1 Basic terminology and examples

A **linear equation in n unknowns** is an equation of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b\,,$$

where $a_1, \ldots, a_n$ and b are given real numbers and $x_1, \ldots, x_n$ are variables.

A **system of m linear equations in n unknowns** is a collection of equations of the form

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & = & b_m \end{array}$$

where the a_{ij}'s and b_i's are all real numbers. We also call such systems $m \times n$ **systems**.

Example 1.1.

$$(a)\ \begin{array}{rcrcl} 2x_1 & + & x_2 & = & 4 \\ 3x_1 & + & 2x_2 & = & 7 \end{array} \qquad (b)\ \begin{array}{rcrcrcl} x_1 & + & x_2 & - & x_3 & = & 3 \\ 2x_1 & - & x_2 & + & x_3 & = & 6 \end{array} \qquad (c)\ \begin{array}{rcrcl} x_1 & - & x_2 & = & 0 \\ x_1 & + & x_2 & = & 3 \\ & & x_2 & = & 1 \end{array}$$

A *solution* of an $m \times n$ system is an ordered n-tuple $(x_1, x_2, \ldots, x_n)$ that satisfies all equations of the system.

The set of all solutions of a system is called its **solution set**, which may be empty, finite or infinite.

Given an $m \times n$ linear system

$$\begin{array}{rcl} a_{11}x_1 + \cdots + a_{1n}x_n & = & b_1 \\ \vdots & & \\ a_{m1}x_1 + \cdots + a_{mn}x_n & = & b_m \end{array}$$

we call the array

$$\left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{array}\right)$$

the **augmented matrix** of the linear system, and the $m \times n$ matrix

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

the **coefficient matrix** of the linear system.

2 Solving a system of linear equations

The following elementary operations do not change the solution set of a system of linear equations:

(i) interchanging two equations;

(ii) multiplying an equation by a non-zero scalar;

(iii) adding a multiple of one equation to another.

A system can be solved by performing operations on the augmented matrix. We have the following three operations that can be applied to the augmented matrix, called **elementary row operations**.

Definition 2.1 (Elementary row operations)**.**

Type I	interchanging two rows;
Type II	multiplying a row by a non-zero scalar;
Type III	adding a multiple of one row to another row.

3 Gaussian elimination

Definition 3.1. A matrix is said to be in *row echelon form* if it satisfies the following three conditions:

(i) All zero rows (consisting entirely of zeros) are at the bottom.

(ii) The first non-zero entry from the left in each nonzero row is a 1, called the *leading* 1 for that row.

(iii) Each leading 1 is to the right of all leading 1's in the rows above it.

A row echelon matrix is said to be in *reduced row echelon form* if, in addition it satisfies the following condition:

(iv) Each leading 1 is the only nonzero entry in its column

Gaussian algorithm:

Step 1 If the matrix consists entirely of zeros, stop — it is already in row echelon form.

Step 2 Otherwise, find the the first column from the left containing a non-zero entry (call it a), and move the row containing that entry to the top position.

Step 3 Now multiply that row by $1/a$ to create a leading 1.

Step 4 By subtracting multiples of that row from rows below it, make each entry below the leading 1 zero.

This completes the first row. All further operations are carried out on the other rows.

Step 5 Repeat steps 1-4 on the matrix consisting of the remaining rows

The process stops when either no rows remain at Step 5 or the remaining rows consist of zeros.

A variant of the Gauss algorithm is the Gauss-Jordan algorithm, which brings a matrix to reduced row echelon form:

Gauss-Jordan algorithm

Step 1 Bring matrix to row echelon form using the Gaussian algorithm.

Step 2 Find the row containing the first leading 1 from the right, and add suitable multiples of this row to the rows above it to make each entry above the leading 1 zero.

This completes the first non-zero row from the bottom. All further operations are carried out on the rows above it.

Step 3 Repeat steps 1-2 on the matrix consisting of the remaining rows.