scipy import stats
rvs1 = stats.norm.rvs(loc=5, scale=10, size=500)
rvs2 = stats.norm.rvs(loc=5, scale=10, size=500)
stats.ttest_ind(rvs1, rvs2)
```

17.3.3. Paired

$$t = \frac{\bar{D}}{\frac{s_D}{\sqrt{n}}}$$

- $\bar{D}$: mean of the differences between paired observations
- s_D: standard deviation of the differences
- n: number of pairs

17.4. Chi-square tests

17.4.1. Goodness of Fit Test

Compares an **observed** categorical distribution to a **theoretical** categorical distribution.

$$\mathrm{X}^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$$

- o_i: observed frequency in category i
- e_i: expected frequency in category i

chi2_got.py

```
from scipy import stats
f_obs = np.array([43, 52, 54, 40])
f_exp = np.array([47, 47, 47, 47])
stats.chisquare(f_obs=f_obs, f_exp=f_exp)
```

17.4.2. Test of independence

Compares two **observed** categorical distributions.

$$\mathrm{X}^2 = \sum_{i=1}^{k} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

- o_{ij}: observed frequency in cell (i, j)
- e_{ij}: expected frequency in category (i, j) calculated as:

$$E_{ij} = \frac{R_i \times C_j}{N}$$

- R_i: Row total for row i
- C_j: Column total for column j
- N: Total number of observations

17.5. ANOVA (Analysis of Variance)

17.5.1. One-way

Compares the means of three or more groups based on one independent variable

- H_0: $\mu_1 = \mu_2 = ... = \mu_k$
- H_1: At least one μ_i differs from the others

Step 1: Calculate Between-Group Variation (SS_{between})

$$SS_{\text{between}} = \sum_{i=1}^{k} n_i (\bar{X}_i - \bar{X}_{\text{overall}})^2$$

- n_1: Number of observations in group i
- $\bar{X}_i$: Mean of group i

- $\bar{X}_{\text{overall}}$: Overall mean of all groups

Step 2: Calculate Within-Group Variation (SS_{within})

$$SS_{\text{within}} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$

- X_{ij}: Observation j in group i

Step 3: Calculate Total Variation (SS_{total})

$$SS_{\text{total}} = SS_{\text{between}} + SS_{\text{within}}$$

Step 4: Calculate Mean Squares

- Mean Square Between (MS_{between})

$$MS_{\text{between}} = \frac{SS_{\text{between}}}{k-1}$$

- Mean Square Within (MS_{within})

$$MS_{\text{within}} = \frac{SS_{\text{within}}}{N-k}$$

- N: total number of observations
- k: number of groups

Step 5: Calculate the F-statistic

$$F = \frac{MS_{\text{between}}}{MS_{\text{within}}}$$

Step 6: Decision Rule

- Compare the F-statistic to the critical value from the F-distribution table (based on chosen significance level α).
- Alternatively, compare the p-value to the significance level α.
- Reject H_0 if the F-statistic is greater than the critical value or if the p-value is less than α, indicating that at least one group mean is significantly different.

17.5.2. Two-way

18. Regression Analysis:

18.1. Simple linear regression

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

- y: dependent variable
- x: independent variables
- β_0: intercept (value of y when $x = 0$)
- β_1: slope (change in y for a one-unit change in x)
- ε: error term (difference between the actual data points and the predicted values)

Example

Data

x (Hours Studied)	y (Test Score)
1	2
2	3
3	5
4	4
5	6

Step 1: Calculate Means

$$\bar{x} = 3$$

$$\bar{y} = 4$$

Step 2: Calculating Slope β_1

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- The numerator

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = 9$$

- The denominator

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = 10$$

- The slope β_1 is

$$\beta_1 = \frac{9}{10} = 0.9$$

Step 3: Calculate Intercept β_0

$$\beta_1 = \bar{y} - \beta_1 \bar{x} = 1.3$$

Step 4: Calculate p-Value

- Calculate Standard Error of the Slope (SE_{β_1})

$$SE_{\beta_1} = \sqrt{\frac{\sum_{n=1}^{n}(y_i - \hat{y}_i)^2}{(n-2)\cdot\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

- Calculate the Residual Sum of Squares (RSS)

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Calculate the t-statistic for the Slope

$$t = \frac{\beta_1}{SE_{\beta_1}}$$

- Determine Degrees of Freedom

$$\mathrm{df} = n - 2$$

- Look up the p-value corresponding to t with 3 degrees of freedom

Step 5: Calculate R^2 (R^2_{adj})

$$R^2 = \frac{\mathrm{SS_{reg}}}{\mathrm{SS_{total}}}$$

$\mathrm{SS_{reg}}$ (Regression Sum of Squares): sum of the squared differences between the predicted $\hat{y}$ values and the mean of the observed y values.

$$\mathrm{SS_{reg}} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$\mathrm{SS_{total}}$ (Total Sum of Squares): sum of the squared differences between the observed y values and the mean of the observed y values.

$$\mathrm{SS_{total}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Adjusting for the numer of independent variables

$$R^2_{\mathrm{adj}} = 1 - \left(\frac{(1 - R^2)(n-1)}{n - k - 1} \right)$$

- n: number of observations
- k: number of independent variables

18.2. Multiple regression

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_1 + ... + \beta_n x_n + \varepsilon$$

Matrix form:

$$Y = X\beta + \varepsilon$$

Estimating Coefficients (OLS):

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Where:

- $\hat{\beta}$: vector of estimated coefficients

18.3. Logistic Regression

Binary classification (1 or 0, true or false, yes or no) based on one or more predictor variables

Sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

Example

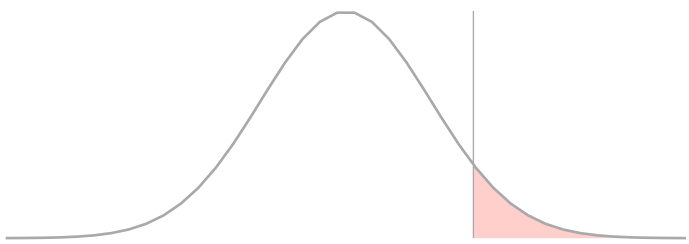
18.4. Model diagnostics

18.4.1. p-Values

Probability of obtaining results at least as extreme as the observed results

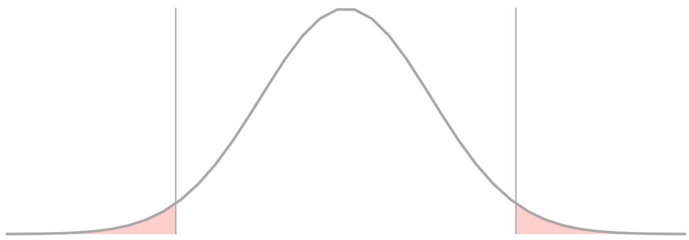
1. One-Tailed

$$p = P(Z > z_{\text{observed}})$$



2. Two-Tailed

$$p = 2 \cdot P(Z > |z_{\text{observed}}|)$$



p_value.py

```
z = 2.1
df = 3
scipy.stats.norm.sf(abs(z), df=df)
```

18.4.2. F-Statistic

$$F = \frac{\mathrm{MSR}}{\mathrm{MSE}}$$

Where:

- Mean Square Regression (MSR):

$$\mathrm{MSR} = \frac{\mathrm{SSR}}{\mathrm{df}_{\mathrm{regression}}}$$

- $\mathrm{df}_{\mathrm{regression}} = p$

Where:

- p: Number of independent
- Mean Square Error (MSE):

$$\mathrm{MSE} = \frac{\mathrm{SSE}}{\mathrm{df}_{\mathrm{error}}}$$

- $\mathrm{df}_{\mathrm{error}} = n - p - 1$

Where:

- p: Number of independent
- n: Number of observations
- 1: Constant (for intercept)

18.4.3. Confidence Intervals (CI)

Range within which we can be confident that the true value (population parameter) lies, based on the sample data

1. Known population standard deviation (σ):

$$CI = \bar{x} \pm z\frac{\sigma}{\sqrt{n}}$$

Where:

- $\bar{x}$: sample mean
- z: z-score corresponding to the desired confidence level
- σ: population standard deviation
- n: sample size

2. Unknown population standard deviation (σ):

$$CI = \bar{x} \pm t\frac{s}{\sqrt{n}}$$

Where:

- $\bar{x}$: sample mean
- t: critical value from t-distribution
- s: sample standard deviation
- n: sample size

19. Correlation

19.1. Pearson

Linear relationships

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \times (y_i - \bar{y})^2}}$$

Interpretation:

- 1: perfect positive linear relationship
- −1: perfect negative linear relationship
- 0: no linear relationship

Example

$$X : [1, 2, 3, 4, 5]$$
$$Y : [2, 4, 6, 8, 10]$$

Step 1: Calculate the Means of X and Y

$$\bar{X} = 3$$
$$\bar{Y} = 6$$

Step 2: Calculate the Differences from the Mean

19.2. Spearman's Rank

Non-linear relationships

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

Where

- ρ: Spearman rank correlation coefficient
- d_i: difference between the ranks of corresponding values
- n: number of observations

Rank: position of a value within a data set when the values are ordered in ascending order

Observation i	X	Rank X	Y	Rank Y	d_i = Rank X - Rank Y	d_i^2
1	3	3	8	5	−2	4
2	1	1	6	3	−2	4
3	4	4	7	4	0	0
4	2	2	4	1	1	1
5	5	5	5	2	3	9

Interpretation

- ρ = 1: Perfect positive correlation
- ρ = −1: Perfect negative correlation
- ρ = 0: No correlation

20. Non-Parametric Statistics

20.1. Mann-Whitney U

Determine whether there is a significant difference between the distributions of two independent samples (used as an alternative to the independent samples t-test when the assumptions of normality are not met)

20.2. Wilcoxon Signed-Rank

Compare two paired samples or to assess whether the median of a single sample is different from a specified value (used as a non-parametric alternative to the paired t-test when the data does not meet the assumptions of normality)

20.3. Kolmogorov-Smirnov

Determine if a sample is drawn from a population with a specific distribution.

It compares the empirical distribution function (EDF) of the sample with the cumulative distribution function (CDF) of the reference distribution.

20.4. Kruskal-Wallis

Determine if there are significant differences between the medians of three or more independent groups (extension of the Mann-Whitney U test, which is used for comparing two groups)

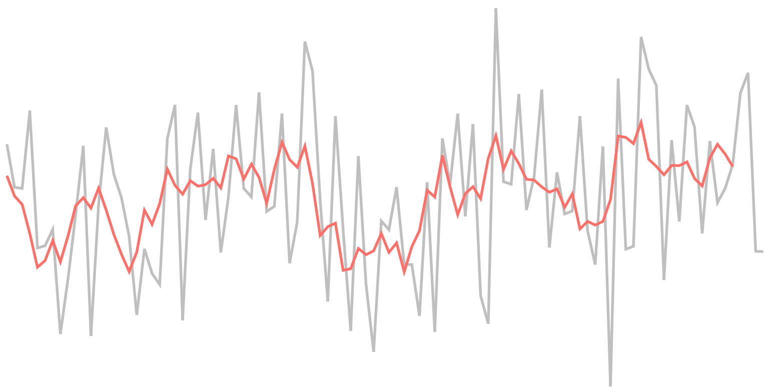
21. Time Series

21.1. SMA (Simple Moving Averages)

For a time series y_t, the SMA of window size k at time t is:

$$s_t = \frac{1}{k}\sum_{i=0}^{k-1} y_{t-i}$$

This is the average of the last k values.



Example

The $k = 3$ day SMA

Data

t	x ($)
1	10
2	12
3	14
4	16
5	18

$$s_3 = \frac{x_{t_1} + x_{t_2} + x_{t_3}}{k} = \frac{10 + 12 + 14}{3} = 12$$

$$s_4 = \frac{x_{t_2} + x_{t_3} + x_{t_4}}{k} = \frac{12 + 14 + 16}{3} = 14$$

$$s_5 = \frac{x_{t_3} + x_{t_4} + x_{t_5}}{k} = \frac{14 + 16 + 18}{3} = 16$$

sma.py

```
pl.col('X').rolling_mean(window_size)
```

21.2. WMA (Weighted Moving Average)

Each data point in the window is assigned a specific weight (usually decrease linearly)

$$\text{WMA} = \frac{\sum_{i=1}^{n}(x_i w_i)}{\sum_{i=1}^{n} w_i}$$

Example

Data

- Day 1: $10
- Day 2: $12
- Day 3: $14
- Day 4: $13
- Day 5: $15

Weights

- 1st most recent: 3
- 2nd most recent: 2
- 3rd most recent: 1

$$\text{WMA for Day 3} = \frac{(14 \times 3) + (12 \times 2) + (10 \times 1)}{3 + 2 + 1} = 12.67$$

$$\text{WMA for Day 4} = \frac{(13 \times 3) + (14 \times 2) + (12 \times 1)}{3 + 2 + 1} = 13.17$$

$$\text{WMA for Day 5} = \frac{(15 \times 3) + (13 \times 2) + (14 \times 1)}{3 + 2 + 1} = 14.7$$

wma.py

21.3. Exponential Smoothing

Gives more weight to recent data points (more responsive to new information)

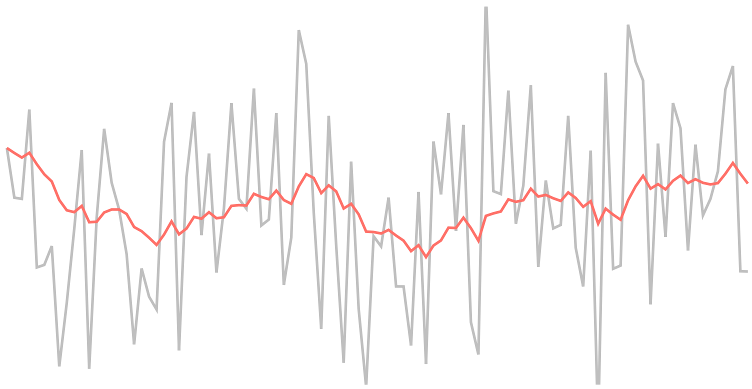
$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}$$

Where:

- s_t: the smoothed value at time t
- x_t: actual value at time t
- α: smoothing factor between 0 and 1

Common way to choose α based on the smoothing window length N:

$$\alpha = \frac{2}{N+1}$$



Example

Data

t	x ($)
1	10
2	12
3	14
4	13
5	15

Calculate Smoothing Factor

$$\alpha = \frac{2}{3+1} = 0.5$$

Calculate SMA for First Value

$$s_3 = \frac{10+12+14}{3}$$

Calculate EMA

$$s_4 = (0.5 \times 13) + (12 \times 0.5) = 12.5$$

$$s_5 = (0.5 \times 15) + (12.5 \times 0.5) = 13.75$$

exp_smooting.py

```
pl.col("X").ewm_mean(span=window_size, adjust=False)
```

21.4. Seasonal Decomposition

- Level: baseline value around which the time series fluctuates
- Trend: ong-term progression or direction of the time series
- Seasonality: regular, repeating patterns or cycles in the time series
- Noise: random fluctuations or irregular variations (cannot be explained by the trend, seasonality, or other components)

Additive Decomposition

When the magnitude of seasonal fluctuations and trend does not change over time

$$Y_t = L_t + T_t + S_t + N_t$$

Where

- L_t: Level at time t
- T_t: Trent at time t
- S_t: Seasonal component at time t
- N_t: Noise (residuals) at time t

Multiplicative Decomposition

When the magnitude of seasonal fluctuations and trend change proportionally over time

$$Y_t = L_t \times T_t \times S_t \times N_t$$

Where

- L_t: Level at time t
- T_t: Trent at time t
- S_t: Seasonal component at time t
- N_t: Noise (residuals) at time t

21.5. ARMA (AutoRegressive Moving Average)

21.6. ARIMA (AutoRegressive Integrated Moving Average)

22. Industrial Engineering & Operations Research

23. Optimization

24. Linear Programming

Function Type	Method	Example
Maximization to Minimization	$\max \quad f(x) \Longleftrightarrow$ $\min \ -f(x)$	$\max \quad 3x+2y \Longleftrightarrow$ $\min -3x-2y$
$\geq$ Constraint to $\leq$	$g_i(x) \geq \quad b_i \Longleftrightarrow$ $-g_i(x) \leq -b_i$	$x+y \geq \quad 5 \Longleftrightarrow$ $-x-y \leq -5$
Equality Constraint	$g_i(x)=b_i \Longleftrightarrow$ $g_i(x) \geq b_i \wedge$ $g_i(x) \leq b_i$	$x+y=4 \Longleftrightarrow$ $x+y \geq 4 \wedge$ $x+y \leq 4$
Slack Variable		$x+y \leq 6$ Add slack variable $s \geq 0$: $x+y+s=6$ s is how much less is being used than the maximum limit of 6
Surplus Variable		$x+y \geq 6$ Add slack variable $s \geq 0$: $x+y-s=6$ s is the amount above the minimum required level 6

25. Integer Programming

Function Type	Method	Example
At **least** k of the items in S	$\sum_{i \in S} x_i \geq k$	Produce at least 3 products: $\sum_{i \in S} x_i \geq 3$
At **most** k of the items in S	$\sum_{i \in S} x_i \leq k$	Open at most 2 facilities: $\sum_{i \in S} x_i \leq 2$
Exactly k of the items in S	$\sum_{i \in S} x_i = k$	Choose exactly one location: $\sum_{i \in S} x_i = 1$
If $x_a=1$, then $x_b=1$	$x_a \leq x_b$	If project A is done, project B must also be done
If $x_a=1$, then $x_b=0$	$x_a+x_b \leq 1$	Projects A and B are mutually exclusive

Function Type	Method	Example
If $x_a = 1$, then $x_b + x_c \leq 0$	$x_b + x_c \leq M(1 - x_a)$	If project A is selected, neither project B nor C can be selected
If $x_j = 0$, then at least k items in S must be selected	$\sum_{i \in S} x_i \geq k(1 - x_j)$	You have a main server represented by x_j. If it is down or not used ($x_j = 0$), then at least k backup servers (in set S) must be activated
At least k of m constraints satisfied	$z_i = \begin{cases} 1 \text{ if } g_i(x) \leq b_i \text{ enforced} \\ 0 \text{ if } g_i(x) \leq b_i \text{ relaxed} \end{cases}$ for $i = 1, \ldots, m$ $g_i(x) - b_i \leq M_i(1 - z_i) \quad \forall i = 1, \ldots, m$ $\sum_{i=1}^{m} z_i \geq k$ $M_i \geq \max_x (g_i(x) - b_i)$	Enforce any k of the m constraints by relaxing up to $m - k$ of them
Fixed (Setup) Costs	$y = \begin{cases} 1 \text{ if is used} \\ 0 \text{ if not used} \end{cases}$ $x \geq 0$ $x \leq M \cdot y$	If facility is used ($y = 1$), then allow production up to M; else $x = 0$
Balanced Flow	$\sum_{\substack{i \in V \\ i \neq k}} x_{ik} = d_k$ $\sum_{\substack{i \in V \\ i \neq k}} x_{jk} = d_k$	Each node k receives and sends exactly d_k units of flow. If $d_k = 1$, this is a TSP or assignment
Unbalanced Flow	$\sum_{\substack{i \in V \\ i \neq k}} x_{ik} - \sum_{\substack{i \in V \\ i \neq k}} x_{jk} = b_k$	Each node k has net inflow minus outflow equal to b_k: • $b_k > 0$: demand • $b_k < 0$: supply • $b_k = 0$: transshipment
Subtours (MTZ)	$1 \leq u_i \leq n - 1 \quad \forall i = 2, \ldots, n$ $u_i - u_j + (n-1)x_{ij} \leq n - 2$ $\forall i \neq j, i, j \in \{2, \ldots, n\}$	
Subtours (SEC)	$\sum_{\substack{i \in S, j \in S \\ j \neq j}} x_{ij} \leq \lvert S\rvert - 1 \quad \forall S \subseteq V, \lvert S\rvert \geq 2$	For every subset S of the nodes V that has at least 2 nodes, the total number of arcs (or paths) within that subset — from node i to node j, where $i \neq j$ must be less than or equal to $\lvert S\rvert - 1$

26. Non-Linear Programming

Function Type	Example	Linearization Method	Example
Binary × Binary Constraint	$z = x \cdot y$ $x, y \in \{0,1\}$	$z \leq x$ $z \leq y$ $z \geq x + y - 1$ $x \in \{0,1\}$	$z = 1 \iff$ $x = y = 1$
Binary × Continuous Constraint	$z = x \cdot y$ $x \in \{0,1\}$ $y \in [L, U]$	$z \leq Ux$ $z \geq Lx$ $z \leq y - L(1 - x)$ $z \geq y - U(1 - x)$	
Continuous × Continuous Constraint	$z = x \cdot y$ $x \in [L_x, U_x]$ $y \in [L_y, U_y]$	$z \geq L_x y + L_y x - L_x L_y$ $z \geq U_x y + U_y x - U_x U_y$ $z \geq L_x y + U_y x - L_x U_y$ $z \geq U_x y + L_y x - U_x L_y$	
Absolute Value	$z = \lvert x \rvert$	$z \geq x$ $z \geq -x$	$z = \lvert x_1 - x_2 \rvert$ $\iff$ $z \geq x_1 - x_2$ $z \geq x_2 - x_1$
Max Constraint	$y \geq \max(x_1, ..., x_n)$	$y \geq x_1 \quad \forall i = 1, ..., n$	$y + x_1 + 3 \geq \max(x_1 - x_3, 2x_2 + 4)$ $\iff$ $\begin{cases} y + x_1 + 3 \geq x_1 - x_3 \\ y + x_1 + 3 \geq 2x_2 + 4 \end{cases}$
Min Constraint	$y \leq \min(x_1, ..., x_n)$	$y \leq x_1 \quad \forall i = 1, ..., n$	$y + x_1 \leq \min(x_1 - x_3, 2x_2 + 4, 0)$ $\iff$ $\begin{cases} y + x_1 \leq x_1 - x_3 \\ y + x_1 \leq 2x_2 + 4 \\ y + x_1 \leq 0 \end{cases}$
Max Min Objective	$\max \min(x_1, ..., x_n)$	$y = \min(x_1, ..., x_n)$ $y \leq x_i \quad \forall i = 1, ..., n$	
Min Max Objective	$\min \max(x_1, ..., x_n)$	$y = \max(x_1, ..., x_n)$ $y \geq x_i \quad \forall i = 1, ..., n$	
Min Min Objective	$\min \min(x_1, ..., x_n)$		
Max Max Objective	$\max \max(x_1, ..., x_n)$		

27. Algorithms

Algorithm	Update Rule	
Gradient Descent	$x_{k+1} = x_k - \alpha \nabla f(x_k)$	
Newton's Method	$x_{k+1} = x_k - [\nabla^2 f(k)]^{-1} \nabla f(x_k)$	

Newton's Method

Approximation	Taylor Expansion	Newton Update
Linear (1D)	$f'(x_{k+1}) \approx f'(x_k) + f''(x_k)(x_{k+1} - x_k)$	$x_{k+1} = x_k - \dfrac{f'(x_k)}{f''(x_k)}$
Linear (nD)	$\nabla f(\mathbf{x}_{k+1}) \approx \nabla f(\mathbf{x}_k) + H_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$	$\mathbf{x_{k+1}} = \mathbf{x}_k - H_k^{-1} \nabla f(\mathbf{x}_k)$
Quadratic (1D)	$f_Q(x) =$ $f(x_k) + f'(x_k)(x - x_k) + \dfrac{1}{2} f''(x_k)(x - x_k)^2$	$x_{k+1} = x_k - \dfrac{f'(x_k)}{f''(x_k)}$
Quadratic (nD)	$m_k(\mathbf{p}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \dfrac{1}{2} \mathbf{p}^T H_k \mathbf{p}$ $\mathbf{p} = \mathbf{x} - \mathbf{x}_k$	$\mathbf{x_{k+1}} = \mathbf{x}_k - H_k^{-1} \nabla f(\mathbf{x}_k)$

27.1. Mathematical Programs

27.1.1. Formulation

$$
\begin{aligned}
\min \quad & f(x_1, x_2, ..., x_n) & & (\text{objective function}) \\
s.t. \quad & g_i(x_1, x_2, ..., x_n) \leq b_i & \forall i \in 1, ..., m \quad & (\text{constraints}) \\
& x_j \in \mathbb{R} & \forall j \in 1, ..., n \quad & (\text{decision variables})
\end{aligned}
$$

- m: Number of constraints
- n: Number of variables
- $x_1, x_2, ..., x_n$: decision variables
- $f : \mathbb{R}^n \to \mathbb{R}$
- $g_i : \mathbb{R}^n \to \mathbb{R}$

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} c_j x_j \\
s.t. \quad & \sum_{j=1}^{n} A_{ij} x_j \leq b_i \quad \forall i = 1, ..., m
\end{aligned}
$$

- A_{ij}: constraint coefficients
- b_i: right-hand-side values (RHS)
- c_j: objective coefficients

Vector

$$
\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & a_i^T x \leq b_i \quad \forall i = 1, ..., m
\end{aligned}
$$

- $a_i \in \mathbb{R}^n$
- $b_i \in \mathbb{R}$
- $c \in \mathbb{R}^n$
- $x \in \mathbb{R}^n$

Matrix

$$
\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & Ax \leq b
\end{aligned}
$$

- $A \in R^{m \times n}$
- $b \in \mathbb{R}^m$

$$
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}
$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 & a_{12}x_2 & \dots & a_{1n}x_n \\ a_{21}x_1 & a_{22}x_2 & \dots & a_{2n}x_n \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}x_1 & a_{m2}x_2 & \dots & a_{mn}x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$c^T x = [c_1, c_2, \dots, c_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{j=1}^{n} c_j x_j$$

27.1.2. Transformation

- **Min and Max**

$$\max f(x) \Leftrightarrow \min -f(x)$$

- **$=$ and $\geq$**

$$g_i(x) \geq b_i \Leftrightarrow -g_i(x) \leq -b_i$$

$$g_i(x) = b_i \Leftrightarrow g_i(x) \geq b_i \wedge g_i(x) \leq b_i$$

- **E.g.,**

$$\begin{array}{lrcrcr} \max & x_1 & - & x_2 & & \\ s.t. & -2x_1 & + & x_2 & \geq & -3 \\ & x_1 & + & 4x_2 & = & 5 \end{array} \Leftrightarrow \begin{array}{lrcrcr} \min & -x_1 & + & x_2 & & \\ s.t. & 2x_1 & - & x_2 & \leq & 3 \\ & x_1 & + & 4x_2 & \leq & 5 \\ & -x_1 & + & 4x_2 & \leq & -5 \end{array}$$

27.1.3. Constraints

- **Sign Constraints**

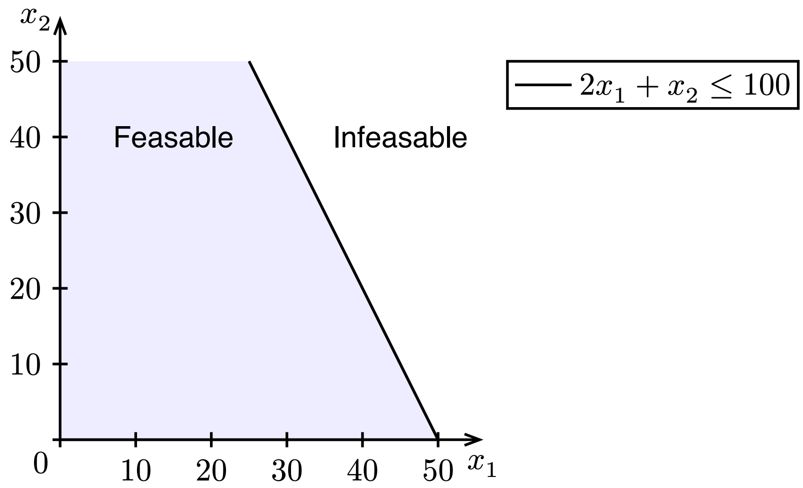
$$x_i \geq 0 \text{ or } x_i \leq 0$$

- **Functional Constraints**

$$\sum_{i=1}^{n} c_i x_i \leq b_i$$

27.1.4. Feasable Solutions

- Feasable solution: Satisfies all constants
- Infeasable solution: violates at least one constraint



27.1.5. Feasable Region and Optimal Solution

- Feasable Region: Set of feasable solutions
- Optimal Solution: Attains largest (maximization) or smallest (minimization) objective value

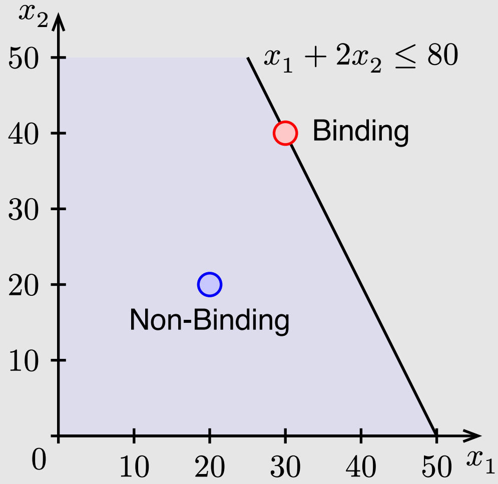
27.1.6. Binding Constraint

Let $g(\cdot) \leq b$ be an inequality constraint and $|(x)$ be a solution. $g(\cdot) \leq b$ is binding at $|(x)$ if $g(|(x)) = b$

Example

$2x_1 + x_2 \leq 100$ is **binding** (**active**) at the point $(x_1, x_2) = (30, 40)$

$2x_1 + x_2 \leq 100$ is **non-binding** (**inactive**) at the point $(x_1, x_2) = (20, 20)$



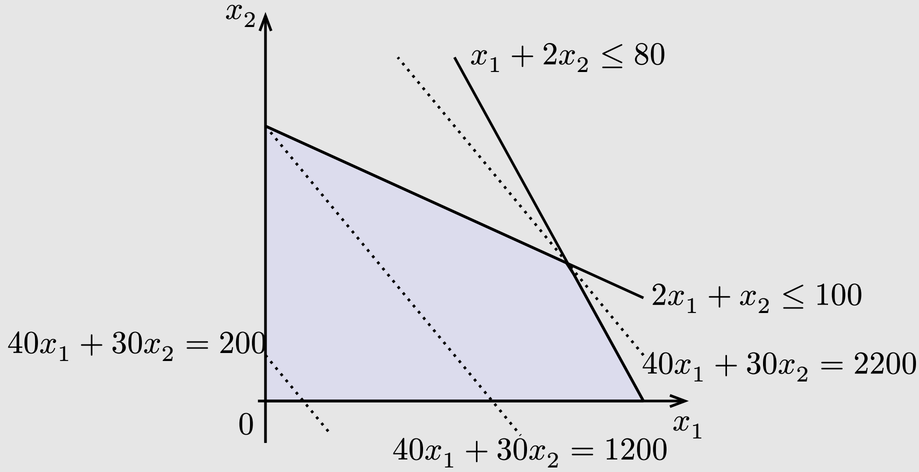
27.1.7. Strict and Weak Inequalities

- Strict: The two sides **cannot be equal** (e.g., $x_1 + x_2 > 5$)
- Weak: The two sides **can be equal** (e.g., $x_1 + x_2 \geq 5$)

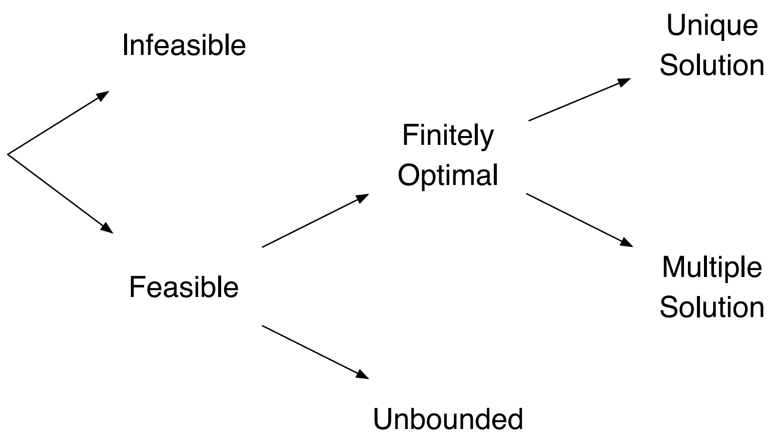
27.1.8. Graphical Approach

Example

$$\begin{array}{rcccccc} \max & 40x_1 & + & 30x_2 & & \\ s.t. & 2\ x_1 & + & x_2 & \leq & 100 \\ & x_1 & + & 2x_2 & \leq & 80 \end{array}$$



27.1.9. Types of LP



27.1.10. LP Formulation

Example

Production

We produce desks and tables

- Producing a desk requires 3 units of wood, 1 hour of labor and 50 minutes of machine time
- Producing a table requires 5 units of wood, 2 hours of labor and 20 minutes of machine time

For each day, we have

- 200 workers that each work 8 hours
- 50 machines that each run for 16 hours
- A supply of 3600 units of wood

Desks and tables are sold \$700 and \$900 per unit, respectively

Everything that is produced is sold

1. Define Variables

x_1: number of desks produced per day

x_2: number of tables produced per day

2. Formulate Objective Function

$$\max \quad 700x_1 + 900x_2$$

3. Formulate Constraints

Resource	Consumption per	Total supply

	Desk	Table	
Wood	3 units	5 units	3600 units
Labor	1 hours	2 hours	200 workers $\times$ 8 hours = 1600 hours
Machine time	50 minutes	20 minutes	50 machines $\times$ 16 hours = 800 hours

$$\begin{array}{rcrcl} 3x_1 & + & 5x_2 & \leq & 3600 \\ x_1 & + & 2x_2 & \leq & 1600 \\ 50x_1 & + & 20x_2 & \leq & 48000 \end{array}$$

4. Complete Formulation

$$\begin{array}{lrcrcll} \max & 700x_1 & + & 900x_2 & & & \\ s.t. & 3x_1 & + & 5x_2 & \leq & 3600 & \text{(wood)} \\ & x_1 & + & 2x_2 & \leq & 1600 & \text{(labor)} \\ & 50x_1 & + & 20x_2 & \leq & 48000 & \text{(machine)} \\ & x_1 & & & \geq & 0 & \\ & & & x_2 & \geq & 0 & \end{array}$$

Compact Formulation

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} P_j x_j \\ s.t. \quad & \sum_{j=1}^{n} A_{ij} x_j \leq R_i \quad \forall i = 1, ..., m \\ & x_j \geq 0 \quad \forall j = 1, ..., n \end{aligned}$$

Where:

- n: number of products
- m: number of resources
- j: indices of products
- i: indices of resources
- P_j: unit sale price of product j
- R_i: supply limit of resource i
- A_{ij}: unit of resource i to produce one unit of product j
- $i = 1, ..., m$
- $j = 1, ..., n$
- x_j: production quantity for product j, $j = 1, ..., n$

model.py

```
from pyomo.environ import *
from pyomo.dataportal import DataPortal

model = AbstractModel()

# Sets
model.Products = Set()
model.Resources = Set()

# Parameters
model.Profit = Param(model.Products)
model.Supply = Param(model.Resources)
model.Consumption = Param(model.Resources, model.Products)

# Variables
model.x = Var(model.Products, domain=NonNegativeReals)

# Objective: Maximize profit
def objective_rule(model):
    return sum(model.Profit[j] * model.x[j] for j in model.Products)
model.OBJ = Objective(rule=objective_rule, sense=maximize)

# Constraints: Do not exceed resource supply
def constraint_rule(model, i):
    return sum(model.Consumption[i, j] * model.x[j] for j in model.Products)
<= model.Supply[i]
model.ResourceConstraint = Constraint(model.Resources, rule=constraint_rule)

# Load data from .dat file
data = DataPortal()
data.load(filename='data.dat', model=model)

# Create an instance of the model
instance = model.create_instance(data)

# Create solver
solver = SolverFactory('glpk')
solver.options['tmlim'] = 60

# Solve with solver timeout (optional)
results = solver.solve(instance, tee=True)

# Display results
instance.display()
```

data.dat

```
set Products := Desk Table ;
set Resources := Wood Labor Machine ;
```

```
param Profit :=
Desk   700
Table  900 ;

param Supply :=
Wood    3600
Labor   1600
Machine 48000 ;

param Consumption:
          Desk Table :=
Wood        3     5
Labor       1     2
Machine    50    20 ;
```

Output:

```
  Variables:
  x : Size=2, Index=Products
      Key   : Lower : Value            : Upper : Fixed : Stale : Domain
      Desk :     0 : 884.210526315789 :  None : False : False :
NonNegativeReals
      Table :     0 : 189.473684210526 :  None : False : False :
NonNegativeReals

Objectives:
  OBJ : Size=1, Index=None, Active=True
      Key  : Active : Value
      None :   True : 789473.6842105257

Constraints:
  ResourceConstraint : Size=3
      Key     : Lower : Body               : Upper
        Labor :  None : 1263.157894736841 :  1600.0
      Machine :  None : 47999.99999999997 : 48000.0
         Wood :  None : 3599.999999999997 :  3600.0
```

)

Example

Multi Period (Inventory)

We produce and sell products

For the coming 4 days, the demand will be

- Days 1: 100
- Days 2: 150

- Days 3: 200
- Days 4: 170

The unit production costs are different for different days

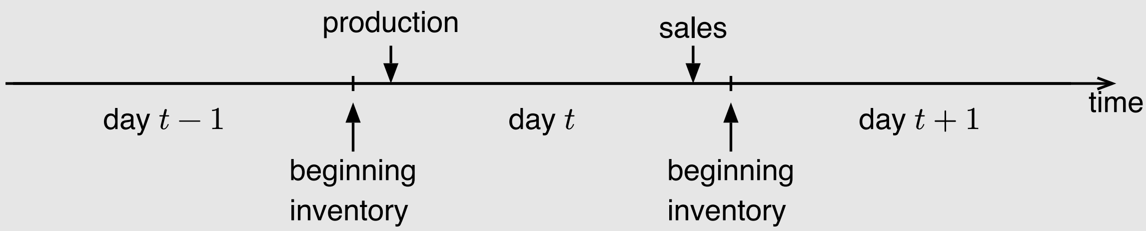
- Days 1: \$9
- Days 2: \$12
- Days 3: \$10
- Days 4: \$12

Prices are all fixed. So maximizing profit is equivalent to minimizing costs.

We may store products and sell them later. Inventory cost is \$1 per unit per day.

E.g., producing 620 units on day 1 to fulfill all demands:

$$\$9 \times 620 + \$1 \times 150 + \$2 \times 200 + \$3 \times 170 = \$6,640$$



$$I_{t-1} + x_t - d_t = I_t$$

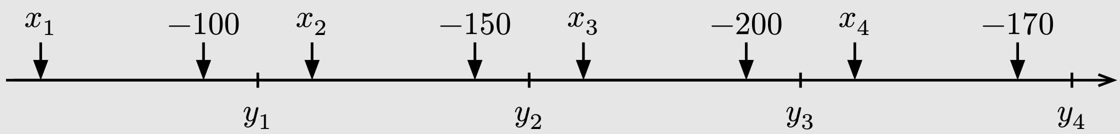
- I_{t-1}: inevntory at the end of day $t-1$ (i.e., beginning inventory for day t)
- x_t: units produced on day t
- d_t: demand (or sales) on dat t
- I_t: inventory at the end of day t

Inventory costs are calculated according to **ending inventory**

- x_t: production quantity of day t, $t = 1, ..., 4$
- y_t: **ending** inventory of day t, $t = 1, ..., 4$

Objective function:

$$\min \quad 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4$$



Inventory balancing constraints:

- Day 1: $x_1 - 100 = y_1$

- Day 2: $y_1 + x_2 - 150 = y_2$
- Day 3: $y_2 + x_3 - 200 = y_3$
- Day 4: $y_3 + x_4 - 170 = y_4$

Demand fulfillment constraints:

- $x_1 \geq 100$
- $y_1 + x_2 \geq 150$
- $y_2 + x_3 \geq 200$
- $y_3 + x_4 \geq 170$

Non-negativity constraints:

- $y_t \geq 0$
- $x_t \geq 0$

Complete Formulation

$$\begin{aligned}
\min \quad & 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4 \\
s.t. \quad & x_1 - 100 = y_1 \\
& y_1 + x_2 - 150 = y_2 \\
& y_2 + x_3 - 200 = y_3 \\
& y_3 + x_4 - 170 = y_4 \\
& x_1 \geq 100 \\
& y_1 + x_2 \geq 150 \\
& y_2 + x_3 \geq 200 \\
& y_3 + x_4 \geq 170 \\
& x_t, y_t \geq 0 \quad \forall t = 1, ..., 4
\end{aligned}$$

Simplification (Redundant Constraints)

- Inventory balancing & non-negativity imply fulfillment
  - E.g., in day 1, $x_1 - 100 = y_1$ and $y_1 \geq 0$ means $x_1 \geq 100$

$$\begin{aligned}
\min \quad & 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4 \\
s.t. \quad & x_1 - 100 = y_1 \\
& y_1 + x_2 - 150 = y_2 \\
& y_2 + x_3 - 200 = y_3 \\
& y_3 + x_4 - 170 = y_4 \\
& x_t, y_t \geq 0 \quad \forall t = 1, ..., 4
\end{aligned}$$

- No need to have ending inventory in period 4 (costly but useless)

$$
\begin{aligned}
\min \quad & 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4 \\
s.t. \quad & x_1 - 100 = y_1 \\
& y_1 + x_2 - 150 = y_2 \\
& y_2 + x_3 - 200 = y_3 \\
& y_3 + x_4 - 170 = 0 \\
& x_t \geq 0 \quad \forall t = 1, ..., 4 \\
& y_t \geq 0 \quad \forall t = 1, ..., 3
\end{aligned}
$$

Compact Formulation

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{4} (C_t x_t + y_t) \\
s.t. \quad & y_t - 1 + x_t - D_t = y_t \quad \forall t = 1, ..., 4 \\
& y_0 = 0 \\
& x_t, y_t \geq 0 \quad \forall t = 1, ..., 4
\end{aligned}
$$

Where:

- D_t: demand on day t
- y_t: ending inventory of day t
- C_t: unit production cost on day t

Example

Personnel Scheduling

Scheduling employees

Each employee must work for 5 consecutive days and then take 2 consectutive rest days

Number of employees required for each day

Mon	Tue	Wed	Thu	Fri	Sat	Sun
110	80	150	30	70	160	120

Seven shifts

- Mon to Fri
- Tue to Sat
- …

Minimize number of employees hired

Let x_i be the number of employees assigned to work from day i for 5 consecutive days

Objective function:

$$\min x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

Demand Fullfilment Constraints

- 110 employeeds needed Monday

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq 110$$

An employee that works on Tues (t_2) or Wed (t_3) does not work on Mon (t_1)

- 80 employees needed Tuesday

$$x_1 + x_2 + x_5 + x_6 + x_7 \geq 80$$

An employee that works on Wed (t_3) or Thu (t_4) does not work on Tue (t_2)

- 120 employees needed Sunday

$$x_3 + x_4 + x_5 + x_6 + x_7 \geq 120$$

An employee that works on Mon (t_1) or Tue (t_2) does not work on Sun (t_7)

Non-Nagativity Constraints

$$x_i \geq 0 \quad \forall i = i, ..., 7$$

Complete Formulation

$$\begin{array}{lccccccccccccccl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & & \\
s.t. & x_1 & + & & & & & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & 110 \\
 & x_1 & + & x_2 & + & & & & & x_5 & + & x_6 & + & x_7 & \geq & 80 \\
 & x_1 & + & x_2 & + & x_3 & + & & & & & x_6 & + & x_7 & \geq & 150 \\
 & x_1 & + & x_2 & + & x_3 & + & x_4 & + & & & & & x_7 & \geq & 30 \\
 & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & & & & & \geq & 70 \\
 & & & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & & & \geq & 160 \\
 & & & & & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & 120
\end{array}$$

$$x_i \geq 0 \forall i = 1, ..., 7$$

model.py

```python
from pyomo.environ import *
from pyomo.dataportal import DataPortal

model = AbstractModel()

# Sets
model.Days = Set(ordered=True)           # Days 1..7 (Mon..Sun)
model.Shifts = Set(ordered=True)         # Shifts starting on days 1..7

# Parameters
model.Demand = Param(model.Days)         # Daily staffing requirements
model.Cover = Param(model.Days, model.Shifts, within=Binary)  # Coverage
matrix (1 if shift j covers day i)

# Variables
model.x = Var(model.Shifts, domain=NonNegativeReals)
```

```python
# Objective: Minimize total employees hired
def obj_rule(model):
    return sum(model.x[s] for s in model.Shifts)
model.OBJ = Objective(rule=obj_rule, sense=minimize)

# Constraints: Cover daily demand
def demand_rule(model, d):
    return sum(model.Cover[d, s] * model.x[s] for s in model.Shifts) >=
model.Demand[d]
model.DemandConstraint = Constraint(model.Days, rule=demand_rule)

# Load data from .dat file
data = DataPortal()
data.load(filename='data.dat', model=model)

# Create an instance of the model
instance = model.create_instance(data)

# Create solver
solver = SolverFactory('glpk')
solver.options['tmlim'] = 60

# Solve with solver timeout (optional)
results = solver.solve(instance, tee=True)

# Display results
instance.display()
```

data.dat

```
set Days := Mon Tue Wed Thu Fri Sat Sun ;
set Shifts := s1 s2 s3 s4 s5 s6 s7 ;

param Demand :=
Mon 110
Tue 80
Wed 150
Thu 30
Fri 70
Sat 160
Sun 120 ;

# Each shift covers 5 consecutive days starting on the shift's day
# 1 if shift s_j covers day d_i, 0 otherwise

param Cover:
        s1 s2 s3 s4 s5 s6 s7 :=
Mon     1  0  0  1  1  1  1
Tue     1  1  0  0  1  1  1
Wed     1  1  1  0  0  1  1
```

```
Thu     1  1  1  1  0  0  1
Fri     1  1  1  1  1  0  0
Sat     0  1  1  1  1  1  0
Sun     0  0  1  1  1  1  1 ;
```

Output:

```
  Variables:
  x : Size=7, Index=Shifts
      Key : Lower : Value              : Upper : Fixed : Stale : Domain
        s1 :     0 : 3.33333333333333 :  None : False : False :
NonNegativeReals
        s2 :     0 :             40.0 :  None : False : False :
NonNegativeReals
        s3 :     0 : 13.3333333333333 :  None : False : False :
NonNegativeReals
        s4 :     0 :              0.0 :  None : False : False :
NonNegativeReals
        s5 :     0 : 13.3333333333333 :  None : False : False :
NonNegativeReals
        s6 :     0 : 93.3333333333333 :  None : False : False :
NonNegativeReals
        s7 :     0 :              0.0 :  None : False : False :
NonNegativeReals

Objectives:
  OBJ : Size=1, Index=None, Active=True
      Key  : Active : Value
      None :   True : 163.33333333333323

Constraints:
  DemandConstraint : Size=7
      Key : Lower : Body               : Upper
      Fri :  70.0 :  69.99999999999993 :  None
      Mon : 110.0 : 109.99999999999993 :  None
      Sat : 160.0 :  159.9999999999999 :  None
      Sun : 120.0 :  119.9999999999999 :  None
      Thu :  30.0 :  56.66666666666663 :  None
      Tue :  80.0 : 149.99999999999994 :  None
      Wed : 150.0 : 149.99999999999994 :  None
```

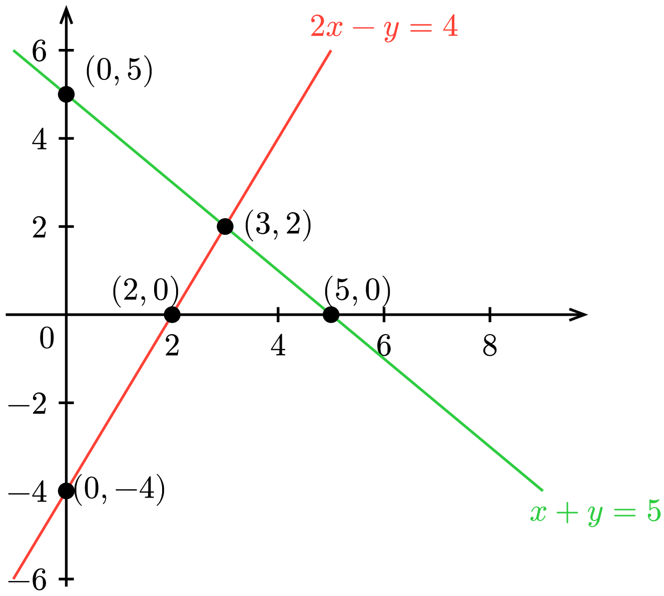
27.2. Simplex Method

27.2.1. Linear Algebra Review

$$x + y = 5$$
$$2x - y = 4$$

Row View

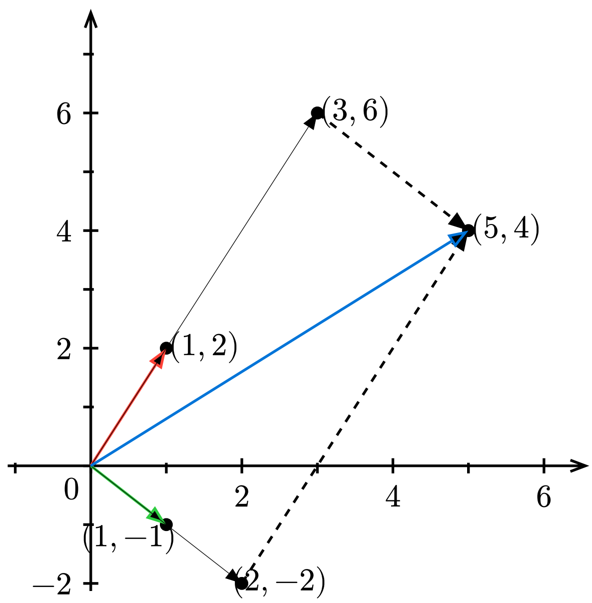
Solution: Intersection of n lines or (hyper)planes



Column View

Solution: Combination of LHS column vectors to form the RHS vector

$$x\begin{bmatrix}1\\2\end{bmatrix} + y\begin{bmatrix}1\\-1\end{bmatrix} = \begin{bmatrix}5\\4\end{bmatrix}$$



A linear system is **singluar** if there is no unique solution

- Row view: the n (hyper)planes do not intersect at exactly one point
- Column view: the n vectors do not span a complete n-dimensional space

27.2.2. Gaussian Elimination

Given a system $Ax = b$, or augmented matrix $[A \mid b]$, use the following 3 rules:

- Row Swapping: $R_i \longleftrightarrow R_j$
- Row Scaling: $k \cdot R_i \longrightarrow R_i$

- Row Replacement: $R_i - k \cdot R_j \longrightarrow R_i$

To solve the system:

1. Transform the augmented matrix into row echelon form (REF) using these operations:
    - The first nonzero entry (pivot) in each row is to the right of the pivot in the row above
    - All entries below a pivot are zero
    - Rows of all zeros (if any) are at the bottom
2. Back-substitute starting from the last nonzero row to find the solution

Example

Non-singular Case

System of equations

$$\begin{aligned} 2y &= 4 \\ 1x + 1y &= 3 \end{aligned}$$

Augmented matrix

$$\left[\begin{array}{cc|c} 0 & 2 & 4 \\ 1 & 1 & 3 \end{array}\right]$$

Swapping ($R_1 \longleftrightarrow R_2$)

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 2 & 4 \end{array}\right]$$

Scaling ($\frac{1}{2}R_2 \longrightarrow R_2$)

Row Echelon Form (REF)

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 1 & 2 \end{array}\right]$$

Replacement ($R_1 - R_2 \longrightarrow R_1$)

Reduced Row Echelon Form (RREF)

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right]$$

Result

$$x = 1, \quad y = 2$$

Example

Singular Case

System of equations

$$\begin{aligned} x + y &= 2 \\ 2x + 2y &= 4 \end{aligned}$$

Augmented matrix

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 2 & 2 & 4 \end{array}\right]$$

Replacement ($R_2 - 2R_1 \longrightarrow R_2$)

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 0 & 0 \end{array}\right]$$

Infinitely many solutions E.g.

- $x = 0, \quad y = 2$
- $x = 1, \quad y = 1$
- $x = 2, \quad y = 0$
- etc.

Rows are **linearly dependent**

The coefficient matrix

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

A has **determinant** 0, so it's **not invertible**

27.2.3. Inverse

$$A^{-1}A = I \quad \wedge \quad AA^{-1}A = I$$

A^{-1} is **unique**

Finding A^{-1}: Gauss-Jordan Elimination

A square matrix is **nonsingular** iif it is **invertible**

$$[A \mid I] \longrightarrow [I \mid A^{-1}]$$

Example

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$[A \mid I] = \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right]$$

Replacement ($R_2 - 2R_1 \longrightarrow R_2$)

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & -3 & -2 & 1 \end{array}\right]$$

Scale ($-\frac{1}{3}R_2 \longrightarrow R_2$)

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & \frac{2}{3} & -\frac{1}{3} \end{array}\right]$$

Replacement ($R_1 - 2R_2 \longrightarrow R_1$)

$$\left[\begin{array}{cc|cc} 1 & 0 & -\frac{1}{3} & \frac{2}{3} \\ 0 & 1 & \frac{2}{3} & -\frac{1}{3} \end{array}\right]$$

Result

$$A^{-1} = \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$$

27.2.4. Linear Dependence and Independence

A set of m n-dimensional vectors $x_1, x_2, ..., x_m$ are **linearly dependent** if there exists a non-zero vector $w \in \mathbb{R}^m$ such that:

$$w_1 x_1 + w_2 x_2 + ... + w_m x_m = 0$$

That is, at least one of the vectors can be written as a linear combination of the others.

Example

Dependent

Let:

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Augmented:

$$\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 2 & 4 & 0 \end{array}\right]$$

Elimination ($R_2 - 2R_1 \longrightarrow R_2$)

$$
\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 0 & 0 & 0 \end{array}\right]
$$

Independent

Let:

$$
x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}
$$

Augmented:

$$
\left[\begin{array}{cc|c} 1 & 3 & 0 \\ 2 & 1 & 0 \end{array}\right]
$$

Elimination $(R_2 - 2R_1 \longrightarrow R_2)$

$$
\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 0 & -5 & 0 \end{array}\right]
$$

Back-substitute

- $-5w_2 = 0 \Rightarrow w_2 = 0$
- $w_1 + 3w_2 = 0 \Rightarrow w_1 = 0$

Only solution is $w_1 = w_2 = 0$

In Gaussian elimination, a **row of all zeros**

1. **Homogeneous** System (RHS = 0)

$$
\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 0 & 0 & 0 \end{array}\right]
$$

$$
\begin{aligned} x_1 + 2x_2 &= 0 \\ 0 &= 0 \end{aligned}
$$

- ✅ Represents a **redundant** equation (i.e., no new information) **dependent** set of vectors
- ✅ Does not affect consistency: the system is always **consistent**
- ✅ The system has fewer pivots than variables, hence **infinitely many solutions**

2. **Inconsistent** System (Nonzero RHS)

$$
\left[\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 0 & 5 \end{array}\right]
$$

$$
\begin{aligned} x_1 + 2x_2 &= 3 \\ 0 &= 5 \end{aligned}
$$

- ❌ The second row leads to a **contradiction**
- ❌ The system is **inconsistent** (**no solution**)

27.2.5. Extreme Points

Given that x, x_1, and x_2 are points in the set S:

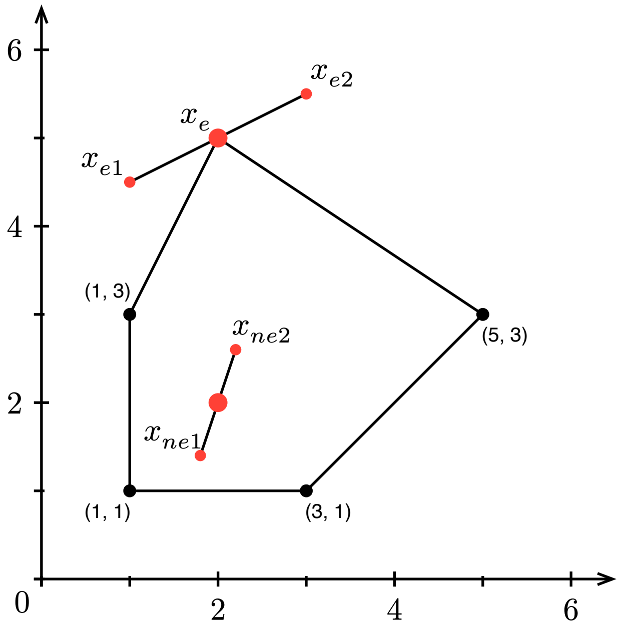
- If you can find two different points x_1 and x_2 in S such that x lies somewhere strictly between them on the straight line connecting x_1 and x_2, then x is not an extreme point
- If no such pair of points exists—meaning you cannot place x anywhere strictly between two other points in S on a straight line—then x is an extreme point

Formal Definition

For a set $S \in \mathbb{R}^n$, a point x is an extreme point if there does not exist a three-tuple (x_1, x_2, λ) such that $x_1 \in S$, $x_2 \in S$, $\lambda \in (0, 1)$, and

$$x = \lambda x_1 + (1 - \lambda) x_2$$

A point x in set $S \subseteq \mathbb{R}^n$ is extreme if it cannot be written as a strict **convex combination** of two different points $x_1, x_2 \in S$.



Convex Combination

Let $x_1, x_2, ..., x_k \in \mathbb{R}^n$. A convex combination of these vectors is any point of the form:

$$x = \lambda_1 x_1 + \lambda_2 x_2 + ... + \lambda_k x_k$$

where:

- $\lambda_i \geq 0 \quad \forall i$
- $\sum_{i=1}^{k} \lambda_i = 1$

x lies strictly between x_1 and x_2 on the line segment connecting them, not equal to either one

27.2.6. Slack and Suplus

1. Slack

Unused capacity

Example

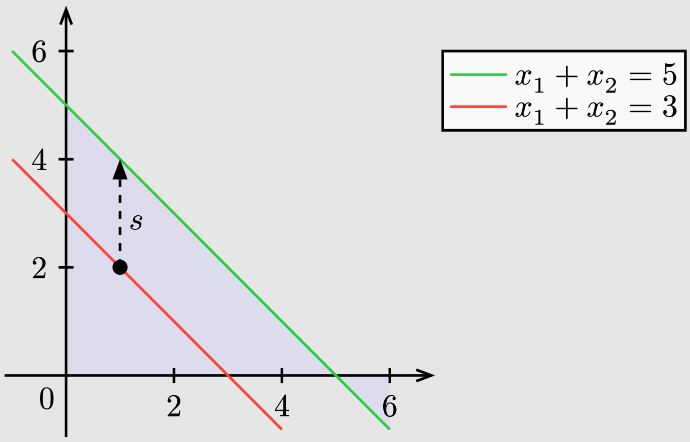
Given a constraint:

$$x_1 + x_2 \leq 5$$

The total **must not exceed** 5. To convert this into an equation, we **add** a slack variable $s \geq 0$:

$$x_1 + x_2 + s = 5$$

- If $x_1 + x_2 = 3$, then $s = 2$
- If $x_1 + x_2 = 5$, then $s = 0$



2. Excess (surplus)

Excess above the required amount

Example

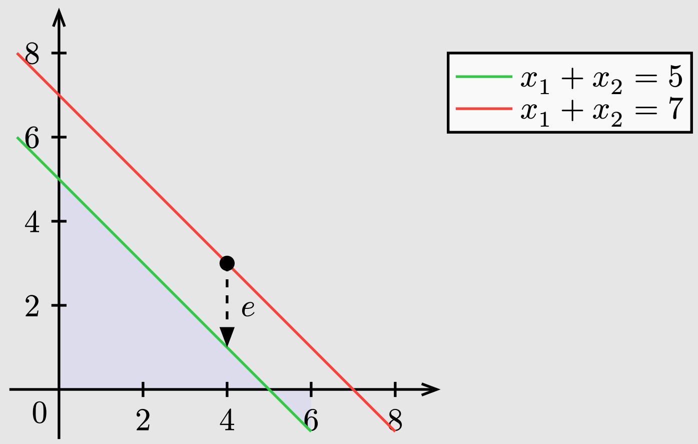
Given a constraint:

$$x_1 + x_2 \geq 5$$

The total **must be at least** 5. To convert this into an equation, we **subtract** a excess variable $s \geq 5$:

$$x_1 + x_2 - e = 5$$

- If $x_1 + x_2 = 7$, then $e = 2$
- If $x_1 + x_2 = 5$, then $e = 0$



Objective Function		
Max	✅	$\max z = c^T x$
Min $\longrightarrow$ Max	$-z$	$\min z = c^T \boldsymbol{x}$ $\Longleftrightarrow$ $\max z = -c^T \boldsymbol{x}$
Constraint Forms		
$\leq$	Slack	$a_1x_1 + a_2x_2 \leq b$ $a_1x_1 + a_2x_2 + s = b$
$\geq$	Excess + Artificial	$a_1x_1 + a_2x_2 \geq b$ $a_1x_1 + a_2x_2 - e + a = b$
$=$	Artificial	$a_1x_1 + a_2x_2 = b$ $a_1x_1 + a_2x_2 + a = b$
Negative b	Multiply by –1	$a_1x_1 + a_2x_2 \leq -b$ $-a_1x_1 - a_2x_2 \geq b$
Variable Bounds		
Non-negativity		$x_i, s_i, a_i \geq 0 \quad \forall i$
Lower Bound ≠ 0	Substitution	$x_1 \geq 5$ $y_1 = x_1 - 5$ $x_1 = y_1 + 5$ $y_1 \geq 0$

Unrestricted Variables	Replace with difference	x_j unrestricted $x_j = x_j' - x_j''$ $x_j', x_j'' \geq 0$
Variable Roles		
Basic Variables	Usually non-zero	Part of the current solution (solved from the constraints)
Non-Basic Variables	Always zero	Temporarily set to 0 so basic variables can be solved
Simplex		
Selection: • Entering variable • Pivot column selection	For **maximization** problems: • Select the column with the **most negative** value in the objective row • This corresponds to the variable that will **increase the objective function** the most For **minimization** problems: • Select the column with the **most positive** value in the objective row • This corresponds to the variable that will **decrease the objective function** the most	

Selection: • Leaving variable • Pivot row	1. Calculate **ratios** for each row (except the objective row) • For each row: b_i/p_i • Only **consider** rows where the pivot column element $p_i > 0$ • **Ignore** rows where the pivot column element $p_i \leq 0$ 2. Select the **minimum ratio**: • The row with the **smallest non-negative ratio** becomes the pivot row • The basic variable in this row is the **leaving variable** (exits the basis)	

,

Example

$$
\begin{aligned}
\max \quad & z = 3x_1 + 2x_2 \\
s.t. \quad & x_1 + x_2 \leq 4 \\
& x_1 \leq 2 \\
& x_2 \leq 3 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Step 1: Convert to Standard Form

$$
\begin{aligned}
\max \quad & z = 3x_1 + 2x_2 \\
s.t. \quad & x_1 + x_2 + s_1 = 4 \\
& x_1 + s_2 = 2 \\
& x_2 + s_3 = 3 \\
& x_1, x_2, s_1, s_2, s_3 \geq 0
\end{aligned}
$$

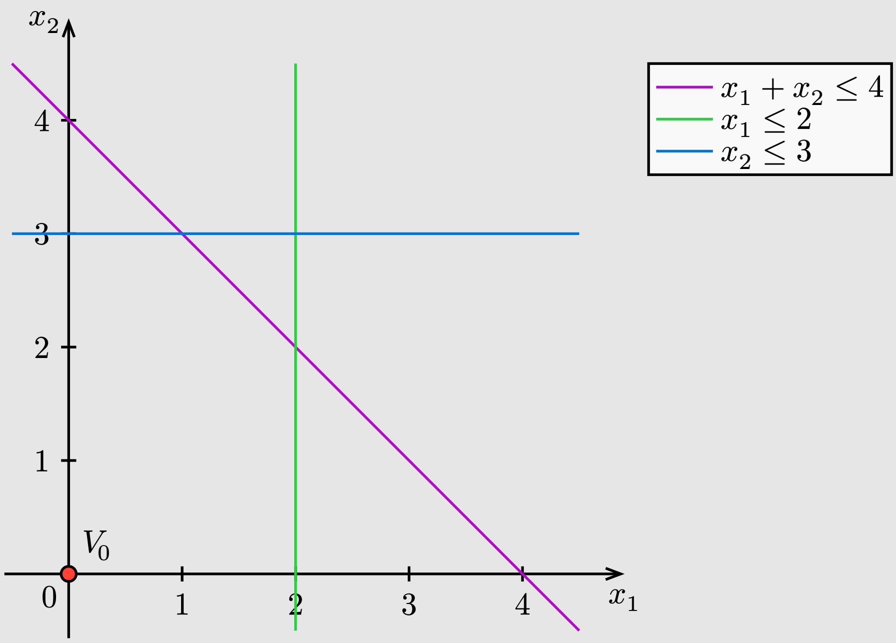
Objective

$$z - 3x_1 - 2x_2 = 0$$

Step 2: Initial Tableau

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	4
s_2	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	0	0	0

- Basic Variables: $s_1 = 4$, $s_2 = 2$, $s_3 = 3$
- Non-basic variables: $x_1 = 0$, $x_2 = 0$
  - $V_0 = (0, 0)$
  - Objective: $z = 3x_1 + 2x_2 = 3(0) + 2(0) = 0$



Step 3: Choose entering variable

- Look at the bottom row (z):
  - Coefficients of $x_1 = -3$, $x_2 = -2$
- Most negative: $x_1 \longrightarrow$ Enter the basis

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	4
s_2	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	0	0	0

Step 4: Determine leaving variable

Look at ratios of b/x_1:

Basis	x_1	x_2	s_1	s_2	s_3	b	b/x_1
s_1	1	1	1	0	0	4	$4/1 = 4$
s_2	1	0	0	1	0	2	$2/1 = 2$
s_3	0	1	0	0	1	3	$3/0 = \infty$
z	–3	–2	0	0	0	0	

- Minimum Ratio
  - s_2 leaves, x_1 enters

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	4
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	3	0	0

Step 5: Pivot on row 2, column x_1

Now eliminate x_1 from other rows:

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	4
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	3	0	0

$R_{s_1} := R_{s_1} - R_{x_1}$

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	4
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	3	0	0

$$R_{s_1} := R_{s_1} - R_{x_1} \quad \Longrightarrow \quad \begin{aligned} x_1 &: \quad 1 - 1 = 0 \\ x_2 &: \quad 1 - 0 = 1 \\ s_1 &: \quad 1 - 0 = 1 \\ s_2 &: \quad 0 - 1 = -1 \\ s_3 &: \quad 0 - 0 = 0 \\ b &: \quad 4 - 2 = 2 \end{aligned}$$

Update R_{s_1}

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	3	0	0

$R_z := R_z - 3R_{x_1}$

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	–3	–2	0	3	0	0

$$R_z := R_z - 3R_{x_1} \quad \Longrightarrow \quad \begin{aligned} x_1 &: \quad -3 + 3(1) = 0 \\ x_2 &: \quad -2 + 3(0) = -2 \\ s_1 &: \quad 0 + 3(0) = 0 \\ s_2 &: \quad 0 + 3(1) = 3 \\ s_3 &: \quad 0 + 3(0) = 0 \\ b &: \quad 0 + 3(2) = 6 \end{aligned}$$

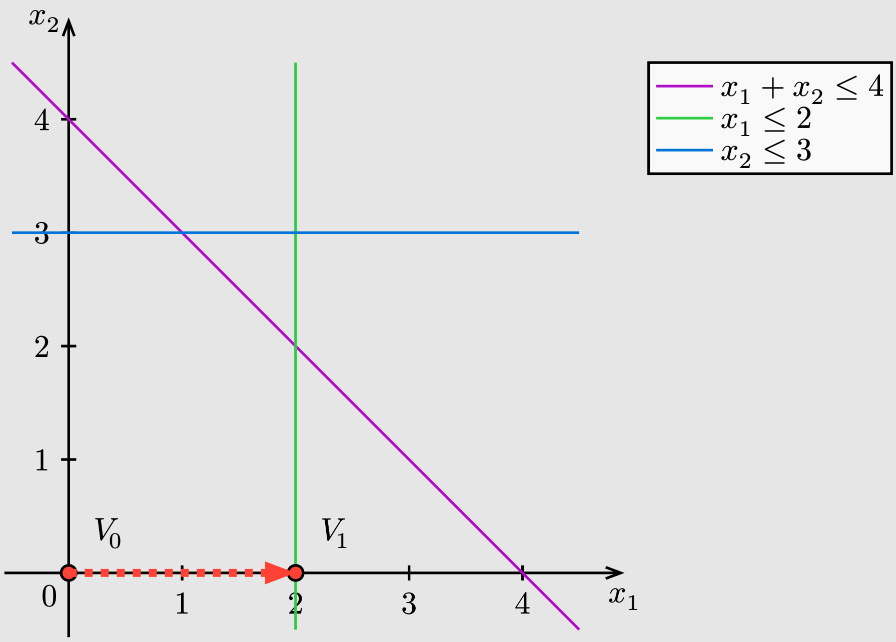
Update R_z

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

- Basic Variables: $x_1 = 2$, $s_1 = 2$, $s_3 = 3$

- Non-basic variables: $x_2 = 0$
  - $V_1 = (2, 0)$
  - Objective: $z = 3x_1 + 2x_2 = 3(2) + 2(0) = 6$

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6



Step 3: Choose entering variable

- Look at the bottom row (z):
  - Coefficients of $x_2 = -2$
- Most negative: $x_2 \longrightarrow$ Enter the basis

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

Step 4: Determine leaving variable

Look at ratios of b/x_2:

Basis	x_1	x_2	s_1	s_2	s_3	b	b/x_2
s_1	0	1	1	–1	0	2	$2/1 = 2$

x_1	1	0	0	1	0	2	$2/0 = \infty$
s_3	0	1	0	0	1	3	$3/1 = 3$
z	0	–2	0	3	0	6	

- s_1 leaves, x_2 enters

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

Step 5: Pivot on row 1, column x_2

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

Now eliminate x_2 from other rows:

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

$R_{s_3} := R_{s_3} - R_{x_2}$

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	1	0	0	1	3
z	0	–2	0	3	0	6

$$R_{s_3} := R_{s_3} - R_{x_2} \quad \Longrightarrow \quad \begin{aligned} x_1 &: \quad 0 - 0 = 0 \\ x_2 &: \quad 1 - 1 = 0 \\ s_1 &: \quad 0 - 1 = -1 \\ s_2 &: \quad 0 - (-1) = 1 \\ s_3 &: \quad 1 - 0 = 1 \\ b &: \quad 3 - 2 = 1 \end{aligned}$$

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	0	–1	1	1	1
z	0	–2	0	3	0	6

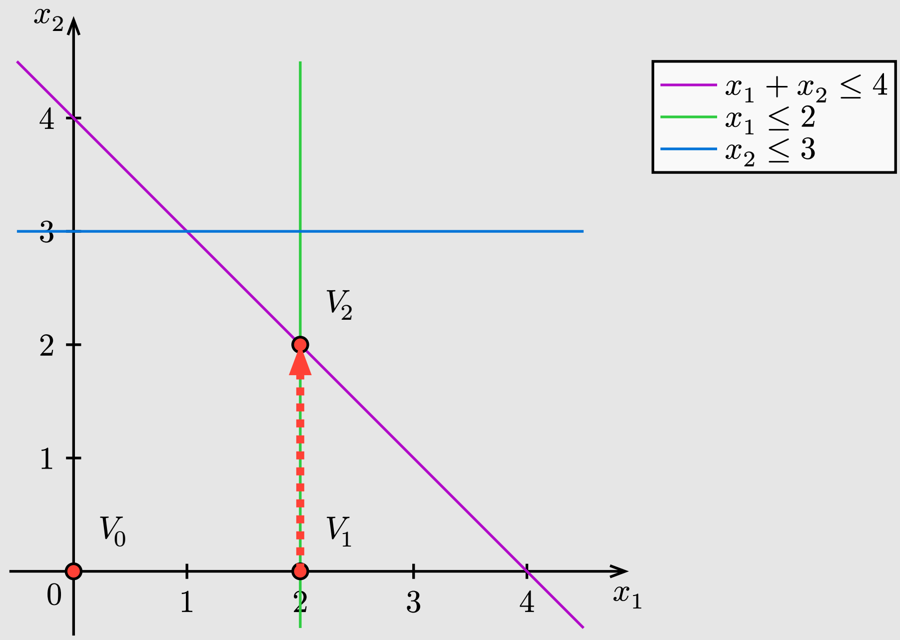
$$R_z = R_z - (-2)R_{x_2} = R_z + 2R_{x_2}$$

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	0	–1	1	1	1
z	0	–2	0	3	0	6

$$R_z = R_z + 2R_{x_2} \quad \Longrightarrow \quad \begin{aligned} x_1 &: \quad 0 + 2(0) = 0 \\ x_2 &: \quad -2 + 2(1) = 0 \\ s_1 &: \quad 0 + 2(1) = 2 \\ s_2 &: \quad 3 + 2(-1) = 1 \\ s_3 &: \quad 0 + 2(0) = 0 \\ b &: \quad 6 + 2(2) = 10 \end{aligned}$$

Basis	x_1	x_2	s_1	s_2	s_3	b
x_2	0	1	1	–1	0	2
x_1	1	0	0	1	0	2
s_3	0	0	–1	1	1	1
z	0	0	2	1	0	10

- Basic Variables: $x_1 = 2$, $x_2 = 2$, $s_3 = 1$
- Non-basic variables: $s_1 = 0$, $s_2 = 0$
  - $V_2 = (2, 2)$
  - Objective: $z = 3x_1 + 2x_2 = 3(2) + 2(2) = 10$



Example

Problem

$$\begin{aligned} \min \quad & z = x_1 + 2x_2 & \\ s.t. \quad & x_1 + x_2 \leq 1 & \text{(slack)} \\ & 2x_1 - x_2 \geq 2 & \text{(surplus + artificial)} \\ & -x_1 + 3x_2 \leq -1 & \text{(negative } b\text{)} \\ & x_1 - x_2 = 1 & \text{(artificial)} \\ & x_1 \geq 0 & \\ & x_2 & \text{(unrestricted)} \end{aligned}$$

Standard Form

Step 1: Handle Unrestricted Variable x_2

Since x_2 is unrestricted:

$$x_2 = x_2' - x_2''$$
$$x_2', x_2'' \geq 0$$

Rewrite the objective:

$$\begin{aligned} z &= x_1 + 2(x_2' - x_2'') \\ &= x_1 + 2x_2' - 2x_2'' \end{aligned}$$

Step 2: Slack Constraint ($\leq$)

$$x_1 + x_2 \leq 1$$

Rewrite with $x_2' - x_2''$:

$$x_1 + (x_2' - x_2'') \leq 1$$

Add slack variable:

$$x_1 + x_2' - x_2'' + s_1 = 1$$
$$s_1 \geq 0$$

Step 3: Surplus + Artificial Constraint ($\geq$)

$$2x_1 - x_2 \geq 2$$

Rewrite with $x_2' - x_2''$:

$$2x_1 - (x_2' - x_2'') \geq 2 \implies 2x_1 - x_2' + x_2'' \geq 2$$

Convert to equality by subtracting a surplus variable and adding an artificial variable:

$$2x_1 - x_2' + x_2'' - s_2 + a_1 = 2$$
$$s_2, a_1 \geq 0$$

Step 4: Negative RHS Constraint

$$-x_1 + 3x_2 \leq -1$$

Rewrite with $x_2' - x_2''$:

$$-x_1 + 3(x_2' - x_2'') \leq -1 \implies -x_1 + 3x_2' - 3x_2'' \leq -1$$

Multiply both sides by -1 to get a positive RHS and flip inequality:

$$x_1 - 3x_2' + 3x_2'' \geq 1$$

Convert to equality by subtracting a surplus variable and adding an artificial variable:

$$x_1 - 3x_2' + 3x_2'' - s_3 + a_2 = 1$$
$$s_3, a_2 \geq 0$$

Step 5: Artificial Equality Constraint ($=$)

$$x_1 - x_2 = 1$$

Rewrite with $x_2' - x_2''$:

$$x_1 - (x_2' - x_2'') = 1 \implies x_1 - x_2' + x_2'' = 1$$

Add an artificial variable:

$$x_1 - x_2' + x_2'' + a_3 = 1$$
$$a_3 \geq 0$$

Final Standard Form

$$
\begin{aligned}
\min \quad & z = x_1 + 2x_2' - 2x_2'' \\
s.t. \quad & x_1 + x_2' - x_2'' + s_1 = 1 \\
& 2x_1 - x_2' + x_2'' - s_2 + a_1 = 2 \\
& x_1 - 3x_2' + 3x_2'' - s_3 + a_2 = 1 \\
& x_1 - x_2' + x_2'' + a_3 = 1 \\
& x_1, x_2', x_2'', s_1, s_2, s_3, a_1, a_2, a_3 \geq 0
\end{aligned}
$$

Phase I

Objective

$$
\begin{aligned}
\min \quad & w = a_1 + a_2 + a_3 \\
s.t. \quad & x_1 + x_2' - x_2'' + s_1 = 1 && \text{(slack)} \\
& 2x_1 - x_2' + x_2'' - s_2 + a_1 = 2 && \text{(surplus + artificial)} \\
& x_1 - 3x_2' + 3x_2'' - s_3 + a_2 = 1 && \text{(negative } b\text{)} \\
& x_1 - x_2' + x_2'' + a_3 = 1 && \text{(artificial)} \\
& x_1, x_2', x_2'', s_1, s_2, s_3, a_1, a_2, a_3 \geq 0 && \text{(unrestricted)}
\end{aligned}
$$

Deriving Phase I Objective Row R_w

1. Objective function before substitution

$$w = a_1 + a_2 + a_3$$

2. Express each artificial variable in terms of other variables from their constraints

- R_{a_1}

$$
\begin{aligned}
2x_1 - x_2' + x_2'' - s_2 + a_1 &= 2 \\
a_1 &= -2x_1 + x_2' - x_2'' + s_2 + 2
\end{aligned}
$$

- R_{a_2}

$$
\begin{aligned}
x_1 - 3x_2' + 3x_2'' - s_3 + a_2 &= 1 \\
a_2 &= -x_1 + 3x_2' - 3x_2'' + s_3 + 1
\end{aligned}
$$

- R_{a_3}

$$
\begin{aligned}
x_1 - x_2' + x_2'' + a_3 &= 1 \\
a_3 &= -x_1 + x_2' - x_2'' + 1
\end{aligned}
$$

3. Substitute into w:

$$
\begin{aligned}
w &= a_1 + a_2 + a_3 \\
&= (-2x_1 + x_2' - x_2'' + s_2 + 2) + (-x_1 + 3x_2' - 3x_2'' + s_3 + 1) + (-x_1 + x_2' - x_2'' + 1) \\
&= (-2x_1 - x_1 - x_1) + (x_2' + 3x_2' + x_2') + (-x_2'' - 3x_2'' - x_2'') + (s_2 + s_3) + (2 + 1 + 1) \\
&= -4x_1 + 5x_2' - 5x_2'' + s_2 + s_3 + 4
\end{aligned}
$$

4. Rewrite as a constraint (bring all variables to the left):

$$w + 4x_1 - 5x_2' + 5x_2'' - s_2 - s_3 = 4$$
$$w = -4x_1 + 5x_2' - 5x_2'' + s_2 + s_3 + 4$$

So in tableau row R_w:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	1
a_1	2	–1	1	0	–1	0	1	0	0	2
a_2	1	–3	3	0	0	–1	0	1	0	1
a_3	1	–1	1	0	0	0	0	0	1	1
w	4	–5	5	0	–1	–1	0	0	0	4

Perform First Pivot

1. Pivot Column

Look at the w row (objective row) for the most negative coefficient:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	1
a_1	2	–1	1	0	–1	0	1	0	0	2
a_2	1	–3	3	0	0	–1	0	1	0	1
a_3	1	–1	1	0	0	0	0	0	1	1
w	4	–5	5	0	–1	–1	0	0	0	4

The most negative is $-5 \longrightarrow$ pivot column is x_2'

2. Ratio

- Compute ratios (b/x_2') for rows with positive pivot entries:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b	b/x_2'
s_1	1	1	–1	1	0	0	0	0	0	1	1
a_1	2	–1	1	0	–1	0	1	0	0	2	✗
a_2	1	–3	3	0	0	–1	0	1	0	1	✗
a_3	1	–1	1	0	0	0	0	0	1	1	✗
w	4	–5	5	0	–1	–1	0	0	0	4	

- Minimum ratio is 1, so we pivot on row s_1:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b	b/x_2'
s_1	1	1	–1	1	0	0	0	0	0	1	1
a_1	2	–1	1	0	–1	0	1	0	0	2	✗

a_2	1	–3	3	0	0	–1	0	1	0	1	✗
a_3	1	–1	1	0	0	0	0	0	1	1	✗
w	4	–5	5	0	–1	–1	0	0	0	4	

3. Pivot $s_1 \longrightarrow x_2'$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
x_2'	1	1	–1	1	0	0	0	0	0	1
a_1	2	–1	1	0	–1	0	1	0	0	2
a_2	1	–3	3	0	0	–1	0	1	0	1
a_3	1	–1	1	0	0	0	0	0	1	1
w	4	–5	5	0	–1	–1	0	0	0	4

4. Gauss-Jordan Elimination

Example

Step 3: Pivot $a_1 \longrightarrow x_1$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	5
x_1	3	–1	1	0	–1	0	1	0	0	4
a_2	1	–2	2	0	0	–1	0	1	0	6
a_3	1	–1	1	0	0	0	0	0	1	2
w	–5	4	–4	0	1	1	0	0	0	12

Gauss-Jordan Elimination

$$R_{x_1} = \frac{1}{3} R_{x_1}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	5
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	1	–2	2	0	0	–1	0	1	0	6
a_3	1	–1	1	0	0	0	0	0	1	2
w	–5	4	–4	0	1	1	0	0	0	12

$$R_{a_2} = R_{a_2} - 1R_{x_1}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b

s_1	1	1	–1	1	0	0	0	0	0	5
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
a_3	1	–1	1	0	0	0	0	0	1	2
w	–5	4	–4	0	1	1	0	0	0	12

$$R_{a_3} = R_{a_3} - 1R_{x_1}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	5
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
a_3	0	–2/3	2/3	0	1/3	0	–1/3	0	1	2/3
w	–5	4	–4	0	1	1	0	0	0	12

$$R_w = R_w + 5R_{x_1}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	1	1	–1	1	0	0	0	0	0	5
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
a_3	0	–2/3	2/3	0	1/3	0	0-1/3	0	1	2/3
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

$$R_{s_1} = R_{s_1} - 1R_{x_1}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
a_3	0	–2/3	2/3	0	1/3	0	–1/3	0	1	2/3
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

Step 3: Perform the Second Pivot

1. Look at the w row (objective row) for the most negative coefficient:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3

a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
a_3	0	–2/3	2/3	0	1/3	0	0-1/3	0	1	2/3
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

- The most negative is $-\frac{7}{3}$ for x_2''
  - Pivot column is x_2''

2. Ratio
   - Compute ratios (b/x_2'') for rows with positive pivot entries:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b	b/x_2''
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3	✗
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3	$\frac{4}{3} \div \frac{1}{3} = 4$
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3	$\frac{14}{3} \div \frac{5}{3} = 2.8$
a_3	0	–2/3	2/3	0	1/3	0	0-1/3	0	1	2/3	$\frac{2}{3} \div \frac{2}{3} = 1$
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3	

   - Minimum ratio is 1, so we pivot on row a_3:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b	b/x_2''
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3	✗
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3	$\frac{4}{3} \div \frac{1}{3} = 4$
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3	$\frac{14}{3} \div \frac{5}{3} = 2.8$
a_3	0	–2/3	2/3	0	1/3	0	0-1/3	0	1	2/3	$\frac{2}{3} \div \frac{2}{3} = 1$
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3	

Step 4: Pivot $a_3 \longrightarrow x_2''$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
x_2''	0	–2/3	2/3	0	1/3	0	–1/3	0	1	2/3
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

Gauss-Jordan Elimination

$$R_{x_2''} = R_{x_2''}/(2/3)$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	4/3	–4/3	1	1/3	0	–1/3	0	0	11/3

x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

$$R_{s_1} = R_{s_1} - \left(-\frac{4}{3}\right) R_{x_2''} = R_{s_1} + \frac{4}{3} R_{x_2''}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	0	0	1	1	0	–1	0	2	5
x_1	1	–1/3	1/3	0	–1/3	0	1/3	0	0	4/3
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

$$R_{x_1} = R_{x_1} - \frac{1}{3} R_{x_2''}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	0	0	0	–1/2	0	1/2	0	–1/2	1
x_1	1	0	0	0	–1/2	0	1/2	0	–1/2	1
a_2	0	–5/3	5/3	0	1/3	–1	–1/3	1	0	14/3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

$$R_{a_2} = R_{a_2} - \frac{5}{3} R_{x_2''}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	0	0	0	–1/2	0	1/2	0	–1/2	1
x_1	1	0	0	0	–1/2	0	1/2	0	–1/2	1
a_2	0	0	0	0	–1/2	–1	1/2	1	–5/2	3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	7/3	–7/3	0	–2/3	1	5/3	0	0	56/3

$$R_w = R_w - \left(-\frac{7}{3}\right) R_{x_2''} = R_w + \frac{7}{3} R_{x_2''}$$

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	0	0	0	–1/2	0	1/2	0	–1/2	1

x_1	1	0	0	0	–1/2	0	1/2	0	–1/2	1
a_2	0	0	0	0	–1/2	–1	1/2	1	–5/2	3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	0	0	0	1/2	1	1/2	0	7/2	21

Step 5: Perform the Third Pivot

1. Look at the w row (objective row) for the most negative coefficient:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b
s_1	0	0	0	0	–1/2	0	1/2	0	–1/2	1
x_1	1	0	0	0	–1/2	0	1/2	0	–1/2	1
a_2	0	0	0	0	–1/2	–1	1/2	1	–5/2	3
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1
w	0	0	0	0	–1/6	1	1/2	0	7/2	21

- The most negative is $-\frac{1}{6}$ for s_2
  - Pivot column is s_2

2. Ratio

- Compute ratios (b/s_2) for rows with positive pivot entries:

Basis	x_1	x_2'	x_2''	s_1	s_2	s_3	a_1	a_2	a_3	b	b/s_2
s_1	0	0	0	0	–1/2	0	1/2	0	–1/2	1	❌
x_1	1	0	0	0	–1/2	0	1/2	0	–1/2	1	❌
a_2	0	0	0	0	–1/2	–1	1/2	1	–5/2	3	❌
x_2''	0	–1	1	0	1/2	0	–1/2	0	3/2	1	2
w	0	0	0	0	–1/6	1	1/2	0	7/2	21	

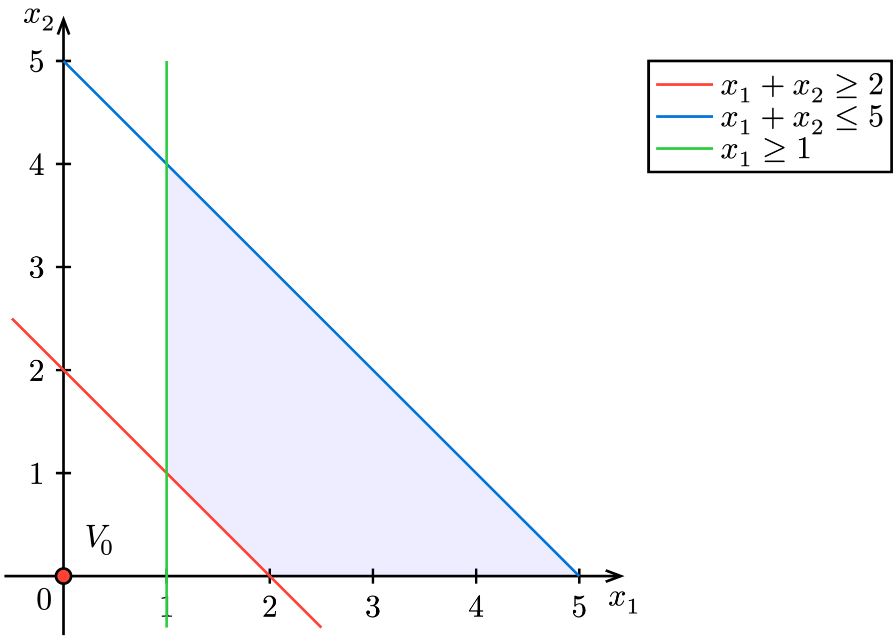
Two-Phase

$$
\begin{array}{llcccl}
\max & z = 2x_1 & + & x_2 & & \\
s.t. & x_1 & + & x_2 & \geq & 2 \\
 & x_1 & & & \geq & 1 \\
 & x_1 & + & x_2 & \leq & 5 \\
 & x_1, x_2 \geq 0 & & & &
\end{array}
$$

$$\begin{array}{lrcrcrcrcrcl}
\max & z = 2x_1 & + & x_2 & + & 0e_1 & + & 0e_2 & + & 0s_1 & & \\
s.t. & x_1 & + & x_2 & - & e_1 & & & & & \geq & 2 \\
 & x_1 & & & & & - & e_2 & & & \geq & 1 \\
 & x_1 & + & x_2 & & & & & + & s_1 & \leq & 5
\end{array}$$

$$z - 2x_1 - x_2 = 0$$

Basis	x_1	x_2	e_1	e_2	s_1	b
???	1	1	-1	0	0	2
???	1	0	0	-1	0	1
s_1	1	1	0	0	1	5
z	–2	–1	0	0	0	0



$$\begin{array}{lrcrcrcrcrcl}
\min & w = 0x_1 & + & 0x_2 & + & 0e_1 & + & 0e_2 & + & 0s_1 & & \\
s.t. & x_1 & + & x_2 & - & e_1 & & & & & \geq & 2 \\
 & x_1 & & & & & - & e_2 & & & \geq & 1 \\
 & x_1 & + & x_2 & & & & & + & s_1 & \leq & 5
\end{array}$$

Degeneracy

Degeneracy occurs when a basic variable takes the value zero in a basic feasible solution. This leads to ties in the minimum ratio test, which determines the leaving variable during a pivot.

Example

Problem

$$
\begin{aligned}
\max \quad & z = x_1 + 2x_2 \\
s.t. \quad & x_1 + x_2 \leq 3 \\
& x_2 \leq 2 \\
& \frac{1}{2}x_1 + x_2 \leq 2.5 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

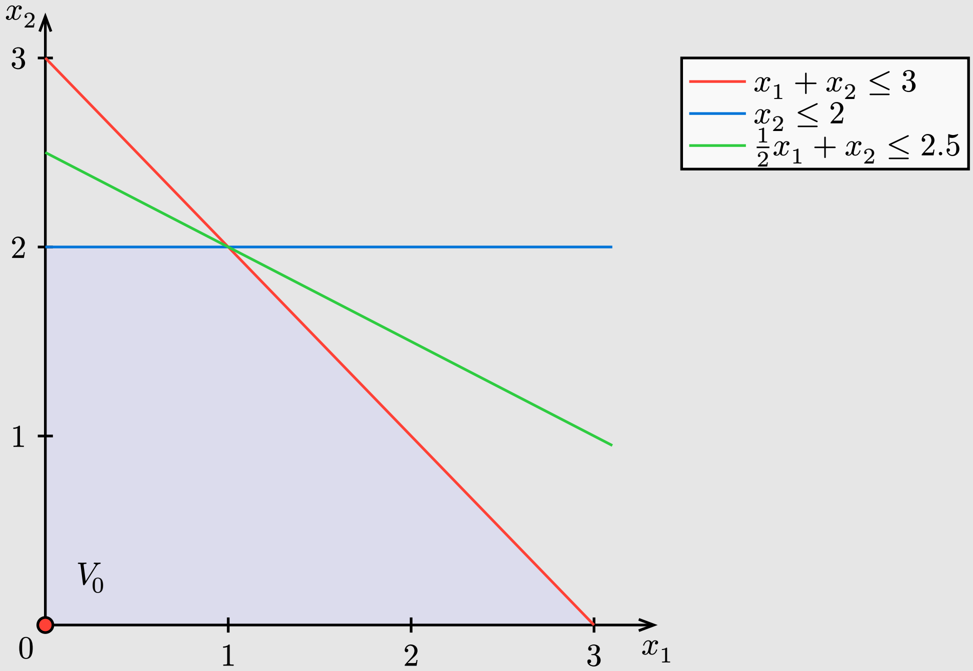
Standard Form

$$
\begin{gathered}
z - x_1 - 2x_2 = 0 \\
x_1 + x_2 + s_1 = 3 \\
x_{2_{s_2}} = 2 \\
\frac{1}{2}x_1 + x_2 + s_3 \\
x_1, x_2, s_1, s_2, s_3 \geq 0
\end{gathered}
$$

Initialize Tableau

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	3
s_2	0	1	0	1	0	2
s_3	1/2	1	0	0	1	2.5
w	–1	–1	0	0	0	0

$$
\begin{gathered}
x_1 = 0 \\
x_2 = 0
\end{gathered}
$$



Minimum Test

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	3
s_2	0	1	0	1	0	2
s_3	1/2	1	0	0	1	2.5
w	–1	–1	0	0	0	0

Ratio Test

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	3
s_2	0	1	0	1	0	2
s_3	1/2	1	0	0	1	2.5
w	–1	–1	0	0	0	0

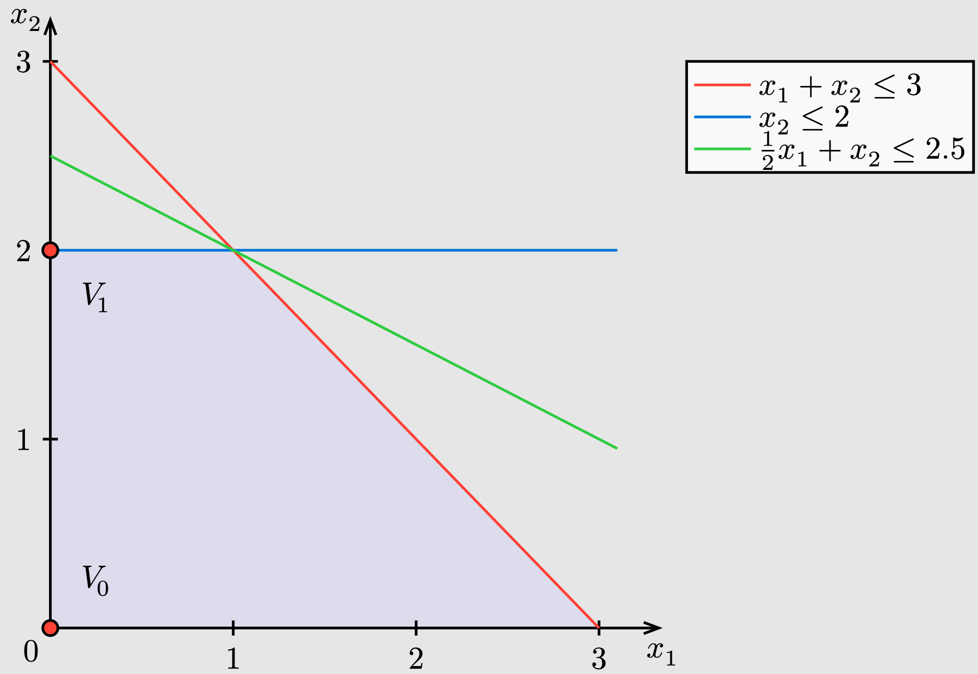
Pivot

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	1	1	0	0	3
x_2	0	1	0	1	0	2
s_3	1/2	1	0	0	1	2.5
w	–1	–1	0	0	0	0

Gaussian Elimination

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	0	1	–1	0	1
x_2	0	1	0	1	0	2
s_3	1/2	0	0	–1	1	0.5
w	–1	0	0	2	0	4

$$x_1 = 0$$
$$x_2 = 2$$



Minimum Test

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	0	1	–1	0	1
x_2	0	1	0	1	0	2
s_3	1/2	0	0	–1	1	0.5
w	–1	0	0	2	0	4

Ratio Test

Basis	x_1	x_2	s_1	s_2	s_3	b
s_1	1	0	1	–1	0	1
x_2	0	1	0	1	0	2
s_3	1/2	0	0	–1	1	0.5
w	–1	0	0	2	0	4

$$\left.\begin{array}{l} \frac{1}{1} \;\; = 1 \\[1em] \frac{0.5}{0.5} = 1 \end{array}\right\} \text{Tie}$$

Pivot

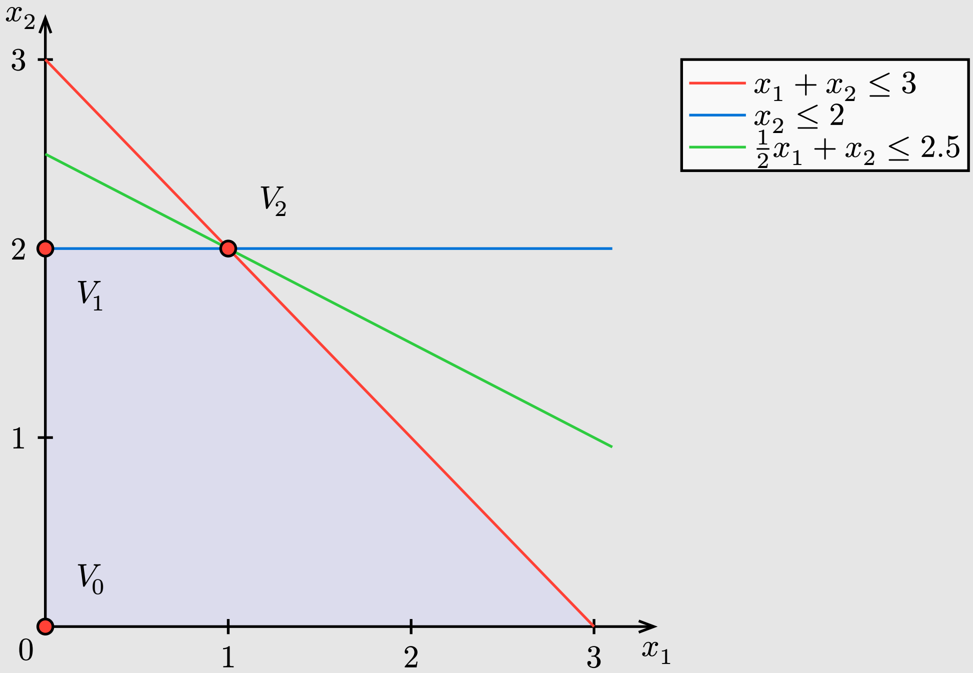
Basis	x_1	x_2	s_1	s_2	s_3	b
x_1	1	0	1	–1	0	1
x_2	0	1	0	1	0	2
s_3	1/2	0	0	–1	1	0.5

w	–1	0	0	2	0	4

Gaussian Elimination

Basis	x_1	x_2	s_1	s_2	s_3	b
x_1	1	0	1	–1	0	1
s_2	0	1	0	0	0	2
s_3	0	0	–1/2	–1/2	1	0
w	0	0	1	1	0	5

If after elimination a basic variable has a value of 0, it indicates degeneracy.



Degeneracy comes from redundant constraints

Bland's Rule: Choose the variable with the lowest index in the basis

- x_1 over x_2
- x_2 over s_1

Unbounded Solution

When identifying the entering variable, look at its column in the tableau. If all entries in that column (above the objective row) are ≤ 0, the problem is unbounded

Can't perform ratio test $\longrightarrow$ no constraint limits the entering variable $\longrightarrow$ objective function increases without bound

Example

Problem

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 - x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Standard Form

$$\begin{aligned} \max \quad & z - x_1 - x_2 = 0 \\ s.t. \quad & x_1 - 2x_2 + s_1 = 1 \\ & x_1, x_2, s_1 \geq 0 \end{aligned}$$

Initialize Tableau

Basis	x_1	x_2	s_1	b
s_1	1	–1	1	1
z	–1	–1	0	0

Minimum and Ratio Test

Basis	x_1	x_2	s_1	b
s_1	1	–1	1	1
z	–1	–1	0	0

Pivot

Basis	x_1	x_2	s_1	b
x_1	1	–1	1	1
z	–1	–1	0	0

Gaussian Elimination

Basis	x_1	x_2	s_1	b
x_1	1	–1	1	1
z	0	–2	1	1

Minimum and Ratio Test

Basis	x_1	x_2	s_1	b
x_1	1	–1	1	1
z	0	–2	1	1

Check for Optimality

Objective row has –2 for x_2 $\longrightarrow$ not optimal

Entering variable: x_2 (most negative coefficient)

Unboundedness Detection

Looking at the x_2 column:

Entry in constraint row: –1 (which is ≤ 0)

All entries in x_2 column are ≤ 0

Conclusion: Cannot perform ratio test $\longrightarrow$ Problem is unbounded

Alternative Solutions

Example

Problem

$$\begin{aligned} \max \quad & 2x_1 + 4x_2 \\ s.t. \quad & x_1 + x_2 \geq 3 \\ & \frac{1}{2}x_1 + x_2 \geq 2.5 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Standard Form

$$\begin{gathered} z - 2x_1 - x_2 = 0 \\ x_1 + x_2 + s_1 = 3 \\ \frac{1}{2}x_1 + x_2 + s_2 = 2.5 \\ x_1, x_2, x_3 \geq 0 \end{gathered}$$

Initialize Tableau

Basis	x_1	x_2	s_1	s_2	b
s_1	1	1	1	0	3
s_2	1/2	1	0	1	2.5
z	–2	–4	0	0	0

Minimum Test

Basis	x_1	x_2	s_1	s_2	b
s_1	1	1	1	0	3
s_2	1/2	1	0	1	2.5
z	–2	–4	0	0	0

Ratio Test

Basis	x_1	x_2	s_1	s_2	b

s_1	1	1	1	0	3
s_2	1/2	1	0	1	2.5
z	–2	–4	0	0	0

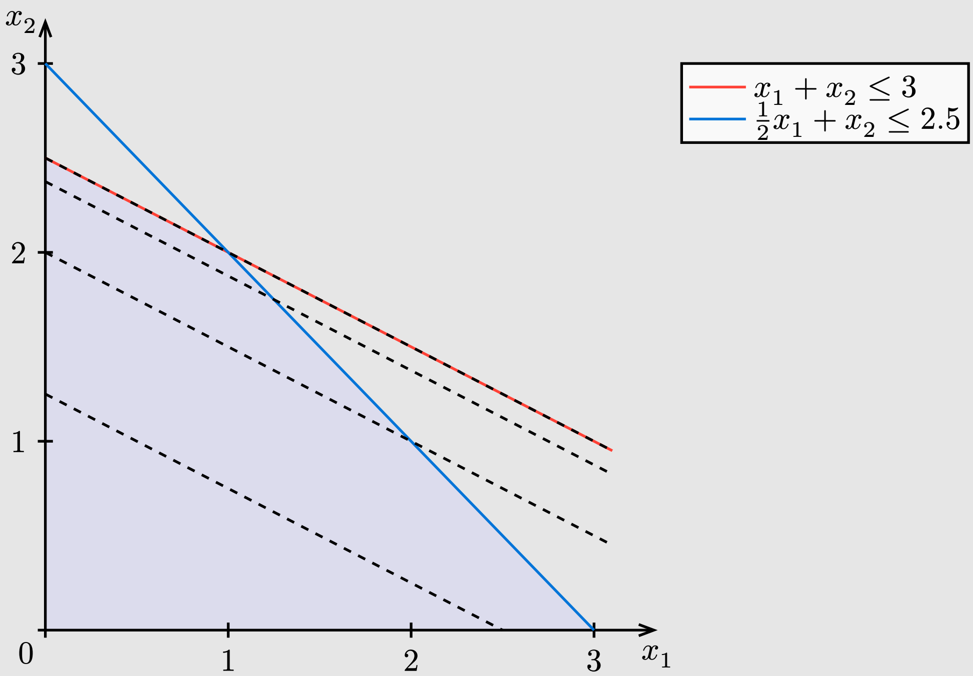
Pivot

Basis	x_1	x_2	s_1	s_2	b
s_1	1	1	1	0	3
x_2	1/2	1	0	1	2.5
z	–2	–4	0	0	0

Gaussian Elimination

Basis	x_1	x_2	s_1	s_2	b
s_1	1/2	0	1	–1	0.5
x_2	1/2	1	0	1	2.5
z	0	0	0	4	10

In a maximization or minimization linear programming problem, if there is a 0 in the z row (objective function row) of the final (optimal) simplex tableau in a non-basic column (i.e. a variable not currently in the solution), then there are multiple optimal solutions.



27.2.7. Standard Form

- **Equalities**: Convert inequalities (using slack, surplus, or artificial variables)
- **Non-negative variables**: If a variable is unrestricted or can be negative, replace it with the difference of two non-negative variables

Example

$$2x_1 + 3x_2 \leq -4$$
$$\Longleftrightarrow$$
$$-2x_1 - 3x_2 \geq 4$$

Nonpositive

$$2x_1 + 3x_2 \leq 4, \quad x_1 \leq 0$$
$$\Longleftrightarrow$$
$$-2x_1 + 3x_2 \leq 4, \quad x_1 \geq 0$$

Free (Unrestricted)

$$2x_1 + 3x_2 \leq 6, \quad x_1 \text{ urs}$$
$$\Longleftrightarrow$$
$$2x_1' - 2x_1'' + 3x_2 \leq 4, \quad x_1', x_1'' \geq 0$$

27.2.8. Simplex Method

Example

$$\begin{aligned} \max \quad & 7x_1 + 6x_2 \\ s.t. \quad & 2x_1 + 4x_2 \leq 16 \\ & 3x_1 + 2x_2 \leq 12 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad & 7x_1 + 6x_2 + 0s_1 + 0\,s_2 \\ s.t. \quad & 2x_1 + 4x_2 + s_1 = 16 \\ & 3x_1 + 2x_2 + s_2 = 12 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

		x_1	x_2	s_1	s_2	
Basis	c_B	7	6	0	0	b
s_1	0	2	4	1	0	16
s_2	0	3	2	0	1	12

z_j	$0 \times 2 + 0 \times 3 = 0$	0	0	0	0
$c_j - z_j$	7	6	0	0	

$$x_1 = 0, x_2 = 0, s_1 = 16, s_2 = 12$$

		x_1	x_2	s_1	s_2		
Basis	c_B	7	6	0	0	b	Ratio
s_1	0	2	4	1	0	16	16/2 = 8
s_2	0	3	2	0	1	12	12/3 = 4
	z_j	0	0	0	0	0	
	$c_j - z_j$	7	6	0	0		

---

1. Formulation

Objective Function

$$\begin{aligned} Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\ &= c^Tx \end{aligned}$$

Constraints

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\ x_1, x_2, \dots, x_n \geq 0 \end{gathered}$$

Or,

$$\begin{gathered} Ax \leq, =, \geq b \\ x \geq 0 \end{gathered}$$

- $\leq$ for maximization
- $\geq$ for minimization

2. Convert to Canonical Form

In **slack variable form**

$$\begin{gathered} a_{11} + x_1 + a_{12}x_2 + \dots + a_{1n}x_n + s_1 = b_1 \\ a_{21} + x_1 + a_{22}x_2 + \dots + a_{2n}x_n + s_2 = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + s_m = b_m \end{gathered}$$

$$x_1, x_2, \dots, x_n \geq 0$$

$$s_1, s_2, \ldots, s_m \geq 0$$

Or **matrix notation**,

$$\begin{aligned} \max \quad & z = c^T x \\ s.t. \quad & Ax + Is = b_i \\ & x \geq 0 \\ & s \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad z \quad &= \quad c^T \quad x \\ &= [c_1, c_2, \ldots, c_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \end{aligned}$$

$$\underset{A}{\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}} \underset{x}{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}} + \underset{I}{\begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}} \underset{s}{\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}} = \underset{b}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}$$

3. Set up Simplex Tableau

	x_1	x_2	...	x_n	s_1	s_2	...	s_m	RHS
Constraint 1	a_{11}	a_{12}	...	a_1n	1	0	...	0	b_1
Constraint 2	a_{21}	a_{22}	...	a_2n	0	1	...	0	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Constraint n	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m
Objective Function	$-c_1$	$-c_2$	...	$-c_n$	0	0	...	0	0

- $-$ for maximization
- $+$ for minimization

4. Perform Pivot Operation

- **Identify the Pivot Column**: Choose the column with the most:
  - Maximization: Negative coefficient
  - Minimization: Positive coefficient (or multiply the objective by −1 and treat it as a maximization)

in the objective function row (this indicates which variable will enter the basis).

- **Identify the Pivot Row**: Calculate the ratio of RHS to the pivot column's coefficients (only where coefficients are positive). The row with the smallest ratio determines the variable to leave the basis.

- **Pivot**: Perform row operations to change the pivot element to 1 and other elements in the pivot column to 0. This involves dividing the pivot row by the pivot element and then adjusting the other rows to zero out the pivot column.

5. Iterate

Repeat the pivot operations until there are no more negative coefficients in the objective function row, indicating that the current solution is optimal.

6. Read the Solution

The final tableau will give the values of the variables at the optimal solution. The basic variables are the variables corresponding to the columns with the identity matrix in the final tableau, while non-basic variables are set to zero.

Example

$$\text{Maximize :} \quad Z = 3x_1 + 2x_2$$

s.t.

$$x_1 + x_2 \leq 4$$

$$2x_1 + x_2 \leq 5$$

$$x_1, x_2 \geq 0$$

1. Convert to Canonical Form

Introduce slack variables s_1 and s_2 to convert inequalities into equalities:

$$x_1 + x_2 + s_1 = 4$$

$$2x_1 + x_2 + s_2 = 5$$

$$x_1, x_2, s_1, s_2 \geq 0$$

Initialize Simplex Tableau

	x_1	x_2	s_1	s_2	RHS
Constraint 1	1	1	1	0	4
Constraint 2	2	1	0	1	5
Objective Function	–3	–2	0	0	0

2. Identify the Pivot Column

In the objective function row, the coefficients are $[-3, -2, 0, 0]$. The most negative coefficient is -3, which is in the x_1 column. Therefore, x_1 will enter the basis.

3. Identify the Pivot Row

Calculate the ratio of RHS to the pivot column's coefficients (where the coefficients are positive):

- For Row 1: $\frac{4}{1} = 4$ (pivot column coefficient is 1)

- For Row 2: $\frac{5}{2} = 2.5$ (pivot column coefficient is 2)

The smallest ratio is 2.5, so Row 2 will be the pivot row. This means that s_2 will leave the basis.

4. Pivot

Perform row operations to make the pivot element 1 and zero out other elements in the pivot column.

Pivot Element: The element at the intersection of Row 2 and x_1 column is 2.

Steps:

1. Make Pivot Element 1: Divide all elements in Row 2 by the pivot element (2):

$$\begin{aligned} \text{New Row 2} &= \frac{1}{2} \times \text{Old Row 2} \\ &= [1, 0.5, 0, 0.5, 2.5] \end{aligned}$$

2. Zero Out Pivot Column in Other Rows:
   - For Row 1: Subtract 1 times New Row 2 from Row 1:

$$\begin{aligned} \text{New Row 1} &= \text{Old Row 1} - [1, 0.5, 0, 0.5, 2.5] \\ &= [0, 0.5, 1, -0.5, 1.5] \end{aligned}$$

   - For Objective Function: Add 3 times New Row 2 to the Objective Function row

$$\begin{aligned} \text{New Objective Function} &= \text{Old Objective Function} + 3 \times [1, 0.5, 0, 0.5, 2.5] \\ &= [0, -0.5, 0, 1.5, 7.5] \end{aligned}$$

Updated Simplex Tableau

	x_1	x_2	s_1	s_2	RHS
Constraint 1	0	0.5	1	–0.5	1.5
Constraint 2	1	0.5	0	0.5	2.5
Objective Function	0	–0.5	0	1.5	7.5

27.3. Matrix Notation

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$$\begin{aligned} A &\in \mathbb{R}^{m \times n} \\ b &\in \mathbb{R}^{m \times 1} \\ c &\in \mathbb{R}^{n \times 1} \\ x &\in \mathbb{R}^{n \times 1} \end{aligned}$$

$$\begin{aligned} \max \quad & c_B^T x_B + c_N^T x_N \\ s.t. \quad & B x_B + N x_N = b \\ & x_B, x_N \geq 0 \end{aligned}$$

$$c^T = [c_B^T, c_N^T]$$
$$A = [B, N]$$

Where:

- x_B: Basic variables (in the basis)
- x_N: Non-basic variables (not in the basis)
- c_B: Coefficients of basic variables in the objective function
- c_N: Coefficients of non-basic variables in the objective function
- B: Columns of A corresponding to **basic** variables
- N: Columns of A corresponding to **non-basic** variables
- b: Right-hand side constants

Example

Standard form:

$$\begin{aligned} \max \quad & x_1 \\ s.t. \quad & 2x_1 - x_2 + x_3 = 4 \\ & 2x_1 + x_2 + x_4 = 8 \\ & x_2 + x_3 = 3 \\ & x_1, x_2, x_3, x_4 \geq 0 \quad \forall i = 1, ..., 5 \end{aligned}$$

$$c^T = [1, 0, 0, 0, 0]$$

$$A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}$$

Given $x_B = (x_1, x_4, x_5)$ and $x_N = (x_2, x_3)$:

$$c_B^T = [1, 0, 0], \qquad c_N^T = [0, 0]$$

$$B = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad N = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}$$

Objective Function

$$1x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 = 0$$

$$x_B^T x_b + x_N^T x_N = 0$$

Constraints

$$\begin{aligned} 2x_1 - 1x_2 + 1x_3 + 0x_4 + 0x_5 &= 4 \\ 2x_1 + 1x_2 + 0x_3 + 1x_4 + 0x_5 &= 8 \\ 0x_1 + 1x_2 + 0x_3 + 0x_4 + 1x_5 &= 3 \end{aligned}$$

$$A_B x_B + A_N x_N = b$$

Problem

$$\begin{aligned} \max \quad & c_B^T x_B + c_N^T x_N \\ s.t. \quad & B x_B + N x_N = b \\ & x_B, x_N \geq 0 \end{aligned}$$

Rearange the the terms in the constrains:

$$\begin{aligned} \max \quad & c_B^T x_B + c_N^T x_N \\ s.t. \quad & x_B = B^{-1}(b - N x_N) \\ & x_B, x_N \geq 0 \end{aligned}$$

Replace x_B in the objective function:

$$\begin{aligned} \max \quad & c_B^T [B^{-1}(b - N x_N)] + c_N^T x_N \\ s.t. \quad & x_B = B^{-1}(b - N x_N) \\ & x_B, x_N \geq 0 \end{aligned}$$

Rearrange the terms in the objectcive function:

$$\begin{aligned} \max \quad & c_B^T B^{-1} b - (c_B^T B^{-1} N - c_N^T) x_N \\ s.t. \quad & x_B = B^{-1}(b - N x_N) \\ & x_B, x_N \geq 0 \end{aligned}$$

The standard form LP becomes:

$$\begin{aligned} \max \quad & c_B^T B^{-1} b - (c_B^T B^{-1} N - c_N^T) x_N \\ s.t. \quad & x_B = B^{-1}(b - N x_N) \\ & x_B, x_N \geq 0 \end{aligned}$$

Rearange te terms of the constrains:

$$\begin{aligned} \max \quad & c_B^T B^{-1} b - (c_B^T B^{-1} N - c_N^T) x_N \\ s.t. \quad & I x_B + B^{-1} N x_N = B^{-1} b \\ & x_B, x_N \geq 0 \end{aligned}$$

Ignore the sign constraints and let z be the objectctive value:

$$\begin{array}{ccccc} z & & + & (c_B^T B^{-1} N - c_N^T) x_N & = & c_B^T B^{-1} b \\ & I x_B & + & B^{-1} N x_N & = & B^{-1} b \end{array}$$

The Simplex Tableau is:

$$\begin{array}{c|c|cc} 0 & c_B^T B^{-1} N - c_N^T & c_B^T B^{-1} b & 0 \\ \hline I & B^{-1} N x_N & B^{-1} b & 1, \ldots, m \end{array}$$

basic non-basic RHS

Example

Step 1. Standard form

$$\begin{array}{llllllllllll} \max & x_1 \\ s.t. & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\ & 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\ & & & x_2 & & & & & + & x_5 & = & 3 \\ & x_i \geq 0 & & \forall i = 1, \ldots, 5 \end{array}$$

Matrix notation:

$$c^T = [1, 0, 0, 0, 0], \qquad A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix}$$

Step 2. Initial Basis

$$B = (x_1, x_4, x_5) \qquad N = (x_2, x_3)$$

$$B = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad N = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Cost split:

$$c_B^T = [1, 0, 0] \qquad c_N^T = [0, 0]$$

Compute inverse of B:

$$B^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad x_B = B^{-1}b = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix}$$

So the basic feasible solution (BFS) is:

$$x = (x_1, x_2, x_3, x_4, x_5) = (2, 0, 0, 4, 3)$$

$$z = c_B^T B^{-1} b = [1, 0, 0] \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = 2$$

Step 3. Reduced costs and entering variable

$$\begin{aligned} \bar{c}_N^T &= c_B^T B^{-1} N - c_N^T \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \end{aligned}$$

Since $\bar{c}_2 = -\frac{1}{2} < 0$, the entering variable is x_2

Step 4. Direction, ratio test and leaving variable

For $x_B = (x_1, x_4, x_5)$, we have:

$$d = B^{-1} N_2 = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 2 \\ 1 \end{bmatrix}$$

Current basic solution:

$$x_B = A_B^{-1} b = \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix}$$

The minimum ratios test:

$$\frac{x_4}{2} = \frac{4}{2} = 2 \qquad \frac{x_5}{1} = \frac{3}{1} = 3$$

Smallest ratio is $2 \Rightarrow x_4$ leaves the basis

Step 5. Pivot to the new basis

$$B = (x_1, x_2, x_5) \qquad N = (x_3, x_4)$$

$$B = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Cost split:

$$c_B^T = [1, 0, 0] \qquad c_N^T = [0, 0]$$

Compute inverse of B:

$$B^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \qquad x_B = B^{-1} b = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}$$

So the new basic feasible solution (BFS) is:

$$x = (x_1, x_2, x_3, x_4, x_5) = (3, 2, 0, 0, 1)$$

$$z = c_B^T x_B = [1, 0, 0] \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = 3$$

Step 6. Check optimality (reduced costs)

$$
\begin{aligned}
\bar{c}_N^T &= c_B^T B^{-1} N - c_N^T \\
&= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \end{bmatrix}
\end{aligned}
$$

No negative reduced costs $\Rightarrow$ optimal

$$x^* = (3, 2, 0, 0, 1), \qquad z^* = 3$$

Example

Problem:

$$
\begin{array}{rrcrcl}
\max & 2x_1 & + & 3x_2 & & \\
s.t. & x_1 & + & x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 6 \\
 & x_1, x_2 \geq 0 & & & &
\end{array}
$$

Standard form:

$$
\begin{array}{rrcrcrcrcl}
\max & 2x_1 & + & 3x_2 & & & & & & \\
s.t. & x_1 & + & x_2 & + & s_1 & & & = & 4 \\
 & x_1 & + & 2x_2 & & & + & s_2 & = & 6 \\
 & x_1, x_2, s_1, s_2 \geq 0 & & & & & & & &
\end{array}
$$

Let $x_B = (s_1, s_2)$ and $x_N = (x_1, x_2)$

$$c_B = [0, 0] \qquad c_N = [2, 3]$$

$$A_B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad A_N = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

-2	-3	0	0	0
1	1	1	0	4
1	2	0	1	6
basic		non-basic		RHS

$A_B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $A_B^{-1}A_N = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ $A_B^{-1}b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$ $c_B^T A_B^{-1} A_N - c_N^T = [-2 \quad -3]$ $c_B^T A_B^{-1} b = 0$	
Let $x_B = (x_1, x_2)$ and $x_N = (s_1, s_2)$ $c_B = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad c_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $A_B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $b = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$ $A_B^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$ $A_B^{-1}A_N = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$ $A_B^{-1}b = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $c_B^T A_B^{-1} A_N - c_N^T = [1 \quad 1]$ $c_B^T A_B^{-1} b = 10$	$\begin{array}{cccc\|c} 0 & 0 & 1 & 1 & 10 \\ \hline 1 & 0 & 2 & -1 & 2 \\ 0 & 1 & -1 & 1 & 2 \\ \text{basic} & & \text{non-basic} & & \text{RHS} \end{array}$

Example

Step 1. Canonical form

$$
\begin{array}{lrcrcr}
\max & x_1 & + & 3x_2 & & \\
s.t. & -x_1 & + & x_2 & \leq & 3 \\
 & -x_1 & + & 2x_2 & \leq & 8 \\
 & 3x_1 & + & x_2 & \leq & 18 \\
 & & x_1, x_2 \geq 0 & & &
\end{array}
$$

Step 2. Standard form

$$
\begin{array}{lrcrcrcrcrcl}
\max & x_1 & + & 3x_2 & & & & & & & & \\
s.t. & -x_1 & + & x_2 & + & s_1 & & & & & = & 3 \\
 & -x_1 & + & 2x_2 & & & + & s_2 & & & = & 8 \\
 & 3x_1 & + & x_2 & & & & & + & s_3 & = & 18 \\
 & \multicolumn{11}{l}{x_1, x_2, s_1, s_2 \geq 0}
\end{array}
$$

Matrix notation:

$$
c^T = \begin{bmatrix} 1 & 3 & 0 & 0 & 0 \end{bmatrix}
$$

$$
A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix}
$$

Step 3. Initial Basis

$$
B = (s_1, s_2, s_3) \qquad N = (x_2, x_2)
$$

$$
B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad N = \begin{bmatrix} -1 & 1 \\ -1 & 2 \\ 3 & 1 \end{bmatrix}
$$

$$
c_B^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \qquad c_N^T = \begin{bmatrix} 1 & 3 \end{bmatrix}
$$

Compute B^{-1}:

$$
B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

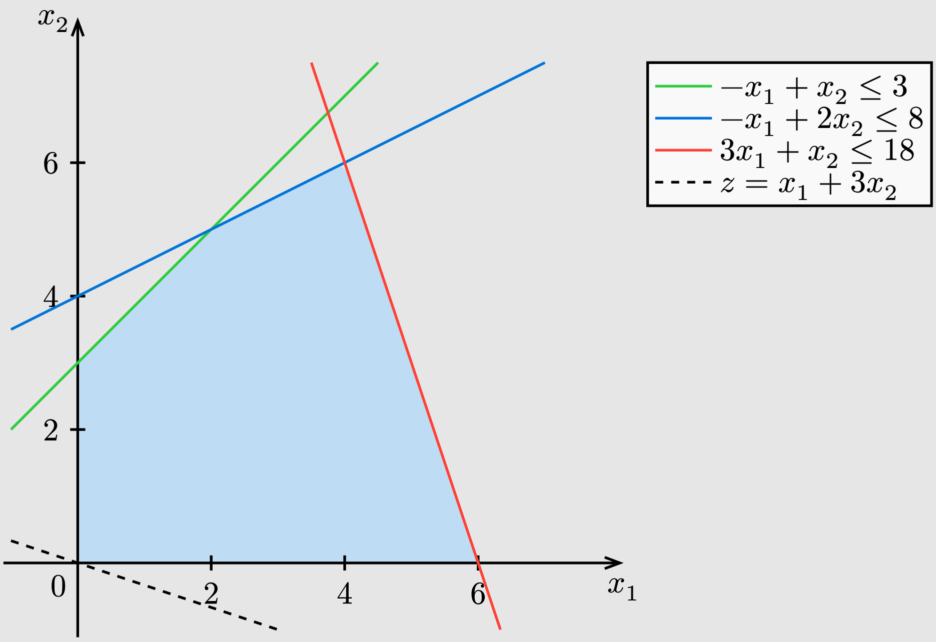
Compute basic feasible solution:

$$
x_B = B^{-1}b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}
$$

Current basic feasible solution:

$$
x = (x_1, x_2, s_1, s_2, s_3) = (0, 0, 3, 8, 18)
$$

$$z = c_B^T B^{-1} b = c_B^T x_B = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix} = 0$$



Step 4. Reduced costs and **entering variable**

$$\begin{aligned} \bar{c}_N^T &= c_B^T B^{-1} N - c_N^T \\ &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 2 \\ 3 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 3 \end{bmatrix} \\ &= \begin{bmatrix} -1 & -3 \end{bmatrix} \end{aligned}$$

Since $\bar{c}_1 = -1 < 0$, the **entering variable** is x_1

Step 5. Direction, ratio test and **leaving variable**

For $x_B = (s_1, s_2, s_3)$, we have:

$$d = B^{-1} N_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 3 \end{bmatrix}$$

Current basic solution:

$$x_B = B^{-1}b = \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix}$$

The minimum ratios test:

$$\frac{s_3}{3} = \frac{18}{3} = 6$$

Smallest ratio is 3 $\Rightarrow s_3$ **leaves the basis** (only consider ratios where the corresponding $d_i > 0$)

Step 6. Pivot to the new basis

$$B = (s_1, s_2, x_1) \qquad N = (s_3, x_2)$$

$$B = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{bmatrix} \qquad N = \begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}$$

$$c_B^T = [0 \quad 0 \quad 1] \qquad c_N^T = [0 \quad 3]$$

Compute B^{-1}:

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0.33 \\ 0 & 1 & 0.33 \\ 0 & 0 & 0.33 \end{bmatrix}$$

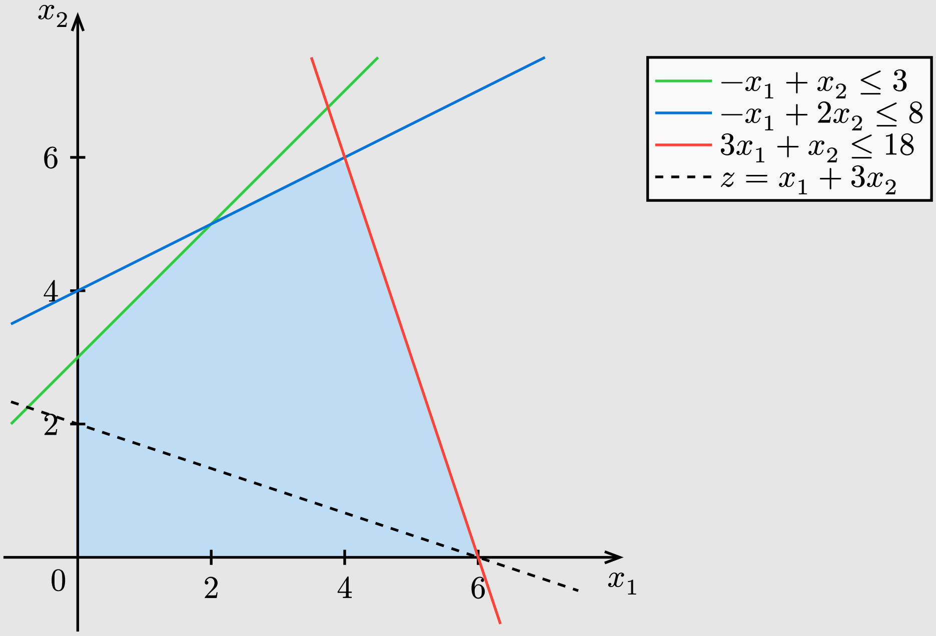
Compute basic feasible solution:

$$x_B = B^{-1}b = \begin{bmatrix} 1 & 0 & 0.33 \\ 0 & 1 & 0.33 \\ 0 & 0 & 0.33 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix} = \begin{bmatrix} 8.94 \\ 13.94 \\ 5.94 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ x_1 \end{bmatrix}$$

Current basic feasible solution:

$$x = (x_1, x_2, s_1, s_2, s_3) = (5.94, 0, 8.94, 13.94, 0)$$

$$z = c_B^T x_B = [0 \quad 0 \quad 1] \begin{bmatrix} 8.94 \\ 13.94 \\ 5.94 \end{bmatrix} = 5.94$$



Step 7. Reduced costs and **entering variable**

$$
\begin{aligned}
\bar{c}_N^T &= c_B^T B^{-1} N - c_N^T \\
&= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.33 \\ 0 & 1 & 0.33 \\ 0 & 0 & 0.33 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 3 \end{bmatrix} \\
&= \begin{bmatrix} 0.33 & -2.67 \end{bmatrix}
\end{aligned}
$$

Since $\bar{c}_2 = -2.67 < 0$, the **entering variable** is x_2

Step 8. Direction, ratio test and **leaving variable**

For $x_B = (s_1, s_2, x_1)$, we have:

$$
d = B^{-1} N_2 = \begin{bmatrix} 1 & 0 & 0.33 \\ 0 & 1 & 0.33 \\ 0 & 0 & 0.33 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.33 \\ 2.33 \\ 0.33 \end{bmatrix}
$$

Current basic solution:

$$
x_B = B^{-1} b = \begin{bmatrix} 8.94 \\ 13.94 \\ 5.94 \end{bmatrix}
$$

The minimum ratios test:

$$
\frac{s_1}{1} = \frac{8.94}{1.33} = 6.75, \qquad \frac{s_2}{2} = \frac{13.94}{2.33} = 6, \qquad \frac{x_1}{1} = \frac{5.94}{0.33} = 18
$$

Smallest ratio is $\frac{13.94}{2} \Rightarrow s_2$ **leaves the basis** (only consider ratios where the corresponding $d_i > 0$)

Step 9. Pivot to the new basis

$$B = (s_1, x_2, x_1) \qquad N = (s_2, s_3)$$

$$B = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & -1 \\ 0 & 1 & 3 \end{bmatrix} \qquad N = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$c_B^T = \begin{bmatrix} 0 & 3 & 1 \end{bmatrix} \qquad c_N^T = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Compute B^{-1}:

$$B^{-1} = \begin{bmatrix} 1 & -0.57 & 0.14 \\ 0 & 0.43 & 0.14 \\ 0 & -0.14 & 0.29 \end{bmatrix}$$

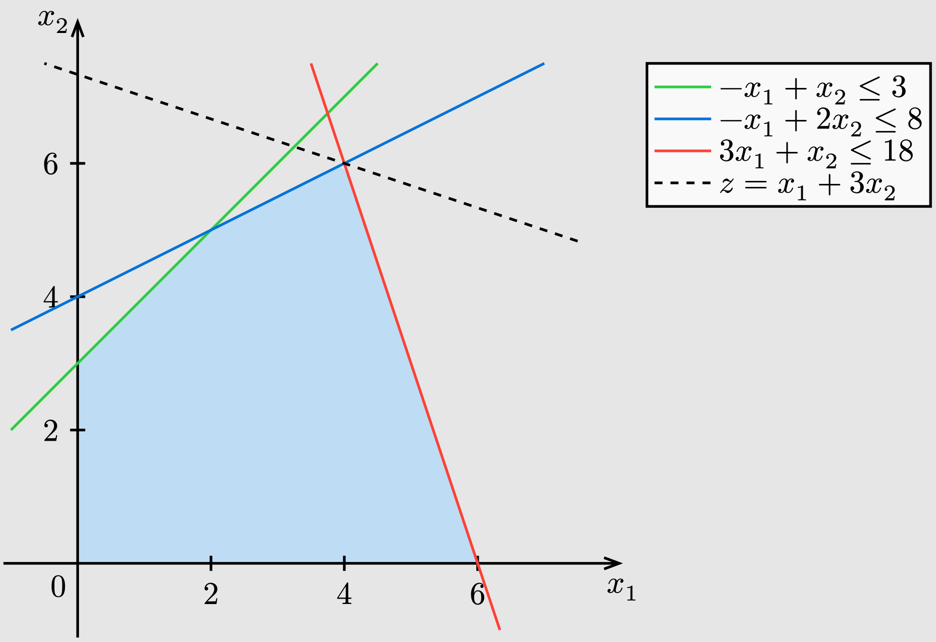
Compute basic feasible solution:

$$x_B = B^{-1}b = \begin{bmatrix} 1 & -0.57 & 0.14 \\ 0 & 0.43 & 0.14 \\ 0 & -0.14 & 0.29 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \\ 18 \end{bmatrix} = \begin{bmatrix} 0.96 \\ 5.96 \\ 4.1 \end{bmatrix} = \begin{bmatrix} s_1 \\ x_2 \\ x_1 \end{bmatrix}$$

Current basic feasible solution:

$$x = (x_1, x_2, s_1, s_2, s_3) = (4, 6, 1, 0, 0)$$

$$z = c_B^T x_B = \begin{bmatrix} 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.96 \\ 5.96 \\ 4.1 \end{bmatrix} = 22$$




```python
Nv = np.array(['x1', 'x2'])
Bv = np.array(['s1', 's2', 's3'])

A = np.array([
    [-1, 1, 1, 0, 0],
    [-1, 2, 0, 1, 0],
    [3, 1, 0, 0, 1],
])

c = np.array([1, 3, 0, 0, 0])

b = np.array([3, 8, 18])

cN = c[:2]
cB = c[2:]

N = A[:,:2]
B = A[:,2:]

print("Nv"), print(Nv)
print()
print("N"), print(N)
print()
print("cN"), print(cN)
print()
print("Bv"), print(Bv)
print()
print("B"), print(B)
print()
print("cB"), print(cB)
print()
print("b"), print(b)
print()
```

```
Binv = np.linalg.inv(B)
bfs = Binv @ b
z = cB @ bfs

print("B^(-1)"), print(Binv)
print()
print("BFS"), print(bfs)
print()
print("z"), print(z)
print()
```

```
reduced_cost = cB @ Binv @ N - cN

print("bar(c)_N^T"), print(reduced_cost)
print()

entering_var_index = 0

print("Entering var"), print(Nv[entering_var_index])
print()

d = Binv @ N[:,entering_var_index]
ratios = bfs / d

print(f"Ratios"), print(ratios)
print()

exiting_var_index = 2

print(f"Exiting variable"), print(Bv[exiting_var_index])
print()

B[:, exiting_var_index], N[:, entering_var_index] = N[:,
entering_var_index].copy(), B[:, exiting_var_index].copy()
cB[exiting_var_index], cN[entering_var_index] = cN[entering_var_index].copy(),
cB[exiting_var_index].copy()
Bv[exiting_var_index], Nv[entering_var_index] = Nv[entering_var_index].copy(),
Bv[exiting_var_index].copy()

print("Nv"), print(Nv)
print()
print("N"), print(N)
print()
print("cN"), print(cN)
print()
print("Bv"), print(Bv)
print()
print("B"), print(B)
print()
print("cB"), print(cB)
print()

Binv = np.linalg.inv(B)
```

```
bfs = Binv @ b
z = cB @ bfs

print("B^(-1)"), print(Binv)
print()
print("BFS"), print(bfs)
print()
print("z"), print(z)
print()
```

```
reduced_cost = cB @ Binv @ N - cN

print("bar(c)_N^T"), print(reduced_cost)
print()

entering_var_index = 1

print("Entering var"), print(Nv[entering_var_index])
print()

d = Binv @ N[:,entering_var_index]
ratios = bfs / d

print(f"Ratios"), print(ratios)
print()

exiting_var_index = 1

print(f"Exiting variable"), print(Bv[exiting_var_index])
print()

B[:, exiting_var_index], N[:, entering_var_index] = N[:,
entering_var_index].copy(), B[:, exiting_var_index].copy()
cB[exiting_var_index], cN[entering_var_index] = cN[entering_var_index].copy(),
cB[exiting_var_index].copy()
Bv[exiting_var_index], Nv[entering_var_index] = Nv[entering_var_index].copy(),
Bv[exiting_var_index].copy()

print("Nv"), print(Nv)
print()
print("N"), print(N)
print()
print("cN"), print(cN)
print()
print("Bv"), print(Bv)
print()
print("B"), print(B)
print()
print("cB"), print(cB)
print()

Binv = np.linalg.inv(B)
bfs = Binv @ b
z = cB @ bfs
```

```
print("B^(-1)"), print(Binv)
print()
print("BFS"), print(bfs)
print()
print("z"), print(z)
print()
```

27.4. Input Output Model (Leontief)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

Where:

- a_{ij}: units of good i needed to produce 1 unit of good j
- $\mathbf{x}$: total outputs for each sector
- $\mathbf{d}$: vector of goods used **externally by consumers**
- $\mathbf{Ax}$: vector of goods used **internally in production**

Technical coefficient: the amount of input required per unit of output

Balance Equation

Total production = production used as inputs for other sectors+final consumption

$$\mathbf{x} = \mathbf{Ax} + \mathbf{d}$$

Rewrite the Equation

1. Subtract $\mathbf{Ax}$ from both sides:

$$\mathbf{x} - \mathbf{Ax} = \mathbf{d}$$

2. Factor out $\mathbf{x}$:

$$\mathbf{x}(\mathbf{I} - \mathbf{A}) = \mathbf{d}$$

Given final demand $\mathbf{d}$ and input requirements $\mathbf{A}$, what total production $\mathbf{x}$ is needed?

2. Solving for Production

$$\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{d}$$

This gives exactly how much each sector must produce to satisfy both internal (inputs) and external (final demand) needs.

$(I - A)^{-1}$: Leontief inverse, each element shows the total output needed from sector i per unit of final demand in sector j, accounting for indirect effects

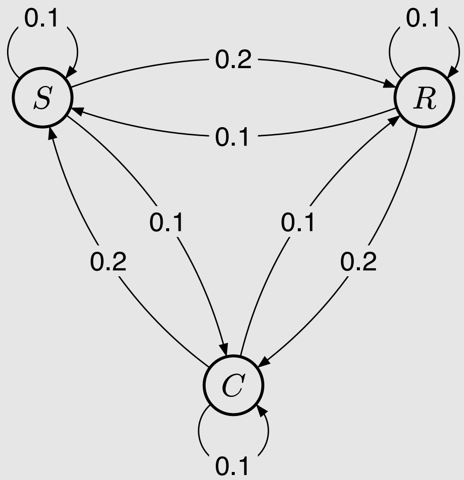
Example

- Steel (S)
- Rubber (R)
- Chemicals (C)

O / I	Steel	Rubber	Chemicals
Steel	0.1	0.2	0.1
Rubber	0.1	0.1	0.2
Chemicals	0.2	0.1	0.1

Resource	Demand
Steel	10
Rubber	20
Chemicals	30

$$\mathbf{A} = \begin{bmatrix} 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.1 \end{bmatrix} \qquad \mathbf{d} = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix}$$



$$\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{d}$$

$$\mathbf{x} = \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 23.42 \\ 34.23 \\ 42.34 \end{bmatrix}$$

- Steel ($x_1 = 23.42$): **total units** needed for final demand and as inputs to other sectors
- Rubber ($x_2 = 34.23$): **total units** for final demand plus inputs to all sectors
- Chemicals ($x_2 = 42.34$): **total units** including intermediate uses

27.4.1. Labor

1. Direct Labor

Each sector requires some amount of direct labor to produce a unit of output. Direct labor coefficient:

- l_k: hours of labor needed to produce **one unit of good** j

2. Indirect Labor

Producing goods often requires inputs from other sectors, which themselves required labor. To account for this:

$$\boldsymbol{\lambda} = \mathbf{l}(\mathbf{I} - \mathbf{A})^{-1}$$

- $\mathbf{l}$: vector of direct labor coefficients
- $\mathbf{A}$: input-output matrix
- $\boldsymbol{\lambda}$: **total (direct and indirect) labor required per unit** of each good

Interpretation: $\lambda_j$is the total socially necessary labor time embedded in one unit of good j

3. Labor as a "Currency"

- People earn labor vouchers corresponding to the hours they work.
- Each good has a price in labor time: λ_j hours per unit.
- If you work h hours, you get h labor vouchers to spend.

Budget constraint for a worker:

$$\sum_j \lambda_j c_j \leq h$$

Where:

- c_j: units of good j the worker wants to consume
- λ_j: labor time required to produce one unit of good j (includes direct and indirect) labor
- h: total labor hours the worker has contributed (i.e., the number of labor vouchers they possess)

Example

Suppose:

- Good 1: $\lambda_1 = 2 \text{ hours/unit}$
- Good 2: $\lambda_1 = 5 \text{ hours/unit}$
- Worker labor: $h = 20 \text{ hours}$

Constraint:

$$2c_1 + 5c_2 \leq 20$$

- If the worker wants 4 units of Good 1: $2 \times 4 = 8 \text{ hours}$
- Then they could afford at most $(20 - 8) \div 5 = 2.4 \text{ units}$ of Good 2

So their consumption plan is limited by their labor contribution

4. Labor Planning

Labor also acts as a constraint for the whole economy:

$$\mathbf{l}\mathbf{x} \leq L$$

Where:

- $\mathbf{x}$: vector of production
- L: total labor available in society
- This ensures that planned production doesn't require more labor than society can provide

Example

1. Stepup

Suppose the economy produces Bread (B) and Clothes (C). Let's define:

- $\mathbf{x} = (x_B, x_C)$: total production of each good
- $\mathbf{d} = (d_B, d_C)$: final demand (how much society wants to consume)
- $\mathbf{A} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.1 \end{bmatrix}$: input-output matrix
- $\mathbf{l} = (2, 5)$: direct labor per unit (hours) for Bread and Clothes
- $L = 100$: total labor available in society

2. Constraints

Input-Output Balance: production must satisfy both internal (inputs) and final demand

$$x_B \geq 0.1x_B + 0.2x_C + d_B$$
$$x_C \geq 0.3x_B + 0.1x_C + d_C$$

Labor Constraint: total labor used cannot exceed available labor

$$2x_B + 5x_C \leq 100$$

Non-negativity: you can't produce negative amounts

$$x_B \geq 0 \qquad x_C \geq 0$$

3. Objective: Maximize production

$$\max \quad w_B d_B + w_C d_C$$

Where:

- w_B, w_C: societal priority weights (e.g., bread might be more essential than clothes)
- d_B, d_C: quantities actually supplied to meet demand

Interpretation

- The LP decides how much of each product to product (x_B, x_C)
- It respects labor limits $(2x_B + 5x_C \leq 100)$
- It respects interdependencies (inputs for other products) via the matrix A
- It maximizes fulfillment of product demand, possibly weighted by social priorities

So, no abstract utility function needed — the LP is about physical goods, labor, and inputs

27.5. IP (Integer Programming)

27.6. Selection and Logical Relations on Binary Variables $x_i \in \{0, 1\}$

27.6.1. Cardinality Constraints

Given a subset $S \subseteq \{1, ..., n\}$:

- At least k of the items in S

$$\sum_{i \in S} x_i \geq k$$

Example

We want to select at least 2 of items 1, 2, 3

$$x_1 + x_2 + x_3 \geq 2$$

- At most k of the items in S

$$\sum_{i \in S} x_i \leq k$$

Example

We want to select at most 2 of items 1, 2, 3

$$x_1 + x_2 + x_3 \leq 2$$

- Exactly k of the items in S

$$\sum_{i \in S} x_i = k$$

Example

We want to select exactly 2 of items 1, 2, 3

$$x_1 + x_2 + x_3 = 2$$

27.6.2. Logical OR

Given a subset $S \subseteq \{1, ..., n\}$:

- At least 1 item in S

$$\sum_{i \in S} x_i \geq 1$$

Example

Select item 1 or item 2, or both

$$x_1 + x_2 \geq 1$$

27.6.3. Conditional Selection: "If X, then Y"

(Implication Constraints)

- If $x_a = 1$, then $x_b = 1$

$$x_a \leq x_b$$

Example

If item 1 is selected, then item 2 must also be selected

$$x_1 \leq x_2$$

x_a	x_b	Valid?
0	0	✅
0	1	✅
1	0	✅
1	1	❌

- If $x_a = 1$, then $x_b = 0$

$$x_a + x_b \leq 1$$

Example

If you choose project A, then you cannot choose project B (mutually exclusive).

x_a	x_b	Valid?
0	0	✅
0	1	✅
1	0	✅
1	1	❌

- If $x_a = 1$, then $x_b + x_c \leq 0$ (generalized)

$$x_b + x_c \leq M(1 - x_a)$$

Example

For binary variables $M = 2$ works

If you choose project A, then you must not choose either project B or C.

x_a	x_b	x_c	Valid?
1	0	0	✅
1	1	0	❌
1	1	1	❌
0	1	1	✅
0	0	1	✅

27.6.4. Conditional OR: “If not X, then select Y and Z”

- If $x_j = 0$, then at least k items in S must be selected (generalized)

$$\sum_{i \in S} x_i \geq k(1 - x_j)$$

Example

If you do not choose item 1, then you must choose both item 2 and item

If $x_1 = 0$, then $x_2 = x_3 = 1$

$$x_2 + x_3 \geq 2(1 - x_1)$$

x_1	x_2	x_3	LHS	RHS	Valid?
0	1	1	2	2	✅
0	1	0	1	2	❌
0	0	0	0	2	❌
1	0	0	0	0	✅
1	1	0	1	0	✅
1	1	1	2	0	✅

27.6.5. XOR / Exclusive Conditions

- Exactly one item in a set S

$$\sum_{i \in S} x_i = 1$$

- Select one or the other but not both (i.e., $x_i \oplus x_k$)

$$x_a + x_b = 1$$

Example

You can select item 1 or item 2, but not both

x_1	x_2	Sum	Valid?
0	1	1	✅
1	0	1	✅
1	1	2	❌
0	0	0	❌

27.7. Constraint Activation and Relaxation via Binary Indicators $z_i \in \{1, 0\}$

27.7.1. Base Case: At Least One of Two Constraints Satisfied

You are given 2 constraints

$$g_1(x) \leq b_1$$
$$g_2(x) \leq b_2$$

We want at least one of these two constraints to be satisfied.

Binary Indicator Variables

Define:

$$z_i = \begin{cases} 1 & \text{if } g_{i(x)} \leq b_i \text{ is enforced} \\ 0 & \text{if } g_{i(x)} \leq b_i \text{ may be violated} \end{cases} \quad \text{for } i = 1, 2$$

Let M_1 and M_2 be upper bounds on how much each constraint's LHS $g_{i(x)}$ can exceed b_i:

$$M_i \geq \max_x \left(g_{i(x)} - b_i\right)$$

Big-M Reformulation

We relax each constraint as follows:

$$g_1(x) - b_1 \leq M_1(1 - z_1)$$
$$g_2(x) - b_2 \leq M_2(1 - z_2)$$

When $z_i = 1$, the original constraint is enforced

When $z_i = 0$, the original constraint can be violated by up to M_i

At Least One Constraint Enforced

We require:

$$z_1 + z_2 \geq 1$$

27.7.2. General Formulation for m Constraints with At Least k Satisfied

1. Introduce a binary variable $z_i \in \{0, 1\}$ for each constraint $g_{i(x)} \leq b_i$, where:
   - $z_i = 1$: the constraint is enforced (i.e., $g_{i(x)} \leq b_i$)
   - $z_i = 0$: the constraint may be violated
2. Use “big-M” constraint to relax $g_{i(x)} \leq b_i$ when $z_i = 0$, like:

$$g_{i(x)} - b_i \leq M_i(1 - z_i), \quad \forall i = 1, ..., m$$

Where $M_i \geq \max_x \left(g_{i(x)} - b_i\right)$

3. Impose a condition that at least k of the m constraints must be satisfied:

$$\sum_{i=1}^{m} z_i \geq k$$

27.8. Fixed Charge Constraints (Setup Costs)

- n: Number of factories
- K_i: Capacity of factory i
- C_i: Unit production cost at factory i
- S_i: Fixed setup cost for factory i
- D: Total Demand

Decision Variables

- x_i: production quantity at factory i, $i = 1, ..., n$
- y_i: Binary variable indicating whether factory i is used

$$y_i = \begin{cases} 1 & \text{if factory } i \text{ is used (i.e., } x_i > 0) \\ 0 & \text{if factory } i \text{ is not used (i.e, } x_i = 0) \end{cases}$$

Objective Function

$$\min \underbrace{\sum_{i=1}^{n} C_i x_i}_{\text{Production Cost}} + \underbrace{\sum_{i=1}^{n} S_i y_i}_{\text{Setup Cost}}$$

Capacity Constraints

$$x_i \leq K_i \quad \forall i = 1, ..., n$$

Demand Fullfilment

$$\sum_{i=1}^{n} x_i \geq D$$

Logical Linking of x_i and y_i

To ensure $y_i = 1$ if any production occurs at factory i:

$$x_i \leq K_i y_i \quad \forall i = 1, ..., n$$

- If $x_i > 0$, this forces $y_i = 1$
- If $x_i = 0$, y_i can be 0 or 1 (but to minimize cost, y_i will be 0)

Binary & Non-negative Constraints

$$y_i \in \{0, 1\}, \quad x_i \geq 0 \quad \forall i = 1, ..., n$$

General Form (Big-M Formulation)

$$x \leq My$$

Where M must be set to the upper bound of x

27.9. Facility Location

Problem	**Definition**
Set Covering	Select the minimum number of facilities such that every demand point is within a specified coverage distance of at least one facility
Maximum Covering	Select a fixed number of facilities to maximize the number of demand points covered within a specified distance
Fixed Charge Location	Determine which facilities to open and how to assign demand to them to minimize the total cost

27.9.1. Set Covering Problem

Question: How to allocate as few facilities as possible to cover all demand nodes?

- I: Set of demand nodes
- J: Set of location nodes
- $d_{ij} > 0$: Distance between demand node $i \in I$ and location node $j \in J$
- s: Maximum allowable distance (a facility at j can cover demand node i if $d_{ij} < s$)
- Demand i is "covered" by location j if $d_{ij} < s$
- $w_j > 0$: Cost of building facility at location j (usually set to $w_j = 1$ for minimizing number of facilities, but can reflect other costs)
- $a_{ij} = \begin{cases} 1 & \text{if } d_{ij} < s \\ 0 & \text{otherwise} \end{cases}$: Indicator of whether location j can cover demand i
- $x_j = \begin{cases} 1 & \text{if facility is built at location } j \\ 0 & \text{otherwise} \end{cases}$

Complete Formulation

$$
\begin{aligned}
\min \quad & \sum_{j \in J} w_j x_j \\
s.t. \quad & \sum_{j \in J} a_{ij} x_j \geq 1 \qquad \forall i \in I \\
& x_j \in \{0,1\} \qquad \forall j \in J
\end{aligned}
$$

27.9.2. Maximum Covering Problem

Question: How to allocate at most p facilities to cover as many demand nodes as possible?

- I: Set of demand nodes
- J: Set of location nodes
- $d_{ij} > 0$: Distance between demand node $i \in I$ and location node $j \in J$
- $p \in \mathbb{N}$: maximum number of facilities
- $w_j > 0$: Cost of building facility at location j (usually set to $w_j = 1$ for minimizing number of facilities, but can reflect other costs)
- $x_j = \begin{cases} 1 & \text{if facility is built at location } j \\ 0 & \text{otherwise} \end{cases}$
- $a_{ij} = \begin{cases} 1 & \text{if } d_{ij} < s \\ 0 & \text{otherwise} \end{cases}$: Indicator of whether location j can cover demand i
- $y_i = \begin{cases} 1 & \text{if demand } i \in I \text{ is covered by any facility} \\ 0 & \text{otherwise} \end{cases}$

Complete Formulation

$$
\begin{aligned}
\min \quad & \sum_{i \in I} w_i y_i \\
s.t. \quad & \sum_{j \in J} a_{ij} x_j \geq y_i \qquad \forall i \in I \\
& \sum_{j \in J} x_j \leq p \qquad \forall j \in J \\
& x_j \in \{0,1\} \qquad \forall j \in J \\
& y_i \in \{0,1\} \qquad \forall i \in I
\end{aligned}
$$

Fixed Charge Location Problems

Question: How to allocate some facilities to minimize the total shipping and construction costs?

Uncapacitated Facility Location Problem (UFL)

- I: Set of demand nodes
- J: Set of location nodes
- $h_i > 0$: Demand size at demand node i
- d_{ij}: Unit shipping cost from location node j to demand node i

- $f_j > 0$: Fixed construction cost at location j
- $x_j = \begin{cases} 1 & \text{if facility is built at location } j \\ 0 & \text{otherwise} \end{cases}$
- $y_{ij} = \begin{cases} 1 & \text{if demand } i{\in}I \text{ is served by facility at location } j{\in}J \\ 0 & \text{otherwise} \end{cases}$

Complete Formulation

$$\begin{aligned}
\min \quad & \underbrace{\sum_{i\in I}\sum_{j\in J} h_i d_{ij} y_{ij}}_{\text{Shipping Costs}} + \underbrace{\sum_{j\in J} f_j x_j}_{\text{Fixed Costs}} \\
s.t. \quad & y_{ij} \le x_j \quad \forall i \in I, j \in J \\
& \sum_{j\in J} y_{ij} = 1 \quad \forall i \in I \\
& x_j \in \{0,1\} \quad \forall j \in J \\
& y_{ij} \in \{0,1\} \quad \forall i \in I
\end{aligned}$$

Capacitated Facility Location Problem (CFL)

If locations have a capacity, we may add constraint

$$\sum_{i\in I} h_i y_{ij} \le K_j \quad \forall j \in J$$

Where:

- $K_j > 0$: Capacity of location j

Example

$$\begin{aligned}
\min \quad & \underbrace{\sum_{j=1}^{5} f_j x_j}_{\text{Fixed Costs}} + \underbrace{\sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij} y_{ij}}_{\text{Shipping Costs}} \\
s.t. \quad & \sum_{i=1}^{5} y_{ij} \le K_j x_j && \forall j = 1,\dots,5 && \text{(Capacity Constraint)} \\
& \sum_{j=1}^{5} y_{ij} \ge D_i && \forall i = 1,\dots,5 && \text{(Demand Constraint)} \\
& x_j \in \{0,1\} && \forall j = 1,\dots,5 && \text{(Binary Constraint)} \\
& y_{ij} \ge 0 && \forall i = 1,\dots,5, j = 1,\dots,5 && \text{(Positivity Constraint)}
\end{aligned}$$

Parameters

- f_j: weekly operating cost of distribution center j
- c_{ij}: shipping cost per book from distribution center j to region i
- K_j: capacity of distribution center j
- D_i: book demand of region i

Decision variable

- x_{ij}: binary variable

$$x_{ij} = \begin{cases} 1 & \text{if a distribution center is build at location } j \\ 0 & \text{otherwise} \end{cases}$$

- y_{ij}: number of books shipped from distribution center j to region i

Supply

	WA	NV	NE	PA	FL	Demand (D_i)
NorthWest	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	8000
SouthWest	y_{21}	y_{22}	y_{23}	y_{24}	y_{25}	12000
MidWest	y_{31}	y_{32}	y_{33}	y_{34}	y_{35}	9000
SouthEast	y_{41}	y_{42}	y_{43}	y_{44}	y_{45}	14000
NorthEast	y_{51}	y_{52}	y_{53}	y_{54}	y_{55}	17000

Fixed Costs (f_j) and Capacity (K_j)

	x_1	x_2	x_3	x_4	x_5
Operation Cost (f_i)	40000	30000	25000	40000	30000
Capacity (K_i)	20000	20000	15000	25000	15000

Shipping Cost (c_j)

	WA	NV	NE	PA	FL
NorthWest	2.4	3.25	4.05	5.25	6.95
SouthWest	3.5	2.3	3.25	6.05	5.85
MidWest	4.8	3.4	2.85	4.3	4.8
SouthEast	6.8	5.25	4.3	3.25	2.1
NorthEast	5.75	6	4.75	2.75	3.5

Binary Decision Variable

WA	x_1
NV	x_2
NE	x_3
PA	x_4
FL	x_5

model.py

model.py

```
from pyomo.environ import *
from pyomo.dataportal import DataPortal

model = AbstractModel()

# Sets
model.I = Set(doc='Regions (demand points)')
model.J = Set(doc='Candidate distribution centers')

# Parameters
model.D = Param(model.I, within=NonNegativeReals, doc='Demand at region i')
model.K = Param(model.J, within=NonNegativeReals, doc='Capacity at
distribution center j')
model.f = Param(model.J, within=NonNegativeReals, doc='Fixed cost to open
center j')
model.c = Param(model.I, model.J, within=NonNegativeReals, doc='Shipping cost
from j to i')

# Decision Variables
model.x = Var(model.J, within=Binary, doc='1 if center j is opened')
model.y = Var(model.I, model.J, within=NonNegativeReals, doc='Units shipped
from j to i')

# Objective Function: Minimize total cost
def total_cost_rule(model):
    fixed = sum(model.f[j] * model.x[j] for j in model.J)
    shipping = sum(model.c[i, j] * model.y[i, j] for i in model.I for j in
model.J)
    return fixed + shipping
model.TotalCost = Objective(rule=total_cost_rule, sense=minimize)

# Constraints

# Capacity Constraint: Total shipped from center ≤ capacity if opened
def capacity_rule(model, j):
    return sum(model.y[i, j] for i in model.I) <= model.K[j] * model.x[j]
model.CapacityConstraint = Constraint(model.J, rule=capacity_rule)

# Demand Constraint: Total received by region ≥ its demand
def demand_rule(model, i):
    return sum(model.y[i, j] for j in model.J) >= model.D[i]
model.DemandConstraint = Constraint(model.I, rule=demand_rule)

# Load data from .dat file
data = DataPortal()
data.load(filename='data.dat', model=model)

# Create an instance of the model
instance = model.create_instance(data)

# Create solver
solver = SolverFactory('glpk')
solver.options['tmlim'] = 60
```

```
# Solve with solver timeout (optional)
results = solver.solve(instance, tee=True)

# Display results
instance.display()
```

data.dat

```
set I := NorthWest SouthWest MidWest SouthEast NorthEast;
set J := WA NV NE PA FL;

param: D :=
NorthWest 8000
SouthWest 12000
MidWest 9000
SouthEast 14000
NorthEast 17000;

param: f :=
WA 40000
NV 30000
NE 25000
PA 40000
FL 30000;

param: K :=
WA 20000
NV 20000
NE 15000
PA 25000
FL 15000;

param c:
        WA     NV     NE     PA     FL :=
NorthWest 2.4   3.25   4.05   5.25   6.95
SouthWest 3.5   2.3    3.25   6.05   5.85
MidWest   4.8   3.4    2.85   4.3    4.8
SouthEast 6.8   5.25   4.3    3.25   2.1
NorthEast 5.75  6      4.75   2.75   3.5 ;
```

Output:

```
Variables:
x : 1 if center j is opened
    Size=5, Index=J
    Key : Lower : Value : Upper : Fixed : Stale : Domain
     FL :     0 :   1.0 :     1 : False : False : Binary
     NE :     0 :   0.0 :     1 : False : False : Binary
     NV :     0 :   1.0 :     1 : False : False : Binary
```

```
     PA :     0 :   1.0 :     1 : False : False : Binary
     WA :     0 :   0.0 :     1 : False : False : Binary
y : Units shipped from j to i
    Size=25, Index=I*J
    Key                 : Lower : Value   : Upper : Fixed : Stale : Domain
      ('MidWest', 'FL') :     0 :  1000.0 :  None : False : False :
NonNegativeReals
      ('MidWest', 'NE') :     0 :     0.0 :  None : False : False :
NonNegativeReals
      ('MidWest', 'NV') :     0 :     0.0 :  None : False : False :
NonNegativeReals
      ('MidWest', 'PA') :     0 :  8000.0 :  None : False : False :
NonNegativeReals
      ('MidWest', 'WA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthEast', 'FL') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthEast', 'NE') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthEast', 'NV') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthEast', 'PA') :     0 : 17000.0 :  None : False : False :
NonNegativeReals
    ('NorthEast', 'WA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthWest', 'FL') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthWest', 'NE') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthWest', 'NV') :     0 :  8000.0 :  None : False : False :
NonNegativeReals
    ('NorthWest', 'PA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('NorthWest', 'WA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthEast', 'FL') :     0 : 14000.0 :  None : False : False :
NonNegativeReals
    ('SouthEast', 'NE') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthEast', 'NV') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthEast', 'PA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthEast', 'WA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthWest', 'FL') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthWest', 'NE') :     0 :     0.0 :  None : False : False :
NonNegativeReals
    ('SouthWest', 'NV') :     0 : 12000.0 :  None : False : False :
NonNegativeReals
    ('SouthWest', 'PA') :     0 :     0.0 :  None : False : False :
NonNegativeReals
```

```
    ('SouthWest', 'WA') :     0 :     0.0 :  None : False : False :
NonNegativeReals

Objectives:
  TotalCost : Size=1, Index=None, Active=True
      Key  : Active : Value
      None :   True : 268950.0

Constraints:
  CapacityConstraint : Size=5
      Key : Lower : Body : Upper
      FL :  None :  0.0 :   0.0
      NE :  None :  0.0 :   0.0
      NV :  None :  0.0 :   0.0
      PA :  None :  0.0 :   0.0
      WA :  None :  0.0 :   0.0
  DemandConstraint : Size=5
      Key       : Lower   : Body    : Upper
        MidWest :  9000.0 :  9000.0 :  None
      NorthEast : 17000.0 : 17000.0 :  None
      NorthWest :  8000.0 :  8000.0 :  None
      SouthEast : 14000.0 : 14000.0 :  None
      SouthWest : 12000.0 : 12000.0 :  None
```

27.10. Machine Scheduling

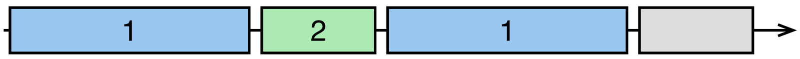
	Problem	**Definition**
One Machine	Single Machine Serial Production	Scheduling jobs one after another on a single machine
Multiple Machines	Multiple Parallel Machines	Scheduling jobs on two or more identical machines working in parallel; each job can be assigned to any machine
	Flow Shop	All jobs follow the same sequence of machines; each job visits every machine in the same order
	Job Shop	Each job has its own specific sequence of machines to follow

- Job Splitting
    - Non-premptive problems

  Once a job starts processing on a machine, it must continue without interruption until completion

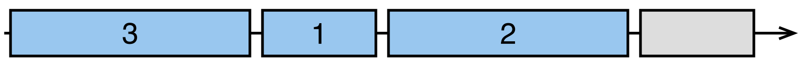
    - Premptive problems

A job can be interrupted and resumed later, possibly on a different machine, without losing progress



- Performance Metrics
  - **Makespan**: The sum of completion times of all jobs, possibly weighted by job importance
  - **(Weighted) total completion time**: The sum of completion times of all jobs, possibly weighted by job importance (measures overall throughput)
  - **(Weighted) number of delayed jobs**: The total number of jobs completed after they are due, possibly weighted by job priority
  - **(Weighted) total lateness**: The sum of differences between each job's completion time and due date (can be negative or positive), possibly weighted
  - **(Weighted) total tardiness**: The sum of positive delays (i.e., lateness when a job finishes after its due date), possibly weighted. Negative values (early completions) are treated as zero.
  - …

27.10.1. Single Machine Serial Production



Notation

- n: Total number of jobs
- $J = \{1, 2, ..., n\}$: Set of jobs
- $j \in J$: A job
- p_j: processing time of job j
- x_j: completion time of job j
- $z_{ij} \in \{0, 1\}$: Binary variable

$$z_{ij} = \begin{cases} 1 & \text{if job } j \text{ is before job } i \\ 0 & \text{otherwise} \end{cases}$$

Job Order Constraints

We must ensure that either job i comes before job j or vice versa, but not both

If job i is before job j ($z_{ij} = 0$)

$$x_j \geq x_i + p_j$$

If job j is before job i ($z_{ij} = 1$):

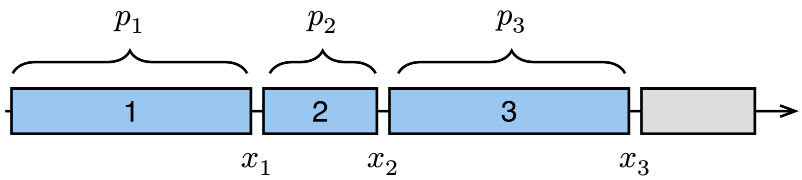
$$x_i \geq x_j + p_i$$

These two cases can be rewritten using a big-M formulation:

$$\begin{aligned} x_i + p_j - x_j &\leq M z_{ij} \\ x_j + p_i - x_i &\leq M(1 - z_{ij}) \end{aligned}$$

Where:

- $M = \sum_{j \in J} p_j$



Complete Formulation

$$\begin{aligned} \min \quad & \sum_{j \in J} x_j \\ s.t. \quad & x_i + p_j - x_j \leq M z_{ij} && \forall i \in J, j \in J, i < j \\ & x_j + p_i - x_i \leq M(1 - z_{ij}) && \forall i \in J, j \in J, i < j \\ & x_j \geq p_j && \forall j \in J \\ & x_j \geq 0 && \forall j \in J \\ & z_{ij} \in \{0, 1\} && \forall i \in J, j \in J, i < j \end{aligned}$$

Interpretation

If jobs are run in order $1, 2, ..., n$, the completion times are:

$$\begin{aligned} x_1 &= p_1 \\ x_2 &= p_1 + p_2 \\ x_3 &= p_1 + p_2 + p_3 \\ &\vdots \\ x_n &= \sum_{i=1}^{n} p_i \end{aligned}$$

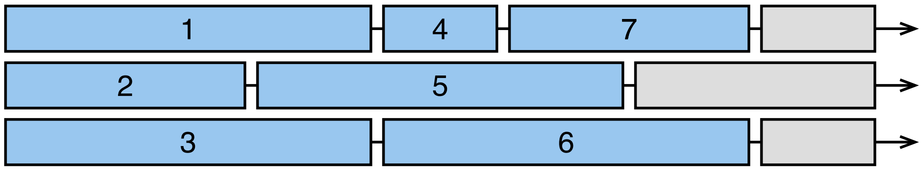
So, the completion time of a job depends on the sum of processing times of all jobs before it in the schedule

Optional: Release Times

If jobs have release times R_j, meaning each job j cannot start before R_j, then we add:

$$x_j \geq R_j + p_j \quad \forall j \in J$$

27.10.2. Multiple Parallel Machines



We aim to assign jobs to machines such that:

- Each job is assigned to exactly one machine.
- A job can be processed on any machine.
- The objective is to minimize the makespan, which is the time when the last machine finishes processing.

Notation

- $J = \{1, 2, ..., n\}$: set of jobs
- $I = \{1, 2, ..., m\}$: set of machines
- p_j: processing time of job $j \in J$
- x_{ij}: a binary variable that indicates whether job j is assigned to machine i:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } i \\ 0 & \text{otherwise} \end{cases}$$

- The completion time of machine i is the total processing time of all jobs assigned to it

$$C_i = \sum_{j \in J} p_j x_{ij}$$

- The makespan w is the maximum completion time accross all machines:

$$w \geq \sum_{j \in J} p_j x_{ij} \quad \forall i \in I$$

Objective

Minimize makespan w

Complete Formulation

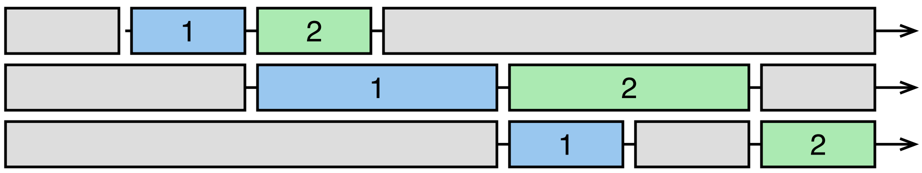
$$\begin{aligned} \min \quad & w \\ s.t. \quad & \sum_{j \in J} p_j x_{ij} \leq w & \forall i \in I \\ & \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\ & x_{ij} \in \{0, 1\} & \forall i \in I, j \in J \end{aligned}$$

- The first constraint ensures the makespan is at least as large as each machine's workload
- The second ensures each job is assigned to one and only one machine

model.py

```
from pyomo.environ import *

model = AbstractModel()

model.MACHINES = Set()
model.JOBS = Set()

model.p = Param(model.JOBS, within=PositiveReals)

model.x = Var(model.MACHINES, model.JOBS, domain=Binary)
model.w = Var(domain=NonNegativeReals)

def obj_rule(m):
    return m.w
model.Obj = Objective(rule=obj_rule, sense=minimize)

def job_assignment_rule(m, j):
    return sum(m.x[i, j] for i in m.MACHINES) == 1
model.JobAssignment = Constraint(model.JOBS, rule=job_assignment_rule)

def machine_completion_rule(m, i):
    return sum(m.p[j] * m.x[i, j] for j in m.JOBS) <= m.w
model.MachineCompletion = Constraint(model.MACHINES,
rule=machine_completion_rule)
```

27.10.3. Flow Shop



27.10.4. Job Shop

27.11. Vehicle Routing

27.11.1. Traveling Salesperson

- Let $G = (V, E)$ be a directed complete graph
- Let d_{ij} be the distance (cost) from node i to j
- Let x_{ij} be a binary decision variable:

$$x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Objective

Minimize the total travel cost

$$\min \sum_{(i,j)\in E} d_{ij} x_{ij}$$

Constraints

1. Flow Balancing

For node $k \in V$

- One **incoming** edge:

$$\sum_{i\in V, i\neq k} = 1$$

- One **outgoing** edge:

$$\sum_{j\in V, j\neq k} = 1$$

2. Subtours

a. Miller-Tucker-Zemlin (MTZ)

- Let $u_i s$: be the position of node i in the tour (e.g., $u_i = k$ if node i is the kth node to be passed on the tour)

$$\begin{aligned} & u_1 = 1 & \\ & 2 \leq u_i \leq n & \forall i \in V \setminus \{1\} \\ & u_i - u_j + 1 \leq (n-1)\big(1 - x_{ij}\big) & \forall (i,j) \in E, i \neq 1, j \neq 1 \end{aligned}$$

b. Subtour Elimination Constraint (SEC)

For every subset $S \subset V$ with $|S| \geq 2$:

$$\sum_{\substack{i\in S, j\in S \\ i\neq j}} x_{ij} \leq |S| - 1 \quad \forall S \subset V, |S| \geq 2$$

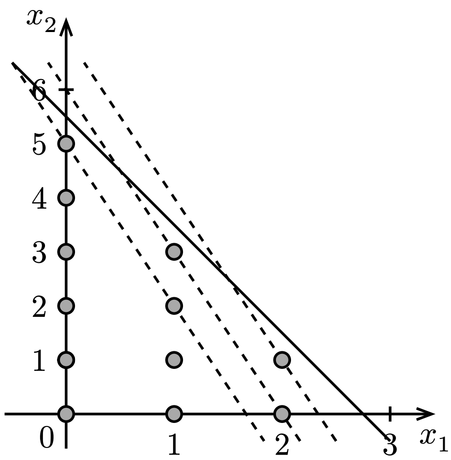
This ensures that no subset of nodes forms a closed tour independent of the main tour.

Complete Formulation

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} d_{ij}x_{ij} & \\
s.t. \quad & \sum_{i\in V, i\neq k} = 1 & \forall k \in V \\
& \sum_{j\in V, j\neq k} = 1 & \forall k \in V \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in E \\
& \text{MTZ or SEC} &
\end{aligned}$$

27.12. Branch-and-Bound

Integer Program



Linear Relaxation:

IP $\longrightarrow$ LP

Decompose an IP into mutltiple LPs

Definition 0: Linear Relaxation

For a Given LP, its linear relaxation is the resulting LP after removing all integer constraints

Example

Integer Problem:

$$\begin{array}{llllll}
\max & 3x_1 & + & x_2 & & \\
s.t. & 4x_1 & + & 2x_2 & \leq & 11 \\
& & {\color{red} x_i \in \mathbb{Z}_+} & & \forall i = 1, 2 &
\end{array}$$

Linear Relaxation:

$$\begin{array}{lrcrcr} \max & 3x_1 & + & x_2 & & \\ s.t. & 4x_1 & + & 2x_2 & \leq & 11 \\ & \multicolumn{5}{c}{x_i \geq 0 \quad \forall i = 1,2} \end{array}$$

Example

Integer Program:

$$\begin{array}{lrcrcrcrcr} \max & 16x_1 & + & 22x_2 & + & 12x_3 & + & 8x_4 & & \\ s.t. & 5x_1 & + & 7x_2 & + & 4x_3 & + & 3x_4 & \leq & 10 \\ & \multicolumn{9}{l}{x_i \in \{0,1\} \quad \forall i = 1, \dots, 4} \end{array}$$

Linear Relaxation:

$$\begin{array}{lrcrcrcrcr} \max & 16x_1 & + & 22x_2 & + & 12x_3 & + & 8x_4 & & \\ s.t. & 5x_1 & + & 7x_2 & + & 4x_3 & + & 3x_4 & \leq & 10 \\ & \multicolumn{9}{l}{x_i \in [0,1] \quad \forall i = 1, \dots, 4} \end{array}$$

$$x_i \in [0,1] \quad \Longleftrightarrow \quad x_i \geq 0 \wedge x_i \leq 1$$

Linear relaxation provides a bound

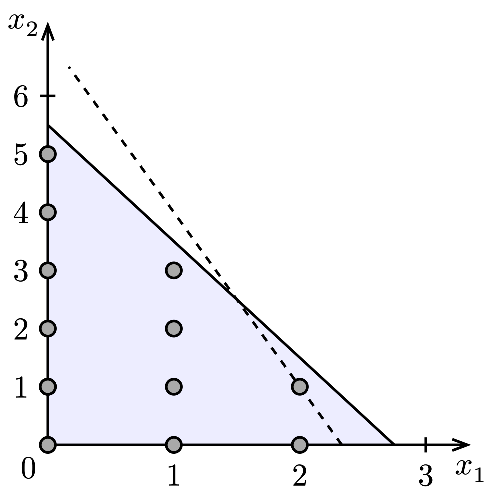
- z^*: Optimal value of the Integer Program (IP)
- z': Optimal value of the Linear Relaxation (LP)

1. Minimization

- For **minimization** IP, linear relaxation provides a **lower bound**
- The feasible region of the LP contains all feasible solutions of the IP (i.e., superset)
- Therefore, the LP can achieve a value that is at most as large as the IP
- Conclusion:

$$z' \leq z^*$$

So, linear relaxation provides a lower bound on the optimal value of the integer program

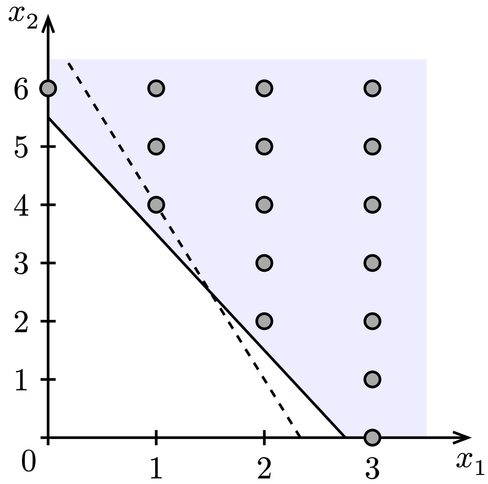


2. Maximization

- For **maximization** IP, linear relaxation provides a **upper bound**
- The LP may find a solution with an objective value greater than or equal to that of any feasible integer solution.
- Conclusion:

$$z' \geq z^*$$

- So, linear relaxation provides an upper bound on the optimal value of the integer program



Problem Type	**Linear Relaxation Provides**	**Inequality Between LP and IP**
Minimization IP	Lower Bound	$z' \leq z^*$
Maximization IP	Upper Bound	$z' \geq z^*$

- If the linear relaxation is infeasible or unbounded then the IP is infeasible or unbounded
- If an optimal solution to the linear relaxation is feasible,

Let x' be an optimal solutions to the linear relaxation of an IP. If x' is feasible to the IP, it is the optimal to the IP

Example

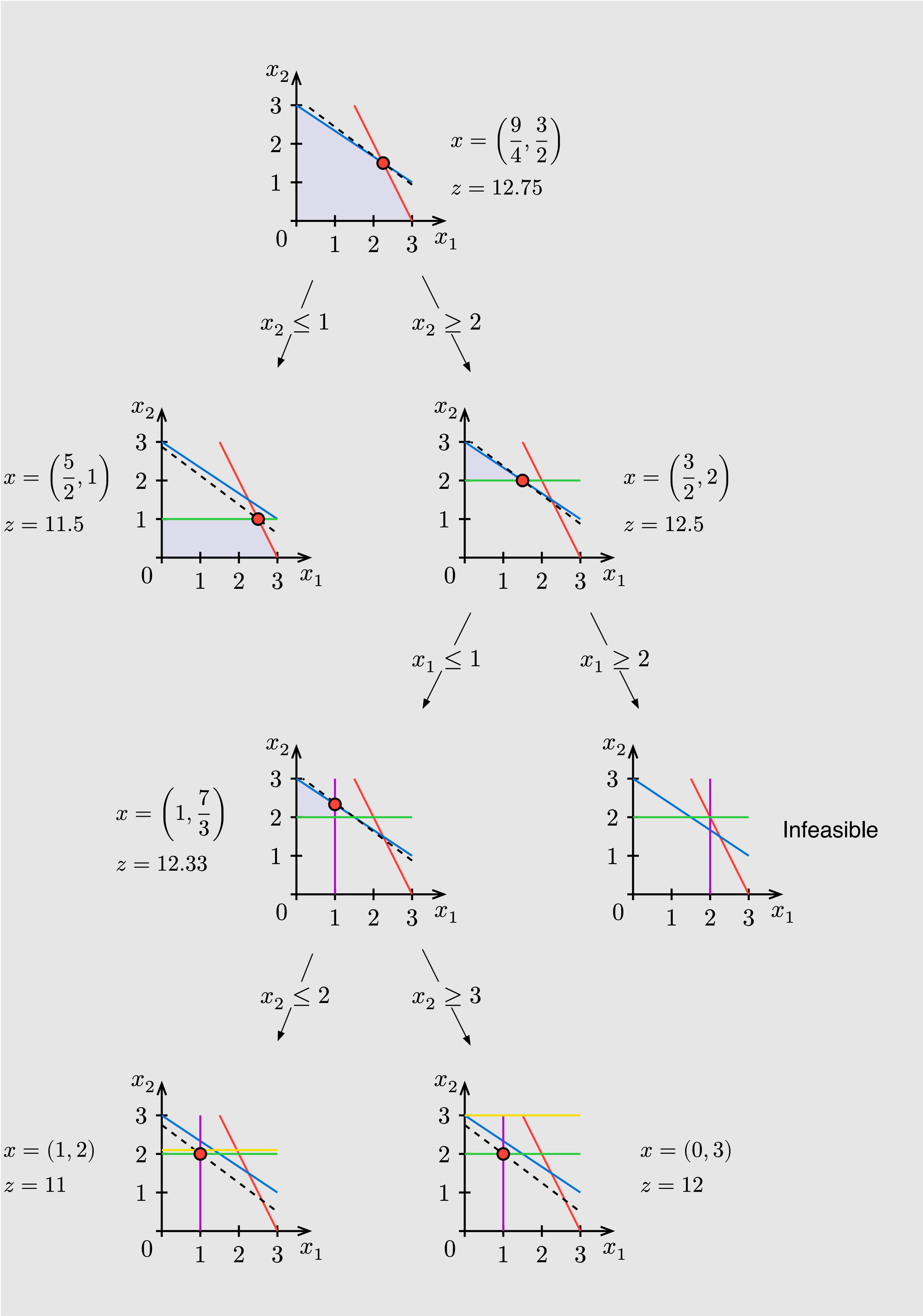
$$\begin{array}{llcccl} \max & z = 3x_1 & + & 4x_2 & & \\ s.t. & 2x_1 & + & x_2 & \leq & 6 \\ & 2x_1 & + & 3x_2 & \leq & 9 \\ & & x_i \in \mathbb{N}_+ & & \forall i = 1, 2 & \end{array}$$

$$
\begin{array}{lrcrcr}
\max & z = 3x_1 & + & 4x_2 & & \\
s.t. & 2x_1 & + & x_2 & \leq & 6 \\
& 2x_1 & + & 3x_2 & \leq & 9 \\
\end{array}
$$

$$x_i \geq 0 \quad \forall i = 1, 2$$

$$x_1 = \frac{9}{4}, \quad x_2 = \frac{3}{2}$$

$$z = 12.75$$


$x = \left(\frac{9}{4}, \frac{3}{2}\right)$
$z = 12.75$
$x_2 \leq 1$
$x_2 \geq 2$
$x = \left(\frac{5}{2}, 1\right)$
$z = 11.5$
$x = \left(\frac{3}{2}, 2\right)$
$z = 12.5$
$x_1 \leq 1$
$x_1 \geq 2$
$x = \left(1, \frac{7}{3}\right)$
$z = 12.33$
Infeasible
$x_2 \leq 2$
$x_2 \geq 3$
$x = (1, 2)$
$z = 11$
$x = (0, 3)$
$z = 12$

27.13. MIP (Mixed Integer Programming)

Optimizing (maximizing or minimizing) a linear **objective function** subject to linear equality or inequality **constraints**. **Decision variables** can take any continuous real or integer values

Minimize (or Maximize):

$$c^T x$$

Subject to:

- $Ax \leq b$
- $x_i \in \mathbb{Z}$ for some i
- $x_j \in \mathbb{R}$ for the remaining j

Where:

- x is the vector of decision variables
- c is the vector of objective function coefficients
- A is the constraint coefficient matrix
- b is the vector of constraint right-hand side values
- x_i represents integer variables
- x_j represents continuous variables

Example

Objective Function

Maximize

$$c^T x$$

$$\begin{bmatrix} 3 & 2 & -5 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Constraints

$$Ax \leq b$$

$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 0 & -5 & 0 \\ 0 & 1 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \leq \begin{bmatrix} 8 \\ 12 \\ 0 \\ 0 \end{bmatrix}$$

27.14. Non-Linear Programming

27.14.1. EOQ

Parameters

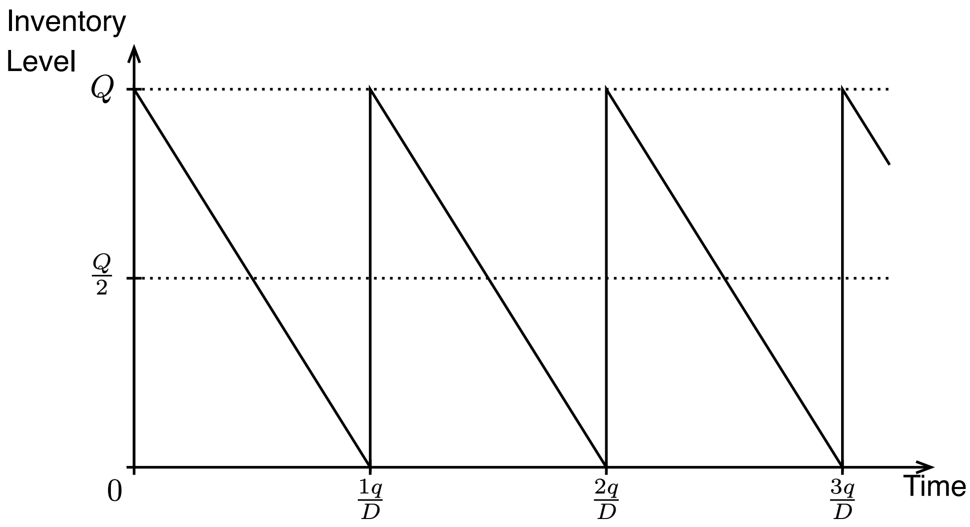
- D: annual demand (units)
- K: unit ordering cost ($)
- h: unit holding cost per year ($)
- p: unit purchasing cost ($)

Decision Variable

- Q: order quantity per order (units)

Objective

Minimize annual total cost



- $\frac{q}{2}$: average inventory level
- $h \times \frac{q}{2} = \frac{hq}{2}$: annual holding cost
- pD: annual purchasing cost
- $\frac{D}{q}$: number of orders in a year
- $K \times \frac{D}{q} = \frac{KD}{q}$: annual ordering cost

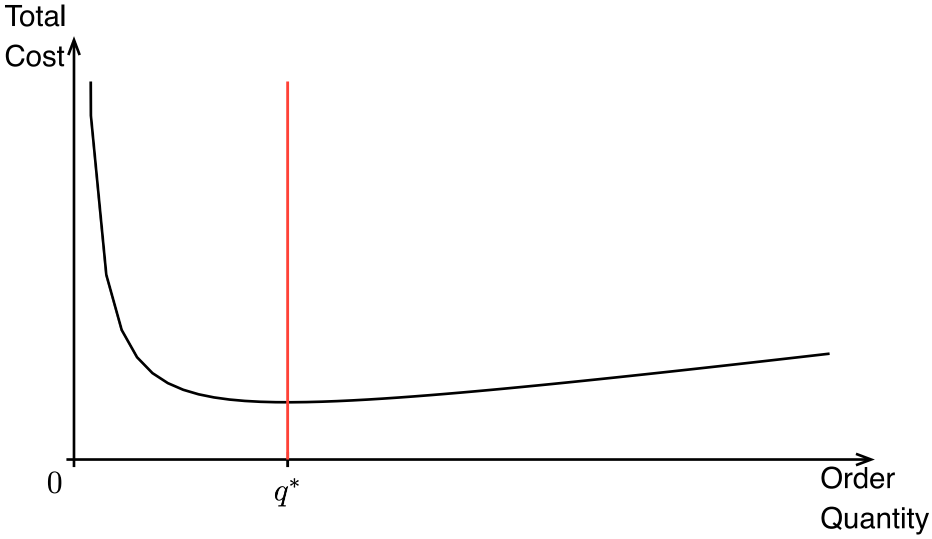
Objective Function

$$\min_{q \geq 0} \underbrace{\frac{KD}{q}}_{\substack{\text{Ordering} \\ \text{Cost}}} + \underbrace{pD}_{\substack{\text{Purchasing} \\ \text{Cost}}} + \underbrace{\frac{hq}{2}}_{\substack{\text{Inventory} \\ \text{Cost}}}$$

pD is just a constant, so:

$$TC(q) = \underbrace{\frac{KD}{q}}_{\substack{\text{Ordering} \\ \text{Cost}}} + \underbrace{\frac{hq}{2}}_{\substack{\text{Inventory} \\ \text{Cost}}}$$

- As q **increases**:
  - Inventory costs increase
  - Ordering cost decease
- As q **decreases**:
  - Inventory costs decrease
  - Ordering cost increase



27.14.2. Portfolio Optimization

Objective

Invest B dollars in n stocks while managing risk and meeting required return

Notation

- B: budget ($)
- R: required minimum expected return
- For each stock $i \in \{1, ..., n\}$
  - p_i: current price
  - μ_i: expected future price
  - σ_i^2: variance of ruturn per share
  - σ_{ij}: Covariance between assets i and j
  - x_i: number of shares to buy (decision variable)

Model

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sigma_i^2 x_i^2 + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sigma_{ij} x_i x_j && \text{(Total risk)} \\
s.t. \quad & \sum_{i=1}^{n} p_i x_i \leq B && \text{(Budget constraint)} \\
& \sum_{i=1}^{n} \mu_i x_i \geq R && \text{(Return constraint)} \\
& x_i \geq 0 \quad \forall i = 1, ..., n && \text{(No short-selling)}
\end{aligned}
$$

27.15. Linearizing Maximum/Minimum Functions

1. When the **maximum** function is on the **smaller side** of inequality

General Form

$$y \geq \max(x_1, x_2) \qquad \Longleftrightarrow \qquad y \geq x_1 \wedge y \geq x_2$$

- This rule holds regardless of whether x_1, x_2, y are variables, constants, or expressions
- You're ensuring that y is greater than or equal to all values in the maximum function, so it must be greater than or equal to each term individually

Generalized form with more than two elements

$$y \geq \max(x_1, x_2, ..., x_n) \qquad \Longleftrightarrow \qquad y \geq x_i \quad \forall i = 1, ..., n$$

Example

$$y + x_1 + 3 \geq \max(x_1 - x_3, 2x_2 + 4) \quad \Longleftrightarrow \quad \begin{cases} y + x_1 + 3 \geq x_1 - x_3 \\ y + x_1 + 3 \geq 2x_2 + 4 \end{cases}$$

2. When the **minimum** function is on the **larger side** of inequality

General Form

$$y \leq \min(x_1, x_2) \qquad \Longleftrightarrow \qquad y \leq x_1 \wedge y \leq x_2$$

- This rule holds regardless of whether x_1, x_2, y are **variables**, **constants**, or **expressions**
- You're ensuring that y is smaller than or equal to all values in the minimum function, so it must be smaller than or equal to each term individually

Generalized form with more than two elements

$$y \leq \min(x_1, x_2, ..., x_n) \qquad \Longleftrightarrow \qquad y \leq x_i \quad \forall i = 1, ..., n$$

Example

$$y + x_1 \leq \min(x_1 - x_3, 2x_2 + 4, 0) \quad \Longleftrightarrow \quad \begin{cases} y + x_1 \leq x_1 - x_3 \\ y + x_1 \leq 2x_2 + 4 \\ y + x_1 \leq 0 \end{cases}$$

Cases Where Linearization Does Not Apply

- $y \leq \max(x_1, x_2) \not\Leftrightarrow y \leq x_1 \wedge y \leq x_2$
- $y \geq \max(x_1, x_2) \not\Leftrightarrow y \geq x_1 \wedge y \geq x_2$
- Maximum or minimum function in an equality

$$y = \max(x_1, x_2) \quad \text{or} \quad y = \min(x_1, x_2)$$

27.16. Linearize Objective Function

1. **Minimize** a **Maximum** Function

$$
\min \quad \max(x_1, x_2) \quad \Longleftrightarrow \quad \begin{aligned} \min \quad & w \\ s.t. \quad & w \geq x_1 \\ & w \geq x_2 \end{aligned}
$$

2. **Maximize** a **Minimum** Function

$$
\begin{aligned} \max \quad & \min(x_1, x_2, 2x_3 + 5) + x_4 \\ s.t. \quad & 2x_1 + x_2 - x_4 \leq x_3 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \max \quad & w + x_4 \\ s.t. \quad & w \leq x_1 \\ & w \leq x_2 \\ & w \leq 2x_3 + 5 \\ & 2x_1 + x_2 - x_4 \leq x_3 \end{aligned}
$$

Cases Where Linearization Does Not Apply

- $\max \max(x_1, x_2)$
- $\min \min(x_1, x_2)$

Absolute Function

The absolute value is equivalent to a maximum function:

$$
|x| = \max(x, -x)
$$

Thus, it can be linearized when it appears on the smaller side of an inequality:

$$
|x| \leq y \quad \Longleftrightarrow \quad \begin{cases} x \leq y \\ -x \leq y \end{cases}
$$

Example

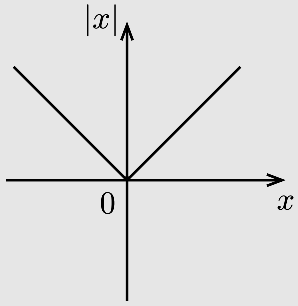
We want to allocate \$1000 to 2 people in a fair way

- **Fairness criterion**: Minimize the difference between the amounts each person receives.
- Let x_i be amount to allocate to person i for $i = 1, 2$

We write the problem as:

$$
\begin{aligned} \min \quad & |x_2 - x_1| \\ s.t. \quad & x_1 + x_2 = 1000 \\ & x_i \geq 0 \quad \forall i = 1, 2 \end{aligned}
$$

- The absolute value makes the objective **nonlinear**



Linearizing the Problem

We now reformulate the problem to make it linear

Step 1.: Introduce a new variable

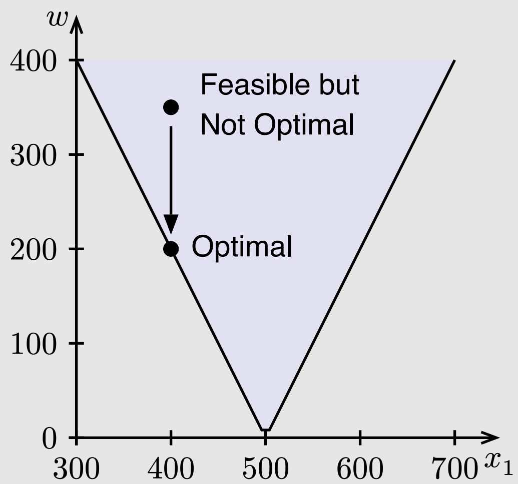
Let w be the absolute difference: $w = |x_2 - x_1|$

$$\begin{aligned} \min \quad & w \\ s.t. \quad & x_1 + x_2 = 1000 \\ & w = |x_2 - x_1| \\ & x_i \geq 0 \quad \forall i = 1, 2 \end{aligned}$$

Step 2. Relax the equality

We replace the equality with an inequality:

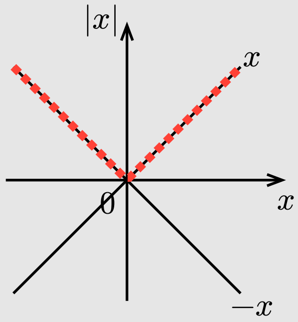
$$\begin{aligned} \min \quad & w \\ s.t. \quad & x_1 + x_2 = 1000 \\ & w \geq |x_2 - x_1| \\ & x_i \geq 0 \quad \forall i = 1, 2 \end{aligned}$$



Step 3. Replace absolute value with maximum

Recall:

$$|x_2 - x_1| = \max(x_2 - x_1, x_1 - x_2)$$



Therefore:

$$w \geq |x_2 - x_1| \quad \Leftrightarrow \quad w \geq \max(x_2 - x_1, x_1 - x_2) \quad \Leftrightarrow \quad w \geq x_2 - x_1 \wedge w \geq x_1 - x_2$$

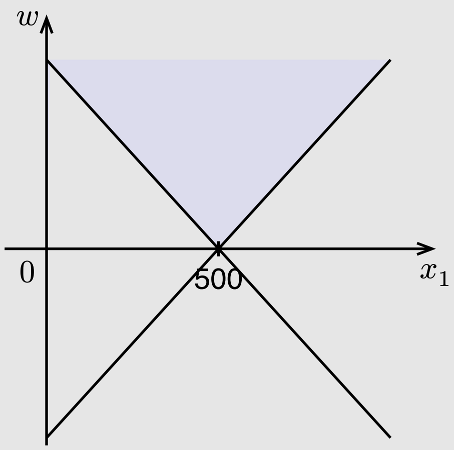
Now the problem is fully linear:

$$\begin{array}{ll} \min & w \\ s.t. & x_1 + x_2 = 1000 \\ & w \geq x_2 - x_1 \\ & w \geq x_1 - x_2 \\ & x_i \geq 0 \quad \forall i = 1, 2 \end{array}$$

Reformulating in One Variable

using $x_2 = 1000 - x_1$, we can express everything in terms of x_1:

$$\begin{array}{ll} \min & w \\ s.t. & w \geq 1000 - 2x_1 \\ & w \geq 2x_1 - 1000 \\ & x_1 \geq 0 \end{array}$$



We are minimizing w, which represents the difference between the two allocations. The solution occurs when the two inequalities intersect — that is, when both are equal:

$$1000 - 2x_1 = 2x_1 - 1000 \quad \Rightarrow \quad x_1 = 500$$

So:

- $x_1 = 500$

- $x_2 = 1000 - x_1 = 500$
- $w = |500 - 500| = 0$

This is the fair allocation: each person receives \$500, and the difference between the two amounts is 0.

27.17. Linearizing Products of Decision Variables

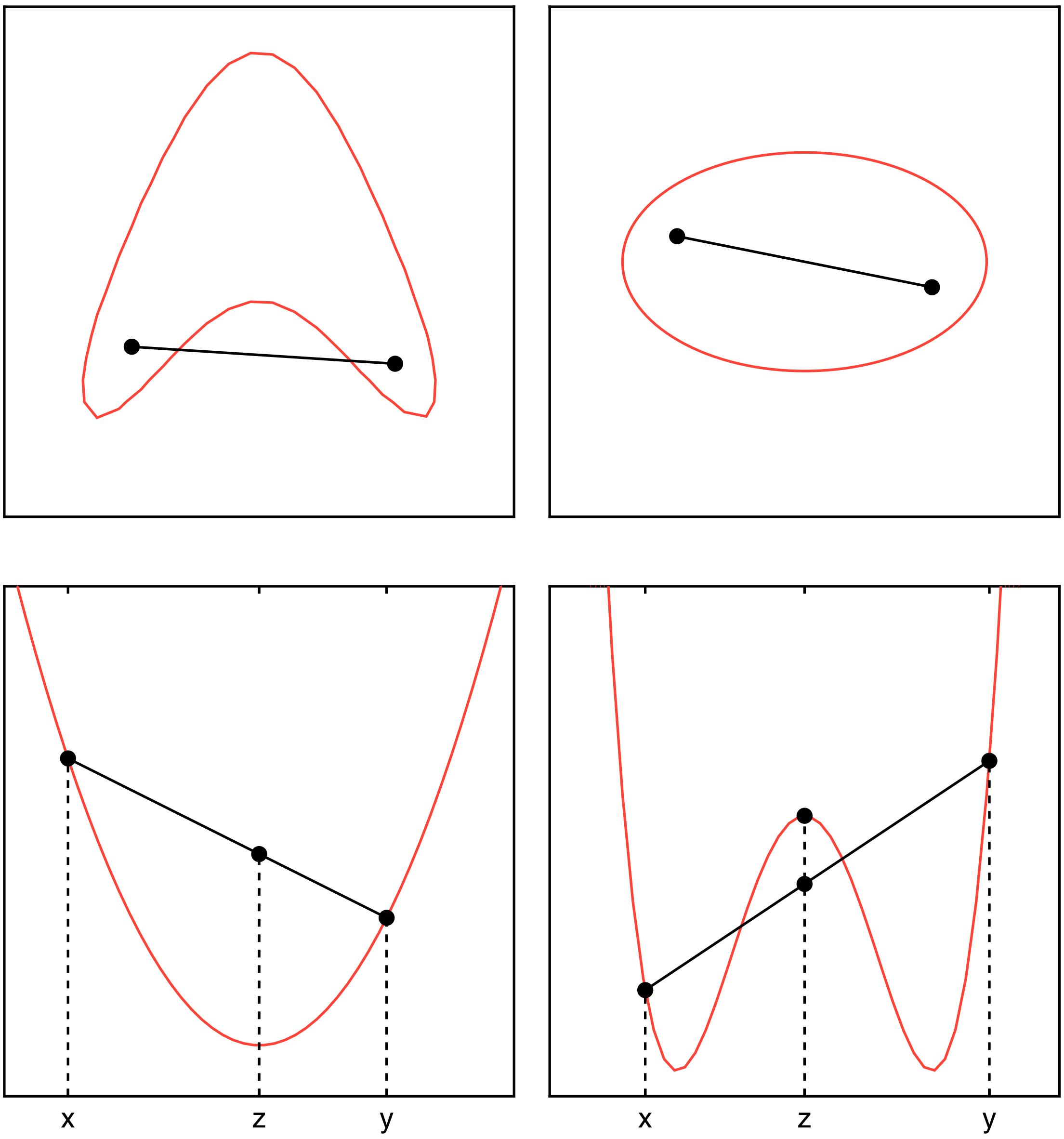
Products of decision variables can be linearized if:

- A binary and a continuous variable
- Two binary variables

Cannot be linearized if:

- Two continuous variables

27.18. Convexity



27.19. Gradient Descent

27.19.1. Gradient

The gradient tells us the direction of steepest increase

For a function

$$f(x_1, x_2, \dots, x_n)$$

the gradient of f, written ∇f, is:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right)$$

- The **magnitude** of the gradient vector tells you how fast the function is increasing (steepness)
- The **direction** of the gradient vector points to where the function increases most rapidly (direction)

Example

Function

$$f(x_1, x_2) = x_1^2 + x_2^2$$

Compute $\frac{\partial d}{\partial x_1}$

- Differentiate f w.r.t. x_1 (treat x_2 as constant):

$$\frac{\partial}{\partial x_1}(x_1^2 + x_2^2) = \frac{\partial}{\partial x_1}(x_1^2) + \frac{\partial}{\partial x_1}(x_2^2)$$

- Since x_2^2 is constant w.r.t. x_1, its derivative is 0:

$$\frac{\partial}{\partial x_1}(x_2^2) = 0$$

- The derivative of x_1^2 w.r.t. x_1 is:

$$\frac{\partial}{\partial x_1}(x_1^2) = 2x_1$$

- So,

$$\frac{\partial f}{\partial x_1} = 2x_1$$

Compute $\frac{\partial f}{\partial x_2}$

- Differentiate f w.r.t. x_2 (treat x_2 as constant):

$$\frac{\partial}{\partial x_2}(x_1^2 + x_2^2) = \frac{\partial}{\partial x_2}(x_1^2) + \frac{\partial}{\partial x_2}(x_2^2)$$

- Since x_1^2 is constant w.r.t. x_2, its derivative is 0:

$$\frac{\partial}{\partial x_2}(x_1^2) = 0$$

- The derivative of x_2^2 w.r.t. x_2 is:

$$\frac{\partial}{\partial x_2}(x_2^2) = 2x_2$$

So,

$$\frac{\partial f}{\partial x_2} = 2x_2$$

Combine to form gradient

$$\nabla f(x_1, x_2) = (2x_1, 2x_2)$$

Evaluate at a point

$$x = (1, 2)$$

$$\nabla f(1, 2) = (2 \times 1, 2 \times 2) = (2, 4)$$

Interpretation

- The gradient vector $(2, 4)$ at $(1, 2)$ points in the **direction** where the function f **increases most rapidly**
- Its **magnitude** $\sqrt{2^2 + 4^2} = \sqrt{20} \approx 4.47$ measures the **steepness** of that increase
- **Moving one unit along the x_1** axis alone would **increase the function by the partial derivative with respect to x_1**, which is 2
- **Moving one unit along the x_2** axis alone would **increase the function by the partial derivative with respect to x_2**, which is 4
- **Moving from $(1, 2)$ in the direction of $(2, 4)$**, the function value will rise fastest and at a rate roughly proportional to **4.47 per unit distance moved**

27.19.2. Algorithm

Find the minimum of a function

In **gradient descent**, we move in the **opposite direction** of the gradient to minimize a function

Direction of **maximum increase**:

$$\nabla f(x)$$

Direction of **maximum decrease**:

$$-\nabla f(x)$$

1. **Initialize**: Start with an initial guess for the parameters
2. **Compute Gradient**: Find the gradient of the function at the current parameters
3. **Update Parameters**: Adjust the parameters by moving in the opposite direction of the gradient, scaled by the learning rate
4. **Repeat**: Continue the process until the parameters converge to a minimum or the changes are minimal

Update Rule:

$$x^{k+1} \leftarrow x^k - \alpha_k \nabla f\left(x^k\right)$$

Where:

- x^k: current iterate (point in domain of f)
- α_k: **step size** or **learning rate** at iteration k
- $\nabla f\left(x^k\right)$: **gradient** of the funtion f with respect to θ
- x^{k+1}: next iterate (point in domain of f)

gradient_descent.py

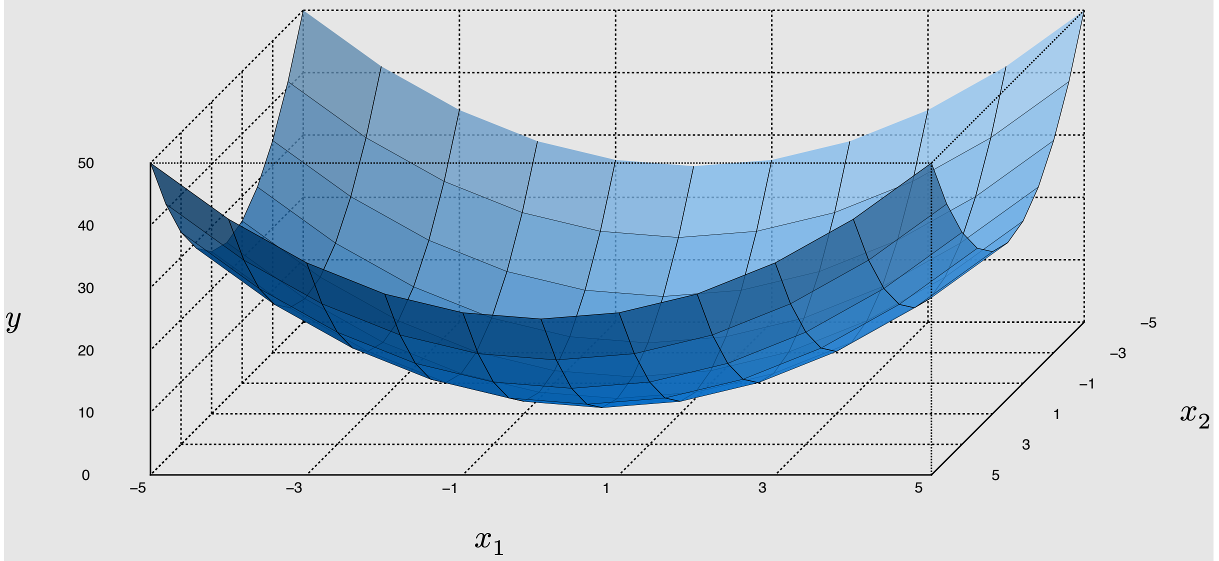
```
import numpy as np

def gradient_descent(
  f,               # Function to minimize, f(x)
  grad_f,          # Gradient of f, grad_f(x)
  x0,              # Initial guess for x
  alpha=0.01,      # Learning rate
  max_iter=1000,   # Maximum number of iterations
  tol=1e-6         # Tolerance for stopping
):
    x = x0.copy()
    for _ in range(max_iter):
        grad = grad_f(x)
        if np.linalg.norm(grad) < tol:
            break
        x -= alpha * grad
    return x

def f(x):
  """Example: minimize f(x) = x^2 + 2x + 1"""
  return x**2 + 2*x + 1

def grad_f(x):
    """Gradient of f(x) = x^2 + 2x + 1 is f'(x) = 2x + 2"""
    return 2*x + 2

gradient_descent(
    f,
```

```
    grad_f,
    x_initial,
    alpha=0.1,
    max_iter=100
)
```

Example

Function

$$x_1^2 + x_2^2$$



Gradient

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2x_1, 2x_2)$$

Learning Rate

$$\alpha = 0.1$$

Iteration 1

- Initial Parameter Values:

$$x_1 = 1 \qquad x_2 = 2$$

- Gradient Calculation

$$\nabla f(x) = (2 \times 1, 2 \times 2) = (2, 4)$$

- Parameter Update:

$$x \leftarrow x - \alpha \nabla f(x)$$

$$x_1 \leftarrow 1 - 0.1 \times 2 = 0.8$$
$$x_2 \leftarrow 2 - 0.1 \times 4 = 1.6$$

- Updated Parameter Values:

$$x_1 = 0.8 \qquad x_2 = 1.6$$

Iteration 2

- Current Parameter Values

$$x_1 = 0.8 \qquad x_2 = 1.6$$

- Gradient Calculation:

$$\nabla f(x) = (2 \times 0.8, 2 \times 1.6) = (1.6, 3.2)$$

- Parameter Update:

$$x \leftarrow x - \alpha \nabla f(x)$$

$$x_1 \leftarrow 0.8 - 0.1 \times 1.6 = 0.64$$
$$x_2 \leftarrow 1.6 - 0.1 \times 3.2 = 1.28$$

- Updated Parameter Values:

$$x_1 = 0.64 \qquad x_2 = 1.28$$

Iteration 3

- Current Values

$$x_1 = 0.64 \qquad x_2 = 1.28$$

- Gradient Calculation:

$$\nabla f(x) = (2 \times 0.64, 2 \times 1.28) = (1.28, 2.56)$$

- Parameter Update:

$$x \leftarrow x - \alpha \nabla f(x)$$

$$x_1 \leftarrow 0.64 - 0.1 \times 1.28 = 0.512$$
$$x_2 \leftarrow 1.28 - 0.1 \times 2.56 = 1.024$$

- Updated Parameter Values:

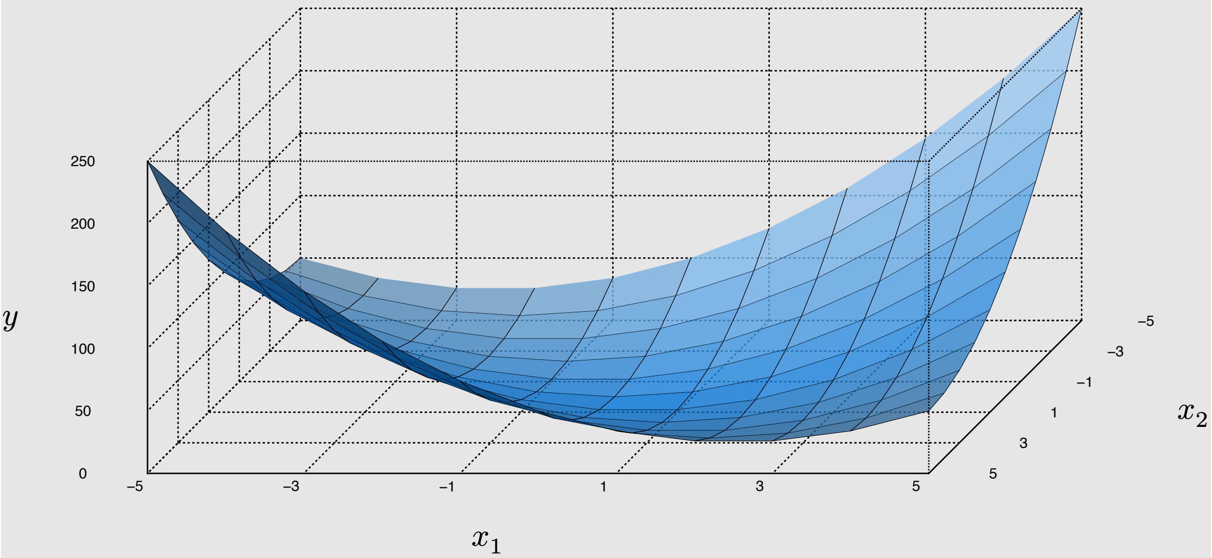
$$x_1 = 0.512 \qquad x_2 = 1.024$$

Example

Problem Setup

Minimize the quadratic function:

$$\min_{x\in\mathbb{R}^2} \quad f(x) = 4x_1^2 - 4x_1x_2 + 2x_2^2 \quad \text{where} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Gradient of $f(x)$

The gradient is:

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

Step 1: Compute $\frac{\partial f}{\partial x_1}$

Take the derivative of each term with respect to x_1 (treat x_2 as a constant):

- $\frac{\partial}{\partial x_1}(4x_1^2) = 8x_1$
- $\frac{\partial}{\partial x_1}(-4x_1x_2) = -4x_2$
- $\frac{\partial}{\partial x_1}(2x_2^2) = 0$

So:

$$\frac{\partial f}{\partial x_1} = 8x_1 - 4x_2$$

Step 2: Find $\frac{\partial f}{\partial x_2}$

Take the derivative of each term with respect to x_2 (treat x_1 as a constant):

- $\frac{\partial}{\partial x_2}(4x_1^2) = 0$
- $\frac{\partial}{\partial x_2}(-4x_1 x_2) = -4x_1$
- $\frac{\partial}{\partial x_2}(2x_2^2) = 4x_2$

So:

$$\frac{\partial f}{\partial x_2} = -4x_1 + 4x_2$$

So the gradient of $f(x)$ is:

$$\nabla f(x) = \begin{bmatrix} 8x_1 - 4x_2 \\ -4x_1 + 4x_2 \end{bmatrix}$$

Optimal Solution

$$x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$f(x^*) = 0$$

Gradient Descent Iterations

Iteration 1

- Initial guess:

$$x^0 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \Rightarrow \quad f(x^0) = 4(2^2) - 4(2)(3) + 2(3^2) = 10$$

- Gradient at x^0:

$$\nabla f(x^0) = \begin{bmatrix} 8(2) - 4(3) \\ -4(2) + 4(3) \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

- Line Search:

$$\begin{aligned} x(\alpha_0) = x^0 - \alpha_0 \nabla f(x^0) &= \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \alpha_0 \begin{bmatrix} 4 \\ 4 \end{bmatrix} \\ &= \begin{bmatrix} 2 - 4\alpha_0 \\ 3 - 4\alpha_0 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} f(x(\alpha_0)) &= 4{x_1}^2 - 4x_1 x_2 + 2{x_2}^2 \\ &= 4(2 - 4\alpha_0)^2 - 4(2 - 4\alpha_0)(3 - 4\alpha_0) + 2(3 - 4\alpha_0)^2 \\ &= 32\alpha_0^2 - 32\alpha_0 + 10 \end{aligned}$$

- Minimizing:

$$\alpha_0 = \text{argmin } f(x(\alpha))$$

$$\frac{\partial}{\partial \alpha} f(x(\alpha_0)) = \frac{\partial}{\partial \alpha}(32\alpha_0^2 - 32\alpha_0 + 10) = 0$$

$$= 64\alpha_0 - 32 = 0$$

$$\alpha_0 = \frac{1}{2}$$

- Update Step:

$$x^1 = x^0 - a_0 \nabla f(x^0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Check Progress:

1. New point:

$$x^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

2. Function value decreases $f\left(x^{k+1}\right) < f\left(x^k\right)$

$$f(x^1) = 2 < f(x^0) = 10$$

3. Gradient at new point:

$$\nabla f(x^1) = \begin{bmatrix} 8(0) - 4(1) \\ -4(0) + 4(1) \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \end{bmatrix}$$

4. Gradient magnitude $\|\nabla f\left(x^{k+1}\right)\| < \|\nabla f\left(x^k\right)\|$

$$\|\nabla f(x^1)\| = 4\sqrt{2} = \|\nabla f(x^0)\|$$

Note: Gradient magnitude stays the same in this iteration due to the specific structure of this quadratic function

Iteration 2

- New Point:

$$x^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Gradient at x_1:

$$\nabla f(x^1) = \begin{bmatrix} 8(0) - 4(1) \\ -4(0) + 4(1) \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \end{bmatrix}$$

$$a_1 = \text{argmin } f(x(\alpha_1))$$

- Line search:

$$x(\alpha_1) = x^1 - \alpha_1 \nabla f(x^1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \alpha_1 \begin{bmatrix} -4 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 4\alpha_1 \\ 1 - 4\alpha_1 \end{bmatrix}$$

$$\begin{aligned} f(x(\alpha_1)) &= 4{x_1}^2 - 4x_1x_2 + 2{x_2}^2 \\ &= 4(4\alpha_1)^2 - 4(4\alpha_1)(1 - 4\alpha_1) + 2(1 - 4\alpha_1)^2 \\ &= 160\alpha_1^2 - 32\alpha_1 + 2 \end{aligned}$$

- Minimize:

$$\alpha_1 = \text{argmin } f(x(\alpha_1)) = \frac{1}{10}$$

$$\frac{\partial}{\partial \alpha} f(x(\alpha_1)) = \frac{\partial}{\partial \alpha}(160\alpha_1^2 - 32\alpha_1 + 2) = 0$$

$$= 320\alpha_1 - 32 = 0$$

$$\alpha_1 = \frac{1}{10}$$

- Update Step:

$$x^2 = x^1 - \alpha_1 \nabla f(x^1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{1}{10} \begin{bmatrix} -4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$$

- Improvement:

1. New point:

$$x_2 = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$$

2. Function value decreases $f\left(x^{k+1}\right) < f\left(x^k\right)$

$$f(x^2) = 0.4 < f(x^1) = 2$$

3. Gradient at new point:

$$\nabla f(x_2) = \begin{bmatrix} 8(0.4) - 4(0.6) \\ -4(0.4) + 4(0.6) \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix}$$

4. Gradient magnitude $\|\nabla f\left(x^{k+1}\right)\| < \|\nabla f\left(x^k\right)\|$

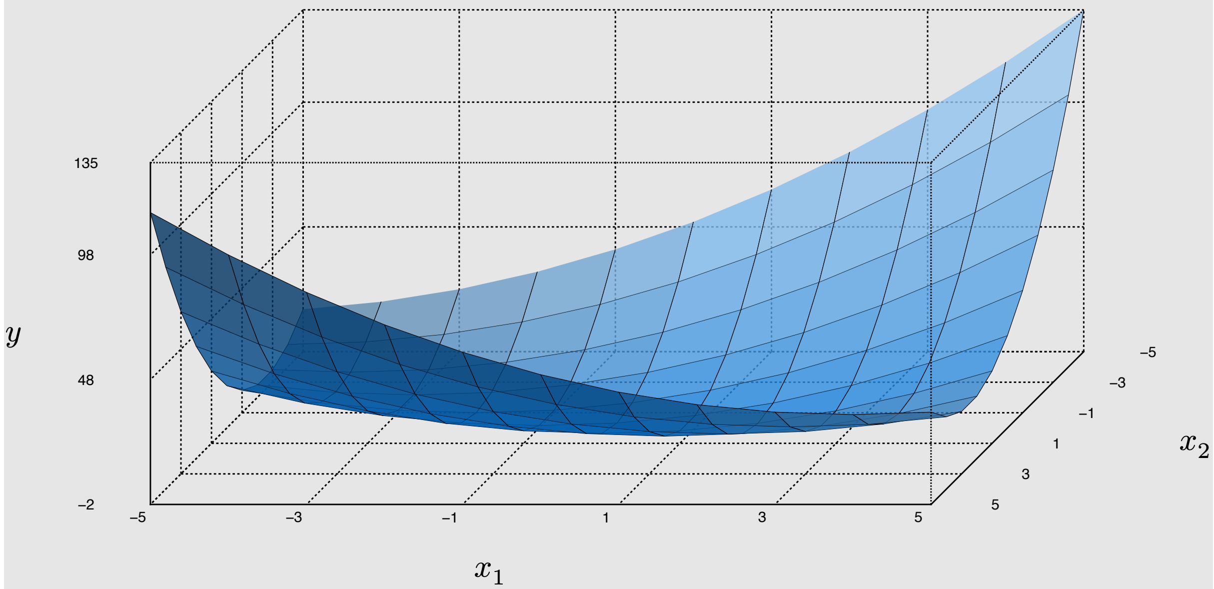
$$\|\nabla f(x^2)\| = \|(0.8, 0.8)\| = \frac{4\sqrt{2}}{5}$$

Example

Problem Setup

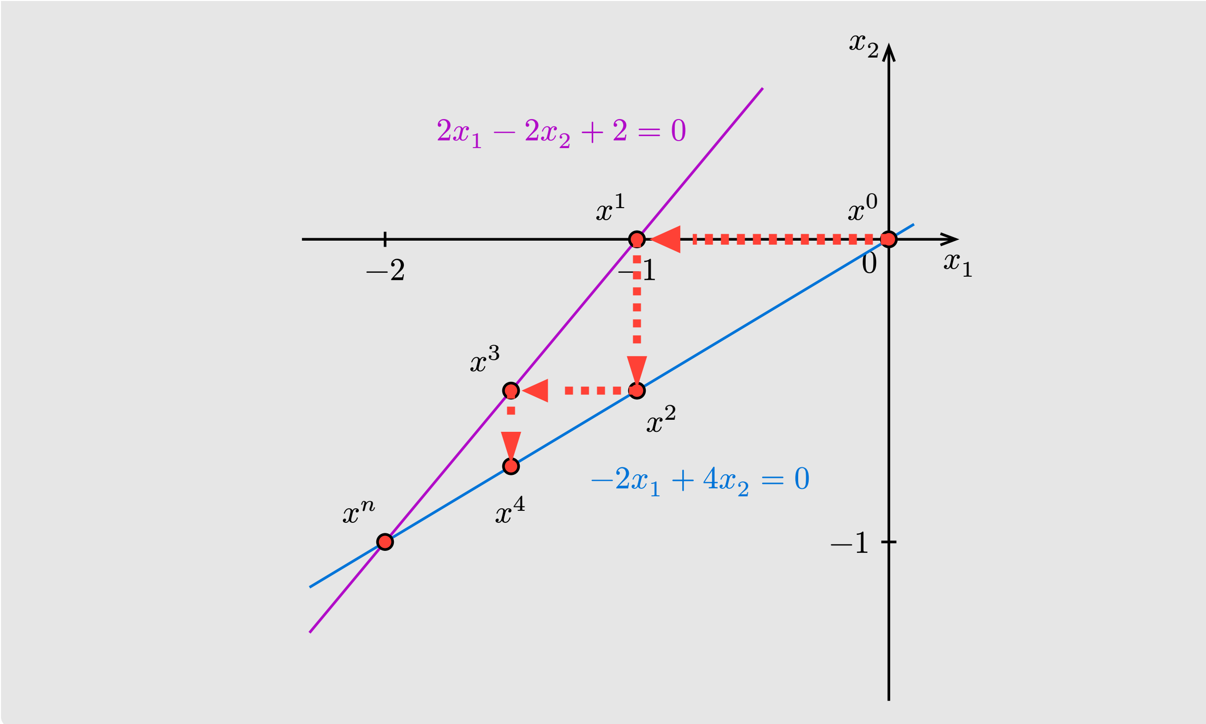
Minimize the quadratic function:

$$\min_{x \in \mathbb{R}^2} \quad f(x) = x_1^2 - 2x_1x_2 + 2x_2^2 + 2x_1 \quad \text{where} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Gradient of $f(x)$

$$\nabla f(x_1, x_2) = \begin{bmatrix} 2x_1 - 2x_2 + 2 \\ -2x_1 + 4x_2 \end{bmatrix}$$



27.20. Newton's Method

Newton's method is an iterative numerical technique to find roots of a function (i.e., solutions to $f(x) = 0$).

Step 1. At each step, approximate $f(x)$ by its tangent line at the current guess x_k:

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

Step 2. Set this approximation equal to 0:

$$0 = f(x_k) + f'(x_k)(x - x_k)$$

Step 3. Solve for x:

$$x = x_k - \frac{f(x_k)}{f'(x_k)}$$

Step 4. This gives the update rule:

$$x = x_k - \frac{f(x_k)}{f'(x_k)}$$

27.20.1. Newton's Method for Nonlinear Optimization

Used to find the **roots of the first derivative** $f'(x) = 0$, which correspond to the **local minima/maxima of the original function** $f(x)$.

Finding **stationary points** of a function $f(x)$, i.e., points where the gradient (derivative) is zero:

- 1D: $f'(x) = 0$
- nD: $\nabla f(\mathbf{x}) = \mathbf{0}$

It can be derived from either linear approximation of the gradient or quadratic approximation of the function

27.20.1.1. 1D Linear Approximation

1. Start with the first-order (linear) Taylor expansion of the derivative around x_k:

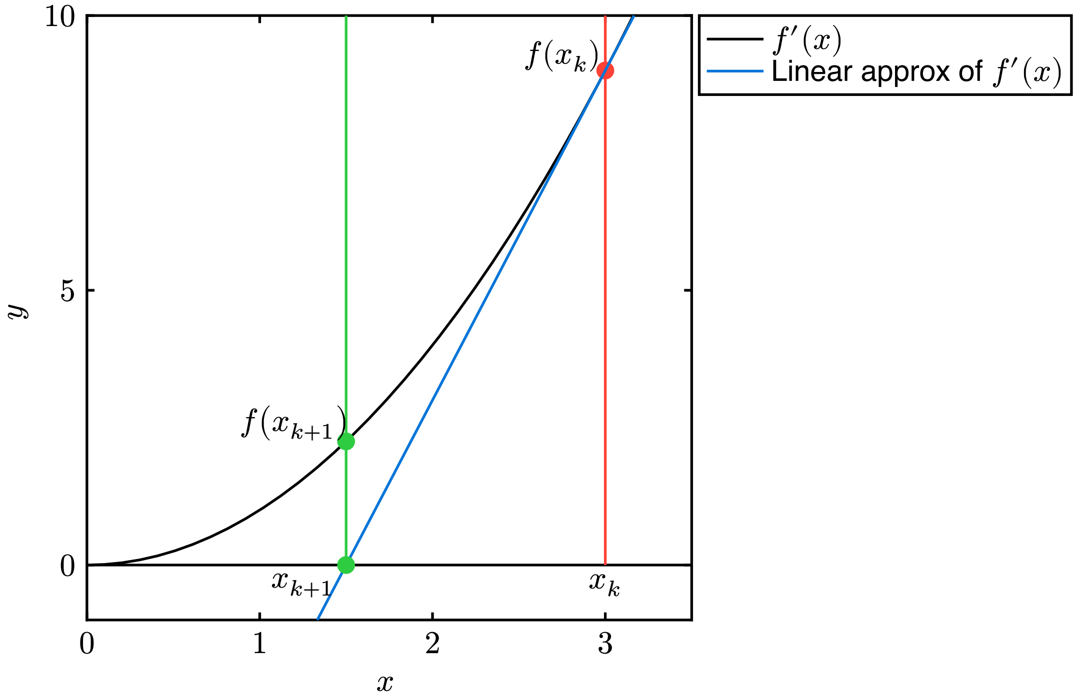
$$f'(x_{k+1}) \approx f'(x_k) + f''(x_k)(x - x_k)$$

2. Stationary point condition: Set the derivative to zero to find a candidate minimum or maximum

$$0 = f'(x_k) + f''(x_k)(x - x_k)$$

3. Solve for the next iterate x_{k+1}:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$



27.20.1.2. nD Linear Approximation

1. Start with the first-order (linear) Taylor expansion of the gradient around x_k:

$$\nabla f(\mathbf{x}_{k+1}) \approx \nabla f(\mathbf{x}_k) + H_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

where $H_k = \nabla^2 f(\mathbf{x}_k)$ is the Hessian matrix

2. Stationary point condition: Set the gradient to zero to find a candidate minimum or maximum:

$$\mathbf{0} = \nabla f(\mathbf{x}_k) + H_k(\mathbf{x} + (k+1) - \mathbf{x}_k)$$

3. Solve for the next iterate $\mathbf{x}_{k+1}$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H_k^{-1} \nabla f(\mathbf{x}_k)$$

27.20.1.3. 1D Quadratic Approximation

1. Start with the second-order (quadratic) Taylor expansion of the function around x_k:

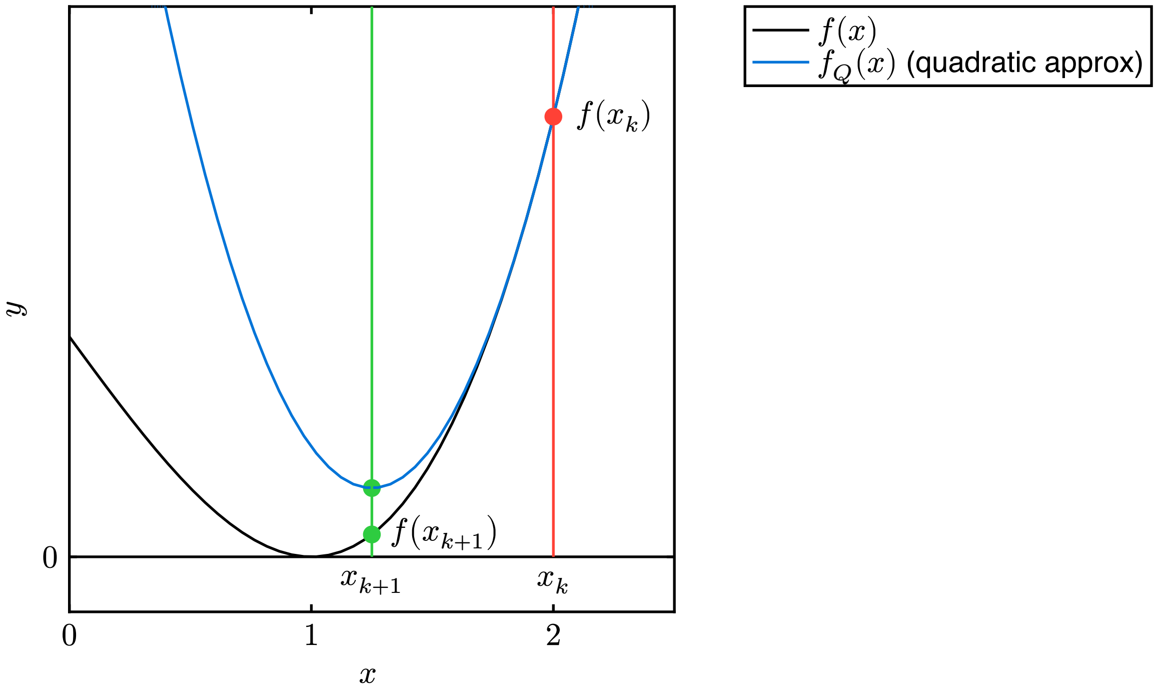
$$f_Q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$

2. Minimize the quadratic approximation: Set the derivative of $f_Q(x)$ to zero:

$$\frac{\mathrm{d}}{\mathrm{d}x} f_Q(x) = f'(x_k) + f''(x_k)(x - x_k) = 0$$

3. Solve for the next iterate x_{k+1}:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$



27.20.1.4. nD Quadratic Approximation

1. Start with the second-order (quadratic) Taylor expansion of the function around x_k:

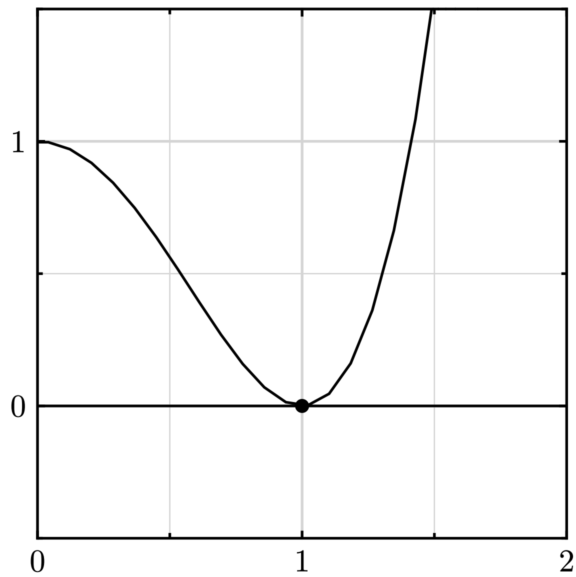
$$m_k(\mathbf{p}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T H_k \mathbf{p}$$
$$\mathbf{p} = \mathbf{x} - \mathbf{x}_k$$

2. Minimize the quadratic approximation: Set the gradient of $m_k(\mathbf{p})$ with respect to $\mathbf{p}$ to zero:

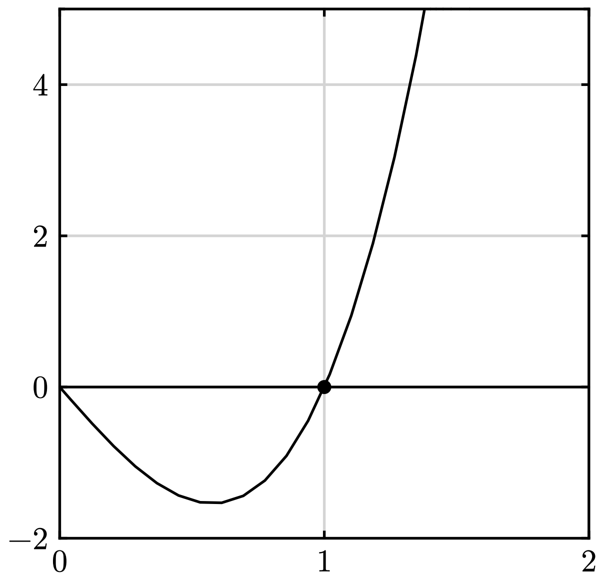
$$\nabla m_k(\mathbf{p}) = \nabla f(\mathbf{x}_k) + H_k \mathbf{p} = \mathbf{0}$$

3. Solve for the step $\mathbf{p}_k$ and next iterate $\mathbf{x}_{k+1}$:

$$\mathbf{p}_k = -H_k^{-1} \nabla f(\mathbf{x}_k), \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$$



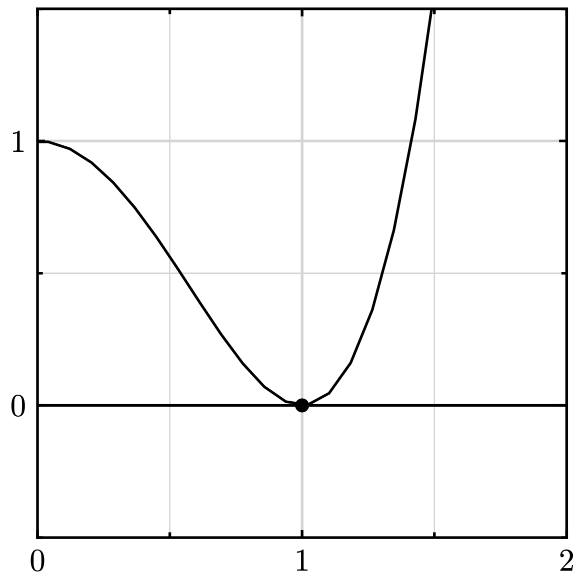
$f(x)$



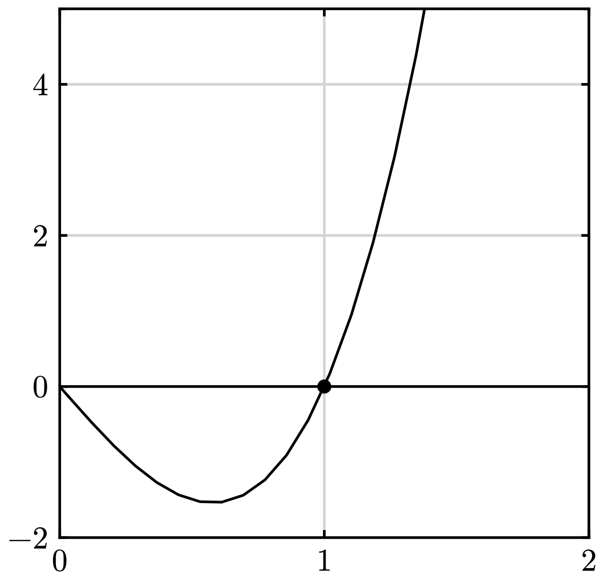
$f'(x)$

27.20.1.5. Quadratic Approximation

$$f_Q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$$



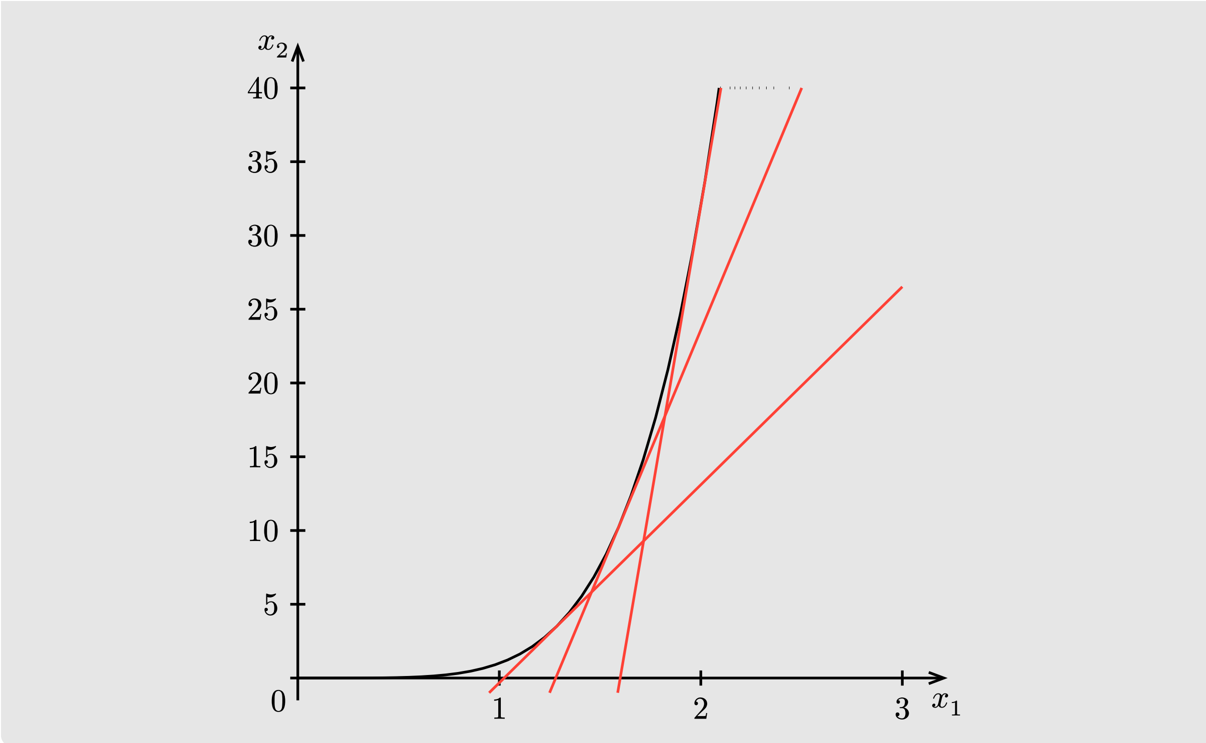
$f(x)$



$f'(x)$

Example

$$f(x) = x^5 \qquad f'(x) = 5x^4$$



Hello

27.20.2. Multi-Dimensional

second-order iterative algorithm

uses both the

- first derivative (gradient)
- second derivative (Hessian)

Step 1. Start with initial guess x_0

Step 2. At each iteration k:

- Compute gradient $\nabla f(x_k)$
- Compute Hessian $\nabla^2 f(x_k)$
- Update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1} \nabla f(\mathbf{x}_k)$$

$$\nabla f(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}_k) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}_k) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}_k) \end{bmatrix} \in \mathbb{R}^n \qquad \nabla^2 f(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}_k) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}_k) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_2^2}(\mathbf{x}_k) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{x}_k) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_n^2 \partial x_2}(\mathbf{x}_k) & \dots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}_k) \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Instead of computing the inverse explicitly, we solve the linear system:

$$\nabla^2 f(\mathbf{x}_k)\mathbf{p}_k = \nabla f(\mathbf{x}_k)$$

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}_k) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}_k) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_2^2}(\mathbf{x}_k) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{x}_k) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}_k) & \frac{\partial^2 f}{\partial x_n^2 \partial x_2}(\mathbf{x}_k) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}_k) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}_k) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}_k) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}_k) \end{bmatrix}$$

$$\begin{gathered} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}_k)p_1 + \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}_k)p_2 + \ldots + \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}_k)p_n = \frac{\partial f}{\partial x_1}(\mathbf{x}_k) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}_k)p_1 + \frac{\partial^2 f}{\partial x_2^2}(\mathbf{x}_k)p_2 + \ldots + \frac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{x}_k)p_n = \frac{\partial f}{\partial x_2}(\mathbf{x}_k) \\ \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}_k)p_1 + \frac{\partial^2 f}{\partial x_n^2 \partial x_2}(\mathbf{x}_k)p_2 + \ldots + \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}_k)p_n = \frac{\partial f}{\partial x_n}(\mathbf{x}_k) \end{gathered}$$

Then update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{p}_k$$

Example

$$f(x, y) = x^2 + xy + y^2$$

Step 1. Compute Gradient

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x + y \\ x + 2y \end{bmatrix}$$

Step 2. Compute Hessian

$$\nabla^2 f(x, y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} =$$

Step 0. Setup

We want to minimize a function $f(x, y)$

At some iteration k, we compute:

- Gradient

$$g = \nabla f(x_k, y_k) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

- Hessian

$$H = \nabla^2 f(x_k, y_k) = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

We need the Newton step vector p from:

$$Hp = g$$

so that the update is:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - p$$

Step 1. We need the Newton step p from:

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

Which is two equations:

$$ap_1 + bp_2 = g_1$$
$$bp_1 + cp_2 = g_2$$

Step 2. Solve without inversion

This is a simple linear system.

For example, solve the first equation for p_1:

$$p_1 = \frac{g_1 - bp_2}{a}$$

Plug into the second:

$$b\left(\frac{g_1 - bp_2}{a}\right) + cp_2 = g_2$$
$$\frac{bg_1}{a} - \frac{b^2}{a} p_2 + cp_2 = g_2$$
$$\left(c = \frac{b^2}{a}\right) p_2 = g_2 - \frac{bg_1}{a}$$
$$p_2 = \frac{ag_2 - bg_1}{ac - b^2}$$

Then substitute back for p_1:

$$p_1 = \frac{g_1 - bp_2}{a}$$

Step 3. Newton update

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

27.21. Heuristic Algorithms

27.22. Linear Programming Duality

Every linear program (primal) has a **unique** and **symmetric** dual problem. For any primal LP, there is a unique dual, whose dual is the primal.

Solving the dual gives you information about the primal.

27.22.1. General Form

Primal

$$\begin{aligned} \text{min / max} \quad & \mathbf{c}^T\mathbf{x} \\ s.t. \quad & A\mathbf{x} \circ \mathbf{b} \\ & \mathbf{x} \diamond 0 \end{aligned}$$

Dual

$$\begin{aligned} \text{max / min} \quad & \mathbf{b}^T\mathbf{y} \\ s.t. \quad & A^T\mathbf{y} \diamond \mathbf{c} \\ & \mathbf{y} \circ 0 \end{aligned}$$

Primal

$$\begin{aligned} \text{min / max} \quad & \sum_{j=1}^{n} c_j x_j \\ s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j \circ b_i, \quad i = 1, ..., m \\ & x_j \diamond 0, \quad j = 1, ..., n \end{aligned}$$

Dual

$$\begin{aligned} \text{max / min} \quad & w = \sum_{i=1}^{m} u_i b_i \\ s.t. \quad & \sum_{i=1}^{m} a_{ji} u_i \diamond c_j, \quad j = 1, ..., n \\ & u_i \circ 0, \quad i = 1, ..., m \end{aligned}$$

Primal

$$\begin{aligned} \text{min / max} \quad & c_1x_1 + c_2x_2 + c_3x_3 \\ s.t. \quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2 \\ & a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \\ & x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \ \text{urs} \end{aligned}$$

Dual

$$\begin{aligned} \text{max / min} \quad & b_1y_1 + b_2y_2 + b_3y_3 \\ s.t. \quad & a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \geq c_1 \\ & a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \leq c_2 \\ & a_{13}y_1 + a_{23}y_2 + a_{33}y_3 = c_3 \\ & y_1 \leq 0, \quad y_2 \geq 0, \quad y_3 \ \text{urs} \end{aligned}$$

27.22.2. Standard Form

Primal

$$\begin{aligned} \text{min / max} \quad & \mathbf{c}^T\mathbf{x} \\ s.t. \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \text{max / min} \quad & \mathbf{b}^T\mathbf{y} \\ s.t. \quad & A^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \ \text{urs} \end{aligned}$$

Primal

$$\begin{aligned} \text{min / max} \quad & \sum_{j=1}^{n} c_j x_j \\ s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, ..., m \\ & x_j \geq 0, \quad j = 1, ..., n \end{aligned}$$

Dual

$$\begin{aligned} \text{max / min} \quad & w = \sum_{i=1}^{m} y_i b_i \\ s.t. \quad & \sum_{i=1}^{m} a_{ji} y_i \leq c_j, \quad j = 1, ..., n \\ & y_i \ \text{urs}, \quad i = 1, ..., m \end{aligned}$$

Primal

$$\begin{aligned} \min / \max \quad & c_1x_1 + c_2x_2 + c_3x_3 \\ s.t. \quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ & a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \max / \min \quad & b_1y_1 + b_2y_2 + b_3y_3 \\ s.t. \quad & a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \geq c_1 \\ & a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \geq c_2 \\ & a_{13}y_1 + a_{23}y_2 + a_{33}y_3 \geq c_3 \\ & y_1, y_2, y_3 \ \text{urs} \end{aligned}$$

Obj	max	min	Obj
Constraint	$\leq$ $\geq$ $=$	≥ 0 ≤ 0 urs.	Variable
Variable	≥ 0 ≤ 0 urs.	$\leq$ $\geq$ $=$	Constraint

Example

Minimization

Primal

$$\begin{array}{rrcrcr} \min & 3x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & 2x_2 & \geq & 4 \\ & 2x_1 & + & x_2 & \geq & 5 \\ & x_1 \leq 0, & & x_2 \ \text{urs} & & \end{array}$$

Dual

$$\begin{array}{rrcrcr} \max & 4y_1 & + & 5y_2 & & \\ s.t. & y_1 & + & 2y_2 & \geq & 3 \\ & 2y_1 & + & y_2 & = & 2 \\ & y_1 \leq 0, & & y_2 \leq 0 & & \end{array}$$

Maximization

Primal

$$\begin{array}{rrcrcr} \max & 3x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & 2x_2 & \leq & 4 \\ & 2x_1 & + & x_2 & \leq & 5 \\ & x_1 \geq 0, & & x_2 \ \text{urs} & & \end{array}$$

Dual

$$\begin{array}{rrcrcr} \min & 4y_1 & + & 5y_2 & & \\ s.t. & y_1 & + & 2y_2 & \geq & 3 \\ & 2y_1 & + & y_2 & = & 2 \\ & y_1 \geq 0, & & y_2 \geq 0 & & \end{array}$$

27.22.3. Weak Duality

1. Dual objective gives a **lower bound** for a **minimization** primal

$$c^T x \geq b^T y$$

2. Dual objective gives an **upper bound** for a **maximization** primal

$$c^T x \leq b^T y$$

Suffciency of optimality

If $\bar{x}$ and $\bar{y}$ are primal and dual feasible and $c^T\bar{x} \leq \bar{y}^T b$, then $\bar{x}$ and $\bar{y}$ are primal and dual optimal

Example

Given a primal feasible solution $\bar{x}$, if we find a dual feasible solution so that their objective values are identical, $\bar{x}$ is optimal (strong duality)

27.22.4. Dual Oprimal solution

If we have solved the primal LP, the dual optimal solution can be obtained

If $\bar{x}$ is primal optimal with basis B, then $\bar{y}^T = c_B^T A_B^{-1}$ is dual optimal

27.22.5. Strong Duality

If both primal and dual are feasible and at least one has an optimal solution:

$\bar{x}$ and $\bar{y}$ are primal and dual optimal iif $\bar{x}$ and $\bar{y}$ are primal and dual feasible and

$$c^T\bar{x} = b^T\bar{y}$$

Example

Primal

$$\begin{array}{llllll} \max & x_1 & & & & \\ s.t. & 2x_1 & - & x_2 & \leq & 4 \\ & 2x_1 & + & x_2 & \leq & 8 \\ & & & x_2 & \leq & 3 \\ & x_1, x_2 \geq 0 & & & & \end{array}$$

$\Leftrightarrow$

Dual

$$\begin{array}{llllllll} \min & 4y_1 & + & 8y_2 & + & 3y_3 & & \\ s.t. & 2y_1 & + & 2y_2 & & & \geq & 1 \\ & -y_1 & + & y_2 & + & y_3 & \geq & 1 \\ & y_1, y_2, y_3 \geq 0 & & & & & & \end{array}$$

Using the simplex method, we obtain the optimal tableau:

–1	0	0	0	0	0
2	–1	1	0	0	$x_3 = 4$
2	1	0	1	0	$x_4 = 8$
0	1	0	0	1	$x_5 = 3$

$\rightarrow \quad \ldots \quad \rightarrow$

0	0	1/4	1/4	0	3
1	0	1/4	1/4	0	$x_1 = 3$
0	1	–1/2	1/2	0	$x_2 = 2$
0	0	1/2	–1/2	1	$x_5 = 1$

- The associated optimal basis is $B = (1, 2, 5)$
- The optimal solution is $\bar{x} = (3, 2)$
- The associated objective value is $z^* = 3$

For the standard form primal LP:

$$c^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Given $x_B = (x_1, x_2, x_5)$ and $x_N = (x_2, x_4)$:

$$c_B^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad A_B = \begin{bmatrix} 2 & -1 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\bar{y} = c_B^T A_B^{-1} = [1 \quad 0 \quad 0] \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix}$$

For $\bar{y} = \left(\frac{1}{4}, \frac{1}{4}, 0\right)$:

- It is dual feasible: $2\left(\frac{1}{4}\right) + 2\left(\frac{1}{4}\right) > 1$ and $-\frac{1}{4} + \frac{1}{4} + 0 \geq 0$
- Its dual objective value $w = 4\left(\frac{1}{4}\right) + 8\left(\frac{1}{4}\right) = 3 = z^*$

Therefore $\bar{y}$ is **dual optimal**

Example

Minimization

Primal

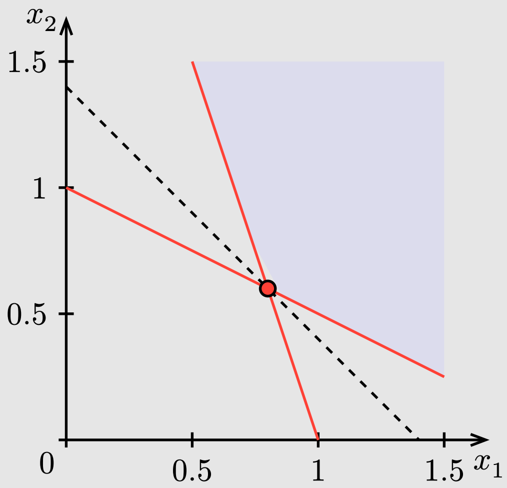
$$\begin{array}{rrcrcl} \min & x_1 & + & x_2 & & \\ s.t. & x_1 & + & 2x_2 & \geq & 2 \\ & 3x_1 & + & x_2 & \geq & 3 \\ & & x_1, x_2 \geq 0 & & & \end{array}$$

Dual

$$\begin{array}{rrcrcl} \max & 2y_1 & + & 3y_2 & & \\ s.t. & y_1 & + & 3y_2 & \leq & 1 \\ & 2y_1 & + & y_2 & \leq & 1 \\ & & y_1, y_2 \geq 0 & & & \end{array}$$

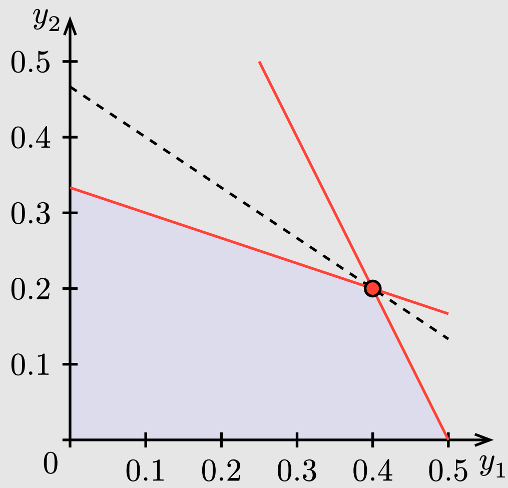
Strong Duality

$$c^T x^* = b^T y^*$$



$x^* = (0.8, 0.6)$

$z^* = 1.4$



$y^* = (0.4, 0.2)$

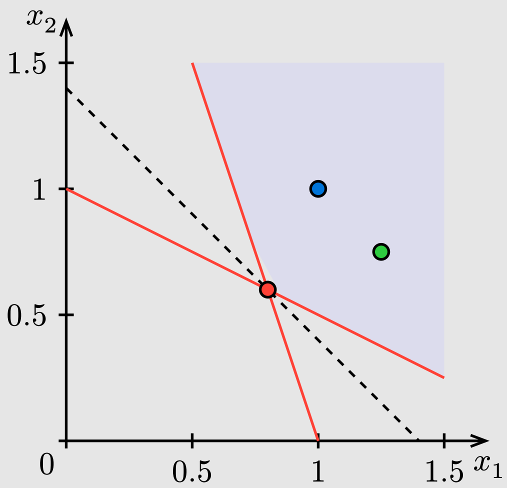
$z^* = 1.4$

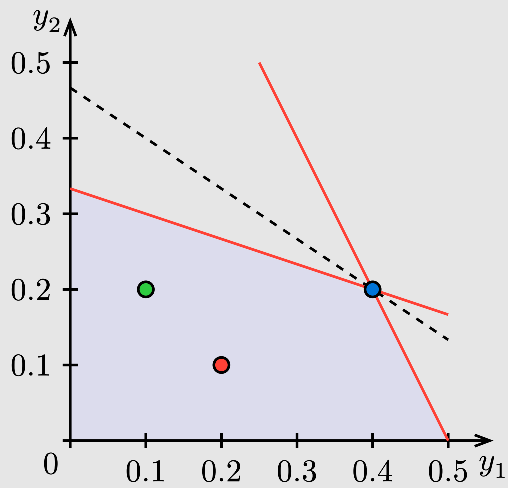
Weak Duality

$$c^T x > b^T y$$

if

$$x \neq x^* \text{ or } y \neq y^*$$





$x^* = (0.8, 0.6)$ $\quad y = (0.2, 0.1)$

$z^* = 1.4 \quad > \quad w = 1$

$x = (1, 1)$ $\quad y^* = (0.4, 0.2)$

$z = 2 \quad > \quad w^* = 1.4$

$x = (1, 1)$ $\quad y = (0.4, 0.2)$

$z = 2 \quad > \quad w = 1.4$

Minimization

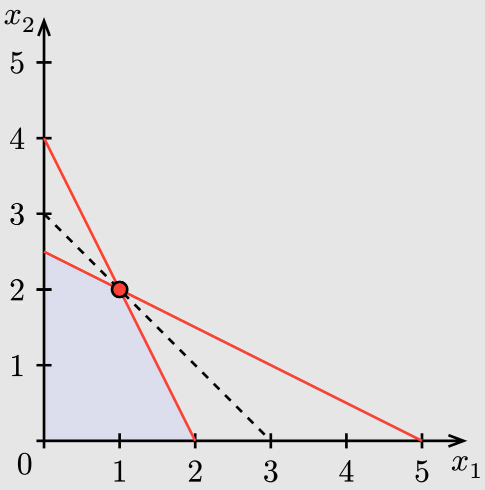
Primal **Dual**

$$\begin{array}{lcccccc} \max & x_1 & + & x_2 & & \\ s.t. & 2x_1 & + & x_2 & \leq & 4 \\ & x_1 & + & 2x_2 & \leq & 5 \\ & x_1, x_2 \geq 0 \end{array}$$

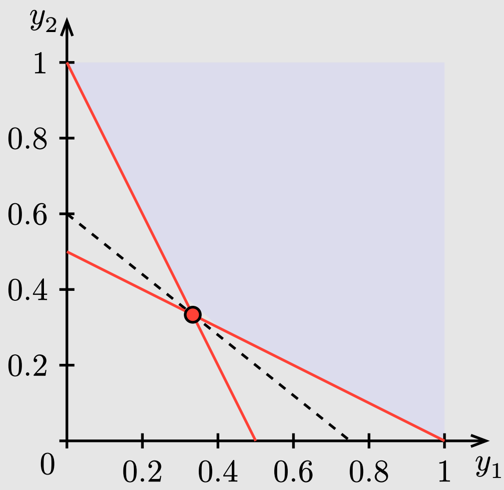
$$\begin{array}{lcccccc} \min & 4y_1 & + & 5y_2 & & \\ s.t. & 2y_1 & + & y_2 & \geq & 1 \\ & y_1 & + & 2y_2 & \geq & 1 \\ & y_1, y_2 \geq 0 \end{array}$$

Strong Duality

$$c^T x^* = b^T y^*$$



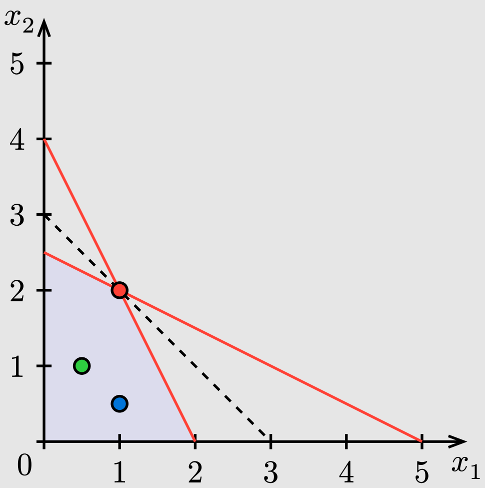
$$x^* = (1, 2)$$
$$z^* = 3$$

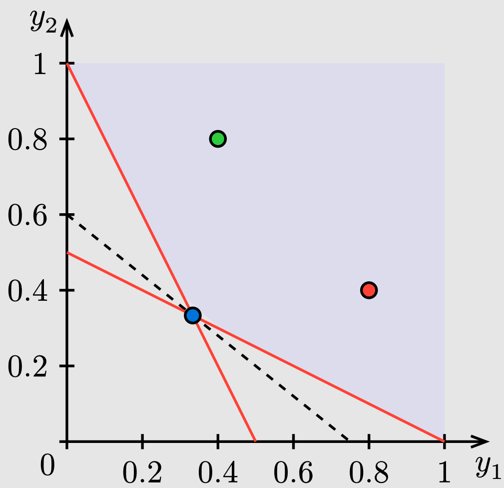


$$y^* = \left(\frac{1}{3}, \frac{1}{3}\right)$$
$$z^* = 3$$

Weak Duality

$$c^T x < b^T y$$
if
$$x \neq x^* \text{ or } y \neq y^*$$





$$x^* = (1, 2) \qquad y = (0.8, 0.4)$$
$$z^* = 3 \quad < \quad w = 5.6$$

$$x = (1, 0.5) \qquad y^* = \left(\frac{1}{3}, \frac{1}{3}\right)$$
$$z = 1.5 \quad < \quad w^* = 3$$

$$x = (0.5, 1) \qquad y = (0.4, 0.8)$$
$$z = 1.5 \quad < \quad w = 5.6$$

27.22.6. Feasibility-Infeasibility Certificates

- If the primal is **infeasible**, the dual is either **infeasible** or **unbounded**
- If the primal is **unbounded**, the dual is necessarily **infeasible**

Primal	Dual		
	Infeasible	Unbounded	Finitely Optimal
Infeasible	✅	✅	❌
Unbounded	✅	❌	❌
Finitely Optimal	❌	❌	✅

- ✅ means possible and ❌ means impossible
- Primal unbounded → no upper bound → dual infeasible
- Primal finitely optimal → finite objective value → dual finitely optimal
- Primal infeasible → dual infeasible or unbounded

Example

Infeasible Primal → Infeasible Dual

Primal

$$\begin{array}{lrcrcl} \min & 2x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & x_2 & \geq & 5 \\ & x_1 & + & x_2 & \leq & 3 \\ & x_1, x_2 \geq 0 & & & & \end{array}$$

Dual

$$\begin{array}{lrcrcl} \max & y_1 & + & y_2 & & \\ s.t. & y_1 & + & y_2 & \leq & 2 \\ & y_1 & + & y_2 & \leq & 2 \\ & y_1, y_2 \geq 0 & & & & \end{array}$$

Infeasible Primal → Unbounded Dual

Unbounded Primal → Infeasible Primal

27.22.7. Strong and Weak Duality

Problem Type	Duality	Weak Duality	Strong Duality
Min Primal **Max** Dual	$c^Tx \geq b^Ty$	$c^Tx > b^Ty$ if $x \neq x^*$ or $y \neq y^*$	$c^Tx^* = b^Ty^*$
Max Primal **Min** Dual	$c^Tx \leq b^Ty$	$c^Tx < b^Ty$ if $x \neq x^*$ or $y \neq y^*$	$c^Tx^* = b^Ty^*$

27.22.8. Complimentary Slackness

Slack variables of the dual LP

$$\begin{array}{rl} \max & c^Tx \\ s.t. & Ax = b \\ & x \geq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{rl} \min & y^Tb \\ s.t. & y^TA \geq c^T \\ & x \text{ urs} \end{array}$$

$$\begin{array}{rl} \min & y^Tb \\ s.t. & y^TA - v^T = c^T \\ & v \geq 0 \end{array}$$

$\bar{x}$ and $(\bar{y}, \bar{v})$ are primal and dual opimal iif they are feasible and $\bar{v}^T\bar{x} = \mathbf{0}$

Proof

We have

$$\begin{aligned} c^T\bar{x} &= (\bar{y}^TA - \bar{v}^T)\bar{x} \\ &= \bar{y}^TA\bar{x} - \bar{v}^T\bar{x} \\ &= \bar{y}^Tb - \bar{v}^T\bar{x} \end{aligned}$$

Therefore $\bar{v}^T\bar{x} = 0$ iif $c^T\bar{x} = \bar{y}^Tb$, i.e., $\bar{x}$ and $(\bar{y}, \bar{v})$ are primal and dual optimal according to strong duality.

- Note that $v^T\bar{x} = 0$ iif $\bar{v}_i\bar{x}_i = 0$ for all i as $\bar{x}_i \geq 0$ and $\bar{v} \geq 0$
- If a dual (primal) constraint is **nonbinding**, the corresponding primal (dual) variable is **zero**

Example

Primal

$$
\begin{array}{lrcrcl}
\max & x_1 & & & & \\
s.t. & 2x_1 & - & x_2 & \leq & 4 \\
 & 2x_1 & + & x_2 & \leq & 8 \\
 & & & x_2 & \leq & 3
\end{array}
$$
$$x_1, x_2 \geq 0$$

$\Leftrightarrow$

Dual

$$
\begin{array}{lrcrcrcl}
\min & 4y_1 & + & 8y_2 & + & 3y_3 & & \\
s.t. & 2y_1 & + & 2y_2 & & & \geq & 1 \\
 & -y_1 & + & y_2 & + & y_3 & \geq & 0
\end{array}
$$
$$y_1, y_2, y_3 \geq 0$$

Let s_i and v_j be the slack variables for the primal and dual LPs:

$$
\begin{array}{lrcrcccccl}
\max & x_1 & & & & & & & & \\
s.t. & 2x_1 & - & x_2 & + & s_1 & & & = & 4 \\
 & 2x_1 & + & x_2 & & & + s_2 & & = & 8 \\
 & & & x_2 & & & & + s_3 & = & 3
\end{array}
$$
$$x_1, x_2 \geq 0, \quad s_1, s_2, s_3 \geq 0$$

$\Leftrightarrow$

$$
\begin{array}{lrcrcrcccl}
\min & 4y_1 & + & 8y_2 & + & 3y_3 & & & & \\
s.t. & 2y_1 & + & 2y_2 & & & - v_1 & & = & 1 \\
 & -y_1 & + & y_2 & + & y_3 & & - v_2 & = & 0
\end{array}
$$
$$y_1, y_2, y_3 \geq 0, \quad v_1, v_2 \geq 0$$

If optimal solution:

$$x_1 v_1 = \mathbf{0}$$
$$x_2 v_2 = \mathbf{0}$$
$$\vdots$$

Similarly, for an optimal solution:

$$s_1 y_1 = \mathbf{0}$$
$$s_2 y_2 = \mathbf{0}$$
$$\vdots$$

Let $(\bar{x}, \bar{s})$ be primal optimal, we have $(\bar{x}, \bar{s}) = (3, 2, 0, 0, 1)$. Let's find the dual optimal solution $(\bar{y}, \bar{v})$ without solving the LP.

According to complimentary slackness $\bar{x_1}, \bar{x_2}, \bar{s_1} > 0$ imply $\bar{v_1} = \bar{v_2} = \bar{y_3} = 0$ since:

$$\bar{x_1}\bar{x_1} = 0$$
$$\bar{x_2}\bar{x_2} = 0$$
$$\bar{s_3}\bar{y_3} = 0$$

The two dual functional equalities are reduced to:

$$2\bar{y_1} + 2\bar{y_2} = 1$$
$$-\bar{y_1} + \bar{y_2} = 0$$

Solving the above equaltions results in $\bar{y_1} = \frac{1}{4}$ and $\bar{y_2} = \frac{1}{4}$.

$(\bar{y}, \bar{v})$ is then guaranteed to be dual optimal

$(z^* = 3 = w^*)$

If a primal solution is positive, the dual slack must be zero and vis versa.

$$x_i \left(c_i - \left(A^T y \right)_i \right) = \mathbf{0} \quad \forall i$$

If:

- $x_i > 0$: dual constraint is tight (equality holds)
- $x_i = 0$: dual constraint is slack (inequality not binding)

27.22.9. Shadow Prices

What if I have an additional unit of a particular resource?

For each resource there is a maximum price we are willing to pay for one additional unit

Definition: Shadow Price

For an LP that has an optimal colution, the **shadow price of a constraint** if the amount the objective value increases when the RHS of that constrint **increases** by 1, **assuming the current optimal basis remains optimal**

Sign or Shadow Price

Objective Function	Constraint		
	$\leq$	$\geq$	$=$
max	≥ 0	≤ 0	Free
min	≤ 0	≥ 0	Free

- If shifting a constraint does not affect the optimal solution, the shadow price is zero. Shadow prices are zero for contraints that are nonbinding at the optimal solution.

Finding all shadow prices:

- m: number of constraints

Example

The solution to the dual program is the shadow price of the primal

What are the shadow prices?

$$\begin{array}{lcccccc} \min & 6x_1 & + & 4x_2 & & \\ s.t. & x_1 & + & x_2 & \geq & 2 \\ & 2x_1 & + & x_2 & \geq & 1 \end{array}$$
$$x_1, x_2 \geq 0$$

Solve the dual LP

$$\begin{array}{lcccccc} \max & 2y_1 & + & y_2 & & \\ s.t. & y_1 & + & 3y_2 & \leq & 6 \\ & y_1 & + & y_2 & \leq & 4 \end{array}$$
$$y_1, y_2 \geq 0$$

The dual optimal solution is $y^* = (4, 0)$

So shadow prices are 4 and 0 respectively

The shadow price is the absolute increate on the objective value given an increate of 1 on the RHS values

Instead of solving m LPs (one for each constraint increase by 1), we can just solve the dual LP

27.23. Minimum Cost Network Flow (MCNF)

1. Maximum Flow
2. Shortest Path

IP $\rightarrow$ LP Relaxation $\rightarrow$ Integer Solution

Network

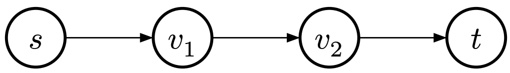
- Nodes (n)
- Edges (u, v)
- Directed
- Undirected

Path

Path from node s to node t is the set of arcs:

$$(s, v_1), (v_1, v_2), (v_2, t)$$

such that s and t are connected



- s: **source**
- t: **destination**

Cycle: Path whose source and destination are the same node

Simple path: A path that is not a cycle

Acyclic network: Network containing no cycle

Flow: actions of sending items through an edge

Flow size: Number of units sent through an edge

Weight: Property on an edge (e.g., distance, cost per unit, etc.)

Weighted network: network whose edges are Weighted

Capacity: Edge constraint

Capacitated network: network whose edges have capacities

Consider the weighted capacitated network $G = (V, E)$

- For node $i \in V$ there is a **supply quantity** b_i
  - $b_i > 0$: i is a **supply** node
  - $b_i < 0$: i is a **demand** node
  - $b_i = 0$: i is a **transhipment** node

- $\sum_{i \in V} b_i = 0$: Total supplies equal total demands
- For edge $(i, j) \in E$, the weight $c_{ij} \geq 0$ is the **cost** of each unit of flow

Objective: Satisfy all demand while minimizing cost

LP Formulation

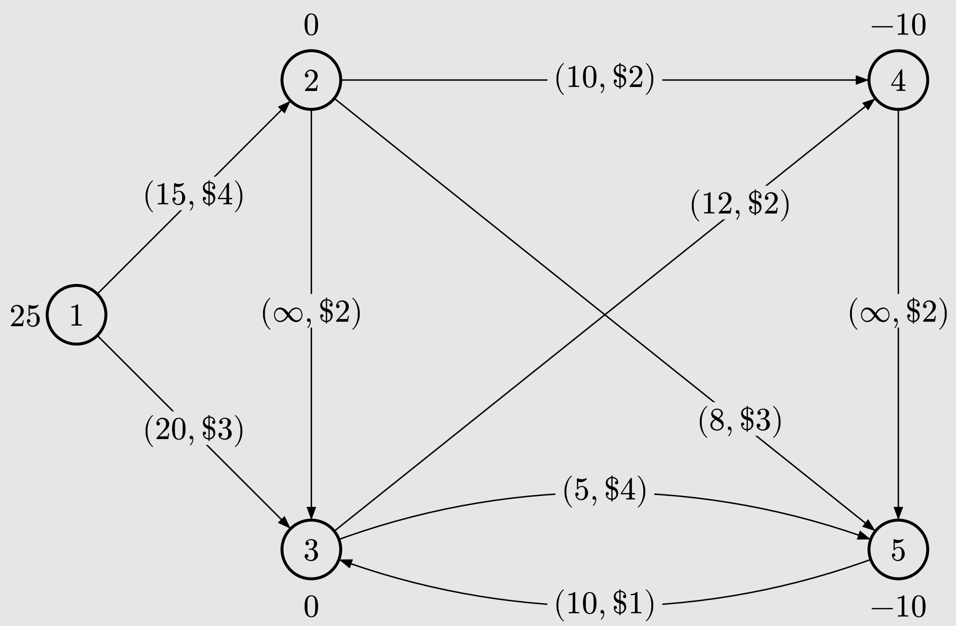
- The number of edges and constraints are equal
- Each variable appears in two constraints (coefficient 1 and coefficient −1)

Example

Decision valiables:

x_{ij}: Flow size of edge (i, j)

for all $(i, j) \in E$



Objective function:

$$\min \quad 4x_{12} + 3x_{13} + 2x_{23} + 2x_{24} + 3x_{25} + 2x_{34} + 1x_{35} + 2x_{45} + 1x_{53}$$

Capacity Constraints

$$\begin{aligned} x_{12} &\leq 15 \\ x_{13} &\leq 20 \\ x_{24} &\leq 10 \\ x_{25} &\leq 8 \\ x_{34} &\leq 12 \\ x_{35} &\leq 5 \\ x_{53} &\leq 10 \end{aligned}$$

Flow Balancing Constraints

- Supply node

$$x_{12} + x_{13} = 25$$

- Transhipment nodes

$$\begin{aligned} x_{12} &= x_{23} + x_{24} + x_{25} \\ x_{13} + x_{23} + x_{53} &= x_{34} + x_{35} \end{aligned}$$

- Demand nodes

$$\begin{aligned} x_{24} + x_{34} &= x_{45} + 10 \\ x_{25} + x_{35} + x_{45} &= x_{53} + 15 \end{aligned}$$

Complete Formulation

$$\begin{array}{lrrrrrrrrrrl} \min & 4x_{12} + 3x_{13} + 2x_{23} + 2x_{24} + 3x_{25} + 2x_{34} + x_{35} + 2x_{45} + 4\,x_{53} \\ s.t. & x_{12} + x_{13} & & & & & & & & = & 25 \\ & -x_{12} & & + x_{23} + x_{24} + x_{25} & & & & & & = & 0 \\ & & -x_{13} - x_{23} & & & + x_{34} + x_{35} & & & - x_{53} & = & 0 \\ & & & -x_{24} & & - x_{34} & & + x_{45} & & = & -10 \\ & & & & -x_{25} & & - x_{35} & - x_{45} & + x_{53} & = & -15 \end{array}$$

$$\begin{aligned} 0 \le x_{12} \le 15 \\ 0 \le x_{13} \le 20 \\ 0 \le x_{24} \le 10 \\ 0 \le x_{25} \le 8 \\ 0 \le x_{34} \le 12 \\ 0 \le x_{35} \le 5 \\ 0 \le x_{53} \le 10 \end{aligned}$$

As long as

- Supply quantities
- Upper bounds

are integers, the solution will be an **integer solution**

The LP relaxation of the IP formulation always gives an integer solution

Total unimodularity gives integer solutions

For a standard form LP $\min\{c^T x \mid Ax = b, x \ge 0\}$, if

- A is totally unimodular and
- $b \in \mathbb{Z}^n$,

then an optimal bfs x^* obtained by the simplex method must satisfy $x^* \in \mathbb{Z}^n$

Proof:

$$x_B = A_B^{-1} b = \frac{1}{\det A_B} A_B^{\text{adj}} b$$

Where:

- A_B^{adj}: adjugate matrix of A_B
- $\left(A_B^{\text{adj}}\right)_{ij}$: determinant of the matrix obtained by removing row j and column i from A_B

If A is totally unimodular, $\det(A_B)$ will be either 1 or −1 for any basis B. x_B is then an integer vector if b is an integer vector.

- If a standard form **LP** has a **totally unimodular coefficient matrix**, an optimal bfs will always be **integer**
- If a standard form **IP** has a **totally unimodular coefficient matrix**, its LP relaxation always gives **integer** solutions

Sufficient conditions for total unimodularity: For matrix A, if:

- all its elements are 1, 0, −1
- each column contains at most two nonzero elements
- rows can be divided into two groups so that for each column two nonzero elements are in the same group if and only if they are different (+ and -)

Then A is totally unimodular

Example

- All its elements are 1, 0, −1
- Each column contains at most two nonzero elements

Constraints:

$$\begin{aligned}
x_{12} + x_{13} \qquad\qquad\qquad\qquad\qquad\qquad\qquad &= 25\\
-x_{12} + x_{23} + x_{24} + x_{25} \qquad\qquad\qquad &= 0\\
-x_{13} - x_{23} + x_{34} + x_{35} - x_{53} &= 0\\
-x_{24} - x_{34} + x_{45} &= -10\\
-x_{25} - x_{35} - x_{45} + x_{53} &= -15
\end{aligned}$$

For simplex:

$$\begin{aligned}
x_{12} + x_{13} \qquad\qquad\qquad\qquad\qquad\qquad\qquad &= 25\\
-x_{12} + x_{23} + x_{24} + x_{25} \qquad\qquad\qquad &= 0\\
-x_{13} - x_{23} + x_{34} + x_{35} - x_{53} &= 0\\
{\color{red}+}x_{24} {\color{red}+} x_{34} {\color{red}-} x_{45} &= {\color{red}+}10\\
{\color{red}+}x_{25} {\color{red}+} x_{35} {\color{red}+} x_{45} {\color{red}-} x_{53} &= {\color{red}+}15
\end{aligned}$$

- Rows can be divided into two groups so that for each column two nonzero elements are in the same group if and only if they are different (+ and -)

$$
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & -1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & -1
\end{bmatrix}
$$

27.24. Transportation Problems

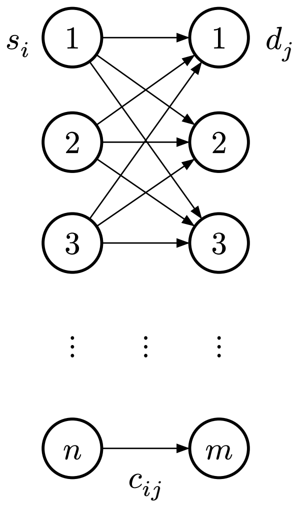
A firm owns n factories that supply one pruduct to m markets

- s_i: Capacity of factory i, $i = 1, ..., n$
- d_j: Demand of market j, $j = 1, ..., m$

Between factory i and market j there is a route

- c_{ij}: Unit cost for shipping one unit from factory i to market j

How to produce and ship the product to fulfill all demands while minimizing the total cost?



if $\sum_{i=1}^{n} s_i = \sum_{j=1}^{n} d_j$

- x_{ij}: Shipping quantity on arc (i, j), $i = 1, ..., n, j = 1, ..., m$

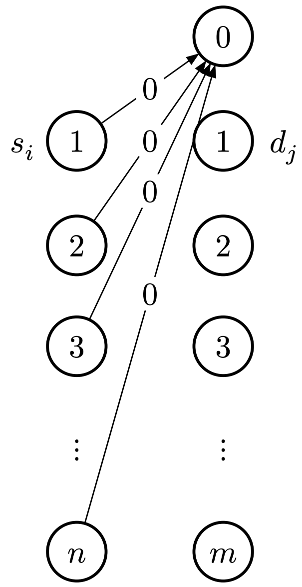
This is a MCNF problem:

- Factories are supply nodes whose supply quantity is s_i
- Markets are demand nodes whose supply quantity is $-d_j$
- No transhipment nodes
- Arc weights are unit transportation costs c_{ij}
- Arcs have unlimited capacities ($u_{ij} = \infty$)

If capacity is greater than demand:

$$\sum_{i=1}^{n} s_i > \sum_{j=1}^{n} d_j$$

- Create a “virtual market” (market 0) whose demand quantity is $d_0 = \sum_{i=1}^{n} s_i - \sum_{j=1}^{n} d_j$
- Arcs $(i, 0)$ have cost $c_{i0} = 0$
- Shipping to market 0 just means some factory capacity is unused



If different factories have different unit production costs c_i^P

- c_{ij} is updated to $c_{ij} + c_i^P$

If different markets habe different unit retailing costs c_j^R

- c_j is updated to $c_{ij} + c_j^R$

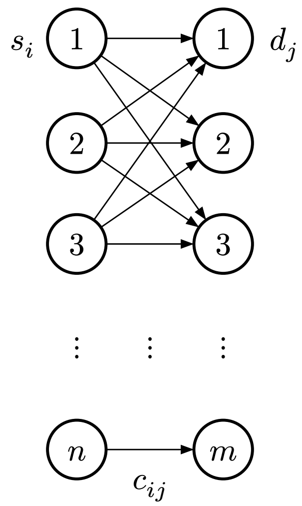
27.25. Assignment Problems

- A manager assignes n jobs to m workers
- The assigment is one-to-one (one job to one worker)
  - Jobs cannot be split
- c_{ij}: cost for worker j to complete job i

How to minimize total cost?

Special case of Transportation Problem

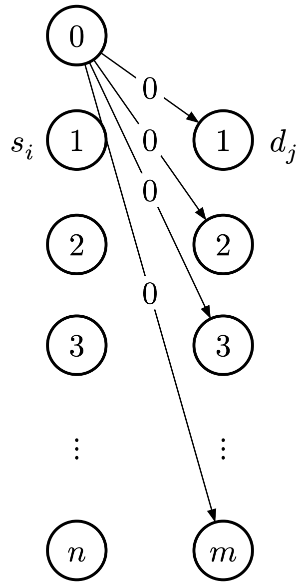
- Jobs are factories and workers are markets
- Each factory produces one item and each market demands one item
- Cost of shipping one item from factory i to maket j is c_{ij}



What if there are fewer jobs then workers:

$$\sum_{i=1}^{n} s_i < \sum_{j=1}^{n} d_j$$

- Create a “virtual job” (job 0)
- Arcs $(i, 0)$ have cost $c_{i0} = 0$
- Assigning job 0 just means some workers are unused



IP Formulation

- I: set of factories / jobs
- J: set of markets / workers

Transportation Problem

Assignment Problem

$$
\begin{aligned}
\min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
s.t. \quad & \sum_{j=1}^{m} x_{ij} = s_i \quad \forall i \in I \\
& \sum_{i=1}^{n} x_{ij} = d_j \quad \forall j \in J \\
& x_{ij} \in \mathbb{Z}^+ \quad \forall i \in I, j \in J
\end{aligned}
$$

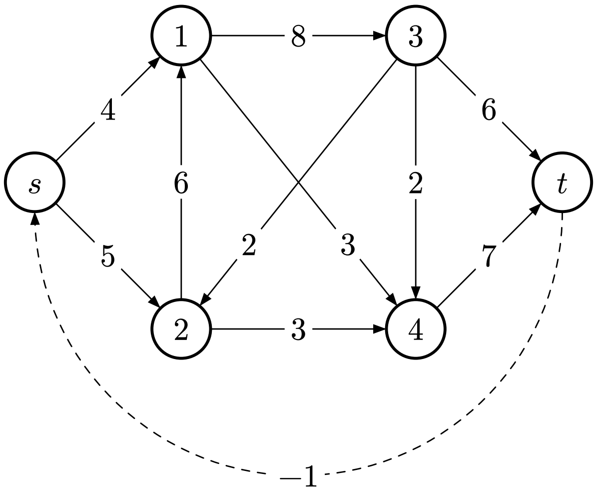
$$
\begin{aligned}
\min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
s.t. \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in I \\
& \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in J \\
& x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J
\end{aligned}
$$

- These matrices are **totally unimodular**

27.26. Transhipment Problems

If there are intermediary nodes in the transportation problem it is a transhipment problem

27.27. Maximum Flow Problems



$$
\begin{aligned}
\min \quad & -x_{ts} \\
s.t. \quad & \sum_{(i,k) \in E} x_{ik} - \sum_{(k,j) \in E} x_{kj} = 0 && \forall k \in V \\
& x_{ij} \leq u_{ij} && \forall (i,j) \in E \\
& x_{ij} \in \mathbb{Z}^+ && \forall (i,j) \in E
\end{aligned}
$$

Where:

- x_{ts}: flow size of the
- u_{ij}:

Objective: Maximize number of units sent from s to t given edge capacities

27.28. Shortest Path Problems

$$
\begin{aligned}
\min \quad & \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \\
s.t. \quad & \sum_{(s,j) \in E} x_{sj} = 1 \\
& \sum_{(i,t) \in E} x_{it} = 1 \\
& \sum_{(i,k) \in E} x_{ik} - \sum_{(k,j) \in E} x_{kj} = 0 \qquad \forall k \in T \\
& x_{ij} \in \{0, 1\} \qquad\qquad\qquad\qquad \forall (i,j) \in E
\end{aligned}
$$

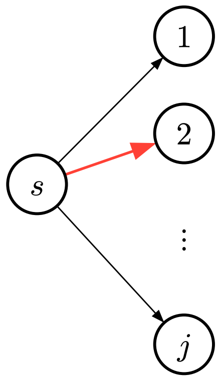
Where:

- s: supply node
- t: demand node
- x_{ij}: weather edge (i, j) is chosen (binary)
- d_{ij}: weight (distance) between nodes i and j

Objective: Get from s to t while minimizing distance

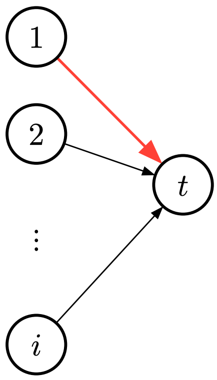
Out of all outgoing edges from s, one must be selected:

$$\sum_{(s,j) \in E} x_{sj} = 1$$



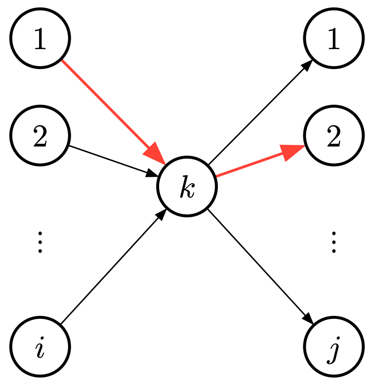
Out of all ingoing edges to t, one must be selected:

$$\sum_{(i,t) \in E} x_{it} = 1$$



For all transhipment nodes T,

$$\sum_{(i,k)\in E} x_{ik} - \sum_{(k,j)\in E} x_{kj} = 0 \quad \forall k \in T$$



Summary

Assigment problems are a special case of **Transportation** problems where:

- $b_i = 1$: net supply at node i

Transportation problems are a special case of **Transhipment** problems where:

- $b_i \neq 0$: net supply at node i

Transhipment problems are a special case of **MCNF** problems where:

- $u_{ij} = \infty$: No capacity on edge (i, j)

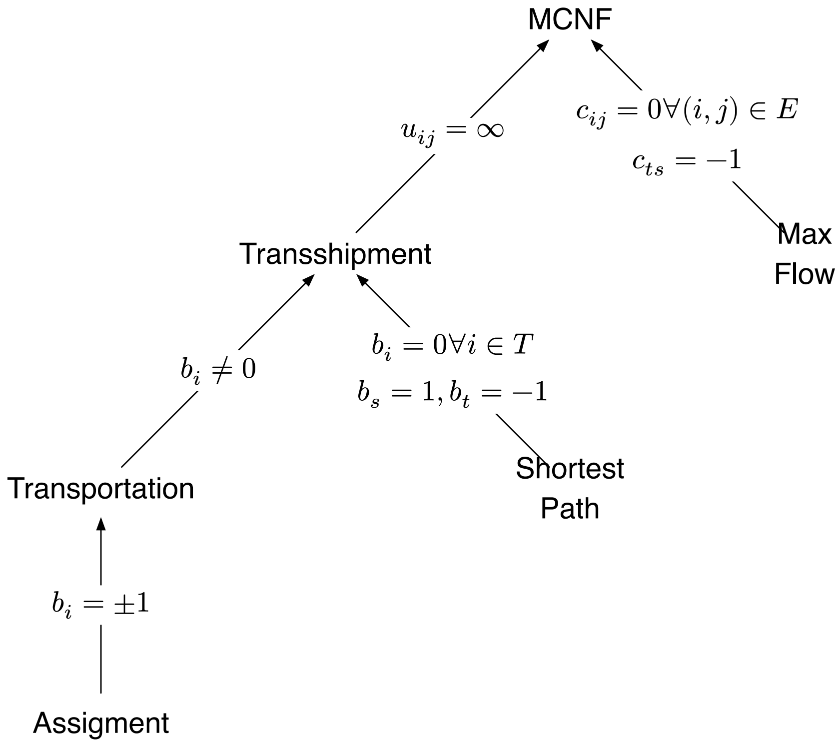
Shortest Path problem is a special case of **Transhipment** problems where:

- $b_s = 1$: One supply node with supply of 1
- $b_t = -1$: One demand node with supply of –1 (demand)
- $b_i = 0 \forall i \in T$: All others are transhipment nodes

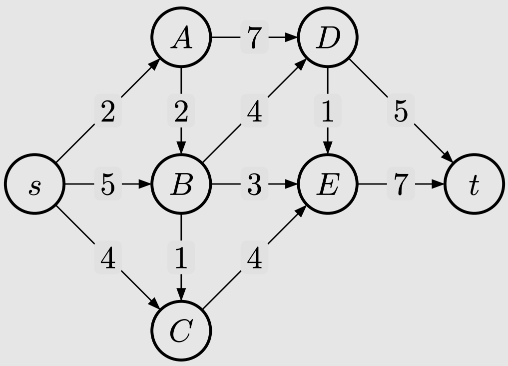
Maximum Flow problems are a special case of **MCNF** problems where:

- $c_{ij} = 0 \forall (i, j) \in E$: All original edges have 0 cost
- $c_{ts} = -1$: One “virtual edge” with cost –1

$$
\begin{aligned}
\min \quad & c^T x \\
s.t. \quad & Ax = b \\
& x \leq u \\
& x \geq 0
\end{aligned}
$$



Example

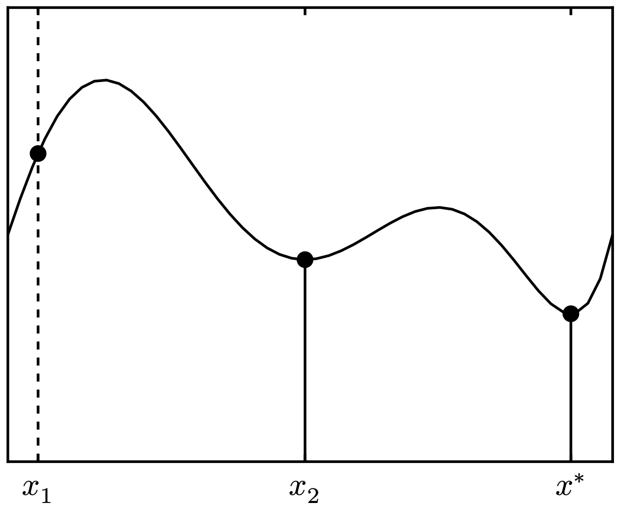


$$
A = \begin{bmatrix}
0 & 2 & 5 & 4 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 7 & 0 & 0 \\
0 & 0 & 0 & 1 & 4 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 7 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

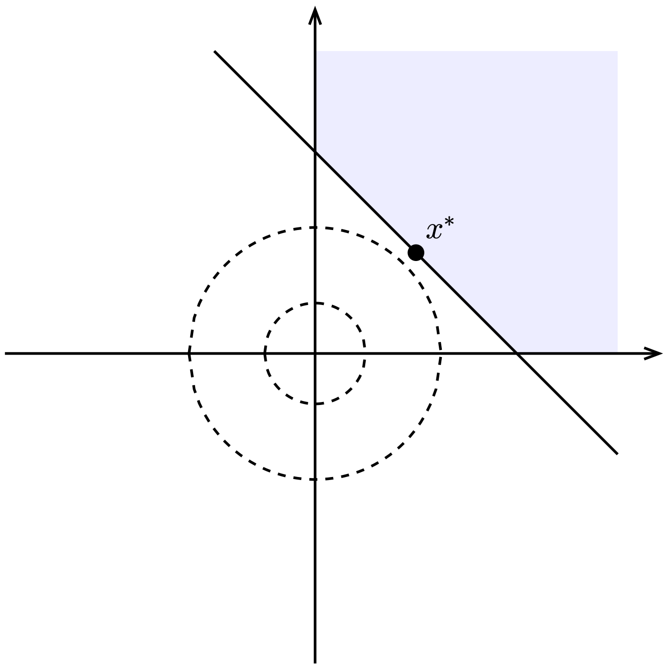
27.29. Convex Analysis

Difficulties of NLP

- A local minimum is not always a global minimum



- An optimal solution may not be an extreme point optimal solution



27.29.1. Convexity

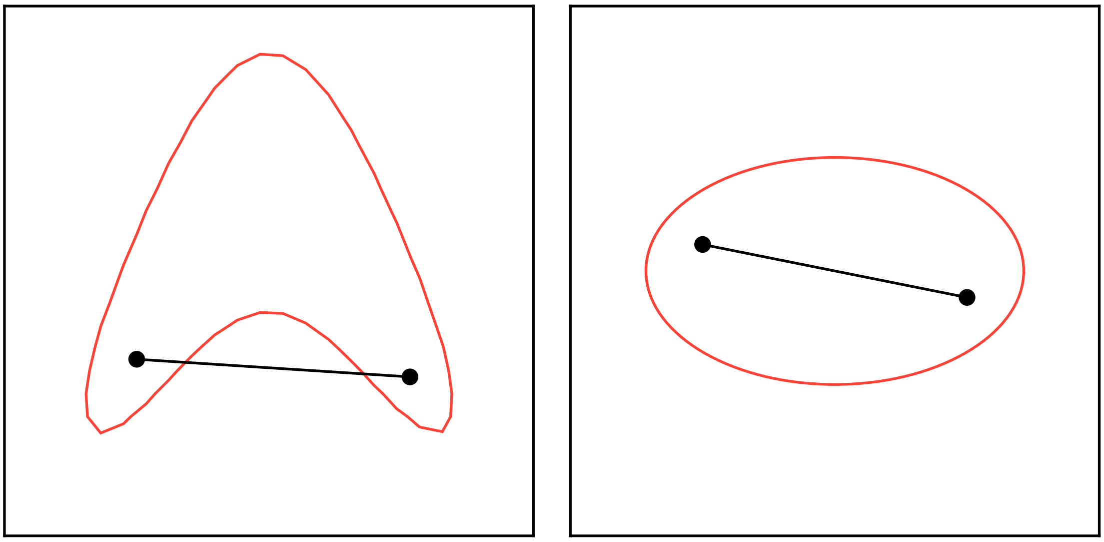
Convex Set

If for whatever two points you select in the set, the line segment connecting them also lies in the set, then the set is convex

A set F is convex if

$$\lambda x_1 + (1-\lambda)x_2 \in F$$

for all $\lambda \in [0,1]$ and $x_1, x_2 \in F$

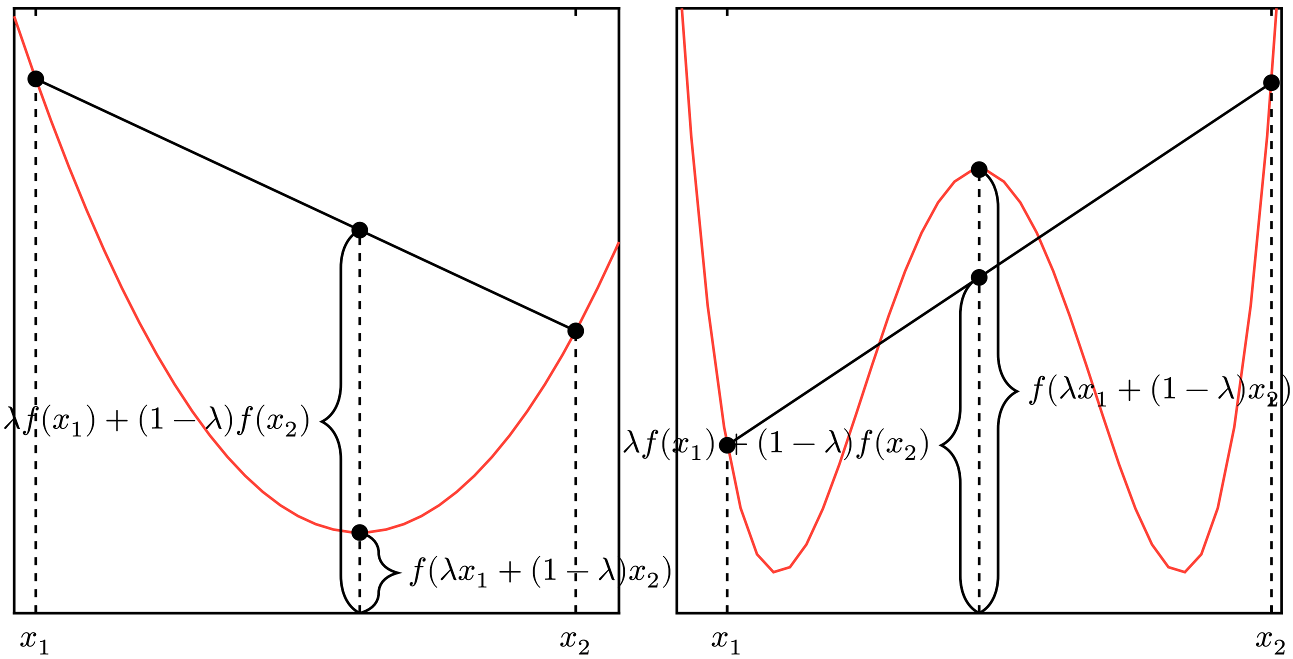


Convex Function

For a convex domain $F \subseteq \mathbb{R}^n$, a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex over F if

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

for all $\lambda \in [0,1]$ and $x_1, x_2 \in F$



Concave Function

For a convex domain $F \in \mathbb{R}^n$, a function $f : \mathbb{R}^n \to \mathbb{R}$ is concave over F if $-f$ is convex

Global Optimality of Convext Functions

Proposition 1. For a convex (concave) function f over a convex doain F, a local minimum (maximum) is a global minimum (maximum)

Two conditions:

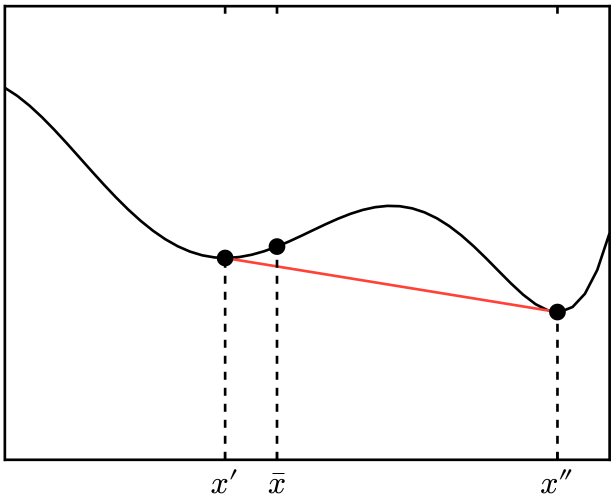
- Function needs to be convex
- Domain (feasible region) need to be convex

Proof

Suppose a local minimum x' is not a global minimum and there exists x'' such that $f(x'') < f(x')$. Consider a small enough $\lambda > 0$ such that $\bar{x} = \lambda x'' + (1-\lambda)x'$ satisfies $f(\bar{x}) > f(x')$. Such $\bar{x}$ exists because x' is a local minimum. Now, note that

$$\begin{aligned} f(\bar{x}) &= f(\lambda x'' + (1-\lambda)x') \\ &> f(x') \\ &= \lambda f(x') + (1-\lambda) f(x') \\ &> \lambda f(x'') + (1-\lambda) f(x') \end{aligned}$$

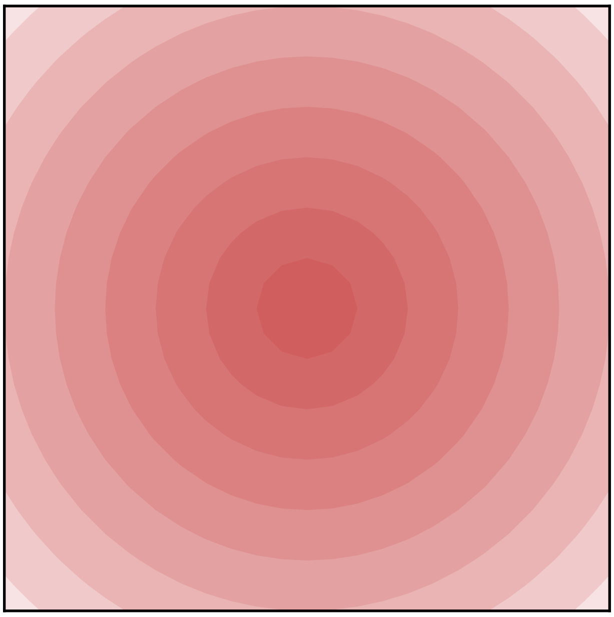
Which violates the fact that $f(\cdot)$ is convex. Therefore, by contradiction, the local minimum x' must be a global minimum.



Minimize a concave function

Proposition 2: For a concave function that has a global minimum over a convex feasible region, there exists a global minimum that is an extreme point

Extreme point: In convex analysis, an extreme point of a convex set C is a point in C that cannot be expressed as a strict convex combination of two other distinct points in C

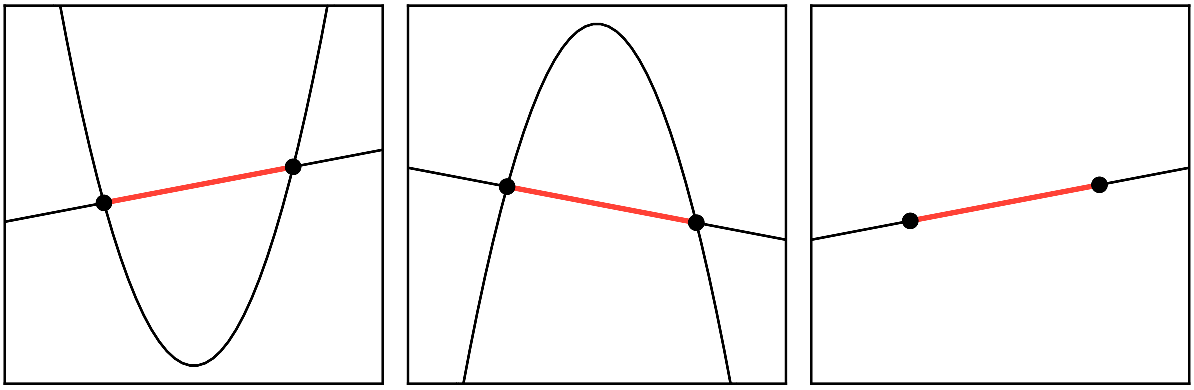


Special Case: LP

When we minimize $f(\cdot)$ over a convez feasible region F:

- If $f(\cdot)$ is **convex**, search for a **local minimum**
- If $f(\cdot)$ is **concave**, search among **extreme points** of F

For any LP we have both



Proposition 3. The feasible region of an LP is convex

The intersections of convex sets are convex

Proposition 4. A linear function $f : \mathbb{R}^n \to \mathbb{R}$ is both convex and concave

Proof:

$$f(\lambda x_1 + (1-\lambda)x_2) = \lambda f(x_1) + (1-\lambda)f(x_2)$$

Let $f(x) = c^T x + b$ be a linear function, $c \in \mathbb{R}^n, b \in \mathbb{R}$, then:

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &= c^T(\lambda x_1 + (1-\lambda)x_2) + b \\ &= \lambda(c^T x_1 + b) + (1-\lambda)(c^T x_2 + b) \\ &= \lambda f(x_1) + (1-\lambda)f(x_2) \end{aligned}$$

Convex Programming

Consider a general NLP

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & g_{i(x)} \leq b_i \quad \forall i = 1, ..., m \end{aligned}$$

If:

- The feasible region $F = \{x \in \mathbb{R}^n \mid g_i(x) \leq b_i \forall i = 1, ..., m\}$ (intersection of all the regions satisfying the constraints)
- f is convex over F

A local minimum is a global minimum

Definition 0: Convex Program (CP)

An NLP is convex if its feasible region is convex and its objective function is convex over the feasible region

Convex Programming:

- Minimizing convex function
- Maximizing concave function

Subject to a convex feasible region

Local min = Global min

For an NLP

$$\min_{x \in \mathbb{R}^n} \{f(x) \mid g_i(x) \leq b_i \forall i - 1, ..., m\}$$

if f and g_is are all convex functions, the NLP is a Convex Program

Proof:

We only need to prove that the feasible region is convex, which is implied if $F_i = \{x \in \mathbb{R}^n \mid g_i(x) \leq b_i\}$ is convex for all i. For two points $x_1, x_2 \in F_i$ and an arbitrary $\lambda \in [0, 1]$, we have

$$\begin{aligned} g_i(\lambda x_1 + (1 - \lambda) x_2) &\leq \lambda g_i(x_1) + (1 - \lambda) g_i(x_2) \\ &\leq \lambda b_i + (1 - \lambda) b_i = b_i \end{aligned}$$

Which implies that F_i is convex. repeating this argument for all i completes the proof

If each constraint independently given a convex feasible region, then their intersection is convex

Definition 0: Affine Combination

An affine combination of points $x_1, x_2, ..., x_k \in \mathbb{R}^n$, is any point of the form:

$$\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_k x_1 + x_k$$

where the coefficients sum to 1:

$$\alpha_1 + \alpha_2 + ... + \alpha_k = 1$$

This is like a linear combination, but with the extra condition that the weights add up to 1.

For a twice differentiable function $f : \mathbb{R} \to \mathbb{R}$ over an interval (a, b):

- f is convex over (a, b) iif $f''(x) > 0$ for all $x \in (a, b)$
- $\bar{x}$ is a local minimum over (a, b), iif $f'(x) = 0$
- If f is concave over (a, b), x^* is a global minimum over (a, b) iif $f'(\bar{x}) = 0$

First order condition (FOC) $f'(x) = 0$

- FOC is necessary for local optimality
- FOC is sufficient for global optimality if f is convex

Example

Economic Order Quantity (EOQ)

Determine the order quantity in each order

- Demand is deterministic and occurs at a constant rate
- Each order incurs a fixed cost (independent of the order size)
- No shortage allowed
- Lead time is zero
- Holding cost is proportional to the average inventory
- Constant holding cost

Parameters:

- D: annual demand (units/year)
- K: unit ordering cost ($/order)
- h: annual holding cost ($/unit/year)
- p: unit purchase cost ($/unit)

Decision Variable:

- q: order quantity (units/order)

Objective:

- Minimize the total annual cost

Average inventory level:

$$\frac{q}{2}$$

Annual holding cost:

$$H = h\left(\frac{q}{2}\right) = \frac{hq}{2}$$

Annual purchase cost:

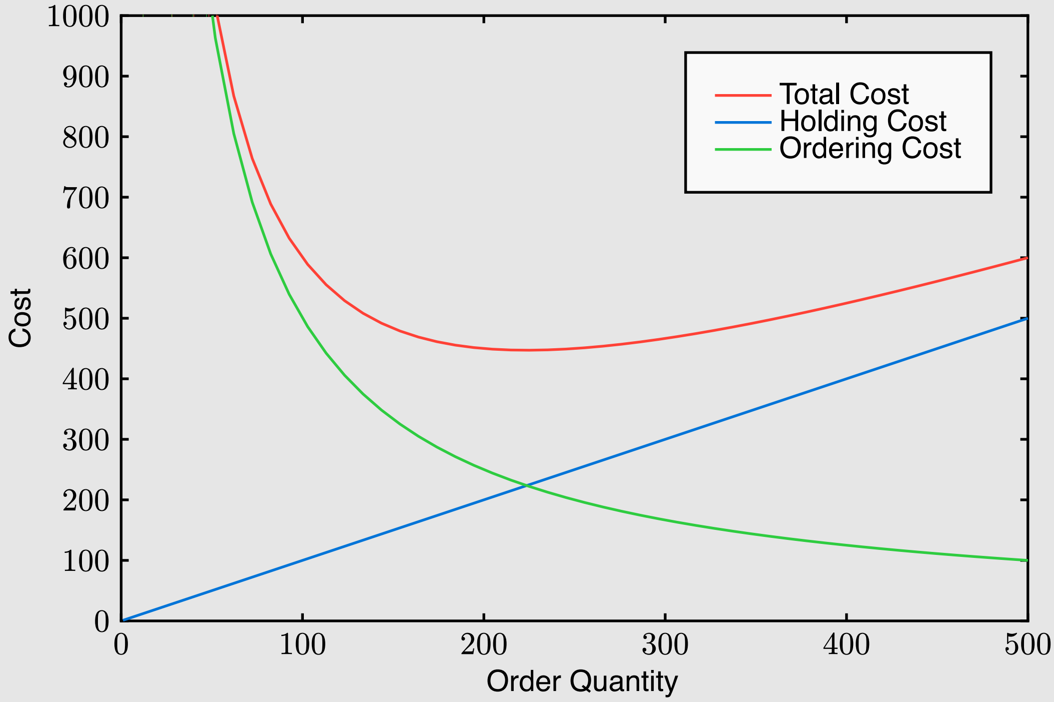
$$pD$$

Annual ordering cost:

$$O = \left(\frac{D}{q}\right)K = \frac{DK}{q}$$

NLP, since pD does not depend on q it does not affect the minimizer:

$$\min_{q>0} f(q) = \left(\frac{D}{q}\right)K + \left(\frac{q}{2}\right)h$$
$$= \frac{DK}{q} + \frac{qh}{2}$$



For:

$$TC(q) = \frac{KD}{q} + \frac{hq}{2}$$

we have

$$TC'(q) = -\frac{KD}{q^2} + \frac{h}{2}$$

and

$$TC''(q) = \frac{2KD}{q^3}$$

Since a twice differentiable function is convex If

$$f''(x) > 0 \quad \forall x \in (a, b)$$

and

$$K > 0, \quad D > 0, \quad q > 0$$

we have

$$TC''(q) = \frac{2KD}{q^3} > 0$$

Therefore, $TC(q)$ is convex

Let q^* be the quantity satifying the FOC:

$$TC'(q^*) = -\frac{KD}{(q^*)^2} + \frac{h}{2} = 0 \quad \Rightarrow \quad q^* = \sqrt{\frac{2KD}{h}}$$

Implications:

- If order cost K increases, order quantity q^* increases
- If demand D increases, order quantity q^* increases
- If holding cost h increases, order quantity q^* decreases

27.30. Multivariate Convex Analysis

An optimal solution either:

- Satisfies the FOC
- Lies on the boundary of the feasible region

If a NLP is a CP, a feasible point satifying the FOC is optimal (any local minimum is a global minimum)

For a function $f : \mathbb{R}^n \to \mathbb{R}$, its ith partial derivative is $\frac{\partial f(x)}{\partial x_i}$

For a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, if all second order partial derivatives are continuous:

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$$

for all $i = 1, ..., n$ and $j = 1, ..., n$.

Single variate case:

- For $f : \mathbb{R} \to \mathbb{R}$, f is convex iif $f''(x) \geq 0$ for all x

Multivariate case:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \qquad \nabla^2 f(x) = H_f = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Example

$$f(x_1, x_2, x_3) = x_1^2 + x_2 x_3 + x_3^3$$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \frac{\partial f(x)}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ x_3 \\ x_2 + 3x_3^2 \end{bmatrix}$$

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f(x)}{\partial x_3 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_3 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_3^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 6x_3 \end{bmatrix}$$

You may ask:

- What is the gradient at a point: $\nabla f(3, 2, 1)$
- What is the Hessian at a point: $\nabla^2 f(3, 2, 1)$

Single Variate Function $f : \mathbb{R} \to \mathbb{R}$	• f is convex in $[a, b]$ if $f''(x) \geq 0$ for all $x \in [a, b]$ • $\bar{x}$ is an interior local minimum if $f'(\bar{x}) = 0$ • If f is convex in $[a, b]$, x^* is a global minimum iif $f'(x^*) = 0$
Multi Variate Function $f : \mathbb{R}^n \to \mathbb{R}$	• f is convex in a convex set $F \subseteq \mathbb{R}^n$ if $\nabla^2 f(x)$ is positive semi-definite for all $x \in F$ • $\bar{x}$ is an interior local minimum if $\nabla f(\bar{x}) = \mathbf{0}$ • If f is convex in a convex set F, x^* is a global minimum iif $\nabla f(x^*) = 0$

Definition 0: Positive Semi-Definite (PSD) Matrix

A symmetric matrix A is positive semi-definite if $\mathbf{x}^T A \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$

Example

Semi-Definite Matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$x^T A x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= 2x_1^2 + 2x_1x_2 + 2x_2^2$$

$$= 2(x_1 + x_1)^2 + x_1^2 + x_2^2 \geq 0$$

Given a function f, when is its Hessian $\nabla^2 f$ PSD?

For a symmetric matrix A, the following statements are equivalent:

- A is **positive semi-definite**
- All **eigenvalues** of A are non-negative
- All **principal minors** of A are non-negative

A's level-k principal minors is the determinant of a $k \times k$ submatrix whose diagonal is a subset of A's diagonal.

A sufficient condition is for A's **leading** principl minors to all **positive**

For a function f:

1. Find **Hessian** $\nabla^2 f(x)$
2. Find **eigenvalues** or **principal minors** of $\nabla^2 f(x)$
3. Determine over what region $\nabla^2 f(x)$ is **PSD**

The function is convex over that region

Example

$$\min_{x \in \mathbb{R}^2} f(x_1, x_2)$$

$$f(x_1, x_2) = x_1^2 + x_2^2 + x_1x_2 - 2x_1 - 4x_2$$

$$\nabla f(x) = \begin{bmatrix} 2x_1 + x_2 - 2 \\ x_1 + 2x_2 - 4 \end{bmatrix} \qquad \nabla^2 f(x) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Find **eigenvalues**

$$Ax = \lambda x \quad \iff \quad (A - \lambda I)x = 0 \quad \iff \quad \det(A - \lambda I) = 0$$

$$\begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 0 \quad \iff \quad 3 - 4\lambda + \lambda^2 = 0 \quad \iff \quad \lambda = 1 \text{ or } 3$$

Or find **leading principal minors**

$$|2| = 2 \quad \text{and} \quad \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix} = 3$$

So $\nabla^2 f(x_1, x_2)$ is PSD and thus $\min_{x \in \mathbb{R}^2} f(x_1, x_2)$ is a CP.

The FOC requires $2x_1^* + x_2^* - 2 = 0$ and $x_1^* + 2x_2^* - 4 = 0$

I.e., $(x_1^*, x_2^*) = (0, 2)$

Example

$$\min_{x \in \mathbb{R}^2} f(x_1, x_2)$$

$$f(x_1, x_2) = x_1^3 + 4x_1x_2 + \frac{1}{2}x_2^2 + x_1 + x_2$$

$$\nabla^2 f(x) = \begin{bmatrix} 6x_1 & 4 \\ 4 & 1 \end{bmatrix}$$

- 1st leading principle minor $6x_1 \geq 0$ ($x_1 \geq 0$)
- 2nd leading principal minor $6x_1 - 16 \geq 0$ ($x_1 \geq \frac{8}{3}$)
- $1 \geq 0$

Therefore, the function is convex iif $x_1 \geq \frac{8}{3}$

Example

Two-Product Pricing

A retailer sells product 1 and 2 at prices p_1 and p_2. For product i the demand q_i is:

$$q_1 = a - p_1 + bp_2$$
$$q_2 = a - p_2 + bp_1$$

where $a > 0$ and $b \in [0, 1)$. The retailer sets p_1 and p_2 to maximize its total profit.

If p_1 and p_2 are substitutes of one another then the price of one will affect the demand of the other.

1. Why $b \in [0, 1)$?

If $b \geq 1$: the other product's price will have the equal or greater impact on your own product's price

If $b < 0$: Complimentary products (rather than substitutes), the higher the other product's price, the lower our demand

2. Formulate problem

$$\max_{p_1, p_2} \quad p_1\underbrace{(a - p_1 + bp_2)}_{q_1} + p_2\underbrace{(a + bp_1 - p_2)}_{q_2}$$

Let

$$f(p) = -[p_1(a - p_1 + bp_2) + p_2(a + bp_1 - p_2)$$

3. Is it a CP?

$$\nabla^2 f(p) = \begin{bmatrix} 2 & -2b \\ -2b & 2 \end{bmatrix}$$

Which is PSD if $b \in [0, 1)$ since

- 1st leading principle minor $2 \geq 0$
- 2nd leading principle minor $4 - 4b^2$ ($4(1-b)(1+b) \geq 0$)

Therefore $f(p)$ is convex and $-f(p)$, the objective function is concave

The problem is a CP

4. Solve problem

$\nabla f(p) = 0$ rquires:

- $-a + 2p_1 - 2bp_2 = 0$
- $-a + 2p_2 - 2bp_1 = 0$

So,

$$p_1 = p_2 = \frac{a}{2(1-b)}$$

5. How does optimal prices change with a and b?

When

- a increases, the two prices increase: price of product increases when demand increases
- b increases, the two prices increase: effective demand becomes larger

Example

Question 1

For each of the following sets, select all that are convex:

1. $S = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 4, x_1 \geq 0, x_2 \leq 0\right\}$

✅ $x_1^2 + x_2^2 \leq 4$, a disk of radius 2, is convex and the $x_1 \geq 0, x_2 \leq 0$ lines are also convex

The intersection of convex sets is convex

2. $S = \left\{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1^2 + x_2^2 \leq 4, x_1 \geq 0, x_2 \leq 0\right\}$

❌ Integer programs are never convex in the usual sense, because the feasible set of integers is discrete

3. $S = \left\{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^2 \leq 4, x_1 \geq 0, x_2 \leq 0\right\}$

✅ $\sum_{i=1}^n x_i^2 \leq 4$, a n-dimensional sphere (hypersphere) of radius 2, is convex and the $x_1 \geq 0, x_2 \leq 0$ lines are also convex

The intersection of convex sets is convex

4. $S = \{x \in \mathbb{R}^n \mid x_1 + x_2 = 10(x_3 + x_4)\}$ where $n \geq 4$

✅ $h(x)$ is a linear function ($h(x) = x_1 + x_2 - 10x_3 - 10x_4$) so $h(x)$ is also affine and is therefore convex

5. $S = \left\{x \in \mathbb{R}^n \mid \max_{i=1,\ldots,n}\{x_i\} = 10\min_{i=1,\ldots,m}\{x_i\}\right\}$

❌ $\min(x)$ and $\max(x)$

Question 2

For each of the following functions, select all that are convex over the given region:

1. $f(x) = 2x^3 - x^2 - 2x + 1$ for $x \in \mathbb{R}$

$f'(x) = 6x^2 - 2x - 2$

$f''(x) = 12x - 2$

❌

2. $f(x) = \begin{cases} -x & \text{if } x<1 \\ -1 & \text{if } x \geq 1 \end{cases} \quad \text{for } x \in \mathbb{R}$

✅

3. $f(x) = x_1^2 - 4x_1x_2 + 3x_2^2 + 3x_1 + 4x_2$ for $x \in \mathbb{R}^2$

❌

4. $f(x) = \sum_{i=1}^{n} (x_i - a)^2$ where $a > 0$, $x \in \mathbb{R}^n$

✅

5. $f(\alpha, \beta) = \sum_{i=1}^{n} (\alpha + \beta x_i - y_i)^2$, where $x_i, y_i, \alpha, \beta \in \mathbb{R}$

✅

Question 3

For the NLP

$$\begin{array}{rl} \min & 2x^3 - x^2 - 2x + 1 \\ s.t. & x \geq -1 \end{array}$$

which of the following statements is correct?

1. This is a convex program.

❌

2. There are two local minimizers.

✅

3. The unique global minimizer is a boundary point.

❌

4. This program is unbounded.

❌

5. None of the above.

❌

Question 4

For each of the following matrices, select all that are positive semi-definite:

1.

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

✅

2.

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

❌

3.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

✅

4.

$$A = \begin{bmatrix} 1 & 2 & 3 & \dots & n \\ 0 & 2 & 3 & \dots & n \\ 0 & 0 & 3 & \dots & n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & n \end{bmatrix}$$

✅

5.

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

❌

Question 5

For the following statements, select all that are **incorrect**:

1. An optimal solution to a linear program must be a boundary point.

❌ not guaranteed if the objective is constant over the feasible region

2. An optimal solution to a linear program must be an extreme point.

✅

3. An optimal solution to a nonlinear program can be an interior point.

❌

4. An optimal solution to a nonlinear program must be an interior point.

✅

5. A global optimal solution to a nonlinear program must be a local optimal solution.

❌

27.31. Sensitivity Analysis

27.31.1. Dual Simplex Method

Example

A company is selling 2 products

- Producing 1 unit of product 1 requires 1 unit of resource 1 and 1 unit of resource 2, which can be sold for \$2
- Producing 1 unit of product 2 requires 1 unit of resource 1 and 2 unit of resource 2, which can be sold for \$3
- Total amount of resources 1 is 4 units
- Total amount of resources 2 is 6 units

$$\begin{array}{rrcrcl} \max & 2x_1 & + & 3x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 4 \\ & x_1 & + & 2x_2 & \leq & 6 \end{array}$$

$$x_1, x_2 \geq 0$$

–2	–3	0	0	0
1	1	1	0	4
1	2	0	1	6

0	–1	2	0	8
1	1	1	0	4
0	1	–1	1	2

0	0	1	1	10
1	0	2	–1	2
0	1	–1	1	2

- An optimal solution $(x_1^*, x_2^*) = (2, 2)$
- The objective is $z^* = 10$

Additional activity

The company now produces a 3rd product.

- 1 unit of product 3 requires 1 unit of resource 2 and is sold for $8

$$\begin{array}{lrcrcrcl} \max & 3x_1 & + & 2x_2 & + & 8x_3 & & \\ s.t. & x_1 & + & 2x_2 & & & \leq & 4 \\ & 2x_1 & + & 4x_2 & + & x_3 & \leq & 6 \end{array}$$

$$x_1, x_2, x_3 \geq 0$$

We don't need to solve a new linear program

The decision variable set changes from (x_1, x_2) to (x_1, x_2, x_3), the solution $(x_1^*, x_2^*, 0)$ is feasible

All we need to do is to check whether we should produce some product 3

- If no, the current solution $(x_1^*, x_2^*, 0)$ is optimal
- If yes, we increase the nonbasic variable x_3 until one basic variable becomes 0

All we need is the (vector of) **reduced costs**:

$$c_B^T B^{-1} N - c_N$$

Where:

- B: basis matrix

- N: non-basis matrix
- c_B: vector of objective coefficients corresponding to the basic variables
- c_N: vector of objective coefficients corresponding to the non-basic variables

For a single nonbasic variable (x_3), ot simplifies to:

$$c_B^T B^{-1} A_j - c_j$$

After we solve the original problem, we have $B^* = (x_1, x_2)$ and $N^* = (s_1, s_2)$

- The optimal basis is B^*

When we add a new decision variable x_3, it is 0 (**nonbasic**) at the beginning

Therefore, to solve the new problem, we may start from the basis $B = (x_1, x_2)$ and the set of nonbasic variables $N = (x_3, s_1, s_2)$

We start with the optimal tableau

$$\begin{array}{cccc|c} 0 & 0 & 1 & 1 & 10 \\ \hline 1 & 0 & 2 & -1 & 2 \\ 0 & 1 & -1 & 1 & 2 \\ \multicolumn{2}{c}{\underbrace{\hspace{3em}}_{\text{basic}}} & \multicolumn{2}{c}{\underbrace{\hspace{3em}}_{\text{nonbasic}}} & \end{array}$$

$$\begin{array}{ccccc|c} 0 & 0 & ? & 1 & 1 & 10 \\ \hline 1 & 0 & ? & 2 & -1 & 2 \\ 0 & 1 & ? & -1 & 1 & 2 \\ \multicolumn{2}{c}{\underbrace{\hspace{3em}}_{\text{basic}}} & \multicolumn{3}{c}{\underbrace{\hspace{6em}}_{\text{nonbasic}}} & \end{array}$$

The vecots of constraint coefficients for nonbasic variables is:

$$B^{-1}N$$

for column j:

$$B^{-1}A_j$$

Where A_j is the coefficient column for x_3:

$$\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

And the vector if reduced costs for nonbasic variables is

$$c_B^T B^{-1} N - c_N^T$$

For column j, that value is

$$c_B^T B^{-1} A_j - c_j$$

$$[2 \quad 3]\begin{bmatrix}-1\\1\end{bmatrix} - 8 = -7$$

$$\begin{array}{ccccc|c} 0 & 0 & -7 & 1 & 1 & 10 \\ \hline 1 & 0 & -1 & 2 & -1 & 2 \\ 0 & 1 & 1 & -1 & 1 & 2 \\ \underbrace{\quad\quad}_{\text{basic}} & & \underbrace{\quad\quad\quad\quad\quad}_{\text{nonbasic}} & & & \end{array}$$

From this new tableau keep iterating

0	0	–7	1	1	10
1	0	–1	2	–1	2
0	1	1	–1	1	2

0	7	0	–6	8	24
1	1	0	1	0	4
0	1	1	–1	1	2

6	13	0	0	8	28
1	1	0	1	0	4
1	2	1	0	1	6

$(x_1^{**}, x_2^{**}, x_3^{**}) = (0, 0, 6)$ with $z^{**} = 48$

Allows asking **what if** questions:

- What if it is \$5 instead of \$8
- What if it takes 2 of resource on instead of 1?
- What if it takes 1 of resource 2 instead of 0?

Additional Constraints

Primal

$$\begin{array}{rrcrcr} \max & 2x_1 & + & 3x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 4 \\ & x_1 & + & 2x_2 & \leq & 6 \\ & & x_1, x_2 \geq 0 & & & \end{array}$$

We add a new contraint:

Dual

$$\begin{array}{rrcrcr} \max & 2x_1 & + & 3x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 4 \\ & x_1 & + & 2x_2 & \leq & 6 \\ & x_1 & & & \leq & 1 \\ & & x_1, x_2 \geq 0 & & & \end{array}$$

We may plug in in the original optimal solution $(x_1^*, x_2^*) = (2, 2)$ into the new constraint

- If it is feasible, it is optimal for the new problem
- In our example, it is not feasible because $2 > 1$

The original optimal tableau

0	0	1	1	10
1	0	2	–1	2
0	1	–1	1	2

The new constraint $x_1 \leq 1$ introduces a new slack variable (s_3)

Include s_3 to be a basic variable

Let $B = (x_1, x_2, s_3)$ and $N = (s_1, s_2)$

We have

$$c_B = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}, \quad c_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 6 \\ 1 \end{bmatrix}$$

We then have

$$B^{-1} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 1 & 0 \\ -2 & 1 & 1 \end{bmatrix}, \quad B^{-1}N = \begin{bmatrix} 2 & -1 \\ -1 & 1 \\ -2 & 1 \end{bmatrix}, \quad c_B^T B^{-1} N - c_N^T = [1 \quad 1], \quad c_B^T B^{-1} b = 10$$

0	0	1	1	0	10

$$
\begin{array}{ccccc|c}
\hline
1 & 0 & 2 & -1 & 0 & 2 \\
0 & 1 & -1 & 1 & 0 & 2 \\
0 & 0 & -2 & 1 & 1 & -1
\end{array}
$$

$$\underbrace{\quad\quad}_{\text{basic}} \quad \underbrace{\quad\quad}_{\text{nonbasic}} \quad \underbrace{\quad}_{\text{basic}}$$

This is an invalid simplex tableau (RHS column column contains a negative value)

- This means $B = (x_1, x_2, s_3)$ is infeasible (as we already know)

Linear Programmin Duality

We know a primal constraint is a dual variable

If a primal LP has one new constraint, it's dual LP will have one new variable

Primal

$$
\begin{array}{rrcrcr}
\max & 2x_1 & + & 3x_2 & & \\
s.t. & x_1 & + & x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 6
\end{array}
$$
$$x_1, x_2 \geq 0$$

Dual

$$
\begin{array}{rrcrcr}
\min & 4y_1 & + & 6y_2 & & \\
s.t. & y_1 & + & y_2 & \geq & 2 \\
 & y_1 & + & 2y_2 & \geq & 3
\end{array}
$$
$$y_1, y_2 \geq 0$$

We add a new contraint:

Primal LP with new constraint

$$
\begin{array}{rrcrcr}
\max & 2x_1 & + & 3x_2 & & \\
s.t. & x_1 & + & x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 6 \\
 & x_1 & & & \leq & 1
\end{array}
$$
$$x_1, x_2 \geq 0$$

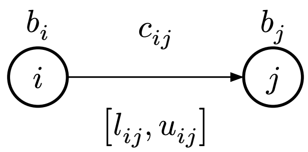
Dual LP with new variable

$$
\begin{array}{rrcrcrcr}
\min & 4y_1 & + & 6y_2 & + & y_3 & & \\
s.t. & y_1 & + & y_2 & + & y_3 & \geq & 2 \\
 & y_1 & + & 2y_2 & & & \geq & 3
\end{array}
$$
$$y_1, y_2, y_3 \geq 0$$

27.32. Netwrok Flow Optimization

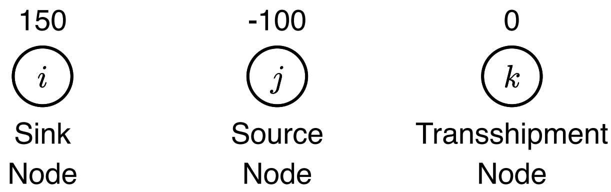
Types of Problems:

- Maximum Flow
- Minimum Cost Flow
- Multi-Commodity Flow



Node i

- Sink node: $b_i > 0$ — Has demand of b_i units
- Source node: $b_i < 0$ — Has supply of $-b_i$ units
- Transspipment node: $b_i = 0$ — Neither supply or demand



Units shipped from node i to node j:

$$x_{ij}$$

Minimize:

$$Z = \sum_{i,j} c_{ij} x_{ij}$$

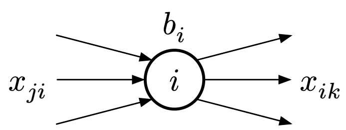
s.t.

$$\sum_k x_{ki} - \sum_l x_{il} = \text{ or } \leq \text{ or } \geq b_1$$

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

Where:

- c_{ij}: Unit cost of flow from node i to node j
- b_i: Demand on node i
- l_{ij}: flow lower bound from i to j
- u_{ij}: Flow upper bound from i to j
- $\sum_k x_{ki}$: Inflow to i
- $\sum_l x_{il}$: Outflow from j



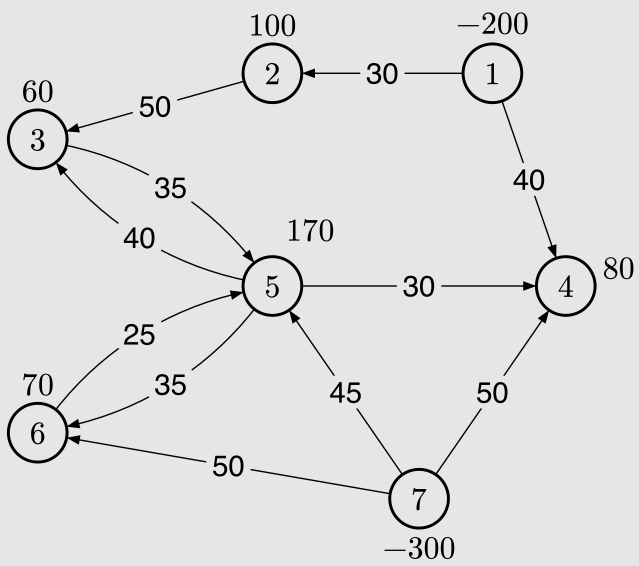
If:

- Total Supply $=$ Total Demand   [Inflow to i] - [Outflow from i] $= b_i$
- Total Supply $>$ Total Demand   [Inflow to i] - [Outflow from i] $\geq b_i$
- Total Supply $<$ Total Demand   [Inflow to i] - [Outflow from i] $\leq b_i$

Important:

- One decision variable x_{ij} for each edge (i, j)
- One flow balancing constraint for each node i

Example



Minimize

$$Z = 30x_{12} + 40x_{14} + 50x_{23} + 35x_{35} + 40x_{53}$$
$$+30x_{54} + 35x_{56} + 25x_{65} + 50x_{74} + 45x_{75} + 50x_{76}$$

s.t.

$x_{12} + x_{14} \leq 200$ (Node 1)

$x_{12} + x_{23} \geq 100$ (Node 2)

$x_{23} + x_{53} - x_{35} \geq 60$ (Node 3)

$x_{14} + x_{54} + x_{74} \geq 80$ (Node 4)

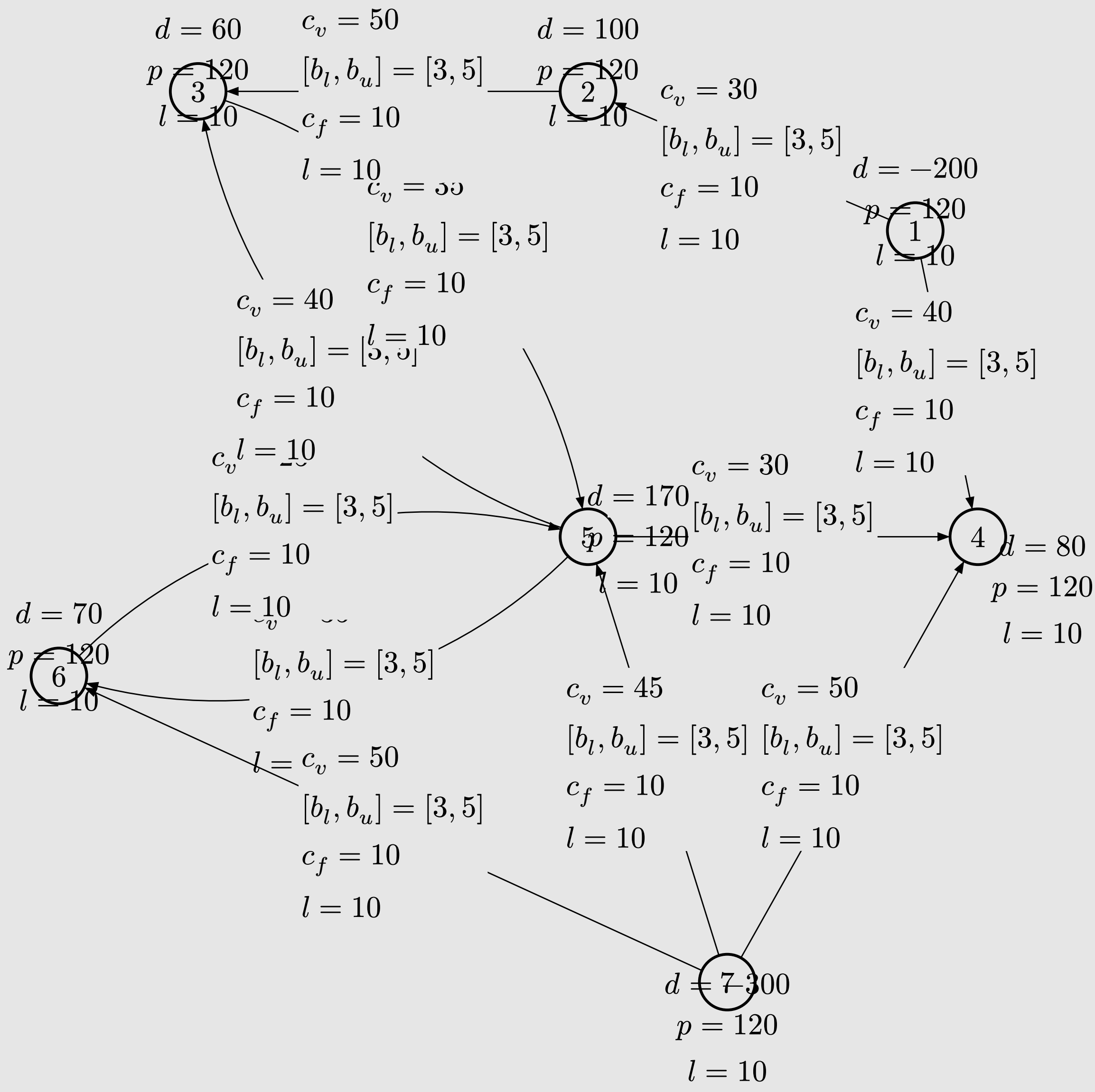
$x_{35} + x_{65} + x_{75} - x_{53} - x_{54} - x_{56} \geq 170$ (Node 5)

$$x_{56} + x_{76} - x_{65} \geq 70 \qquad \text{(Node 6)}$$

$$x_{76} + x_{75} + x_{74} \leq 300 \qquad \text{(Node 7)}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in E$$

Example



Minimize:

$$\sum_{ij}\left(c_{ij}^{f}\cdot y_{ij}\right)+\sum_{ij}\left(c_{ij}^{v}\cdot x_{ij}\right)+\sum_{i}\left(c_{i}^{f}\cdot y_{i}\right)+\sum_{i}(p_i\cdot s_i)+$$

Where:

- $\sum_{ij}\left(c_{ij}^{f}\cdot y_{ij}\right)$: Edge fixed cost contribution
- $\sum_{ij}\left(c_{ij}^{v}\cdot x_{ij}\right)$: Variable cost contribution
- $\sum_{i}\left(c_{i}^{f}\cdot y_{i}\right)$: Node fixed cost contribution
- $\sum_{i}(p_i\cdot s_i)$: Penalty contribution
- $\sum_{ij}\left(l_{ij}\cdot x_{ij}\right)$: Edge lead time weighted by flow
- $\sum_{i} l_i\cdot\sum_{j} x_{ij}$: Node service time weighted by flow

s.t.

$\sum_j x_{ji}-\sum_j x_{ij} = \text{or} \leq \text{or} \geq d_i$	Flow Conservation
$b_{ij}^{l}\leq x_{ij}\leq b_{ij}^{u}$	Lower & Upper Flow Bound
$x_{ij}\leq M\cdot y_{ij}$	Fixed Cost Route
$\sum_i\left(x_{ji}+x_{ij}\right)\leq M\cdot y_i$	Fixed Cost Node
$\sum_i x_{ij}+s_j\geq d_j$	Unmet Demand Penalty
$x_{ij}\geq 0 \quad \forall(i,j)\in E$	Non Negative Flow

Node Properties:

- **Node Type**: Source, sink, or intermediary.
- **Supply (Source)**: Flow capacity of source nodes.
- **Holding Cost (Intermediary)**: Inventory cost for stored goods.
- **Service Time**: Processing time at the node.
- **Demand**: Required flow at sink nodes.
- **Storage Capacity (Intermediary)**: Maximum amount of goods that can be held at a node
- **Penalty for Unfulfilled Demand (Sink)**: Cost for unmet demand.
- **Disruption Risk (All Nodes)**: Probability of a node being unavailable due to unforeseen circumstances

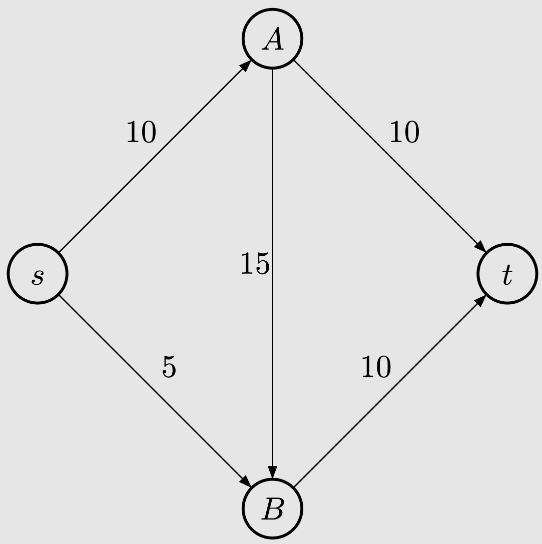
Edge Properties:

- **Edge Type**: Transport mode (air, water, road, rail).
- **Fixed Cost**: Cost incurred for using the edge, regardless of flow.
- **Reliability**: Probability of edge availability.
- **Flow Bounds**: Minimum and maximum allowable flow.
- **Unit Cost**: Cost per unit of flow.
- **Lead Time**: Time it takes for flow to travel along the edge.
- **Environmental Impact**: Account for the carbon footprint of using certain transport modes.

27.33. Ford-Fulkerson

Find augmenting paths in the network and increase the flow until no more augmenting paths can be found

Example



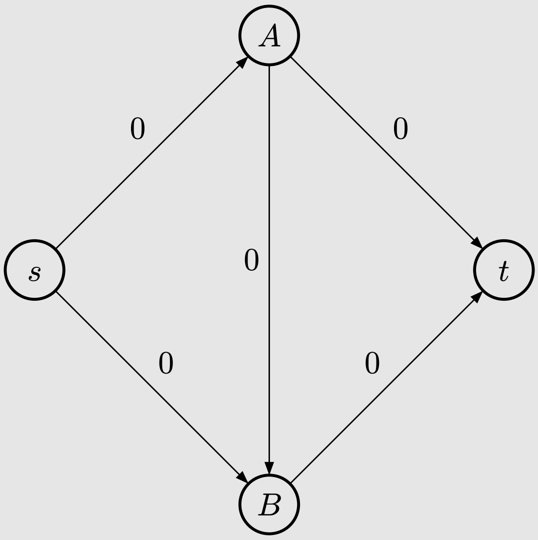
- s: Source
- A & B: Intermediate nodes
- t: Sink

Capacities:

- $s \to t$: 10
- $s \to B$: 5
- $A \to B$: 15
- $A \to t$: 10
- $B \to t$: 10

Step 1: Initialize flow to 0

All flows through the edges are initially set to 0.



Step 2: Find an augmenting path

Find an augmenting path using Depth-First Search (DFS). Start from the source s

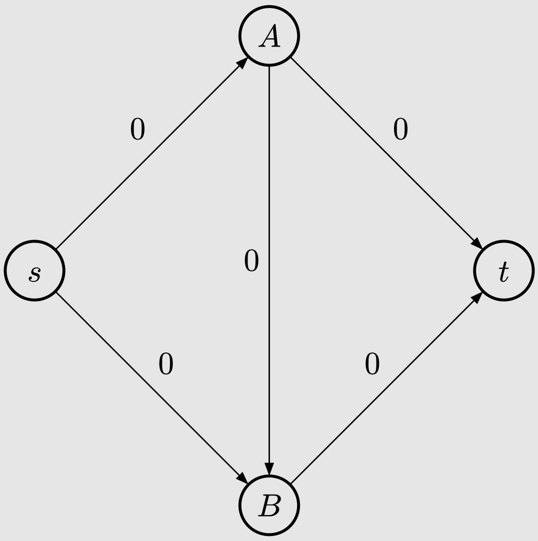
$$s \to A \to t$$

The minimum capacity along this path is 10 (bottleneck on edge $A \to t$)

We can push a flow of 10 units along this path.

Step 3: Update the residual graph

- $s \to A$: Capacity becomes $10 - 10 = 0$ (no residual capacity)
- $A \to t$: Capacity becomes $10 - 10 = 0$ (no residual capacity)



Step 4: Find another augmenting path

Find another augmenting path.

We cannot use $s \to A$ or $A \to t$ because their capacities are 0.

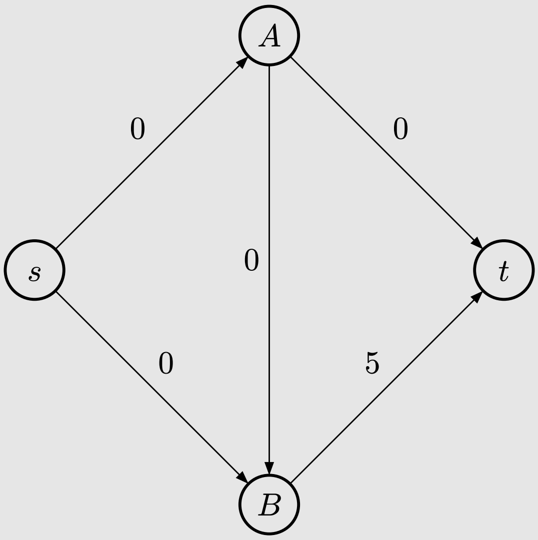
$$s \to B \to t$$

The minimum capacity along this path is 5 (bottleneck on edge $s \to B$)

We can push a flow of 5 units along this path

Step 5: Update the residual graph

- $s \to B$: Capacity becomes $5 - 5 = 0$
- $B \to t$: Capacity becomes $10 - 5 = 5$



Step 6: Termination

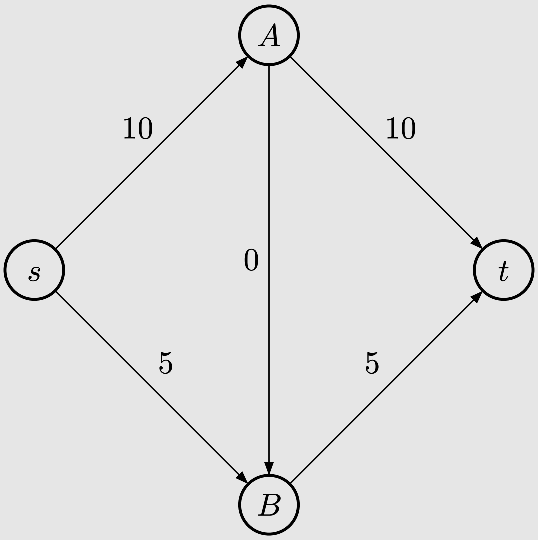
No more augmenting paths from s to t can be found, as all edges from s are fully saturated

Maximum flow:

- $s \to A \to t$: 10 units
- $s \to B \to t$: 5 units

Thus, the maximum flow from source s to sink t is 15 units

Step 7: Final Flow Distribution



27.34. Lagrange Relaxation

2 types of points:

- **Interior** point
- **Boundary** point

27.34.1. Single Variate

For constrained single variate:

- **Check convexity**: Verify whether the objective function is convex by inspecting the second derivative. Convexity ensures that any stationary point is a global minimum
- **Find unconstrained minimizer**: Solve the first-order condition (FOC), $f'(x) = 0$, to identify the unconstrained optimal solution x^*.
- Apply feasibility:
  - If x^* lies within the feasible region, it is the constrained optimum.
  - If x^* is infeasible, the constrained optimum is attained at the feasible boundary point closest to x^* (since convexity guarantees the function grows monotonically away from the minimizer).

Example

$$\min_{x \geq 0} f(x) = x^2 + 2x - 3$$

Derivatives:

$$f'(x) = 2x + 2$$
$$f''(x) = 2$$

Thus, f is convex.

FOC solution:

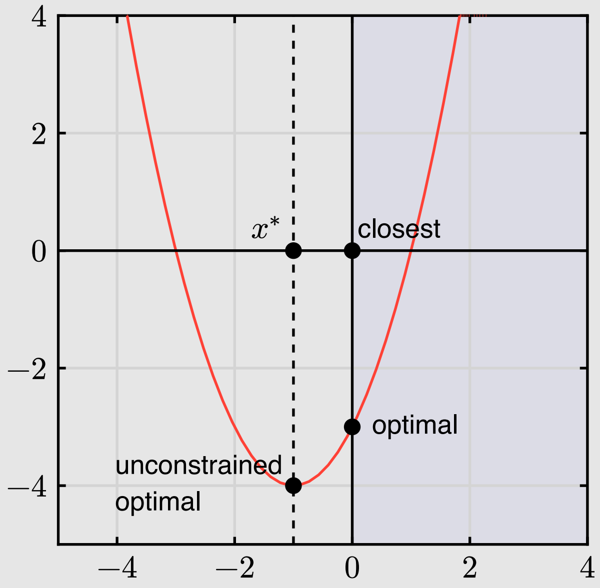
$$f'(x) = 0 \quad \Rightarrow \quad x^* = -1$$

This is the unconstrained minimizer, but it violates $x \geq 0$:

- Since $x^* = -1$ lies outside the feasible set, the closest feasible point is the boundary $x = 0$.

Evaluation:

$$f(0) = -3$$



So the constrained optimum is at $x = 0$ with objective value -3.

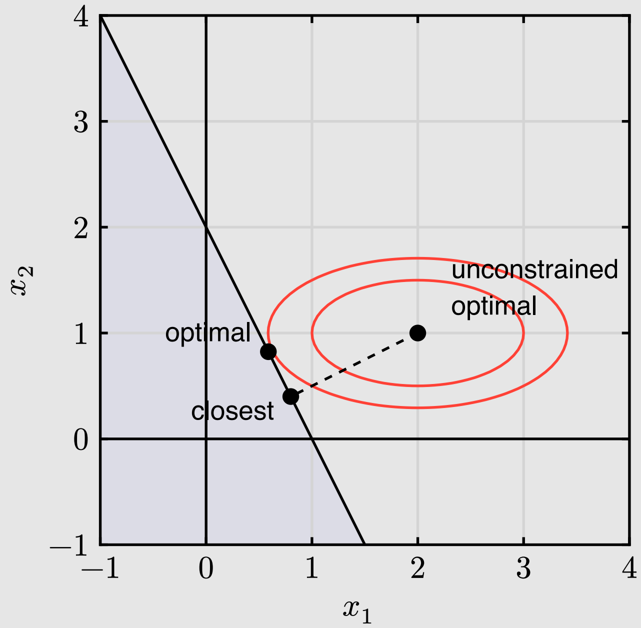
27.34.2. Multi Variate

Example

$$\min_{x \in \mathbb{R}^2} \quad f(x) = (x_1 - 2)^2 + 4(x_2 - 1)^2$$
$$s.t. \quad x_1 + 2x_2 \le 2$$

For this CP, the FOC solution $x = (2, 1)$ is infeasible

The closest feasible solution is **not** optimal



$$z^* = \max_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad g_{i(x)} \leq b_i \quad \forall i = 1, \ldots, m$$

Replace **hard** constraints with **soft** constraints

$$z^L(\lambda) = \max_{x \in \mathbb{R}^n} \quad f(x) + \sum_{i=1}^{m} \lambda_i \left[b_i - g_{i(x)}\right]$$
$$\lambda_i \geq 0$$

Lagrangian

$$\mathcal{L}(x \mid \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \left[b_i - g_{i(x)}\right]$$

λ_i is the Lagrangian multiplier

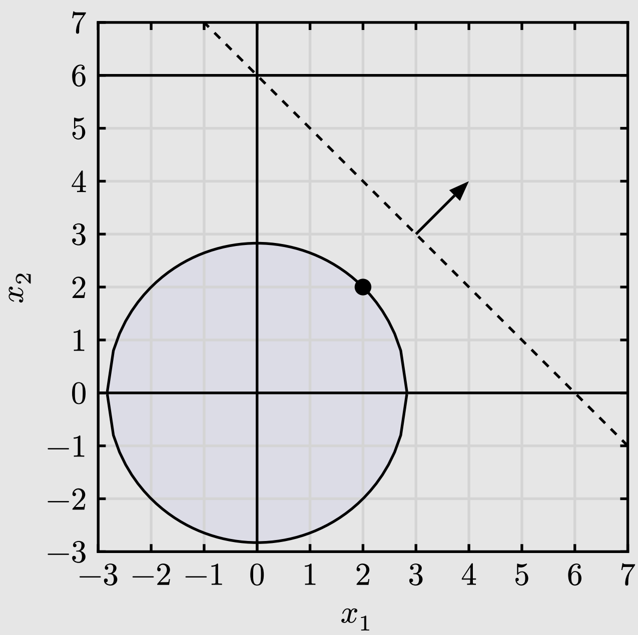
For a minization problem

$$z^* = \min_{x \in \mathbb{R}^n} \quad \left\{ f(x) \mid g_{i(x)} \leq b_i \quad \forall i = 1, \ldots, m \right\}$$

$$z^L(\lambda) = \max_{x \in \mathbb{R}^n} \quad f(x) + \sum_{i=1}^{m} \lambda_i \left[b_i - g_{i(x)}\right]$$
$$\lambda_i \leq 0$$

Example

$$z^* = \max \quad x_1 + x_2$$
$$s.t. \quad x_1^2 + x_2^2 \leq 8$$
$$x_2 \leq 6$$



Original NLP

$$z^* = \max_{x \in \mathbb{R}^2} \{x_1 + x_2 \mid x_1^2 + x_2^2 \leq 8, x_2 \leq 6\}$$

Given the Lagrangian multipliers $\lambda = \{\lambda_1, \lambda_2\} \geq 0$, the Lagrangian is:

$$\mathcal{L}(x \mid \lambda) = \underbrace{x_1 + x_2}_{f} + \lambda_1 \underbrace{(8 - x_1^2 - x_2^2)}_{b_1 - g_1} + \lambda_2 \underbrace{(6 - x_2)}_{b_2 - g_2}$$

We may then solve

$$z^L(\lambda) = \max_{x \in \mathbb{R}^2} \mathcal{L}(x \mid \lambda)$$

given any $\lambda \geq 0$.

E.g.:

-

27.34.3. Bounds

Lagrance relaxation provides bounds for the original NLP

Weak Duality of Lagrangian Relaxation

$$z^L(\lambda) \geq z^* \quad \forall \lambda \geq 0$$

Proof

$$\begin{aligned} z^* &= \max_{x \in \mathbb{R}^n} = \{f(x) \mid g_i(x) \leq b_i \quad \forall i = 1, ..., m\} \\ &\leq \max_{x \in \mathbb{R}^n} = \left\{ f(x) + \sum_{i=1}^m \lambda_i [b_i - g_i(x)] \mid g_i(x) \leq b_i \quad \forall i = 1, ..., m \right\} \\ &\leq \max_{x \in \mathbb{R}^n} = \left\{ f(x) + \sum_{i=1}^m \lambda_i [b_i - g_i(x)] \right\} = z^L(\lambda) \end{aligned}$$

Given that $x^L(\lambda) \geq z^*$ for all $\lambda \geq 0$, the **Lagrange dual program** is defined as:

$$\min_{\lambda \geq 0} z^L(\lambda)$$

The lagrange multipliers are the **dual variables** in NLP

27.35. KKT Conditions

Conditions that a solution must satisfy in order to be optimal for a nonlinear optimization problem

Conditions are necessary for optimality, and sufficient if:

- f is convex
- g_i are convex
- h_j linear

27.35.1. Setup

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & g_i(x) \le 0 \quad \forall i = 1, ..., m \\ & h_j(x) = 0 \quad \forall j = 1, ..., p \end{aligned}$$

- $f(x)$: objective function
- $g_i(x)$: inequality constraints
- $h_j(x)$: equality constraints

27.35.2. KKT Multipliers

We introduce:

- $\lambda_i \ge 0$: for each inequality constraint (Lagrange multipliers)
- $\mu_i \in \mathbb{R}$: for each equality constraint

The Lagrangian is:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x)$$

For n variables and m constrains:

- n primal variables (x) and m dual variables (λ)
- n equalities for dual feasibility
- m equalities for complementary slackness

27.35.3. KKT Conditions

At local optimum x^* there exists multipliers (λ^*, μ^*) such that:

1. **Stationarity**

Minimization:

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0$$

Stationarity is like a generalized version of “set the derivative to zero”, but accounting for the constraints

2. **Primal feasibility**

$$\begin{aligned} g_i(x^*) \le 0 \\ h_j(x^*) = 0 \end{aligned}$$

The solution must satisfy all the original constraints — it must lie inside the feasible region

3. **Dual Feasibility**

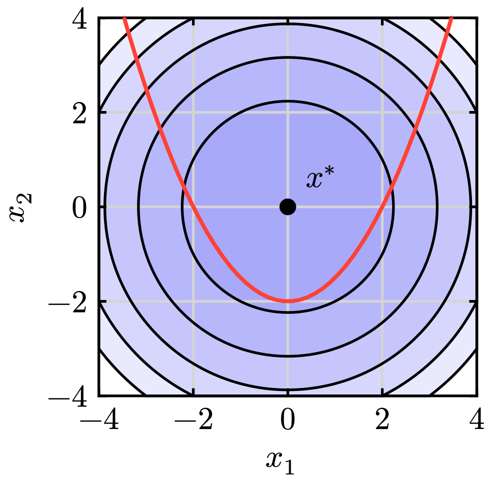
$$\lambda_i \ge 0 \quad \forall i$$

4. **Complementary Slackness**

$$\lambda_i g_i(x^*) = 0 \quad \forall i$$

Inactive (non-binding) constraint

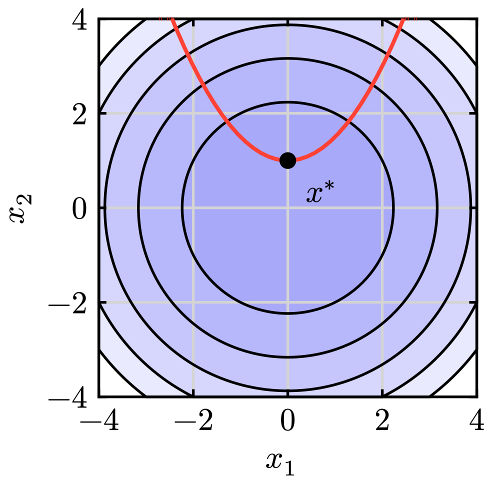
$$g_i(x^*) < 0 \quad \Rightarrow \quad \lambda_i^* = 0$$



- The constraint does not affect the optimum
- The feasible region is “loose” at the optimum — the optimum lies strictly inside it
- Economically, the shadow price is zero: relaxing the constraint wouldn’t change the objective

Active (binding) constraint

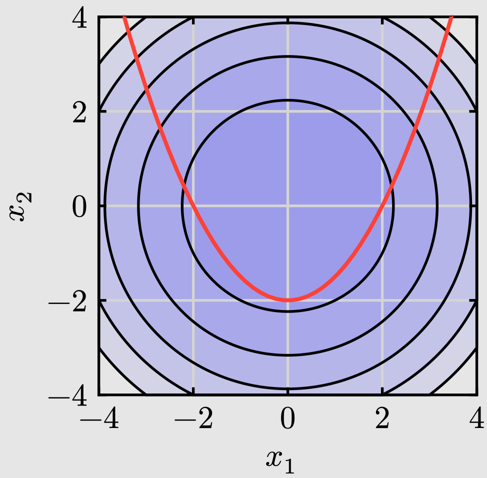
$$g_i(x^*) = 0 \quad \Rightarrow \quad \lambda_i^* \geq 0$$



- The constraint is tight at the optimum — it holds with equality
- λ_i^* is the shadow price:
    - It measures how much the objective value would improve per unit relaxation of the constraint
        - The objective would decrease by λ_i^* per unit of relaxation (if convex)

Example

$$\min \quad x_1^2 + x_2^2$$
$$s.t. \quad \frac{x_1^2}{2} - x_2 \leq 2$$



Step 1. Standardize constraints

Constraints already in the form $g(x) \leq 0$ for minimization

Sign convention $\lambda \geq 0$

Step 2. Formulate Lagrangian

$$\mathcal{L}(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda\left(\frac{x_1^2}{2} - x_2 - 2\right)$$

Step 3. Stationarity

$$\nabla_x \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x_1} \\ \\ \frac{\partial \mathcal{L}}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 + \lambda x_1 \\ \\ 2x_2 - \lambda \end{bmatrix} = 0$$

Solve for x_1:

$$\begin{aligned} 2x_1 + \lambda x_1 &= 0 \\ x_1(2 + \lambda) &= 0 \\ x_1(2 + \lambda) &= \frac{0}{2 + \lambda} \\ x_1 &= 0 \end{aligned}$$

Solve for x_2

$$\begin{aligned} 2x_2 - \lambda &= 0 \\ x_2 &= \frac{\lambda}{2} \end{aligned}$$

Candidate point:

$$(x_1, x_2) = \left(0, \frac{\lambda}{2}\right)$$

Step 4. Primal Feasibility

Constraint:

$$g(x_1, x_2) = \frac{x_1^2}{2} - x_2 - 2 \leq 0$$

Substitute $x_1 = 0$ and $x_2 = \frac{\lambda}{2}$

$$\begin{aligned} \frac{0^2}{2} - \frac{\lambda}{2} - 2 &\leq 0 \\ -\frac{\lambda}{2} - 2 &\leq 0 \\ -\frac{\lambda}{2} &\leq 2 \\ \lambda &\geq -4 \end{aligned}$$

Step 5. Dual Feasibility

$$\lambda \geq 0$$

Step 6. Complementary Slackness

Constraint:

$$g(x_1, x_2) = \frac{x_1^2}{2} - x_2 - 2 \leq 0$$

Condition:

$$\lambda \cdot g(x^*) = 0$$

Candidate point:

$$(x_1, x_2) = \left(0, \frac{\lambda}{2}\right)$$

Evaluate $g(x^*)$:

$$\lambda \cdot g(x^*) = \lambda\left(\frac{x_1^2}{2} - x_2 - 2\right) = 0$$

Substitute $x_1 = 0$ and $x_2 = \frac{\lambda}{2}$

$$\lambda\left(\frac{0^2}{2} - \frac{\lambda}{2} - 2\right) = 0$$

$$\lambda\left(-\frac{\lambda}{2} - 2\right) = 0$$

Solve:

$$\lambda\left(-\frac{\lambda}{2}-2\right)=0 \quad \Rightarrow \quad \lambda=0 \text{ or } \lambda=-4$$

But because of Dual Feasibility $\lambda \geq 0$, so $\lambda = 0$

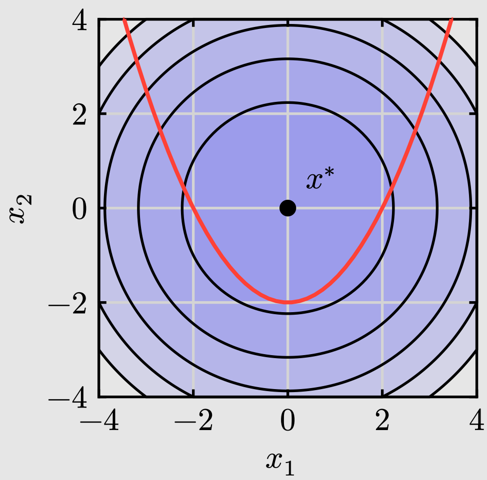
Constraint is non-binding at optimum:

- $\lambda = 0$
- $g(x^*) < 0$

$$\begin{aligned} g(x^*) = \frac{x_1^2}{2} - x_2 - 2 &\leq 0 \\ \frac{0^2}{2} - \frac{\lambda}{2} - 2 &\leq 0 \\ \frac{0^2}{2} - \frac{0}{2} - 2 &\leq 0 \\ -2 &\leq 0 \end{aligned}$$

Step 7. Solve for Optimum

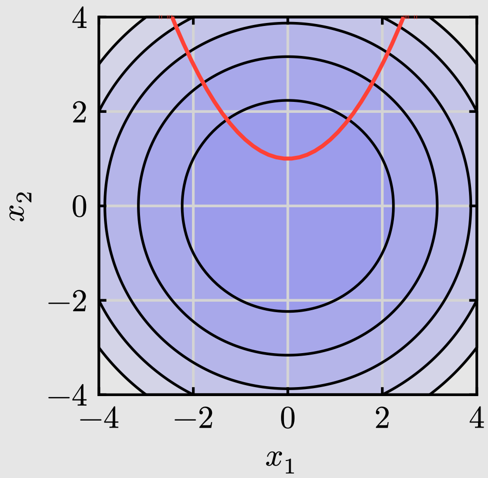
$$\begin{aligned} (x_1, x_2) &= \left(0, \frac{\lambda}{2}\right) \\ &= \left(0, \frac{0}{2}\right) \\ &= (0, 0) \end{aligned}$$



Step 8. Check Convexity

Example

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ s.t. \quad & \frac{x_1^2}{2} - x_2 \leq -1 \end{aligned}$$



Step 1. Standardize constraints

Constraints already in the form $g(x) \le 0$ for minimization

Sign convention $\lambda \ge 0$

Step 2. Formulate Lagrangian

$$\mathcal{L}(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda\left(\frac{x_1^2}{2} - x_2 + 1\right)$$

Step 3. Stationarity

$$\nabla_x \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x_1} \\ \frac{\partial \mathcal{L}}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 + \lambda x_1 \\ 2x_2 - \lambda \end{bmatrix}$$

Solve for x_1:

$$\begin{aligned} 2x_1 + \lambda x_1 &= 0 \\ x_1(2+\lambda) &= 0 \\ x_1 &= \frac{0}{2+\lambda} \\ x_1 &= 0 \end{aligned}$$

Solve for x_2:

$$\begin{aligned} 2x_2 - \lambda &= 0 \\ 2x_2 &= \lambda \\ x_2 &= \frac{\lambda}{2} \end{aligned}$$

Candidate point:

$$(x_1, x_2) = \left(0, \frac{\lambda}{2}\right)$$

Step 4. Primal Feasibility

Constraint:

$$g(x_1, x_2) = \frac{x_1^2}{2} - x_2 + 1 \le 0$$

Substitute $x_1 = 0$ and $x_2 = \frac{\lambda}{2}$:

$$\frac{x_1^2}{2} - x_2 + 1 \le 0$$
$$\frac{0^2}{2} - \frac{\lambda}{2} + 1 \le 0$$
$$-\frac{\lambda}{2} + 1 \le 0$$
$$-\frac{\lambda}{2} \le -1$$
$$\lambda \ge 2$$

Step 5. Dual Feasibility

$$\lambda \ge 0$$

Step 6. Complementary Slackness

Constraint:

$$g(x_1, x_2) = \frac{x_1^2}{2} - x_2 + 1 \le 0$$

Condition:

$$\lambda \cdot g(x^*) = 0$$

Candidate point:

$$(x_1, x_2) = \left(0, \frac{\lambda}{2}\right)$$

Evaluate $g(x^*)$:

$$\lambda \cdot g(x^*) = \lambda\left(\frac{x_1^2}{2} - x_2 + 1\right) = 0$$

Substitute $x_1 = 0$ and $x_2 = \frac{\lambda}{2}$

$$\lambda\left(\frac{0^2}{2} - \frac{\lambda}{2} + 1\right) = 0$$
$$\lambda\left(-\frac{\lambda}{2} + 1\right) = 0$$

Solve:

$$\lambda\left(-\frac{\lambda}{2} + 1\right) = 0 \quad \Rightarrow \quad \lambda = 0 \text{ or } \lambda = 2$$

But because of Primal Feasibility, if $\lambda = 0$, $(x_1, x_2) = (0, 0)$ and $g(0, 0) = 1 \nleq 0$, thus $\lambda = 2$

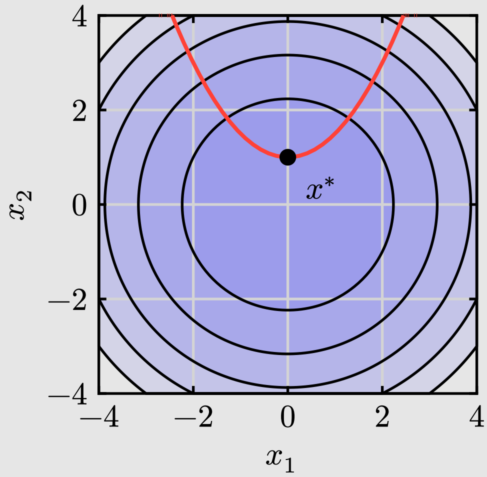
Constraint is binding at optimum:

- $\lambda = 2$
- $g(x^*) = 0$

$$\begin{aligned} g(x^*) = \frac{x_1^2}{2} - x_2 + 1 &= 0 \\ \frac{0^2}{2} - \frac{\lambda}{2} + 1 &= 0 \\ \frac{0^2}{2} - \frac{2}{2} + 1 &= 0 \\ 0 &= 0 \end{aligned}$$

Step 7. Solve for Optimum

$$\begin{aligned} (x_1, x_2) &= \left(0, \frac{\lambda}{2}\right) \\ &= \left(0, \frac{2}{2}\right) \\ &= (0, 1) \end{aligned}$$



Step 8. Check Convexity

27.35.4. Calculating Lagrangian Multipliers

To find the multipliers (λ^*, μ^*) and the optimal point x^*, solve the KKT system of equations:

$$\begin{cases} \nabla_x \mathcal{L}(x, \lambda, \mu) = 0 & \\ g_i(x) \leq 0 & \forall i = 1, ..., m \\ h_j(x) = 0 & \forall i = 1, ..., p \\ \lambda_i \geq 0 & \forall i = 1, ..., m \\ \lambda_i g_i(x) = 0 & \forall i = 1, ..., m \end{cases}$$

Example

Retailer Maximizing Profit under a Capacity Constraint

A retailer sells product 1 and 2 with quantities q_1 and q_2. For product i the market-clearing prices are:

$$p_i = a_i - b_i q_i \quad i = 1, 2$$

Where $a_i, b_i > 0$.

The retailer maximizes **total profit** subject to a **capacity constraint**:

$$q_1 + q_2 \leq K \quad q_1, q_2 \geq 0$$

1. Formulation

$$\begin{aligned} \max_{q_1, q_2 \geq 0} \quad & q_1 \overbrace{(a_1 - b_1 q_1)}^{p_1} + q_2 \overbrace{(a_2 - b_2 q_2)}^{p_2} \\ s.t. \quad & q_1 + q_2 \leq K \end{aligned}$$

The objective is concave because

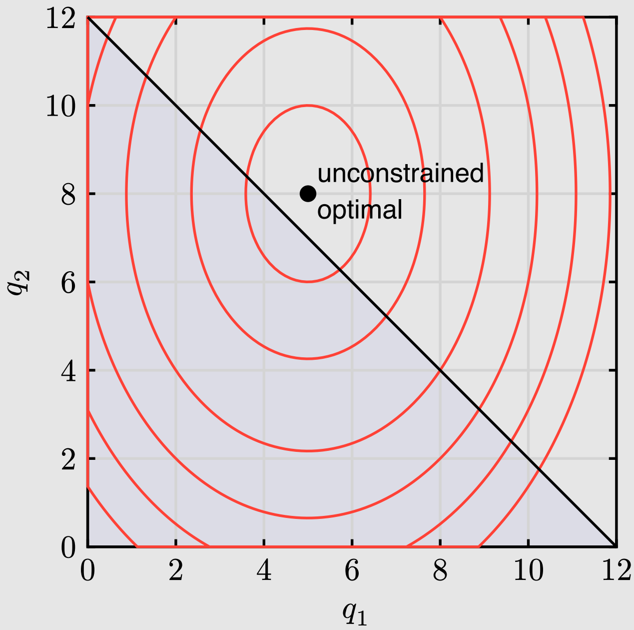
$$-\nabla^2 f(q_1, q_2) = \begin{bmatrix} 2b_1 & 0 \\ 0 & 2b_2 \end{bmatrix}$$

is positive semi-definite.

The unconstrained FOC solution is:

$$q_1 = \frac{a_1}{2b_1}, \quad q_2 = \frac{a_2}{2b_2}$$

If $q_1 + q_2 \leq K$, this solution is feasible.



2. Lagrangian

Introduce KKT multiplier $\lambda \geq 0$ for the inequality constraint:

$$\mathcal{L}(q_1, q_2, \lambda) = \overbrace{q_1(a_1 - b_1 q_1) + q_2(a_2 - b_2 q_2)}^{f} + \lambda\overbrace{(K - q_1 - q_2)}^{b-g}$$

3. KKT Conditions

1. Stationarity

$$\frac{\partial}{\partial q_1}\mathcal{L} = a_1 - 2b_1 q_1 - \lambda = 0$$
$$\frac{\partial}{\partial q_2}\mathcal{L} = a_2 - 2b_2 q_2 - \lambda = 0$$

2. Primal Feasibility

$$q_1 + q_2 \leq K \quad q_1, q_2 \geq 0$$

3. Dual Feasibility

$$\lambda \geq 0$$

4. Complementary Slackness

$$\lambda(K - q_1 - q_2) = 0$$

4. Solving KKT system

Case 1: Unconstrained ($\lambda = 0$)

$$\max_{q_1, q_2 \geq 0} \quad f(q_1, q_2) = q_1(a_1 - b_1 q_1) + q_2(a_2 - b_2 q_2)$$
$$f(q_1, q_2) = a_1 q_1 - b_1 q_1^2 + a_2 q_2 - b_2 q_2^2$$

First-Order Condition

$$\frac{\partial f}{\partial q_1} = a_1 - 2b_1 q_1$$
$$\frac{\partial f}{\partial q_2} = a_2 - 2b_2 q_2$$

Set derivatives to 0 (stationarity)

$$a_1 - 2b_1 q_1 = 0 \quad \Rightarrow \quad q_1^* = \frac{a_1}{2b_1}$$
$$a_2 - 2b_2 q_2 = 0 \quad \Rightarrow \quad q_2^* = \frac{a_2}{2b_2}$$

Unconstrained Solution:

$$q_1^* = \frac{a_1}{2b_1}$$
$$q_2^* = \frac{a_2}{2b_2}$$

The second derivative check confirms it's a maximum because:

$$\frac{\partial^2 f}{\partial q_1^2} = -2b_1 < 0$$
$$\frac{\partial^2 f}{\partial q_2^2} = -2b_2 < 0$$

Feasible if

$$\frac{a_1}{2b_1} + \frac{a_2}{2b_2} \leq K$$

Case 2: Binding Constraint ($\lambda > 0$)

We now include the capacity constraint:

$$q_1 + q_2 = K$$

If it binds (active) then

$$q_1 + q_2 = K$$

Lagrangian

$$\mathcal{L}(q_1, q_2, \lambda) = \overbrace{q_1(a_1 - b_1 q_1) + q_2(a_2 - b_2 q_2)}^{f(q_1, q_2)} + \lambda \overbrace{(K - q_1 - q_2)}^{b - g(q_1, q_2)}$$

First-Order Condition

$$\frac{\partial \mathcal{L}}{\partial q_1} = a_1 - 2b_1 q_1 - \lambda$$
$$\frac{\partial \mathcal{L}}{\partial q_2} = a_2 - 2b_2 q_2 - \lambda$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = K - q_1 - q_2$$

Set derivatives to 0 (stationarity)

$$a_1 - 2b_1 q_1 - \lambda = 0$$
$$a_2 - 2b_2 q_2 - \lambda = 0$$
$$K - q_1 - q_2 = 0$$

When we take the derivative of the Lagrangian with respect to the multiplier λ we recover the constraint itself.

Solve system of equations

$$\begin{cases} a_1 - 2b_1 q_1 - \lambda = 0 \\ a_2 - 2b_2 q_2 - \lambda = 0 \\ q_1 + q_2 = K \end{cases}$$

Solution:

$$q_1 = \frac{2b_2 K + a_1 - a_2}{2(b_1 + b_2)} \qquad q_2 = \frac{2b_1 K + a_2 - a_1}{2(b_1 + b_2)}$$

Feasible if

$$\frac{a_1}{2b_1} + \frac{a_2}{2b_2} > K$$

5. Optimal Solution

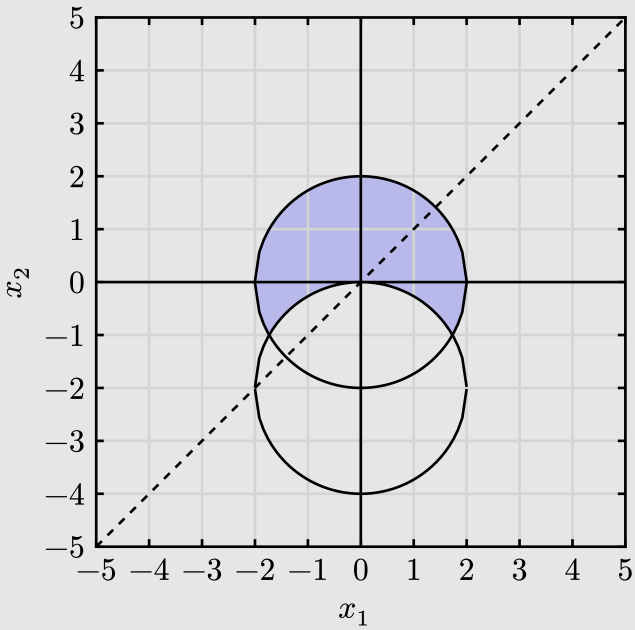
$$(q_1^*, q_2^*) = \begin{cases} \left(\frac{a_1}{2b_1}, \frac{a_2}{2b_2}\right) & \text{if } \frac{a_1}{2b_1} + \frac{a_2}{2b_2} \leq K \\ \left(\frac{2b_2 K + a_1 - a_2}{2(b_1+b_2)}, \frac{2b_1 K + a_2 - a_1}{2(b_1+b_2)}\right) & \text{otherwise} \end{cases}$$

Intuition:
- If capacity K is large enough, the retailer produces unconstrained quantities.
- If K is tight, the total production hits the limit, and the optimal quantities are shared according to the relative parameters a_i, b_i

Example

$$\begin{aligned} \max \quad & f(x) = x_1 - x_2 \\ s.t. \quad & g_1(x) = x_1^2 + x_2^2 \leq 4 \\ & g_2(x) = -x_1^2 - (x_2 + 2)^2 \leq -4 \end{aligned}$$

This NLP is **nonconvex**



1️⃣ Lagrangian

Introduce multipliers $\lambda_1, \lambda_2 \geq 0$ for the inequalities. The Lagrangian is

$$\mathcal{L}(x_1, x_2, \lambda) = \overbrace{x_1 - x_2}^{f} + \lambda_1 \overbrace{(4 - x_1^2 - x_2^2)}^{g_1} + \lambda_2 \overbrace{\left(-4 + x_1^2 + (x_2+2)^2\right)}^{g_2}$$

The solution $\bar{x}$ is a local maximum only if there exists λ such that

2️⃣ KKT Conditions

1. Primal feasibility

$$x_1^2 + x_2^2 \leq 4 \qquad \text{(PF-1)}$$
$$-x_1^2 - (x_2+2)^2 \leq -4 \qquad \text{(PF-2)}$$

2. Dual Feasibility

$$\lambda_1 \geq 0 \qquad \text{(DFS-1)}$$
$$\lambda_2 \geq 0 \qquad \text{(DFS-2)}$$

3. Stationarity

$$\frac{\partial \mathcal{L}}{\partial x_1} = 1 - 2(\lambda_1 - \lambda_2)x_1 = 0 \qquad \text{(DFF-1)}$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = -1 - 2(\lambda_1 - \lambda_2)x_2 + 4\lambda_2 = 0 \qquad \text{(DFF-2)}$$

4. Complimentary Slackness

$$\lambda_1(4 - x_1^2 - x_2^2) = 0 \qquad \text{(CS-1)}$$
$$\lambda_2\left(-4 + x_1^2 + (x_2+2)^2\right) = 0 \qquad \text{(CS-2)}$$

3️⃣ Examine all 4 cases for (λ_1, λ_2)

Case 1. $(\lambda_1 > 0, \lambda_2 > 0)$

Step 1: Complementary Slackness

Since both multipliers are positive, the corresponding constraints must be active:

$$x_1^2 + x_2^2 = 4 \qquad \text{(CS-1)}$$
$$x_1^2 + (x_2+2)^2 = 4 \qquad \text{(CS-2)}$$

Solve this system $\rightarrow$ two candidate points:

$$(x_1, x_2) = \left(\sqrt{3}, -1\right) \quad \text{and} \quad \left(-\sqrt{3}, -1\right)$$

Step 2: Primal Feasibility

1. For $\left(\sqrt{3}, -1\right)$

$$x_1^2 + x_2^2 = \sqrt{3}^2 + (-1)^2 \le 4 \qquad \text{(PF-1)}$$
$$-x_1^2 - (x_2+2)^2 = -\left(\sqrt{3}\right)^2 - (-1+2)^2 \le -4 \qquad \text{(PF-2)}$$

2. For $\left(-\sqrt{3}, -1\right)$

$$x_1^2 + x_2^2 = \left(-\sqrt{3}\right)^2 + (-1)^2 \le 4 \qquad \text{(PF-1)}$$
$$-x_1^2 - (x_2+2)^2 = -\left(-\sqrt{3}\right)^2 - (-1+2)^2 \le -4 \qquad \text{(PF-2)}$$

✅ Both satisfy primal feasibility.

Step 3: Stationarity / First-order conditions (DFF)

Plug each candidate point into the gradient equations and solve the system:

$$1 - 2(\lambda_1 - \lambda_2)x_1 = 0 \qquad \text{(DFF-1)}$$
$$-1 - 2(\lambda_1 - \lambda_2)x_2 + 4\lambda_2 = 0 \qquad \text{(DFF-2)}$$

For $\left(\sqrt{3}, -1\right)$:

$$1 - 2(\lambda_1 - \lambda_2)\left(\sqrt{3}\right) = 0$$
$$-1 - 2(\lambda_1 - \lambda_2)(-1) + 4\lambda_2 = 0$$

Solving the system we get:

$$\lambda_1 = \frac{1}{4} + \frac{1}{4\sqrt{3}}$$
$$\lambda_2 = \frac{1}{4} - \frac{1}{4\sqrt{3}}$$

✅ Both $\lambda_1, \lambda_2 \ge 0 \rightarrow$ dual feasibility satisfied.

For $\left(-\sqrt{3}, -1\right)$:

$$1 - 2(\lambda_1 - \lambda_2)\left(-\sqrt{3}\right) = 0 \Longrightarrow \lambda_1 - \lambda_2 = -\frac{1}{2\sqrt{3}}$$

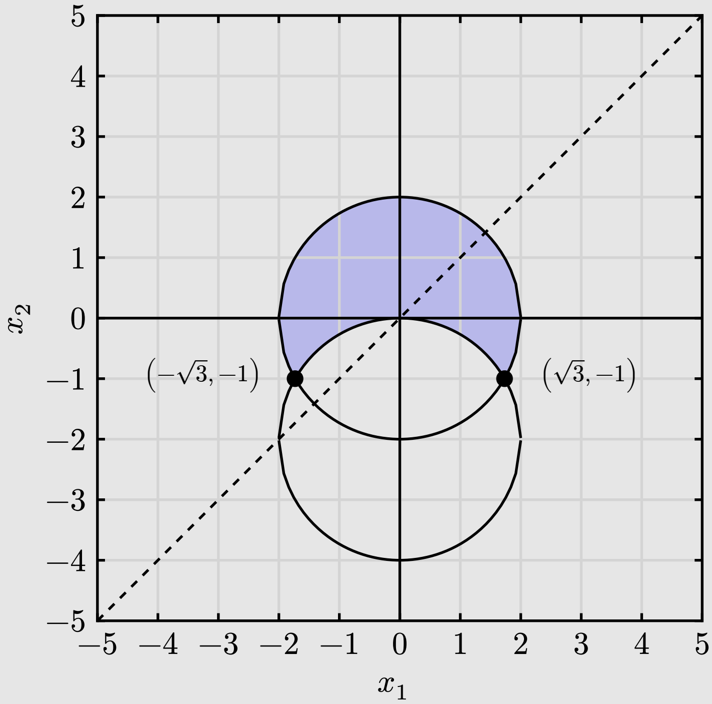
$$-1 - 2(\lambda_1 - \lambda_2)(-1) + 4\lambda_2 = 0 \Longrightarrow \lambda_2 = \frac{1}{4} + \frac{1}{4\sqrt{3}}$$

Solving the system we get:

$$\lambda_1 = \frac{1}{4} - \frac{1}{4\sqrt{3}}$$

$$\lambda_2 = \frac{1}{4} + \frac{1}{4\sqrt{3}}$$

✅ Both $\lambda_1, \lambda_2 \geq 0 \rightarrow$ dual feasibility satisfied.



✅ Both $\left(\sqrt{3}, -1\right)$ and $\left(-\sqrt{3}, -1\right)$ satisfy dual feasibility $\rightarrow$ both are KKT points.

Case 2. $(\lambda_1 > 0, \lambda_2 = 0)$

Step 1. Assumptions (Activity / Inactivity)

- Constraint 1: Active $(x_1^2 + x_2^2 = 4)$
- Constraint 2: Inactive $(x_1^2 + (x_2 + 2)^2 = 4)$

Step 2. Stationarity Condition

The Lagragian:

$$\mathcal{L}(x_1, x_2, \lambda_1, \lambda_2) = (x_1 - x_2) + \lambda_1(4 - x_1^2 - x_2^2) + \lambda_2\left(-4 + x_1^2 + (x_2 + 2)^2\right)$$

Compute the gradients with respect to x_1 and x_2:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x_1} = 1 - 2\lambda_1 x_1 + 2\lambda_2 x_1 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_2} = -1 - 2\lambda_1 x_2 + 2\lambda_2 (x_2 + 2) = 0 \end{cases}$$

Since $\lambda_2 = 0$, the equations simplify to:

$$\begin{cases} 1 - 2\lambda_1 x_1 = 0 \\ -1 - 2\lambda_1 x_2 = 0 \end{cases}$$

Simplify:

$$\begin{cases} x_1 = \frac{1}{2\lambda_1} \\ x_2 = -\frac{1}{2\lambda_1} \end{cases} \Rightarrow \quad x_1 = -x_2$$

Step 3. Substitute into Active Constraint

Using $x_1^2 + x_2^2 = 4$ and $x_2 = -x_1$:

$$x_1^2 + (-x_1)^2 = 2x_1^2 = 4 \quad \Rightarrow \quad x_1^2 = 2 \quad \Rightarrow \quad x_1 = \pm\sqrt{2}, \quad x_2 = \mp\sqrt{2}$$

Two cadidate solutions x_1 and x_2:

$$\left(-\sqrt{2}, \sqrt{2}\right) \quad \text{and} \quad \left(\sqrt{2}, -\sqrt{2}\right)$$

Step 4. Solve for λ_1 and enforce $\lambda_1 > 0$

$$1 + 2\lambda_1 x_1 = 0 \quad \Rightarrow \lambda_1 = -\frac{1}{2x_1}$$

Substitute (x_1, x_2) into the stationarity condition:

$$1 + 2\lambda_1 x_1$$

- For $(x_1, x_2) = \left(\sqrt{2}, -\sqrt{2}\right)$:

$$\lambda_1 = \frac{1}{2\sqrt{2}} > 0 \rightarrow ✅ \ \textbf{Valid} \ (\lambda_1 \geq 0)$$

- For $(x_1, x_2) = \left(-\sqrt{2}, \sqrt{2}\right)$:

$$\lambda_1 = \frac{1}{2\left(-\sqrt{2}\right)} > 0 \rightarrow ❌ \ \textbf{Reject} \ (\text{violates } \lambda_1 \geq 0)$$

So, the surviving candidate is:

$$(x_1, x_2) = \left(\sqrt{2}, -\sqrt{2}\right), \qquad \lambda_1 = \frac{1}{2\sqrt{2}}, \lambda_2 = 0$$

Step 5. Check the (inactive) constraint 2 for primal feasibility

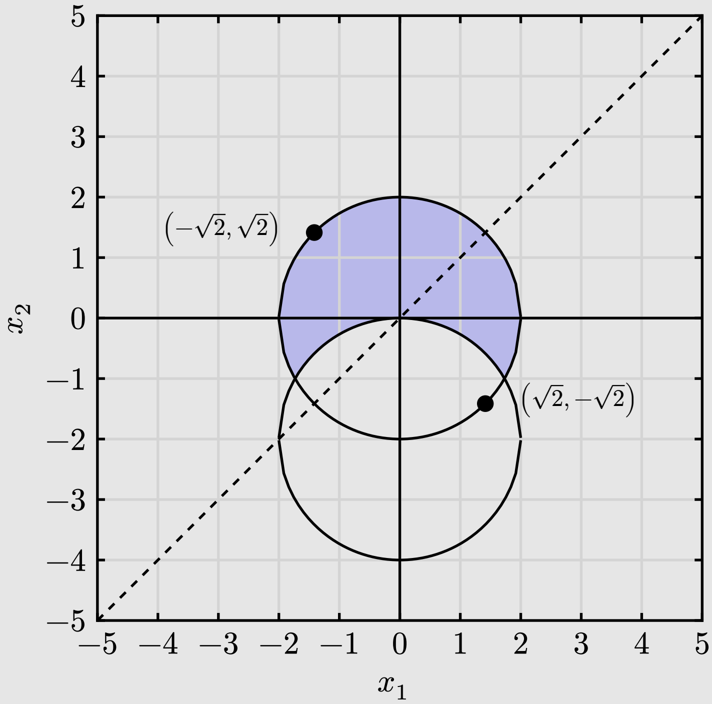
Constraint 2:

$$g_2(x) = -x_1^2 - (x_2 + 2)^2 + 4 \leq 0$$

Plug $x_1 = \sqrt{2}$ and $x_2 = -\sqrt{2}$:

$$g_2(x) = -\left(\sqrt{2}\right)^2 - \left(-\sqrt{2}+2\right)^2 + 4 > 0$$

❌ This violates the inequality $g_2(x) \leq 0$



❌ $(x_1, x_2) = \left(\sqrt{2}, -\sqrt{2}\right)$ is rejected because it violates the inequality $g_2(x) \leq 0$

❌ $(x_1, x_2) = \left(-\sqrt{2}, \sqrt{2}\right)$ is rejected because it gives $\lambda < 0$

Case 3. $(\lambda_1 = 0, \lambda_2 > 0)$

Step 1. Assumptions (Activity / Inactivity)

- Constraint 1: Inactive $(x_1^2 + x_2^2 = 4)$
- Constraint 2: Active $(x_1^2 + (x_2+2)^2 = 4)$

Step 2. Stationarity Condition

The Lagragian:

$$\mathcal{L}(x_1, x_2, \lambda_1, \lambda_2) = (x_1 - x_2) + \lambda_1(4 - x_1^2 - x_2^2) + \lambda_2\left(-4 + x_1^2 + (x_2+2)^2\right)$$

Compute the gradients with respect to x_1 and x_2:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x_1} = 1 - 2\lambda_1 x_1 + 2\lambda_2 x_1 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_2} = -1 - 2\lambda_1 x_2 + 2\lambda_2 (x_2 + 2) = 0 \end{cases}$$

Since $\lambda_1 = 0$, the equations simplify to:

$$\begin{cases} 1 + 2\lambda_2 x_1 = 0 \\ -1 + 2\lambda_2(x_2 + 2) = 0 \end{cases}$$

Simplify:

$$\begin{cases} x_1 = -\frac{1}{2\lambda_2} \\ x_2 = \frac{1}{2\lambda_2} - 2 \end{cases} \quad \Rightarrow \quad x_1 = -x_2$$

Step 3. Substitute into Active Constraint

$$x_1^2 + (x_2 + 2)^2 = 4$$

Substituting x_1 and x_2:

$$\left(-\frac{1}{2\lambda_2}\right)^2 + \left(\frac{1}{2\lambda_2} - 2 + 2\right)^2 = 4 \quad \Rightarrow \quad \lambda_2 = \pm\frac{1}{2\sqrt{2}}$$

Two cadidate solutions x_1 and x_2:

$$\left(-\sqrt{2}, \sqrt{2} - 2\right) \quad \text{and} \quad \left(\sqrt{2}, -\sqrt{2} - 2\right)$$

Substitute (x_1, x_2) into the stationarity condition:

$$-1 + 2\lambda_2(x_2 + 2)$$

- For $(x_1, x_2) = \left(\sqrt{2}, -\sqrt{2} - 2\right)$:

$$\lambda_2 = -\frac{1}{2\sqrt{2}} < 0 \rightarrow ❌ \ \textbf{Reject} \ (\text{violates } \lambda_1 \geq 0)$$

- For $(x_1, x_2) = \left(-\sqrt{2}, \sqrt{2} - 2\right)$:

$$\lambda_2 = \frac{1}{2\sqrt{2}} > 0 \rightarrow ✅ \ \textbf{Valid} \ (\lambda_1 \geq 0)$$

So, the surviving candidate is:

$$(x_1, x_2) = \left(-\sqrt{2}, \sqrt{2} - 2\right), \qquad \lambda_1 = 0, \lambda_2 = \frac{1}{2\sqrt{2}}$$

Step 5. Check the (inactive) constraint 1 for primal feasibility

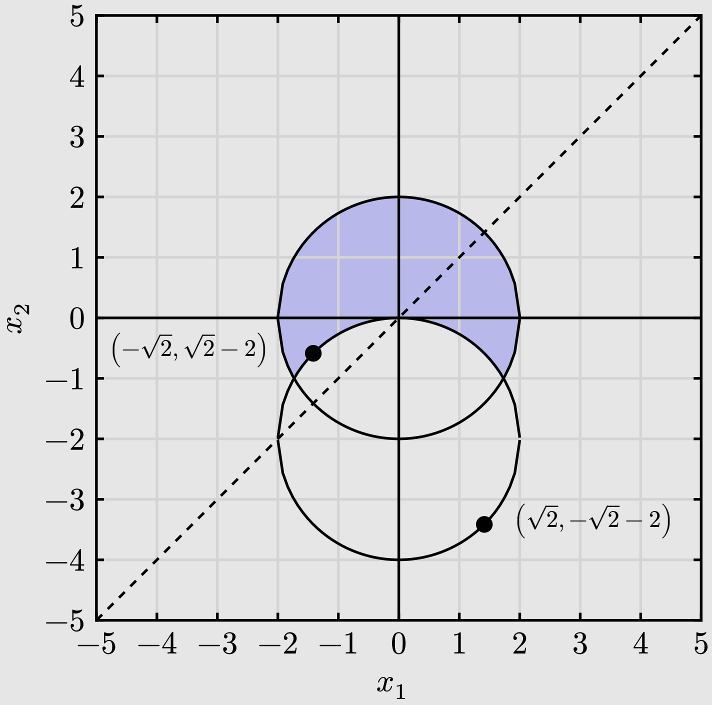
Constraint 1:

$$g_1(x) = x_1^2 + x_2^2 \leq 4$$

Plug $x_1 = -\sqrt{2}$ and $x_2 = \sqrt{2} - 2$:

$$g_1(x) = -\left(\sqrt{2}\right)^2 - \left(\sqrt{2} - 2\right)^2 + 4 \leq 0$$

✅ Satisfies primal feasibility. Constraint 1 is inactive (strict inequality), as required.



✅ $(x_1, x_2) = \left(-\sqrt{2}, \sqrt{2} - 2\right)$ is a valid KKT point under this assumption.

❌ $(x_1, x_2) = \left(\sqrt{2}, -\sqrt{2} - 2\right)$ is rejected because it gives $\lambda < 0$

Case 4. $(\lambda_1 = 0, \lambda_2 = 0)$

Step 1. Assumptions (Activity / Inactivity)

- Both multipliers are zero: $\lambda_1 = 0, \lambda_2 = 0$
- No constraints are forced to be active by complementary slackness

Step 2. Stationarity Condition

With $\lambda_1 = 0$ and $\lambda_2 = 0$, the stationarity equations siplify to:

$$\begin{cases} 1 - 2(\lambda_1 - \lambda_2)x_1 = 1 = 0 \\ -1 - 2(\lambda_1 - \lambda_2)x_2 + 4\lambda_2 = -1 = 0 \end{cases}$$

Step 3. Stationary Check

Both equations $1 = 0$ and $-1 = 0$ are impossible

No solution exists under this assumption

❌ No KKT points exist

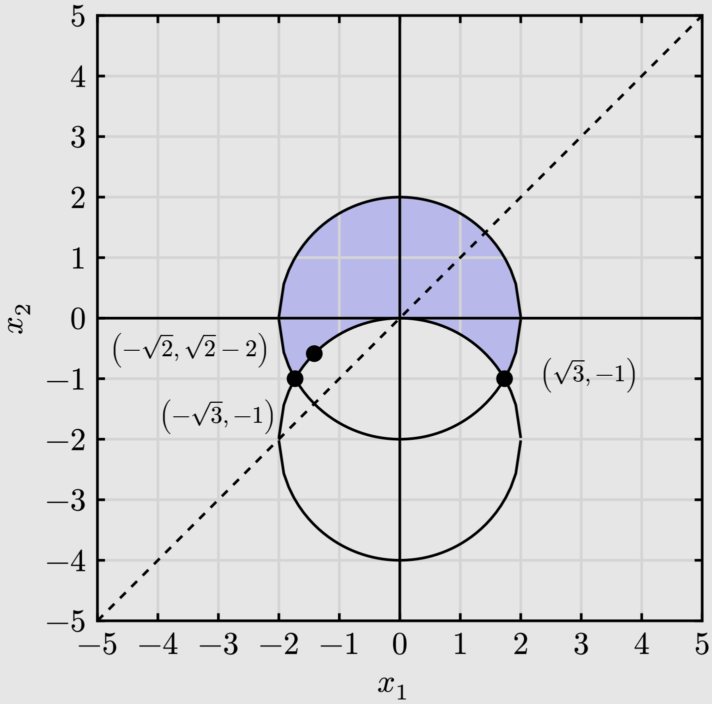
4️⃣ Summary

- $\lambda_1 > 0$ and $\lambda_2 > 0$:
  - $\left(\sqrt{3}, -1\right)$
  - $\left(-\sqrt{3}, 1\right)$

- $\lambda_1 = 0$ and $\lambda_2 > 0$:
  - $\left(-\sqrt{2}, \sqrt{2} - 2\right)$

These are the only candidates for local maxima (and thus global maxima)

Necessay, but not sufficient for non-convex NLPs



27.35.5. Sensitivity Analysis and Shadow Prices

Lagrange multipliers measure how senitive the objective function is to changes in the constraints

27.35.5.1. Shadow Prices

The shadow price of a constraint is the value of the corresponding Lagrange multiplier at the optimal solution:

$$\text{ShadowPrice}_i = \lambda_i^*$$

It measures how much the objective function would improve if the constraint were relaxed by one unit. For example:

$$x_1 + x_2 \leq 12 \quad \rightarrow \quad x_1 + x_2 \leq 13$$

The corresponding shadow price tells us the change in the optimal objective due to this relaxation. The magnitude of the Lagrange multiplier reflects the influence of its associated constraint on the optimal solution.

	Constraint Status	Shadow Price (λ^*)	Impact on Objective

Binding / Active	$g_i(x^*) = 0$	$\lambda^* > 0$	Relaxing improves objective
Inactive / Slack	$g_i(x^*) < 0$	$\lambda^* = 0$	Relaxing has no effect

Example

Primal NLP

$$z^* = \max_{x \in \mathbb{R}^n} \{ f(x) \mid g_i(x) \leq b_i \quad \forall i = 1, \ldots, m \}$$

Lagrange **Dual** Program

$$w^* = \min_{\lambda \geq 0} z^L(\lambda) = \min_{\lambda \geq 0} \left\{ \max_{x \in \mathbb{R}^n} f(x) + \sum_{i=1}^{m} \lambda_i [b_i - g_i(x)] \right\}$$

Convexity

The Lagrange Dual Program function $z^L(\lambda)$ is always convex over $\lambda \in [0, \infty)^n$

Weak Duality

$$w^* \geq z^* \quad \forall \lambda \geq 0$$

Strong Duality

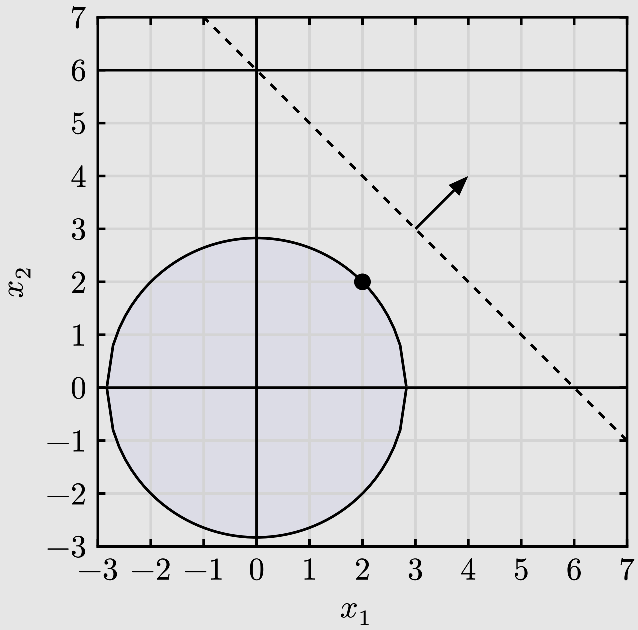
$w^* = z^*$ is the primal NLP is a "regular" **convex** program

Example

$$\begin{array}{rcrrrcl} z^* & = & \max & x_1 & +x_2 & & \\ & & s.t. & x_1^2 & +x_2^2 & \leq & 8 \\ & & & & x_2 & \leq & 6 \end{array}$$

Optimal solution

$$x^* = (2, 2)$$
$$z^* = 4$$



Lagragian relaxation

$$w^* = \min_{\lambda_1 \geq 0, \lambda_2 \geq 0} z^L(\lambda) = \min_{\lambda_1 \geq 0, \lambda_2 \geq 0} \max_{x \in \mathbb{R}^2} \mathcal{L}(x \mid \lambda)$$

Where

$$\mathcal{L}(x \mid \lambda) = x_1 + x_2 + \lambda_1(8 - x_1^2 + x_2^2) + \lambda_2(6 - x_2)$$

To solve the Lagrange dual program:

$$\max_{x \in \mathbb{R}^2} \mathcal{L}(x \mid \lambda) \max_{x \in \mathbb{R}^2} \{x_1 + x_2 + \lambda_1(8 - x_1^2 - x_2^2) + \lambda_2(6 - x_2)\}$$

First-order condition:

$$x_1 = \frac{1}{2\lambda_1} \quad \text{and} \quad x_2 = \frac{1 - \lambda_2}{2\lambda_1}$$

Plugging into $\mathcal{L}(x \mid \lambda)$:

$$z^L(\lambda) = \max_{x \in \mathbb{R}^2} \mathcal{L}(x \mid \lambda) = \frac{1 + (1 - \lambda_2)^2}{4\lambda_1} + 8\lambda_1 + 6\lambda_2$$

The Lagrangian dual program is to look for $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ to minimize $z^L(\lambda)$

The Lagrange dual program:

$$\min_{\lambda_1 \geq 0, \lambda_2 \geq 0} \frac{1 + (1 - \lambda_2)^2}{4\lambda_1} + 8\lambda_1 + 6\lambda_2$$

$z^L(\lambda)$ is convex over $(0, \infty)^2$:

$$\nabla z^L(\lambda) = \begin{bmatrix} -\frac{1+(1-\lambda_2)^2}{4\lambda_1^2} + 8 \\ \\ -\frac{1-\lambda_2}{2\lambda_1} + 6 \end{bmatrix}, \quad \nabla^2 z^L(\lambda) = \begin{bmatrix} \frac{1+(1-\lambda_2)^2}{2\lambda_1^2} & \frac{1-\lambda_2}{2}\lambda_1^2 \\ \\ \frac{1-\lambda_2}{2\lambda_1^2} & \frac{1}{2\lambda_1} \end{bmatrix}$$

Since:

$\frac{1+(1-\lambda_2)^2}{2\lambda_1^2} > 0$ and $\det(\nabla^2 z^L(\lambda)) = \frac{1}{2\lambda_1} > 0$, $z^L(\lambda)$ is convex

To solve

$$\min_{\lambda_1 \geq 0, \lambda_2 \geq 0} \frac{1+(1-\lambda_2)^2}{4\lambda_1} + 8\lambda_1 + 6\lambda_2$$

Apply KKT conditions

λ_1 cannot be binding ($\lambda_1 = 0$), because division by 0, so ignore it

Let $\mu \geq 0$ be the Lagrange multiplier for $\lambda_2 \geq 0$, the Lagrange is

$$\frac{1+(1-\lambda_2)^2}{4\lambda_1} + 8\lambda_1 + 6\lambda_2 - \mu\lambda_2$$

The KKT condition requires an optional solution to satisfy (FOC)

$$-\frac{1+(1-\lambda_2)^2}{4\lambda_1^2} + 8 = 0, \quad -\frac{1-\lambda_2}{2\lambda_1} + 6 - \mu = 0, \quad \mu\lambda = 0$$

- Suppose $\mu > 0$
  - Implies $\lambda_2 = 0$
  - $-\frac{1+(1-\lambda_2)^2}{4\lambda_1^2} + 8 = 0$ requires $\lambda_1 = \frac{1}{4}$
  - $-\frac{1-\lambda_2}{2\lambda_1} + 6 - \mu = 0$ requires $\mu = 4$, which is feasible
- Suppose $\mu = 0$
  - $-\frac{1+(1-\lambda_2)^2}{4\lambda_1^2} + 8 = 0$ requires $\lambda_2 = 1 - 12\lambda_1$
  - Plugging into $-\frac{1+(1-\lambda_2)^2}{4\lambda_1^2} + 8 = 0$ results on $1 + 112\lambda_1^2$ which is impossible
- The only KKT point is $(\lambda_1, \lambda_2) = \left(\frac{1}{4}, 0\right)$
- Plugging into $z^L(\lambda)$ gives us $w^* = 4$ which exactly equals z^*

So strong duality holds, $w^* = z^*$

27.35.6. Lagrange Duality and LP Duality

LP duality is a special case of Lagrange Duality

For a general LP

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Where $A \in \mathbb{R}^{m \times n}$ (m constraints and n variables)

Let $\lambda \in \mathbb{R}^m$ be the Lagrange multipliers, the Lagrange relaxation is

$$
\begin{aligned}
z^*(\lambda) &= \max_{x \geq 0} c^T x + \lambda^T (b - Ax) \\
&= A^T b + \max_{x \geq 0} (c^T - \lambda^T A) x
\end{aligned}
$$

Because of equality constraint, λ is unrestricted in sign

Lagrange dual program

$$\min_{\lambda} z^L(\lambda) = \min_{\lambda} \left\{ \lambda^T b + \max_{x \geq 0} (c^T - \lambda^T A) x \right\}$$

Search for λ that minimizes $z^L(\lambda)$

Dual program is only meaningful if $c^T \leq \lambda^T A$, because if $\left(c^T\right)_i > \left(\lambda^T A\right)_i$ for any i, $\max_{x \geq 0} (c^T - \lambda^T A) x$ will be unbounded because we can increase x_i to ∞

Thus, no choice of λ that violates $c^T \leq \lambda^T A$ may be optimal to the Lagrange dual program

The Lagrange dual program

$$\min_{\lambda} z^*(\lambda) = \min_{\lambda_i c^T \leq \lambda^T A} \left\{ \lambda^T b + \max_{x \geq 0} (c^T - \lambda^T A) x \right\}$$

If λ satisfies $c^T \leq \lambda^T A$ thenwe know $\max_{x \geq 0} (c^T - \lambda^T A) x = 0$

The Lagrange dual program becomes

$$
\begin{aligned}
\min_{\lambda \text{ urs}} \quad & \lambda^T b \\
s.t. \quad & \lambda^T A \geq c^T
\end{aligned}
$$

Which is exactly the dual LP of

$$
\begin{aligned}
\max_{x \geq 0} \quad & c^T x \\
s.t. \quad & Ax = b
\end{aligned}
$$

Example

$$
\begin{aligned}
\min \quad & (x_1 - 4)^2 + (x_2 - 2)^2 \\
s.t. \quad & 2x_1 + x_2 \leq 6
\end{aligned}
$$

1. What are the leading principle minors of the Hessian matrix of the objective function?

$$\nabla f(x_1, x_2) = \begin{bmatrix} 2(x_1 - 4) \\ 2(x_2 - 2) \end{bmatrix} \qquad \nabla^2 f(x_1, x_2) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

2 and 4 ✅

2.

$$\mathcal{L}(x_1, x_2 \mid \lambda) = (x_1 - 4)^2 + (x_2 - 2)^2 + \lambda(6 - 2x_2 - 2)$$

$\lambda \leq 0$ ✅

3. According to the FOC of the Lagragian, what is a necessary condition for an optimal solution

Case 1. $\lambda > 0$

Constraint active: $2x_2 - 2 \leq 6$

FOC (gradient w.r.t. x_1 and x_2):

$$\nabla \mathcal{L}(x_1, x_2 \mid \lambda) = \begin{bmatrix} 2(x_1 - 4) \\ 2(x_2 - 2) - 2\lambda \end{bmatrix}$$

$$2(x_1 - 4) = 0$$
$$x_1 = 4$$

$$2(x_2 - 2) - 2\lambda = 0$$
$$2x_2 - 4 - 2\lambda = 0$$
$$x_2 = \frac{2\lambda + 4}{2}$$
$$x_2 = \lambda + 2$$

$$x_1 = 4$$
$$x_2 = \lambda + 2$$

Substitute into active constraint

$$6 - 2x_2 - 2 = 0$$
$$4 - 2x_2 = 0$$
$$4 - 2(\lambda + 2) = 0$$
$$4 - 2\lambda + 4 = 0$$
$$2\lambda = 0$$
$$\lambda = \frac{0}{-2}$$
$$\lambda = 0$$

Find λ:

$\lambda = 0$

Which violates $\lambda > 0$

Case 2. $\lambda = 0$

$$x_1 = 4$$
$$x_2 = \lambda + 2$$
$$x_2 = 0 + 2$$

$$(x_1, x_2) = (4, 2)$$

$\boldsymbol{x_1 - 2x_2 = 0}$ ✅

Case 2. $\lambda = 0$

4. What is a local optimal solution to the nonlinear program?

$(x_1, x_2) = (\frac{12}{5}, \frac{6}{5})$ ✅

5.

For a linear program, linear programming duality and Lagrange duality is equivalent (i.e., the dual programs obtained through the two ways are identical). ✅

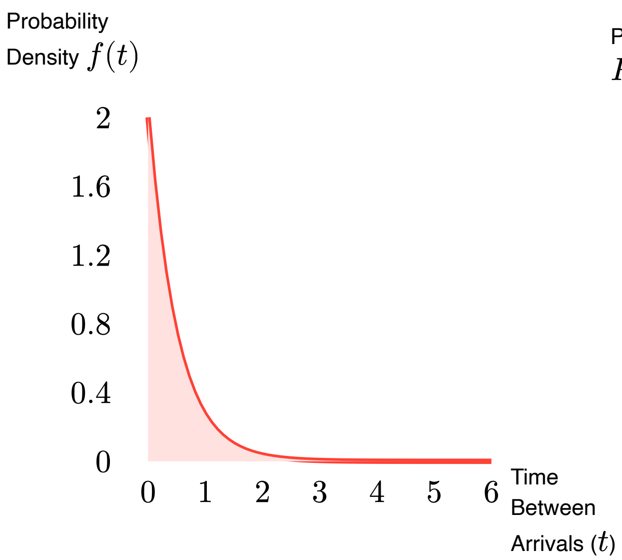
For an unconstrained nonlinear program, the KKT condition is equivalent to the first-order condition. ✅

For an unconstrained nonlinear program, the KKT condition is necessary and sufficient. ✅

28. Queuing Theory

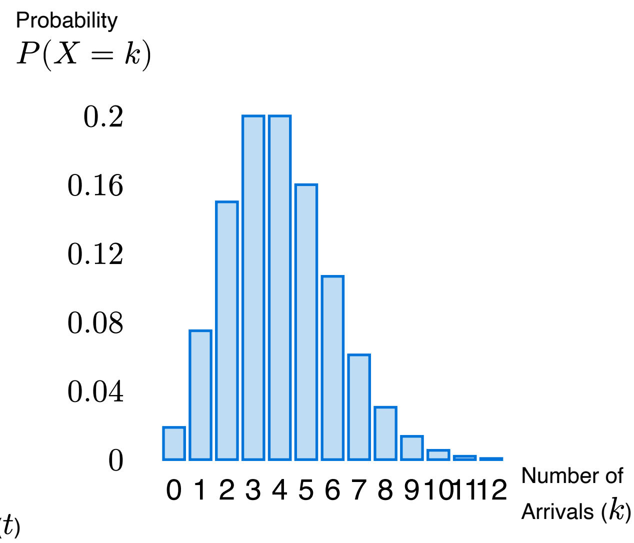
28.1. M/M/1

- Arrival rate (λ): Average number of customers arriving per unit of time
- Service rate (μ): Average number of customers served per unit of time



Exponential Distribution:
Inter-arrival Times

Rate parameter λ = 2 (same as above)



Poisson Distribution:
Number of Arrivals in Time Period $t = 2$

Rate parameter $\lambda = 2$, so $\lambda t = 4$

Key Relationship

Both distributions share the same rate parameter λ = 2

- If inter-arrival $\times \sim \text{Exponential}(\lambda)$, then arrivals in time $t \sim \text{Poisson}(\lambda t)$

- Mean inter-arrival time = $\frac{1}{\lambda}$ = 0.5
- Mean arrivals per unit time = λ = 2
- Mean arrivals in time period 2 = λt = 4

1. Utilization (ρ)

Fraction of time the server is busy

$$\rho = \frac{\lambda}{\mu}$$

2. Average Number of Customers in the System (L)

Average number of customers (both waiting and being served) in the system

$$L = \frac{\rho}{1-\rho}$$

3. Average Number of Customers in the Queue (L_q)

Average number of customers waiting in the queue

$$L_q = \frac{\rho^2}{1-\rho}$$

4. Average Time a Customer Spends in the System (W)

Average time a customer spends in the system (from arrival until they are done being served)

$$W = \frac{1}{\mu - \lambda}$$

5. Average Waiting Time in the Queue (W_q)

Average time a customer spends just waiting in line before being served

$$W_q = \frac{\rho}{\mu - \lambda}$$

6. Probability that the System is Empty (P_0)

Probability that there are zero customers in the system (no one is being served and no one is waiting)

$$P_0 = 1 - \rho$$

7. Probability that n Customers are in the System (P_n)

Probability that there are n customers in the system (either waiting or being served)

$$P_n = (1-\rho) \cdot \rho^n$$

8. Probability the Queue is Full (if the Queue has Limited Capacity) ($P_{n_{\max}}$)

Probability that the system is at full capacity

$$P_{n_{\max}} = (1-\rho) \cdot p^{n_{\max}}$$

9. System Throughput

Rate at which customers are served and leave the system

$$\text{Throughput} = \lambda$$

10. Expected Time in Service (W_s)

Average time a customer spends actually being served (not including waiting time)

$$W_s = \frac{1}{\mu}$$

11. Idel Time ($1 - \rho$)

Fraction of time that the server is idle (i.e., not serving any customers)

$$\text{Idle Time} = 1 - \rho$$

12. Probability of Having to Wait in the Queue (P_w)

Probability that an arriving customer will have to wait before being served, i.e., that the server is busy when the customer arrives

$$P_w = \rho$$

13. Variance of the Number of Customers in the System ($\text{Var}(L)$)

Variance of the number of customers in the system

$$\text{Var}(L) = \frac{\rho}{(1 - \rho)^2}$$

Example

A bank with a single teller

- **Arrival rate (λ)**: On average, 4 customers arrive every 10 minutes ($\lambda = 4$ custommers per 10 minutes)
- **Service Rate (μ)**: The teller can serve 6 customers every 10 minutes ($\mu = 6$ customers per 10 minutes)

1. **Utilization (ρ)**

$$\rho = \frac{\lambda}{\mu} = \frac{4}{6} = 0.67$$

The teller is busy 67% of the time. The remaining 33% of the time, the teller is idle, waiting for the next customer

2. Average Number of Customers in the System (L)

$$L = \frac{\rho}{1 - \rho} = \frac{0.67}{1 - 0.67} = 2$$

On average, there are 2 customers in the coffee shop at any given time, either being served or waiting in line

3. Average Number of Customers in the Queue (L_q)

$$L_q = \frac{\rho^2}{1-\rho} = \frac{0.67^2}{1-0.67} = 1.33$$

On average, about 1.33 customers are waiting in line at any time

4. Average Time a Customer Spends in the System (W)

$$W = \frac{1}{\mu - \lambda} = \frac{1}{6} - 4 = 0.5$$

On average, a customer spends 5 minutes (0.5×10 minutes) in the shop (including both waiting in line and getting served)

5. Average Waiting Time in the Queue (W_q)

$$W_q = \frac{\rho}{\mu - \lambda} = \frac{0.67}{6-4} = 0.33$$

On average, a customer waits 3.3 minutes (0.33×10 minutes) in line before being served by the teller

6. Probability that the System is Empty (P_0)

$$P_0 = 1 - \rho = 1 - 0.67 = 0.33$$

There is a 33% chance that the coffee shop is empty, meaning there is no customer in the queue or being served

7. Probability that n Customers are in the System (P_n)

$$P_n = (1-\rho) \cdot \rho^n = (1-0.67) \cdot 0.67^2 = 0.148$$

There is a 14.8% chance that exactly 2 customers are either in line or being served

8. Probability the Queue is Full (if the Queue has Limited Capacity) ($P_{n_{\max}}$)

$$P_{n_{\max}} = (1-\rho) \cdot p^{n_{\max}} = (1-0.67) \cdot 0.67^5 = 0.028$$

There is a 2.8% chance that the system is full, and no new customers can enter

9. System Throughput

$$\text{Throughput} = \lambda = 4$$

The coffee shop serves 4 customers every 10 minutes, on average

10. Expected Time in Service (W_s)

$$W_s = \frac{1}{\mu} = \frac{1}{6} = 0.167$$

On average, a customer spends 1.67 minutes being served by the teller

11. Idel Time ($1 - \rho$)

$$\text{Idle Time} = 1 - \rho = 1 - 0.67 = 0.33$$

The teller is idle 33% of the time

12. Probability of Having to Wait in the Queue (P_w)

$$P_w = \rho = 0.67$$

There is a 67% chance that a customer will have to wait when they arrive

13. Variance of the Number of Customers in the System ($\text{Var}(L)$)

$$\text{Var}(L) = \frac{\rho}{(1-\rho)^2} = \frac{0.67}{(1-0.67)^2} = 6.12^2$$

The queue length varies significantly, with a variance of 6.12 customers

Costs

- **Cost per Waiting Customer per Unit Time (C_w)**: cost incurred per customer for each unit of time they spend waiting in the queue

$$\text{Total Waiting Cost} = L_q \times C_w \times \text{Unit Time}$$

- **Cost per Idle Server per Unit Time (C_s)**: cost incurred per unit time when the server is not serving customers

$$\text{Total Idle Cost} = (1-\rho) \times C_s \times \text{Unit Time}$$

Total Cost

$$\text{Total Cost} = \text{Total Waiting Cost} + \text{Total Idle Cost}$$

Example

- Arrival Rate (λ): 4 customers per 10 minutes
- Service Rate (μ): 6 customers per 10 minutes
- Cost per Waiting Customer per Hour (C_w): \$10
- Cost per Idle Server per Hour (C_s): \$20
- Operational Time: 1 hour

1. Utilization

$$\rho = \frac{\lambda}{\mu} = \frac{4}{6} = 0.67$$

2. Average Number of Customers in Queue (L_q)

$$L_q = \frac{\rho^2}{1-\rho} = \frac{0.67^2}{1-0.67} = 1.33 \text{ cusommers}$$

3. Total Waiting Cost

$$\text{Total Waiting Cost} = L_q \times C_w \times \text{Unit Time} = 1.33 \times 10 \times 1 = \$13.33$$

4. Idle Time

$$\text{Idle Time} = 1 - \rho = 1 - 0.67 = 0.33$$

5. Total Idle Cost

$$\text{Total Idle Cost} = (1-\rho) \times C_s \times \text{Unit Time} = 0.33 \times 20 \times 1 = \$6.67$$

6. Total Cost

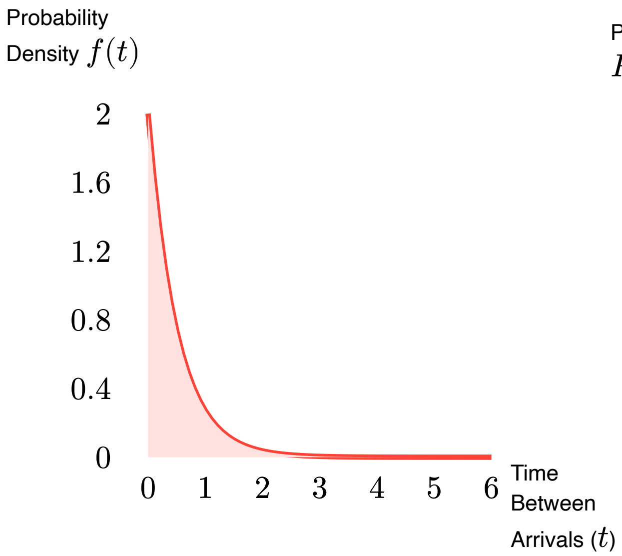
$$\text{Total Cost} = \text{Total Waiting Cost} + \text{Total Idle Cost} = 13.33 + 6.67 = \$20$$

28.2. M/M/c

- M: Memoryliss interarrival times
- M: Memoryless service times
- c: number of servers

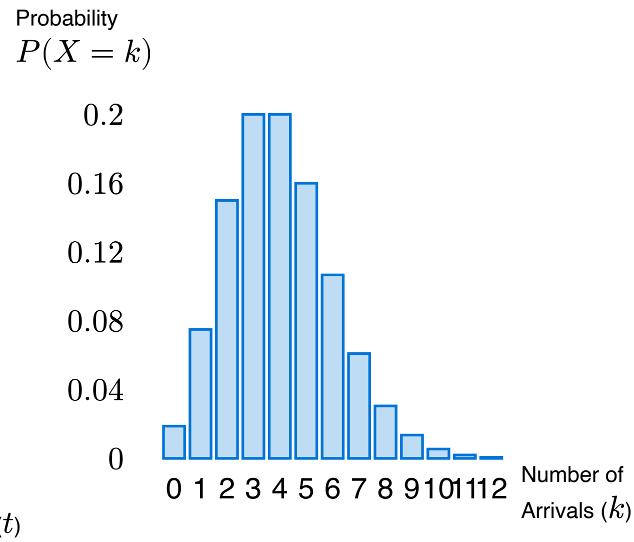
Input parameters:

- c: number of parallel servers in the system
- μ: average service rate of µ (mu) per server
  - Exponential distribution: μ customers per unit of time
- λ: average rate of arrivals per unit of time
  - Poisson process: number of arrivals in a given time period
  - Exponential distribution: time between successive arrivals



Exponential Distribution:
Inter-arrival Times

Rate parameter λ = 2 (same as above)



Poisson Distribution:
Number of Arrivals in Time Period $t = 2$

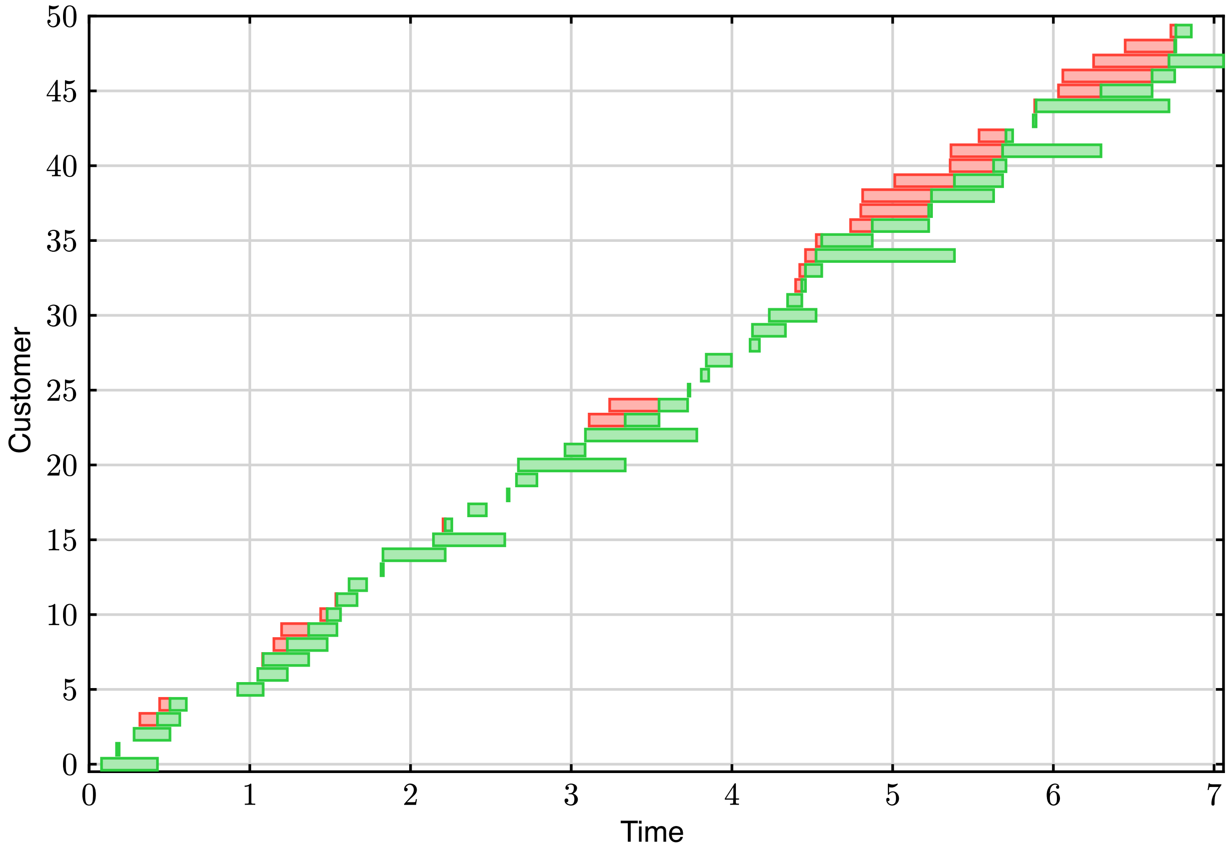
Rate parameter $\lambda = 2$, so $\lambda t = 4$

System Stability

- $\lambda > c\mu$:
  - Arrival rate greater than service rate
  - Queue will grow infinitely
- $\lambda < c\mu$:

- Without variability: no queue or wait time
- With variability: may have wait time

- $\lambda = c\mu$:



Performance Metrics

1. Utilization (ρ)

$$\rho = \frac{\lambda}{c \cdot \mu}$$

Where:

- λ: arrival rate (mean number of arrivals per unit of time)
- μ: service rate (mean number of services completed per server per unit of time)

2. Probability of Zero Customers in the System (Idle Probability) (P_0)

Probability that no customers are in the system

$$P_0 = \left(\sum_{n=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^2}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-p)} \right)^{-1}$$

3. Probability that All Servers are Busy (Blocking Probability or Queueing Probability) (P_w)

Probability that all c servers are busy and a customer will have to wait in the queue

$$P_w = \frac{\frac{\left(\frac{\lambda}{\mu}\right)^c}{c!}}{\sum_{n=0}^{c-1} \frac{\frac{\left(\frac{\lambda}{\mu}\right)^n}{c!}}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-p)}}$$

4. Average Number of Customers in the System (L)

Average number of customers in the system (both in the queue and being served)

$$L = \frac{\lambda}{\mu} + \frac{P_w \cdot \left(\frac{\lambda}{\mu}\right)}{c(1-p)}$$

5. Average Number of Customers in the Queue (L_q)

Average number of customers waiting in the queue (not being served)

$$L_q = P_w \cdot \left(\frac{\rho \cdot \left(\frac{\lambda}{\mu}\right)}{1-\rho}\right)$$

6. Average Time a Customer Spends in the System (W)

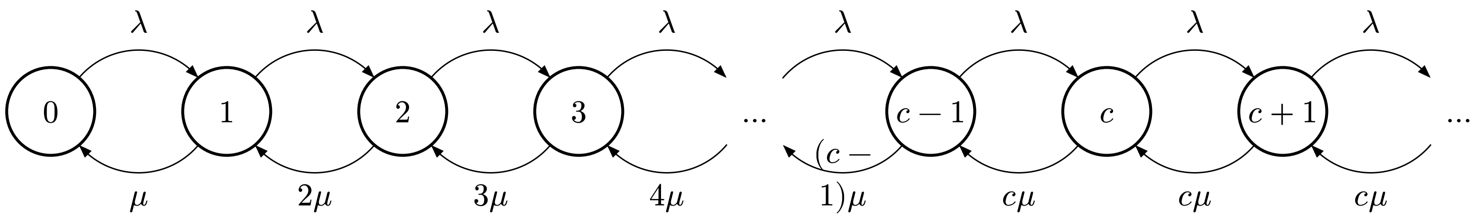
Average time a customer spends in the system (both waiting and being served) (Little's Law)

$$W = \frac{L}{\lambda}$$

7. Average Time a Customer Spends in the Queue (W_q)

Average waiting time a customer spends in the queue before being served

$$W_q = \frac{L_q}{\lambda} = P_w \cdot \frac{1}{c\mu(1-p)}$$



28.3. Little's Law

$$L = \lambda W$$

Where:

- L: average number of items (customers, tasks, etc.) in the system
- λ: average arrival rate (items per unit time)
- W: average time an item spends in the system

Example

Bank where customers arrive at an average rate of 10 customers per hour ($\lambda = 10$ customers / hour). Each customer spends an average of 12 minutes in the bank ($W = \frac{12}{60} = 0.2$ hours).

$$L = \lambda W = 10 \times 0.2 = 2$$

On average, there are 2 customers in the bank at any given time (includes being served and those waiting in line)

29. Control Charts

Chart	Tracks	Data Typse	Sample Size	Statistical Basis	Use Case
c	Number of defects per unit	Count (discrete)	Fixed	Poisson	When you count defects per inspection unit (e.g., scratches, cracks) and sample size is constant
u	Number of defects per unit (standardized)	Count per unit (continuous)	Variable	Poisson	Same as c-chart but for variable sample sizes (defects per unit area, time, etc.)
np	Number of defective units	Count (discrete)	Fixed	Binomial	Each item is either defective or not; count how many items fail inspection
p	Proportion of defective units	Proportion (0-1)	Variable	Binomial	Same as np-chart but with variable

Chart	Tracks	Data Typse	Sample Size	Statistical Basis	Use Case
					sample size — tracks percentage defective
$\bar{x}$	Mean of continuous variable	Continuous	Grouped samples	Normal	Monitors process average (e.g., length, weight, temperature) over time
R	Range of values in a sample	Continuous	Grouped samples	Range distribution (based on Normal)	Used with x̄-chart to monitor variation in process (spread)

29.1. C-charts (Count)

Count of defects (fixed unit size)

Number of defects observed in each sample or inspection unit

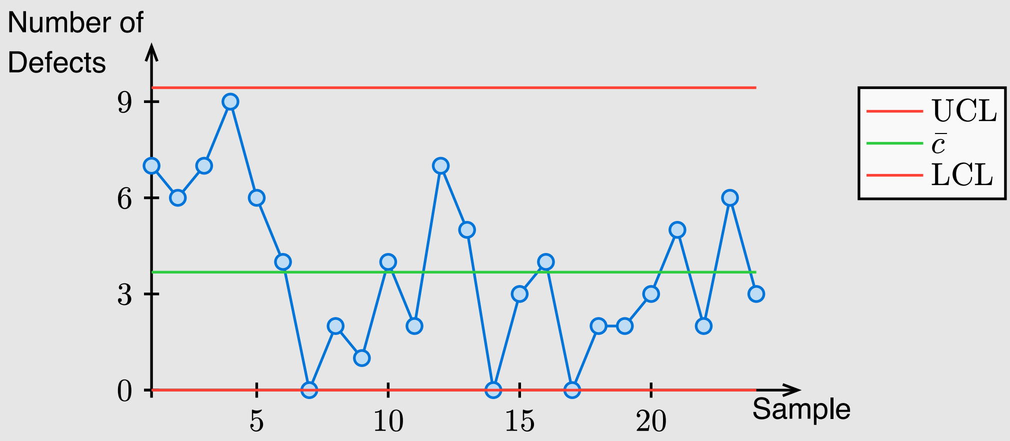
$$\bar{c} = \frac{1}{k}\sum_{i=1}^{k} c_i$$

$$\mathrm{UCL}_c = \bar{c} + 3\sqrt{\bar{c}}$$
$$\mathrm{LCL}_c = \bar{c} - 3\sqrt{\bar{c}}$$

Where:

- c: number of defects
- k: number of samples

Example



This C-chart displays the number of defects identified in each of 25 inspected units, where each unit is of fixed size. The center line represents the average number of defects across all samples, calculated as:

$$\bar{c} = \frac{1}{k} \sum_{i=1}^{k} c_i$$

Control limits are drawn at ±3 standard deviations from the mean, using the square root of $\bar{c}$ to account for Poisson-distributed defect counts:

$$\mathrm{UCL}_c = \bar{c} + 3\sqrt{\bar{c}} \qquad \mathrm{LCL}_c = \max\left(0, \bar{c} - 3\sqrt{\bar{c}}\right)$$

29.2. P-charts (Proportion)

The p-chart monitors the proportion of defectives in each sample, accounting for variable sample sizes.

$$\mathrm{UCL}_{p_i} = \bar{p} + 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n_i}}$$

$$\mathrm{LCL}_{p_i} = \bar{p} - 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n_i}}$$

Where:

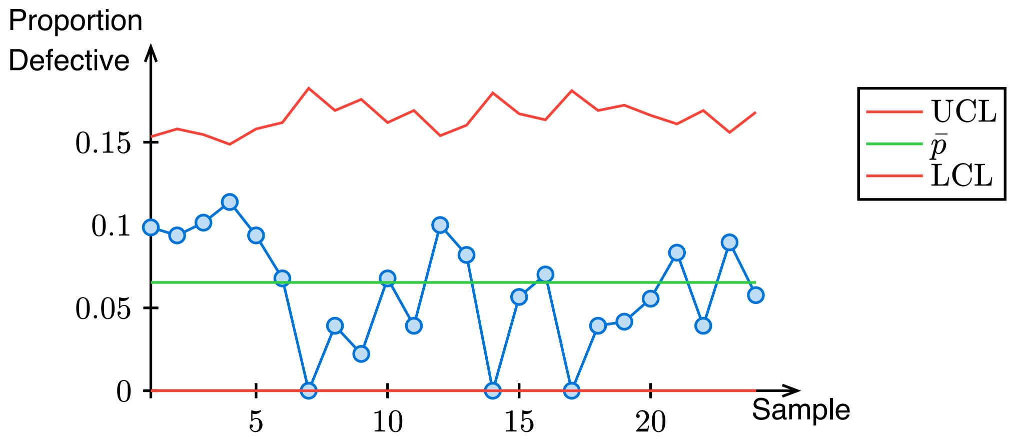
- d_i: The number of defective items observed in sample i
- p_i: sample proportion defective

$$p_i = \frac{d_i}{n_i}$$

- $\bar{p}$: overall proportion defective

$$\bar{p} = \frac{\sum d_i}{\sum n_i}$$

- n_i: sample size for sample i



29.3. U-charts (Unit)

Defects per unit (variable unit size)

$$\text{UCL}_u = \bar{u} + 3\sqrt{\frac{\bar{u}}{n_i}}$$

$$\text{LCL}_u = \bar{u} - 3\sqrt{\frac{\bar{u}}{n_i}}$$

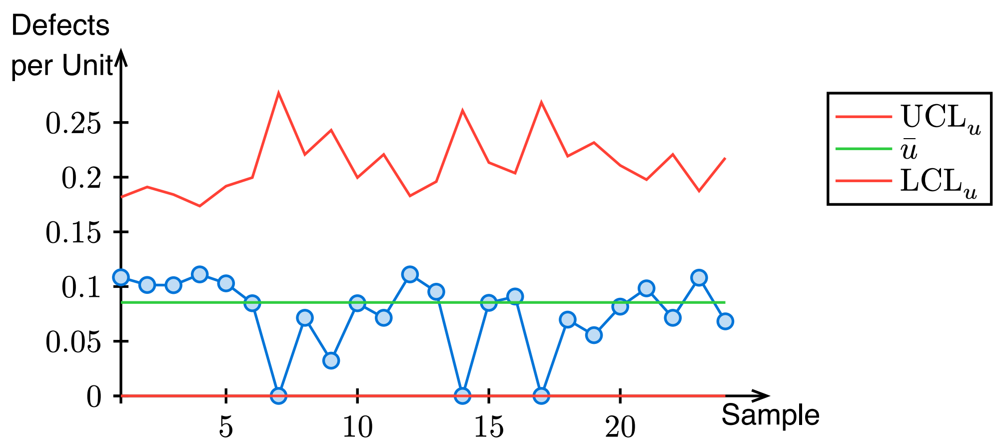
Where:

- c_i: number of defects in sample i
- n_i: sample size of group i
- u_i: defects per unit

$$u_i = \frac{c_i}{n_i}$$

- $\bar{u}$:

$$\frac{\sum c_i}{\sum n_i}$$



29.4. NP-charts (Number Proportion)

Number of defective items (constant sample size)

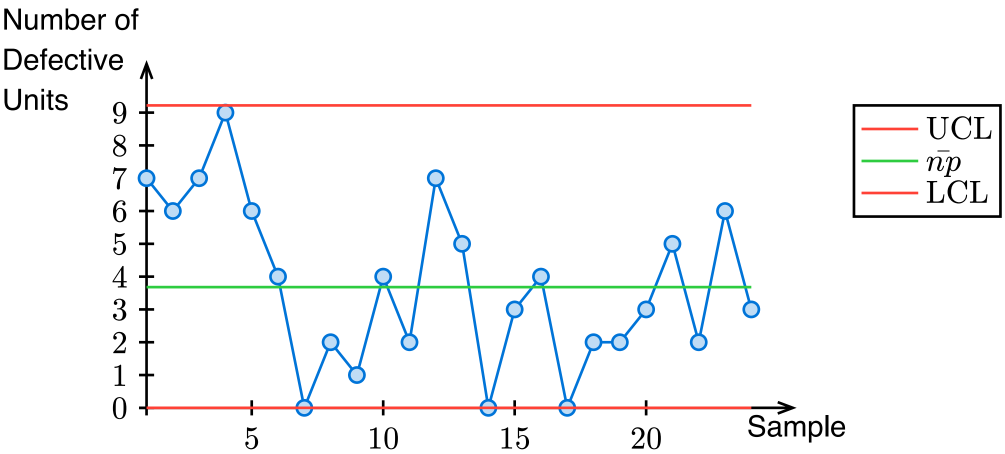
$$\mathrm{UCL}_{np} = n\bar{p} + 3\sqrt{n\bar{p}(1-\bar{p})}$$
$$\mathrm{LCL}_{np} = n\bar{p} - 3\sqrt{n\bar{p}(1-\bar{p})}$$

Where:

- n: count of defective items in each sample
- p: proportion of defectives
- $\bar{p}$:

$$\begin{aligned}\bar{p} &= \frac{1}{k}\sum_{i=1}^{k}\frac{d_i}{n} \\ &= \frac{\sum_{i=1}^{k} d_i}{n \cdot k}\end{aligned}$$

- $\bar{np} = n\bar{p}$



29.5. $\bar{x}$-chart

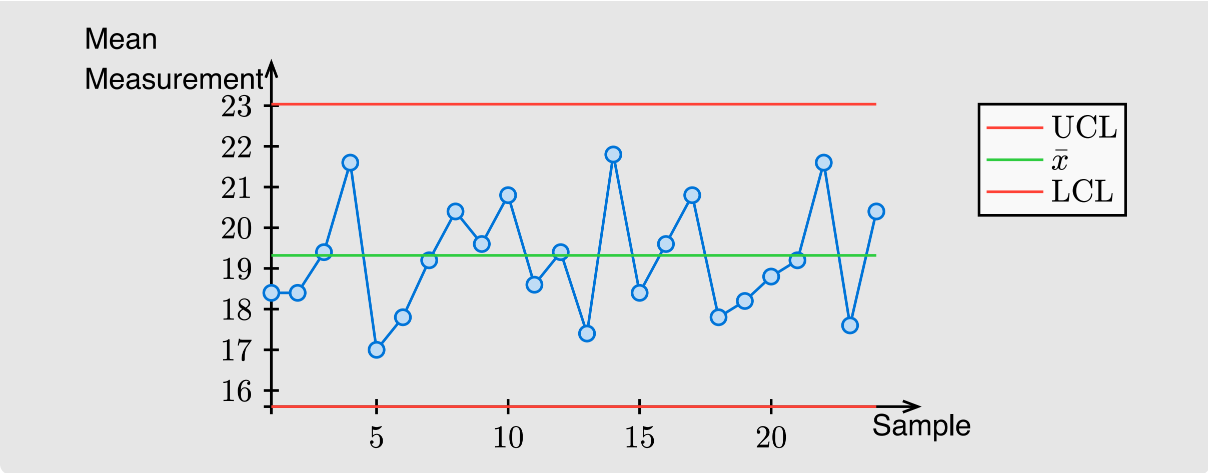
Used to monitor the average of a continuous quality characteristic (e.g., weight, length, temperature), assuming constant sample size per subgroup.

$$\mathrm{UCL}_{\bar{\bar{x}}} = \bar{\bar{x}} + A_2 \cdot \bar{R}$$
$$\mathrm{UCL}_{\bar{\bar{x}}} = \bar{\bar{x}} - A_2 \cdot \bar{R}$$

Where:

- $\bar{x}_i$: average of subgroup i
- $\bar{\bar{x}}$: grand mean of all subgroup means
- $\bar{R}$: average range of all subgroups
- A_2: constant dependent on sample size n (e.g., $A_2 = 0.577$ for $n = 5$)

Example



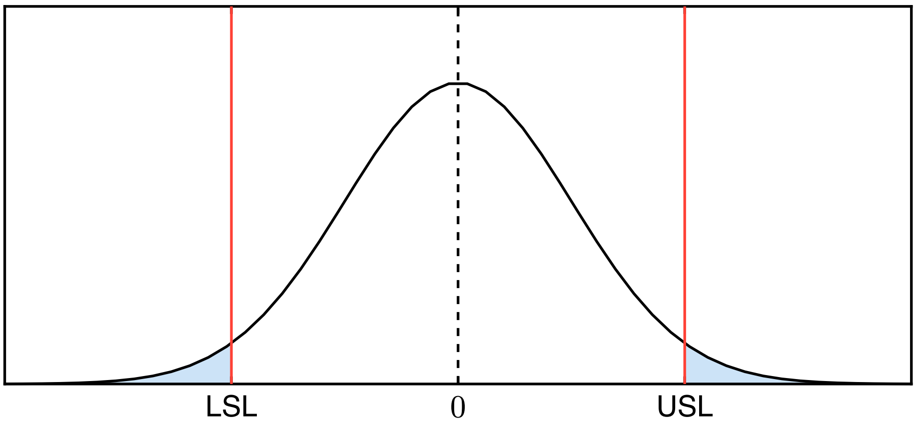
30. Process Capability Analysis

30.1. C_p (Process Capability Index)

Measure how well a process can produce outputs within specified limits

$$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$$

Assumption: Process is **centered** within the specification limits



C_p **Value**	**Meaning**	**Quality Impact**
$C_p < 1$	Process variation exceeds specs	Defects likely, even if centered
$C_p = 1$	Process barely fits within specs	OK if perfectly centered
$C_p > 1$	Process well within specs	Robust, fewer defects

$C_p = 2$	Six Sigma level	World-class performance

Example

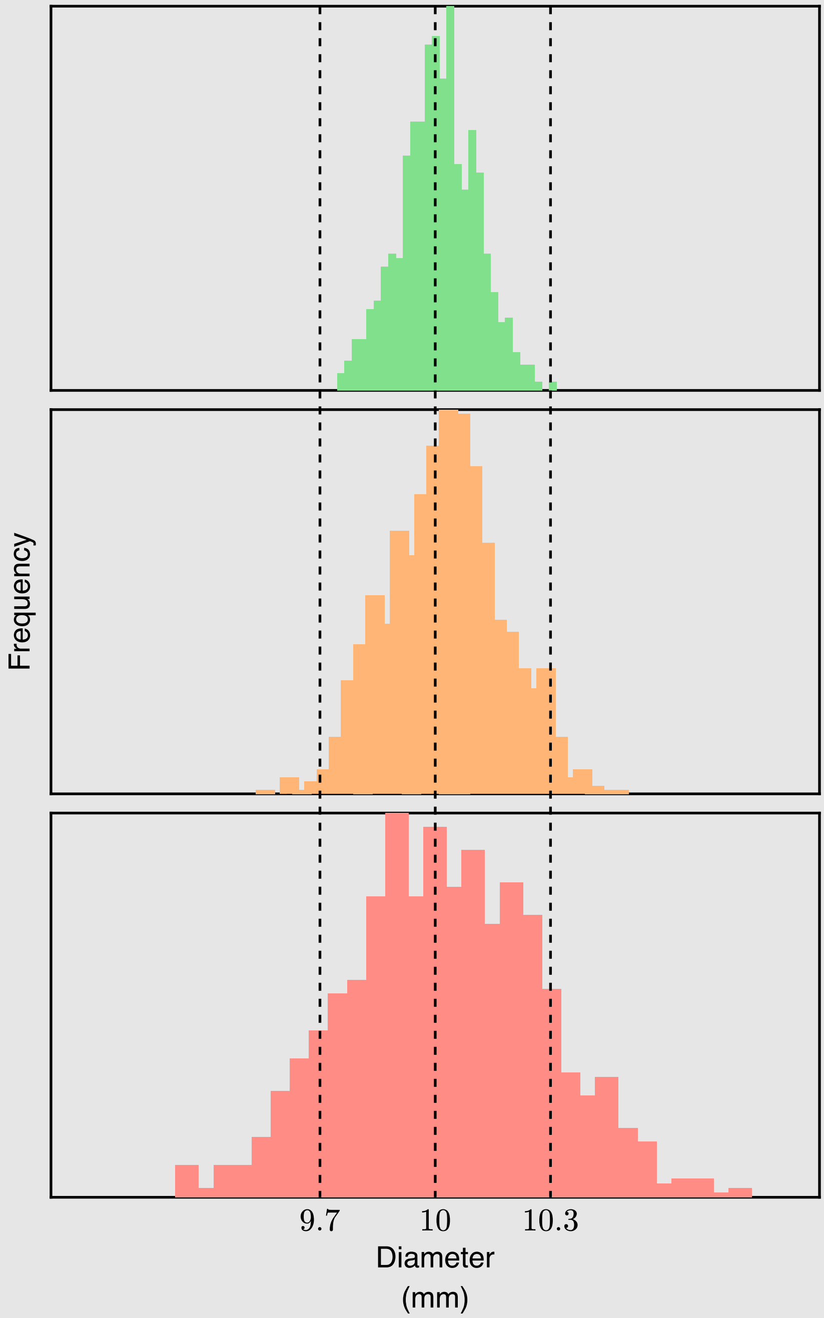
A manufacturing process produces metal rods with a target diameter of 10.0 mm. The specification limits are:

- Target: 10 mm
- Lower Specification Limit (LSL): 9.7 mm
- Upper Specification Limit (USL): 10.3 mm
- Specification Width (Tolerance): 0.6 mm

Let's analyze three different processes with varying levels of capability.

Process	Mean	Std Dev	C_p	Assessment
Process 1	9.998	0.099	1.009	Good
Process 2	10.001	0.147	0.682	Marginal
Process 3	9.994	0.255	0.393	Poor

Process 1	Process 2	Process 3	Six Sigma
$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$	$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$	$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$	$C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$
$= \frac{10.3 - 9.7}{6 \times 0.099}$	$= \frac{10.3 - 9.7}{6 \times 0.147}$	$= \frac{10.3 - 9.7}{6 \times 0.255}$	$= \frac{10.3 - 9.7}{6 \times 0.05}$
$= \frac{0.6}{0.6}$	$= \frac{0.6}{0.9}$	$= \frac{0.6}{1.5}$	$= \frac{0.6}{0.30}$
$= 1.009$	$= 0.682$	$= 0.393$	$= 2$



Step 3: Interpretation

Process	C_p	Within Spec	Defective	DPMO
Process 1	1.009	96.64%	3.36%	33600
Process 2	0.682	84.13%	15.87%	159700
Process 3	0.393	50.00%	50.00%	500000
Six Sigma	2.0	99.9999998%	0.0000002%	0.002

Higher C_p means fewer defects and better process quality. and lower defects per million opportunities (DPMO)

c_p.py

```
cp_values = [2, 1, 0.1]

for cp in cp_values:
    z = 3 * cp  # spec limit in standard deviations
    prob_defective = 2 * (1 - norm.cdf(z))
    within_specs = 1 - prob_defective
    dpmo = prob_defective * 1_000_000

    print({
      "Cp": cp,
      "Sigma limit (±z)": z,
      "Percent within specs": within_specs * 100,
      "Percent defective": prob_defective * 100,
      "DPMO": dpmo
    })
```

Output:

```
[{'Cp': 2,
  'Sigma limit (±z)': 6,
  'Percent within specs': 99.99999980268245,
  'Percent defective': 1.9731754008489588e-07,
  'DPMO': 0.0019731754008489591},
{'Cp': 1,
  'Sigma limit (±z)': 3,
  'Percent within specs': 99.73002039367398,
  'Percent defective': 0.2699796063260207,
  'DPMO': 2699.796063260207},
{'Cp': 0.1,
  'Sigma limit (±z)': 0.30000000000000004,
  'Percent within specs': 23.582284437790534,
  'Percent defective': 76.41771556220947,
  'DPMO': 764177.1556220946}]
```

30.2. C_{pk} (Process Capability Index with Centering)

Minimum distance from mean to nearest spec limit divided by 3 standard deviations. Process capability considering centering and variation.

If process is centered:

$$C_{pk} = C_p = \frac{\text{USL} - \text{LSL}}{6\sigma}$$

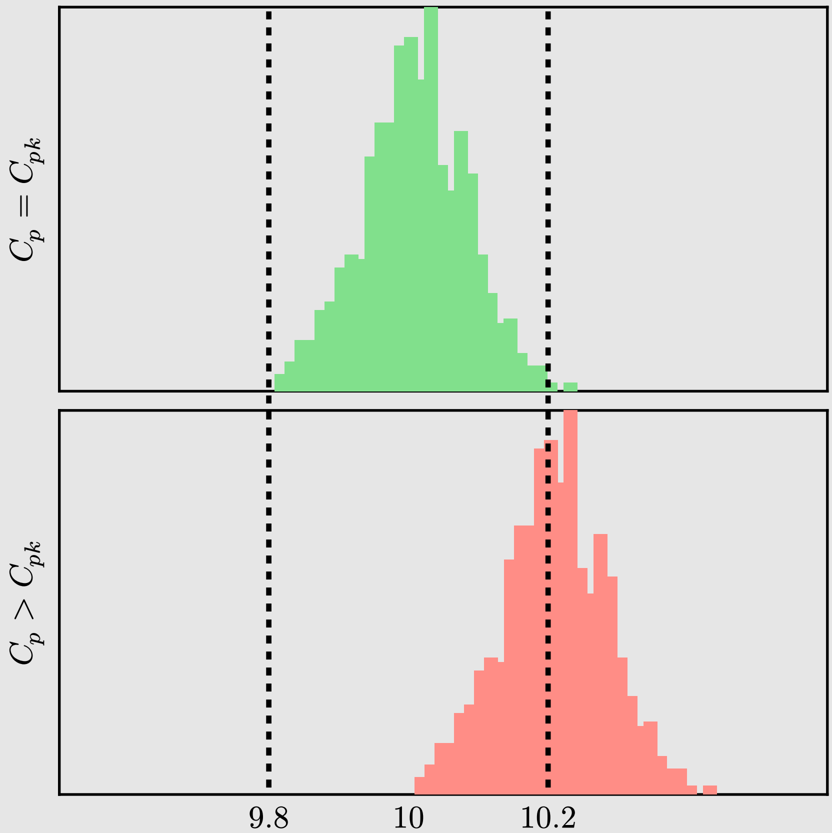
If process mean is off center, the smaller side of the process-to-limit ratio determines C_{pk}:

$$C_{pk} = \text{Min}\left(\frac{\text{USL} - \bar{x}}{3\sigma}, \frac{\bar{x} - \text{LSL}}{3\sigma}\right)$$

If $C_{pk} < C_p$, process is not centered

$C_{\{pk\}}$ Value	Meaning	Quality Impact
$C_{\{pk\}} < 1$	Process not capable, output falls outside specs	Many defects, poor performance
$C_{\{pk\}} = 1$	Process just meets specification limits	Acceptable only if well centered
$C_{\{pk\}} > 1$	Process is capable, centered or nearly so	Fewer defects, reliable quality
$C_{\{pk\}} = 2$	Excellent capability, Six Sigma level	Near-perfect quality

Example



Suppose a company manufactures metal rods, and the specification limits for the diameter of the rods are:

- Target: 10 mm

- Lower Specification Limit (**LSL**): 9.8 mm
- Upper Specification Limit (**USL**): 10.2 mm

The process has:

- A standard deviation σ of 0.05 mm.
- A process mean μ of 10.1 mm.

Step 1: Calculate the distance from the mean to the USL:

$$\frac{\text{USL} - \mu}{3\sigma} = \frac{10.2\ \text{mm} - 10.1\ \text{mm}}{3 \times 0.05\ \text{mm}} = \frac{0.1\ \text{mm}}{0.15\ \text{mm}} = 0.67$$

Step 2: Calculate the distance from the mean to the LSL:

$$\frac{\mu - \text{LSL}}{3\sigma} = \frac{10.1\ \text{mm} - 9.8\ \text{mm}}{3 \times 0.05\ \text{mm}} = \frac{0.3\ \text{mm}}{0.15\ \text{mm}} = 2.00$$

Step 3: Determine C_{pk}:

$$C_{pk} = \min(0.67, 2.00) = 0.67$$

Interpretation:

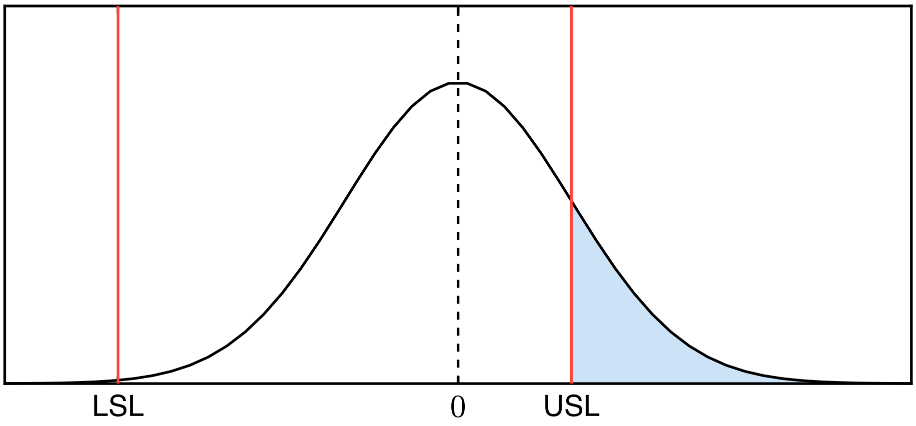
C_p = 1.33 means the process spread (6 σ) fits 1.33 times within the tolerance range (the distance between the Upper Specification Limit and Lower Specification Limit.

- C_p = 1.00: Process mean is exactly at the midpoint of the specification limits, and the process variation fits exactly within these limits. 99.73% of the output will be within specifications, indicating a capable process (3 sigma process).
- $C_p > 1.00$: The higher the C_{pk}, the more capable and stable the process is, meaning it can consistently produce parts within tolerance with minimal risk of defects.
- $C_p < 1$: The process mean is off-center or the variation is wider than the specification limits, or both. A significant portion of the output may fall outside the specification limits.

30.3. C_{pm} (Taguchi Capability Index)

30.4. P_p (Process Performance Index)

$$P_p = \frac{\text{USL} - \text{LSL}}{6\sigma_{\text{overall}}}$$



Example

Suppose a company manufactures metal rods, and the specification limits for the diameter of the rods are:

- Upper Specification Limit (**USL**): 10.2 mm
- Lower Specification Limit (**LSL**): 9.8 mm
- Overall standard deviation (σ_{overall}): 0.06 mm

Step 1: Determine the Specification Width

The specification width is the difference between the USL and LSL.

$$\text{Specification Width} = \text{USL} - \text{LSL} = 10.2 \text{ mm} - 9.8 \text{ mm} = 0.4$$

Step 2: Calculate the Process Performance Index P_p

The formula for P_p is:

$$P_p = \frac{\text{Specification Width}}{6\sigma_{\text{overall}}} = \frac{\text{USL} - \text{LSL}}{6\sigma_{\text{overall}}}$$

Substitute the values:

$$P_p = \frac{0.4 \text{ mm}}{6 \times 0.06 \text{ mm}} = \frac{0.4 \text{ mm}}{0.36 \text{ mm}} = 1.11$$

Interpretation:

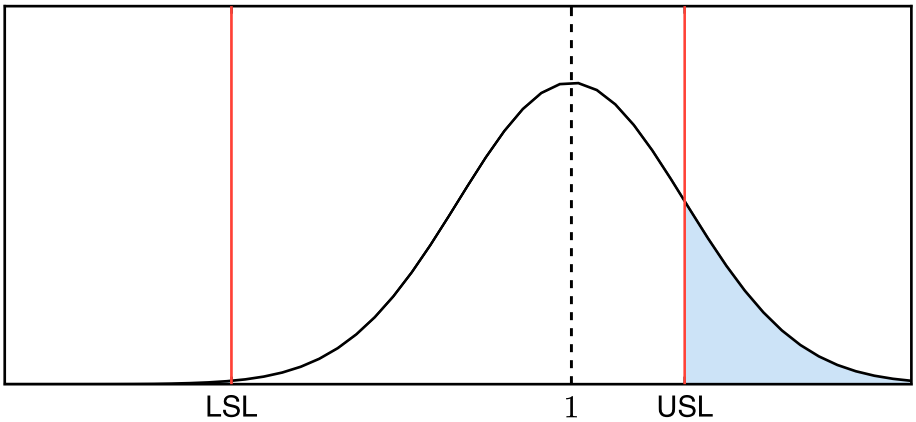
- A P_p of 1.11 indicates that the process performance, considering all sources of variation over time, is capable but less so than the potential capability indicated by C_p. The value being slightly above 1 suggests that the process can generally produce rods within specifications, but there might be more variability in the process compared to the short-term capability measured by C_p
- Decrease from the C_p value (1.33 to 1.11) reflects the impact of additional variability when evaluating the process over a longer time or under different conditions.

Use:

When you need to evaluate the **actual performance** of a process over a longer period, considering all sources of variation, including shifts, drifts, and other long-term factors.

30.5. P_{pk} (Process Performance Index with Centering)

$$P_{pk} = \text{Min}\left(\frac{\text{USL} - \mu_{\text{overall}}}{3\sigma}, \frac{\mu_{\text{overall}} - \text{USL}}{3\sigma}\right)$$



Example

Suppose a company manufactures metal rods, and the specification limits for the diameter of the rods are:

- Upper Specification Limit (**USL**): 10.2 mm
- Lower Specification Limit (**LSL**): 9.8 mm
- Overall standard deviation (σ_{overall}): 0.06 mm
- Overall process mean (μ_{overall}): 10.1 mm

Step 1: Calculate the Distance from the Process Mean to the Specification Limits

Calculate the distance from the overall process mean to both the USL and LSL:

$$\text{USL} - \mu_{\text{overall}} = 10.2 \text{ mm} - 10.1 \text{ mm} = 0.1 \text{ mm}$$

$$\mu_{\text{overall}} - \text{LSL} = 10.1 \text{ mm} - 9.8 \text{ mm} = 0.3 \text{ mm}$$

Step 2: Calculate the Process Performance Index P_k

The formula for P_{pk} is:

$$P_{pk} = \text{Min}\left(\frac{\text{USL} - \mu_{\text{overall}}}{3\sigma}, \frac{\mu_{\text{overall}} - \text{USL}}{3\sigma}\right)$$

Substitute the values:

$$P_{pk} = \text{Min}\left(\frac{0.1 \text{ mm}}{3 \times 0.06 \text{ mm}}, \frac{0.3 \text{ mm}}{3 \times 0.06 \text{ mm}}\right)$$

$$P_{pk} = \text{Min}\left(\frac{0.1 \text{ mm}}{0.18 \text{ mm}}, \frac{0.3 \text{ mm}}{0.18 \text{ mm}}\right)$$

$$P_{pk} = \min(0.56, 1.67) = 0.56$$

Interpretation:

- A P_p of 1.11 indicates that the process performance, considering all sources of variation over time, is capable but less so than the potential capability indicated by C_p. The value being slightly above 1 suggests that the process can generally produce rods within specifications, but there might be more variability in the process compared to the short-term capability measured by C_p
- Decrease from the C_p value (1.33 to 1.11) reflects the impact of additional variability when evaluating the process over a longer time or under different conditions.

Use:

When you need to evaluate the **actual performance** of a process over a longer period, considering all sources of variation, including shifts, drifts, and other long-term factors.

31. Inventory Management

31.1. Newsvendor

determine the optimal order quantity Q^* that minimizes the total expected cost or maximizes the expected profit, based on the trade-off between the overage and underage costs

Assumptions

- Products are separable
- Planning is done for a single period
- Demand is random
- Deliveries are made in advance of demand
- Costs of overage or underage are linear

1. Parameters
   - P: Sale Price
   - C: Purchase Cost
   - S: Unsold Value
   - μ: Mean Demand
   - σ: Standard Deviation Demand
2. Calculate Underage and Overage Costs:
   - Underage Cost (C_u): Profit lost for each unit of demand not met

     $$C_u = P - C$$

   - Overage Cost (C_o): This is the cost of holding an unsold newspaper.

$$C_o = C - S$$

3. Calculate Critical Ratio (CR)

$$CR = \frac{C_u}{C_u - C_o}$$

4. Find z-score: Find the number of standard deviations away from the mean corresponding to the critical ratio:

$$z^* = \Phi^{-1}(CR)$$

Where:

- Φ^{-1}: Inverse of the CDF of the standard normal distribution (PPF)

5. Calculate Optimal Order Quantity (Q^*)

$$Q^* = \mu + z^* \sigma$$

Where:

- z^*: z-score corresponding to the critical ratio CR from the standard normal distribution

Example

Consider a newsvendor selling newspapers:

1. Parameters
    - Sale Price (P): \$3 (per unit)
    - Purchase Cost (C): \$1 (per unit)
    - Unsold Value (S): \$0 (per unit)
    - Mean Demand (μ): 100 (units)
    - Standard Deviation Demand (σ): 20 (units)
2. Calculate Underage and Overage Costs:
    - Underage Cost (C_u): Profit lost for each unit of demand not met

    $$C_u = P - C = 3 - 1 = 2$$

    - Overage Cost (C_o): This is the cost of holding an unsold newspaper.

    $$C_o = C - S = 1 - 0 = 1$$

3. Calculate Critical Ratio (CR)

$$CR = \frac{C_u}{C_u - C_o} = \frac{2}{2+1} = \frac{2}{3} = 0.67$$

4. Find z-score: Find the number of standard deviations away from the mean corresponding to the critical ratio:

$$z^* = \Phi^{-1}(CR) = 0.44$$

5. Calculate Optimal Order Quantity (Q^*)

$$Q^* = \mu + z^*\sigma = 100 + 0.44 \cdot 20 = 108.8$$

newsvendor.py

```
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

# Parameters for the newsvendor example
selling_price = 3  # Selling price per newspaper
purchase_cost = 1   # Purchase cost per newspaper
unsold_value = 0    # Value of unsold newspapers
mu = 100 # Mean of demand
sigma = 20 # Standard deviation of demand

# Calculate underage and overage costs
C_u = selling_price - purchase_cost  # Underage cost
C_o = purchase_cost - unsold_value    # Overage cost

# Calculate the critical ratio
CR = C_u / (C_u + C_o)

# Assume a normal distribution for demand
mu = 50    # Mean demand
sigma = 10 # Standard deviation of demand

# Calculate the optimal order quantity (Q*)
z_star = stats.norm.ppf(CR)  # z-score corresponding to the critical ratio
Q_star = mu + z_star * sigma  # Optimal order quantity
```

31.2. ABC Analysis

categorizes inventory items into three groups (A, B, and C) based on their importance

- A: Top 70-80% of the total annual consumption value
- B: Next 15-25% of the total annual consumption value
- C: Remaining 5-10% of the total annual consumption value

Example

Item	Usage Quantity (Annual)	Unit Cost	Conumption Value (Annual)
I_1	50	\$100	\$5000
I_2	150	\$20	\$3000
I_3	300	\$10	\$3000

Item	Usage Quantity (Annual)	Unit Cost	Conumption Value (Annual)
I_4	400	$5	$2000
I_5	500	$1	$500

Step 1: Calculate Annual Consumption Values

Step 2: Sort Items by Annual Consumption Value (Descending)

Step 3: Calculate Total Annual Consumption Value

$$\text{Total} = \$5,000(I_1) + \$3,000(I_2) + \$3,000(I_3) + \$2,000(I_4) + \$500(I_5) = \$13500$$

Step 4: Calculate Cumulative Consumption Value Percentages

- $I_1 : \frac{5000}{13500} \times 100\% = 37.04\%$
- $I_2 : \frac{3000}{13500} \times 100\% = 22.22\%$
- $I_3 : \frac{3000}{13500} \times 100\% = 22.22\%$
- $I_4 : \frac{2000}{13500} \times 100\% = 14.81\%$
- $I_5 : \frac{500}{13500} \times 100\% = 3.7\%$

Step 5: Cumulative Percentages

- I_1: 37.04%
- $I_1 + I_2$: 59.26%
- $I_1 + I_2 + I_3$: 81.48%
- $I_1 + I_2 + I_3 + I_4$: 96.3%
- $I_1 + I_2 + I_3 + I_4 + I_5$: 100%

Step 6: Categorize Items

- A: I_1, I_2, I_3
- B: I_4
- C: I_5

31.3. Fill Rate

Percentage of customer demand that is satisfied from available inventory

$$F = \frac{U_f}{U_o} \times 100\%$$

Where:

- F: Fill Rate
- U_f: Number of units fulfilled
- U_o: Total number of units ordered

Example

Customers order 100 units of a product and 90 units are fulfilled from stock:

$$F = \frac{90}{100} \times 100\% = 90\%$$

31.4. (OCT) Order Cycle Time

Measures the total time taken from when a customer places an order to when the order is delivered

$$\text{OCT} = T_{\text{order}} + T_{\text{processing}} + T_{\text{production}} + T_{\text{shipping}}$$

Where:

- T_{order}: Order Entry Time (time it takes to receive and log the order)
- $T_{\text{processing}}$: Order Processing Time (time to check inventory, verify details, and prepare for production or shipment)
- $T_{\text{production}}$: Production Time (time to manufacture or prepare the product)
- T_{shipping}: Shipping Time (time it takes to deliver the product from the warehouse to the customer)

31.5. ROP (Reorder Point)

The inventory level at which a new order should be placed to avoid stockouts

$$\text{ROP} = (\text{Average Demand per Period} \times \text{Lead Time})$$

Example

Suppose your business sells 50 units per week, and the lead time for a new order is 2 weeks. Using the formula:

$$\text{ROP} = 50(\text{units per week}) \times 2(\text{weeks}) = 100 \text{ units}$$

This means that when your inventory level drops to 100 units, you should place a new order to avoid running out of stock.

31.6. XYZ Analysis

Categorization based on variability

- X: Low variability
- Y: Moderate variability
- Z: High variability

Coefficient of Variation:

$$CV = \frac{\sigma}{\mu}$$

- X Items: Low CV
  - CV < k_1, where k_1 is the threshold value indicating low variability
- Y Items: Moderate CV
  - CV < k_2, where k_2 is the moderate value indicating low variability
- Z Items: High CV
  - CV < k_3, where k_3 is the threshold value indicating high variability

Example

Step 1: Collect Historical D../ata (12 months)

- Product A: [100, 105, 98, 102, 101, 104, 103, 100, 99, 100, 101, 102]
- Product B: [150, 155, 145, 160, 140, 150, 155, 150, 165, 155, 150, 140]
- Product C: [200, 180, 220, 190, 210, 240, 180, 230, 220, 210, 250, 190]

Step 2: Calculate the Mean and Standard Deviation

- Product A:
  - $\mu = 101$
  - $\sigma = 2$
- Product B:
  - $\mu = 150$
  - $\sigma = 8$
- Product C:
  - $\mu = 210$
  - $\sigma = 25$

Step 3: Calculate the Coefficient of Variation (CV)

- $CV = \frac{2}{101} = 0.0198$
- $CV = \frac{8}{150} = 0.0533$
- $CV = \frac{25}{210} = 0.1190$

Step 4: Categorization

- X: Product A (Low variability)
- Y: Product B (Moderate variability)
- Z: Product C (High variability)

31.7. EOQ (Economic Order Quantity)

Optimal order quantity that **minimizes the total cost**, which includes both **holding costs** and **ordering costs**

$$\mathrm{EOQ} = \sqrt{\frac{2DS}{H}}$$

Where:

- D: Demand per time period
- S: Ordering cost per order
- H: Holding cost per unit, per time period

Example

A company sells widgets and wants to determine the optimal order quantity for inventory.

- The annual demand for widgets (D) is 12,000 units.
- The cost to place an order (S) is $50.

The holding cost per unit per year (H) is $2.

$$\text{EOQ} = \sqrt{\frac{2 \times 12000 \times 50}{2}} = 775$$

The company should order 775 widgets each time they place an order to minimize the total cost, which includes both ordering and holding costs

31.7.1. Perfect Order Rate

Measures the percentage of orders delivered to customers in full, on time, and without any damage

$$\text{Perfect Order Rate} = \frac{\text{Number of Perfect Orders}}{\text{Total Number of Orders}} \times 100\%$$

Where:

- **Number of Perfect Orders**: The number of orders that are delivered on time, complete, and undamaged.
- **Total Number of Orders**: The total number of orders fulfilled within a specific period.

Example

Suppose you received 1,000 orders over a quarter, and 900 of those orders were delivered on time, complete, and without damage.

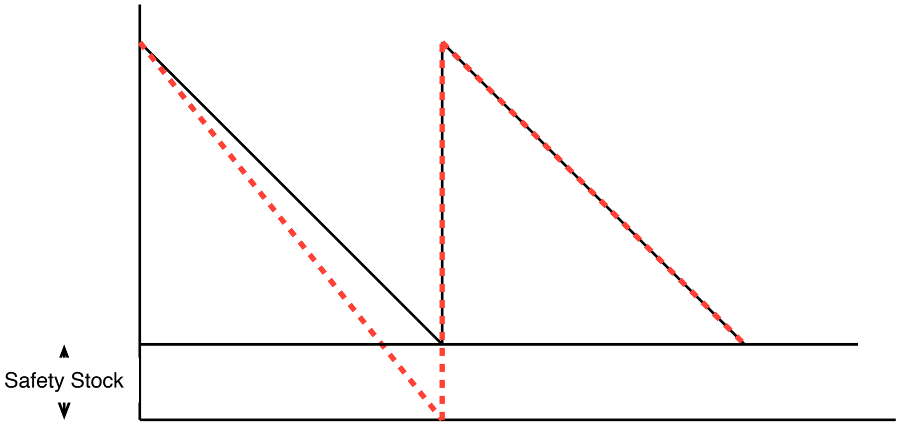
$$\text{Perfect Order Rate} = \frac{900}{1000} \times 100\% = 90\%$$

31.8. Safety Stock

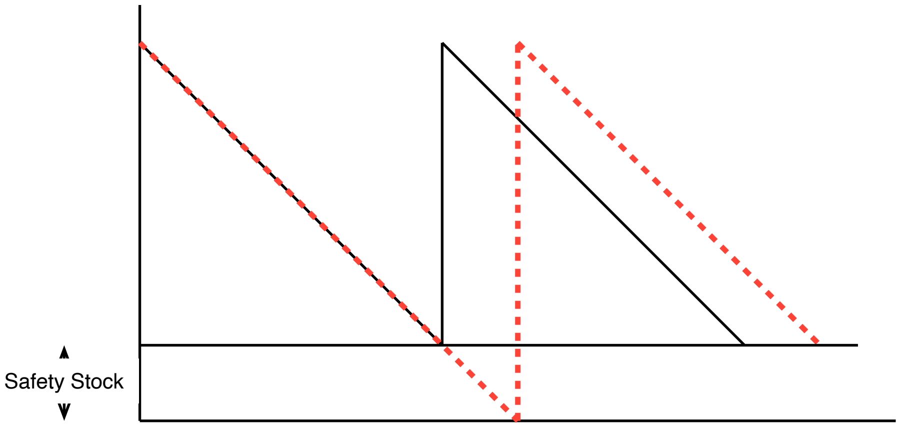
Additional quantity of inventory kept on hand to protect against uncertainties in demand or supply. Buffer to prevent stockouts due to unexpected variations in demand or delays in supply.

2 reasons to keep safety stock:

- Variation in Demand



- Variation in Lead Time



1. Constant Demand & Constant Lead Time

$$\mathrm{SS} = Z \times \sigma_D$$

Where:

- Z: Z-score corresponding to the desired service level
- σ: standard deviation of demand

2. Variable Demand & Constant Lead Time

$$\mathrm{SS} = Z \times \sigma_D \times \sqrt{L}$$

Where:

- Z: Z-score corresponding to the desired service level
- σ_D: Standard deviation of demand per unit of time
- L: Lead time

3. Variable Lead Time & Constant Demand

$$\mathrm{SS} = Z \times \bar{D} \times \sigma_L$$

Where:

- Z: Z-score corresponding to the desired service level
- $\bar{D}$: Average demand
- σ_L: Standard deviation of lead time

4. Variable Demand & Variable Lead Time

$$\text{SS} = Z \times \sqrt{\left(\bar{D}^2 \times \sigma_L^2\right) + \left(L \times \sigma_D^2\right)}$$

Where:

- $\bar{L}$: Average Lead Time (average time it takes to receive inventory after placing an order)
- σ_D^2: Demand Variance (variability in demand during the lead time)
- $\bar{D}$: Average Demand (mean quantity of demand per time period)
- σ_L^2: Lead Time Variance (variability in lead time)

31.9. Risk Pooling

Standard deviation does not scale linearly

Example

- Total Demand D: 100 units / day
- Standard Deviation Demand σ_{total}: 20 units / day
- Lead Time L: 1 day
- Service Level z: 1.65 ($\approx 95\%$)

Case 1: Single Warehouse

One warehouse serves all customers

$$\begin{aligned} \text{SS} &= z \cdot \sigma_{\text{total}} \cdot \sqrt{L} \\ &= 1.65 \cdot 20 \cdot \sqrt{1} \\ &= 33 \text{ units} \end{aligned}$$

Case 2: Four warehouses

Customers are split evenly across 4 warehouses

$$\sigma_{\text{per\_warehouse}} = \frac{\sigma_{\text{total}}}{\sqrt{4}} = \frac{20}{2} = 10$$

Safety stock per warehouse:

$$\begin{aligned} \text{SS}_{\text{per\_warehouse}} &= z \cdot \sigma_{\text{per\_warehouse}} \cdot \sqrt{L} \\ &= 1.65 \cdot 10 \cdot \sqrt{1} \\ &= 16.5 \end{aligned}$$

Total safety stock across 4 warehouses:

$$\text{SS}_{\text{total}} = 4 \cdot 16.5 = 66 \text{ units}$$

32. Causes of variation

32.1. Common

Common Cause Variation: Normal, expected variation inherent to the process; predictable and stable.

Example

- Temperature
- Humidity
- Haterial properties

32.2. Special

Special Cause Variation: Abnormal, unexpected variation due to specific causes; unpredictable and requires immediate action.

External disruptions

Example

- Machine malfunction
- Faulty raw materials
- Staff shortages
- Weather events

33. Algorithms

34. Shortest Path

34.1. Dijkstra’s

Step 1: Initialize

- Mark all nodes as unvisited
- Set the distance to the source as 0
- Set the distance to all other nodes as ∞
- Create a priority queue (or a simple list) to keep track of unvisited nodes and their distances.

Example

Graph nodes: A, B, C, D

Start node: A

Distances:

- $A = 0$
- $B = \infty$
- $C = \infty$
- $D = \infty$

Step 2: Select the Unvisited Node with the Smallest Distance

Among all unvisited nodes, pick the one with the smallest known distance. Let’s call this the current node.

Step 3: Update Neighboring Distances

For each neighbor of the current node:

1. Calculate the tentative distance:

   distance(current) + weight(current $\rightarrow$ neighbor)

2. If this tentative distance is less than the known distance, update it.

Example

If $A \longrightarrow B$ has weight 4 and distance to A is 0:

Tentative distance to B = 0 + 4 = 4

If current distance to B is ∞, update to 4

Step 4: Mark the Current Node as Visited

- Once all neighbors are checked, mark the current node as visited
- Visited nodes are not checked again

Step 5: Repeat Until All Nodes Are Visited

- Go back to Step 2 and repeat.
- Stop when all nodes have been visited or the smallest tentative distance among the unvisited nodes is ∞ (meaning unreachable)

Step 6: Reconstruct the Shortest Paths (Optional)

- To find the actual shortest path, track the previous node for each visited node
- Starting from the destination, backtrack through the previous nodes

Example

35. Organisations

35.0.1. Actor-Network Theory (ANT)

35.0.2. Bounded Rationality

35.1. Bureaucratic Theory

35.2. Embeddedness

35.3. Functional Theory of Stratification

$$R \propto f(S, I)$$

Where:

- R: reward
- S: scarcity, how many people are qualified to perform a role
- I: importance, how critical the role is to the functioning of society

35.4. Institutional Theory

35.4.1. World-Systems Theory

(Wallerstein)

35.4.2. Global Commodity Chains (GCC)

- Producer-driven (autos, electronics)
- Buyer-driven (apparel, retail)

35.4.3. Global Value Chains (GVC)

Governance types:

- market
- modular
- relational
- captive
- hierarchical

35.4.4. Global Production Networks (GPN)

35.4.5. World-Systems Theory

(Wallerstein)

35.4.6. Global Commodity Chains (GCC)

- Producer-driven (autos, electronics)
- Buyer-driven (apparel, retail)

35.4.7. Global Value Chains (GVC)

Governance types:

- market
- modular
- relational
- captive
- hierarchical

35.4.8. Global Production Networks (GPN)

35.5. Resource Dependency Theory (RDT)

$$P_{A \to B} \propto D_{B \to A}$$

Where:

- P: power
- D: dependence

35.6. Scientific Management

- Work Study (Time and Motion Analysis) “One best way”
- Scientific Selection and Training of Workers
- Division of Labor
- Standardization: Tools and procedures
- Performance-Based Incentives

35.6.1. Transaction Costs

Why do firms / organizations exist? Why not many atomized individuals (market)?

Why are there hierarchies (firms / organizations) in markets?

Hierarchy or Market? Make or Buy? Which transactions should be included in the hierarchy, which transactions should be left to the market?

Unit of analysis: Transaction Relation (networks)

Traditional assumption: Using the market has no cost (frictionless)

If trust there would be no cost But the risk of opportunism, shirking exists

Therefore, there are costs:

- Search
- Bargaining
- Monitoring / Enforcement: ensuring that the terms of the agreement are being fulfilled (inspection, quality control, and supervision)
- Information: expenses incurred in acquiring, processing, and disseminating information relevant to the transaction
- Coordination
- Adaptation

Costs are a function of:

- Asset Specificity
- Uncertainty
- Frequency

Efficient alignment hypothesis: Transactions, which vary in their characteristics, should be matched with governance structures that differ in their costs and competencies. Alignment aims to minimize transactions costs.

Questions:

- For you which transactions should be included in the hierarchy and which should be left to the market?
- Furthermore, which transactions included in hierarchies should be in public hierarchies and which in private hierarchies?
- How can this transaction sort approaches be used to predict end symbiosis?

Let the set of transcations be $T = \{1, 2, ..., n\}$ For each transaction $i \in T$, define:

- C_i^M: transaction cost if done via **Market**
- C_i^H: transaction cost if done via **Hierarchy**

Then the total cost accross all transactions is:

$$TC = \sum_{i=1}^{n} \min\{C_i^M, C_i^H\}$$

We want to choose the governance $g_i \in \{M, H\}$ for each i that minimized total transaction cost

Transaction costs can be modeled as a function of key characteristics

$$C_i^g = f^{g(S_i, U_i, F_i)}$$

Where:

- S_i: asset specificity
- U_i: Uncertainty
- F_i: frequency
- g: governance structure (M, H)

Assumptions

$$C_i^M = \alpha_M + \beta_M S_i + \gamma_M U_i + \delta_M F_i$$
$$C_i^H = \alpha_H + \beta_H S_i + \gamma_H U_i + \delta_H F_i$$

Where coefficients reflect **how sensitive each governance mode is to specificity, uncertainty, and frequency**. For instance:

- Market is cheap for low specificity, low uncertainty, low frequency
- Hierarchy becomes cheaper as specificity, uncertainty, or frequency increase

Then the optimal governance for transaction i is:

$$g_i^* = \arg \min_{g \in \{M, H\}} C_i^g(S_i, U_i, F_i)$$

So, for each transaction, compute the three potential costs and pick the governance mode with the lowest cost

Symbiosis prediction

Symbiosis arises when two transactions are co-dependent and jointly internalized in the hierarchy:

- Let i and j be linked transactions
- Joint transaction cost:

$$C_{ij}^{\text{joint}} < C_i^M + C_j^M$$

If this inequality holds, hierarchy creates a **net benefit**, predicting **stable interdependence**, i.e., symbiosis

35.7. Organizational Ecology

36. White Paper

(Project Cybersyn/Synco, Project Red Book, OGAS Project)

- Publicly owned
- Democratically controlled
- Environmentally sustainable
- Fully automated
- Transparent

Supply chain guaranteeing a predefined set of goods and services to all members of society

France is split into 5 strictly nested administrative units:

- National ($\mathcal{N}$)
- Regions ($\mathcal{R}$)
- Departements ($\mathcal{D}$)
- Communes ($\mathcal{C}$)
- IRIS ($\mathcal{I}$)

Objectives:

1. Facility Location
   - **Distribution Centers**: Determine optimal locations of distribution centers at the National, Regional, Departement, Commune, and IRIS levels, subject to demographic demand and geographic accessibility.
   - **Government Stores**: Determine optimal placement of government stores such that all households have equitable access to essential goods.
2. Flow Allocation
   - **Supply Allocation**: Determine how much of each good should be allocated from supply nodes (imports, extraction, or domestic manufacturing plants) to each level of distribution.
   - **Demand Fulfillment**: Ensure that each demand node receives sufficient quantities of goods to meet population needs, respecting fairness and prioritization rules (e.g., hospitals and schools may receive priority allocations).
   - **Reverse Flows**: Incorporate waste recovery, recycling, and reverse logistics so that outputs from demand nodes (waste, byproducts, returns) are routed to processing or extraction nodes when feasible.
3. Systemic Objectives
   - **Equity**: Guarantee universal access across all IRIS units, reducing geographic disparities.
   - **Resilience**: Provide redundancy and flexibility in flows to handle shocks (e.g., disruptions to imports or local production).
   - **Sustainability**: Minimize environmental impact by favoring local production, renewable inputs, and circular waste loops.
   - **Efficiency**: Optimize flows to minimize costs, transport distances, and resource usage, while meeting fairness and sustainability constraints.
   - **Transparency** and **Participation**: Make allocation decisions visible, accountable, and open to democratic oversight at all administrative levels.

36.1. Administrative Hierarchy and Demographics

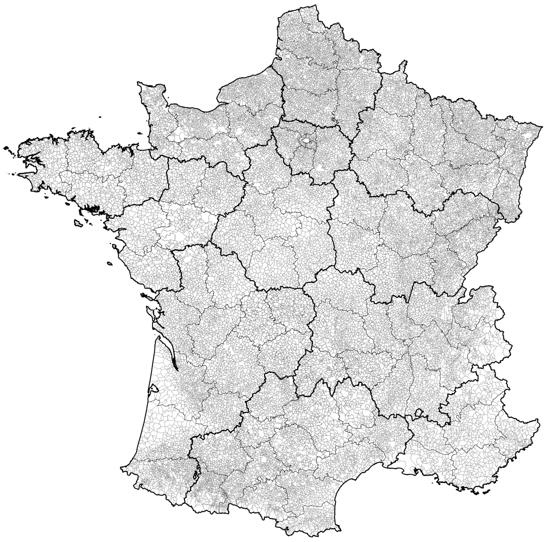


Figure 7: Administrative Hierarchy

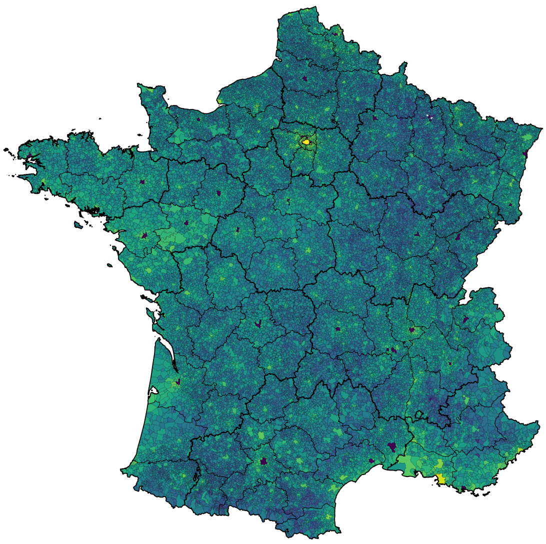


Figure 8: Commune Population Map

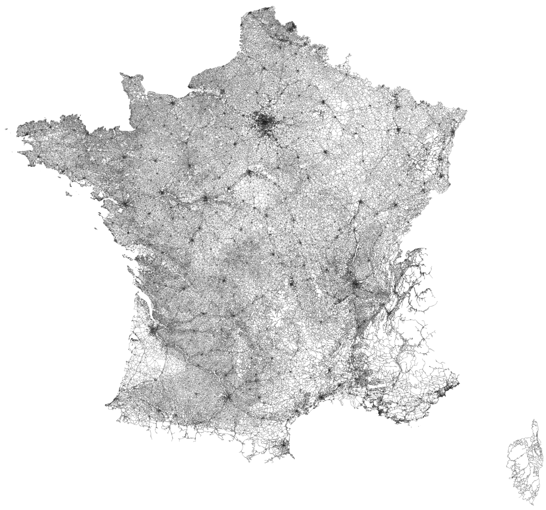


Figure 9: Roads

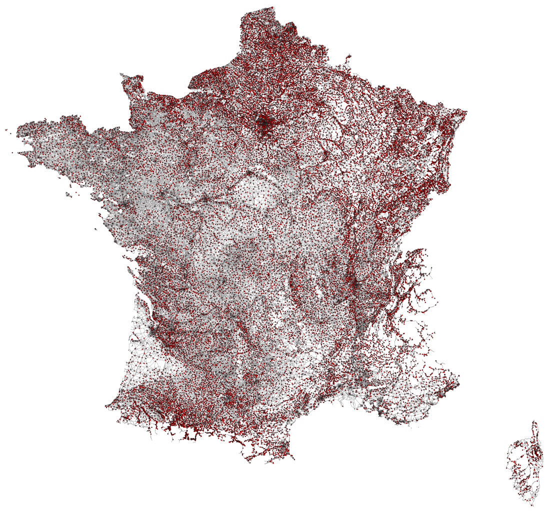


Figure 10: Roads & Commune Hubs

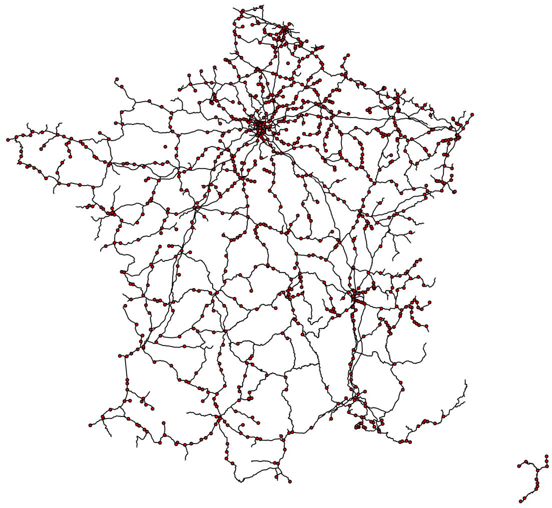


Figure 11: Train Network and Hubs

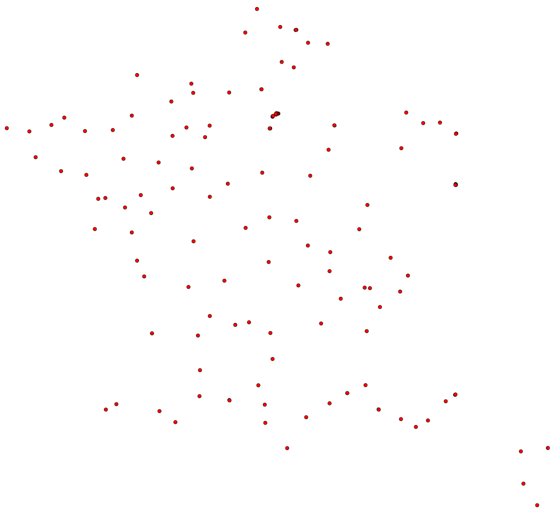


Figure 12: Aeronatic Hubs

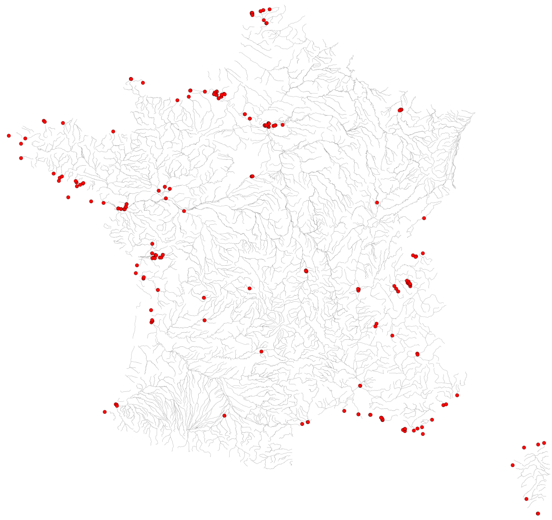


Figure 13: Port Hubs

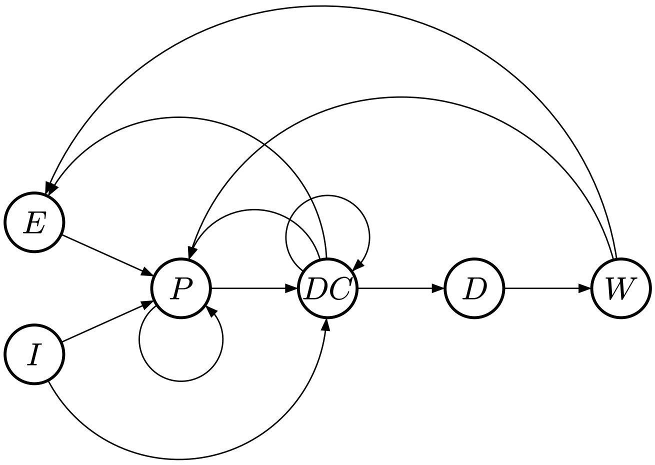
36.2. Schema

Nodes

Category	Type	Attributes	Description	Code
Extraction				
Supply	Import Facility			
	Manufacturing Plant			
Intermediary	National Distribution Center (NDC)			`NDC_`
	Regional Distribution Center (RDC)			`RDC_`
	Department Distribution Center (DDC)			`DDC_`

	Commune Distribution Center (CDC)			CDC_
Demand	Store			
	Household			
	School			
	Hospital			
Waste				

Mode of Transport	Type	Attributes	Desciption	Code
Road	Highway National Local			
Rail	Freight Regional			
Air	Flight Path Cargo Route			
Martime	Shipping Coastal			
Inland Waterway	River Canal			



Raw materials are obtained either from Extraction Nodes (E) or imported through Import Nodes (I). They are then sent to Processing Nodes (P), where they undergo one or more processing stages. The resulting products are delivered to Distribution Nodes (DC).

Distribution Nodes can also receive finished products directly from Import Nodes. They may transfer goods to other Distribution Nodes (at different administrative levels), directly to Demand Nodes (D), or back to Processing and Extraction Nodes—for example, when machinery, spare parts, or energy carriers are required to sustain those activities.

Once products are consumed at Demand Nodes, they generate outputs that flow into Waste Nodes (W). Waste Nodes can (1) dispose of waste, (2) redirect it to Processing Nodes for recovery or recycling, or (3) return it to Extraction Nodes through resource recovery activities such as landfill mining or biogas capture.

Data

IRIS	• https://geoservices.ign.fr/contoursiris • https://geoservices.ign.fr/irisge
Adminitrative Units	• https://geodatafr.github.io/IGN/ADMIN_EXPRESS_Administrative_boundaries/
Roads	• https://transport.data.gouv.fr/datasets/route-500-r
Rail	• https://www.data.gouv.fr/datasets/fichier-de-formes-des-lignes-du-reseau-ferre-national • https://ressources.data.sncf.com/explore/dataset/formes-des-lignes-du-rfn/information/
Train Stations	• https://transport.data.gouv.fr/datasets/liste-des-gares?locale=en
Martime Ports	• https://www.data.gouv.fr/datasets/ports-espace-maritime-francais/ • https://data.opendatasoft.com/explore/dataset/caracteristiques-des-ports-en-france-2021%40pndb/
Inland Ports	
Aeroports	• https://www.data.gouv.fr/datasets/aeroports-francais-coordonnees-geographiques/

Stack

Safe Rust

DataBase	• PSQL (`tokio_postgres`, `sqlx`, `bb8`) • ScyllaDB (`scylla`)
API	• gRPC (`tonic`) • REST (`axum`)
Front End	• `dioxus` • `leptos` • `bevy`

Bibliography