

# Types of Linear Systems

# Types of Linear Systems

We consider a linear system written in matrix form:

$$Ac = y,$$

where:

- $A \in \mathbb{R}^{m \times n}$ ,
- $m =$  number of equations,
- $n =$  number of unknowns.

# Types of Linear Systems

We consider a linear system written in matrix form:

$$Ac = y,$$

where:

- $A \in \mathbb{R}^{m \times n}$ ,
- $m$  = number of equations,
- $n$  = number of unknowns.

## Key idea

The relation between  $m$  and  $n$  fundamentally changes the nature of the problem.

## Square Full-Rank Systems ( $m = n$ )

The system is:

- **square:**  $A \in \mathbb{R}^{n \times n}$ ,
- **full-rank:** columns are linearly independent,
- equivalently:  $\det(A) \neq 0$ .

## Square Full-Rank Systems ( $m = n$ )

The system is:

- **square:**  $A \in \mathbb{R}^{n \times n}$ ,
- **full-rank:** columns are linearly independent,
- equivalently:  $\det(A) \neq 0$ .

### Consequences

- exactly one solution,
- stable and well-defined algebraic structure.

## Square Full-Rank Systems ( $m = n$ )

The system is:

- **square:**  $A \in \mathbb{R}^{n \times n}$ ,
- **full-rank:** columns are linearly independent,
- equivalently:  $\det(A) \neq 0$ .

### Consequences

- exactly one solution,
  - stable and well-defined algebraic structure.
- 
- Each equation adds an independent constraint
  - Geometrically: hyperplanes intersect at a single point

# Solving Square Systems

For square full-rank systems, we can solve exactly using:

- LU decomposition,
- QR decomposition.

# Solving Square Systems

For square full-rank systems, we can solve exactly using:

- LU decomposition,
- QR decomposition.

## Remark

These methods are efficient and numerically stable for well-conditioned matrices.

## Overdetermined Systems ( $m > n$ )

A system is **overdetermined** if:

$$m > n$$

## Overdetermined Systems ( $m > n$ )

A system is **overdetermined** if:

$$m > n$$

This situation is very common in experimental sciences:

- many measurements,
- few unknown parameters,
- noisy or redundant constraints.

## Overdetermined Systems ( $m > n$ )

A system is **overdetermined** if:

$$m > n$$

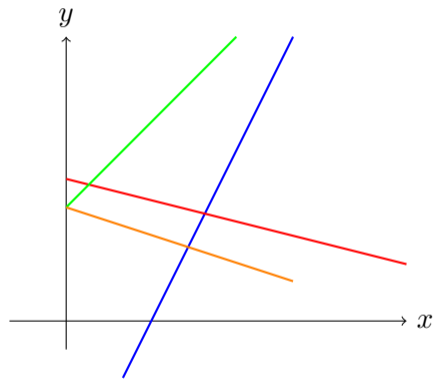
This situation is very common in experimental sciences:

- many measurements,
- few unknown parameters,
- noisy or redundant constraints.

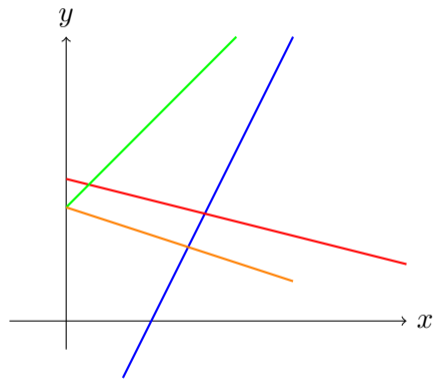
### Key consequence

In general, **no exact solution** satisfies all equations.

## Geometric Interpretation: Overdetermined Case



## Geometric Interpretation: Overdetermined Case



### Interpretation

More equations than variables  $\Rightarrow$  no common intersection point.

## Least-Squares Solution

Instead of solving  $Ac = y$  exactly, we look for:

$$c^* = \arg \min_c \|Ac - y\|^2$$

## Least-Squares Solution

Instead of solving  $Ac = y$  exactly, we look for:

$$c^* = \arg \min_c \|Ac - y\|^2$$

- minimizes the total squared error,
- provides the best approximate solution,
- corresponds to an orthogonal projection.

## Least-Squares Solution

Instead of solving  $Ac = y$  exactly, we look for:

$$c^* = \arg \min_c \|Ac - y\|^2$$

- minimizes the total squared error,
- provides the best approximate solution,
- corresponds to an orthogonal projection.

### Tool

Solved using QR decomposition or normal equations.

## Underdetermined Systems ( $m < n$ )

A system is **underdetermined** if:

$$m < n$$

## Underdetermined Systems ( $m < n$ )

A system is **underdetermined** if:

$$m < n$$

Key property

The system has an **infinite number of solutions**.

## Underdetermined Systems ( $m < n$ )

A system is **underdetermined** if:

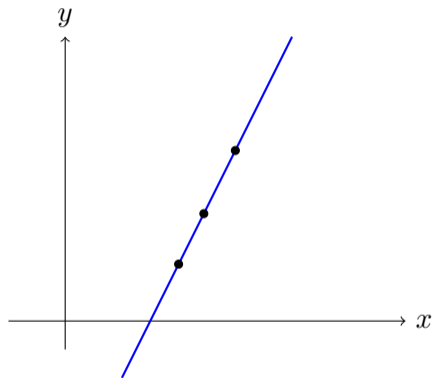
$$m < n$$

### Key property

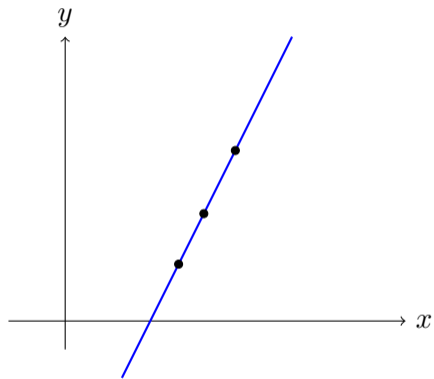
The system has an **infinite number of solutions**.

- not enough information,
- many solutions satisfy the equations.

## Geometric Interpretation: Underdetermined Case



## Geometric Interpretation: Underdetermined Case



### Interpretation

The solution is a whole subspace, not a single point.

# Regularization for Underdetermined Systems

To select a meaningful solution, we add extra constraints.

## Regularization for Underdetermined Systems

To select a meaningful solution, we add extra constraints.

Ridge Regression (L2)

$$c^* = \arg \min_c \|Ac - y\|^2 + \lambda \|c\|_2^2$$

Chooses the **minimum-norm** solution.

## Regularization for Underdetermined Systems

To select a meaningful solution, we add extra constraints.

### Ridge Regression (L2)

$$c^* = \arg \min_c \|Ac - y\|^2 + \lambda \|c\|_2^2$$

Chooses the **minimum-norm** solution.

### LASSO (L1)

$$c^* = \arg \min_c \|Ac - y\|^2 + \lambda \|c\|_1$$

Promotes **sparse** solutions.

# Why This Matters

Underdetermined systems are common in:

- genomics,
- transcriptomics,
- neuroscience,
- high-dimensional regression.

## Why This Matters

Underdetermined systems are common in:

- genomics,
- transcriptomics,
- neuroscience,
- high-dimensional regression.

Regularization is essential to:

- stabilize solutions,
- avoid overfitting,
- improve interpretability.

## Summary

| <b>System Type</b>  | <b>Properties</b>         | <b>Method</b>  |
|---------------------|---------------------------|----------------|
| $m = n$ , full-rank | Unique solution           | LU / QR        |
| $m > n$             | No exact solution         | Least squares  |
| $m < n$             | Infinitely many solutions | Regularization |

## Summary

| <b>System Type</b>  | <b>Properties</b>         | <b>Method</b>  |
|---------------------|---------------------------|----------------|
| $m = n$ , full-rank | Unique solution           | LU / QR        |
| $m > n$             | No exact solution         | Least squares  |
| $m < n$             | Infinitely many solutions | Regularization |

### Take-home message

The shape of  $A$  determines the nature of the solution and the appropriate numerical method.