

## 2 Systems of Linear Equations

A system of equations of the form

$$\begin{array}{rcl} x & + & 2y = 7 \\ 2x & - & y = 4 \end{array}$$

or

$$\begin{array}{rcl} 5p & - & 6q + r = 4 \\ 2p & + & 3q - 5r = 7 \\ 6p & - & q + 4r = -2 \end{array}$$

is called a **system of linear equations**.

**Definition.** An equation involving certain variables is **linear** if each side of the equation is a sum of **constants** and **constant multiples of the variables**.

**Examples.** The equations  $2x = 3y + \sqrt{3}$ ,  $3 = 0$  and  $x + y + z = 0$  are linear.

**Non-examples.** The equations  $y = x^2$  and  $3xy + z + 7 = 2$  are **not** linear since they involve **products** of variables.

### 2.1 Solutions to Systems of Equations

A **solution** to a system of equations is a way of assigning a value to each variable, so as to make all the equations true at once.

**Example.** The system of equations ...

$$\begin{array}{rcl} 4x & - & 2y = 22 \\ 9x & + & 3y = 12 \end{array}$$

... has a solution  $x = 3$  and  $y = -5$ . (In fact in this case this is the **only** solution.)

**Remark.** In general, a system of linear equations can have **no** solutions, a **unique** solution, or **infinitely many** solutions. It is **not** possible for the system to have, for example, exactly 2 solutions; this is a higher dimensional generalisation of the fact that 2 straight lines cannot meet in exactly 2 points (recall Exercise Sheet 0).

**Definition.** The set of all solutions is called the **solution set** or **solution space** of the system.

**Warning.** When we talk about **solving** a system of equations, we mean describing **all** the solutions (or saying that there are none, if the solution set is empty), not just finding one possible solution!

### 2.2 2-dimensional Geometry of Linear Equations

What can solution spaces look like? Consider a linear equation in two variables  $x$  and  $y$ , say:

$$3x + 4y = 4.$$

Each solution consists of an  $x$ -value and a  $y$ -value. We can think of these as the coordinates of a point in 2-dimensional space. For example, a solution to the above equation is  $x = -4$  and  $y = 4$ . This solution gives the point  $(-4, 4)$ .

In general, the set of solutions to a linear equation with variables  $x$  and  $y$  forms a **line** in 2-dimensional space. So each equation corresponds to a line — we say that the equation **defines** the line.

**Systems of Equations.** Now suppose we have a **system** of two such equations. A solution to the system means an  $x$ -value and a  $y$ -value which solve **all the equations at once**. In other words, a solution

corresponds to a point which lies on **all** the lines. So in geometric terms, the solution set to a system of linear equations with variables  $x$  and  $y$  is an **intersection of lines** in the plane.

**Question.** Recall from Exercise Sheet 0 what an intersection of two lines in the plane can look like:

- Usually<sup>4</sup> it is a **single point**. This is why 2 equations in 2 variables usually have a unique solution.
- Alternatively, the lines could be distinct but parallel. Then the intersection is the **empty set**. This is why 2 equations in 2 variables can sometimes have no solutions.
- Or then again, the two lines could actually be the same. In this case the intersection is the **whole line**. This is why 2 equations in 2 variables can sometimes have infinitely many solutions.

Similarly, if we have a system of **more than 2** equations, we are looking for the intersection of all the lines (in other words, the set of points in the plane which lie on all the lines). The intersection of 3 lines is “usually” empty but could also be a single point or a line.

### 2.3 Higher Dimensional Geometry of Linear Equations

Now suppose we have an equation in the variables  $x$ ,  $y$  and  $z$ , say

$$3x + 4y + 7z = 4.$$

This time we can consider a solution as a point in 3-dimensional space. The set of all solutions (assuming there are some) forms a **plane** in space. The solution set for a **system** of  $k$  linear equations will be an **intersection of  $k$  planes**.

**Exercise.** What can the intersection of 2 planes in 3-space look like? Try to imagine all the possibilities, as we did for lines in the previous section. How do they correspond to the possible forms of solution sets of 2 equations with 3 variables?

**Exercise.** Now try to do the same for intersections of 3 planes. (Recall that the intersection of three sets is the set of points which lie in all three). How do the possible geometric things you get correspond to the possible forms of solution sets for 3 equations with 3 variables?

In yet higher dimensions, the solution set to a linear equation in  $n$  variables is a copy of  $(n - 1)$ -dimensional space inside  $n$ -dimensional space. Such a set is called a **hyperplane**. So geometrically, the solution set of  $k$  equations with  $n$  variables will be an **intersection of  $k$  hyperplanes** in  $n$ -dimensional space.

**Remark.** Actually it is not **quite** true to say that every equation defines a hyperplane. Equations like “ $0 = 0$ ” and “ $2 = 2$ ” define the whole space; if **all** equations in a system have this form then the solution space will be the whole space. On the other hand, equations like “ $0 = 1$ ” have no solutions; if **any** equation in a system has this form then the solution space will be empty.

**Remark.** You might be wondering what happens if we drop the requirement that the equations be linear, and allow for example **polynomial** equations? If we do this then lines, planes and hyperplanes get replaced by more general objects called **curves**, **surfaces** and **hypersurfaces** respectively. For example, the solution set to  $x^2 + y^2 + z^2 = 1$  is a sphere, which is a 2-dimensional surface in 3-space. The solution space to a **system** of polynomial equations (an intersection of hypersurfaces) is called an **(affine) algebraic variety**. This part of mathematics is called **algebraic geometry**; you can study it in your third year.

---

<sup>4</sup>Of course “usually” here is a vague term, but I hope you can see intuitively what I mean! If you pick two lines at random it is somehow “infinitely unlikely” that they will be exactly parallel. It is possible to make this intuition precise; this is well beyond the scope of this course but as a (challenging!) exercise you might like to think how you would do so.

## 2.4 Linear Equations and Matrices

Any system of linear equation can be rearranged to put all the constant terms on the right, and all the terms involving variables on the left. This will yield a system something like ...

$$\begin{aligned}4x - 2y &= 3 \\2x + 2y &= 5\end{aligned}$$

... so from now on we will assume all systems have this form. The system can then be written in **matrix form**:

$$\begin{pmatrix} 4 & -2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$

You can check (by multiplying out the matrix equation using the definition of matrix multiplication) that it holds if and only if both of the original (scalar) equations hold. We can express the equations even more concisely as an **augmented matrix**:

$$\left( \begin{array}{cc|c} 4 & -2 & 3 \\ 2 & 2 & 5 \end{array} \right).$$

## 2.5 Elementary Row Operations and Solving Systems of Equations

At school you learnt to solve systems of linear equations by simple algebraic manipulation. This *ad hoc* approach is handy for small examples, but many applications involve bigger systems of equations, and for these it is helpful to have a **systematic method**.

We start with the augmented matrix of our system (see Section 2.4), and we allow ourselves to **modify** the matrix by certain kinds of steps. We can:

- (i) **multiply** a row by a **non-zero** scalar (written  $r_i \rightarrow \lambda r_i$ );
- (ii) **add** a multiple of one row to another (written  $r_i \rightarrow r_i + \lambda r_j$ );
- (iii) **swap** two rows of the matrix (written  $r_i \leftrightarrow r_j$ ).

These are called **elementary row operations**.<sup>5</sup>

**Warning.** Multiplying a row by 0 is definitely **not** an elementary row operation!

**Definition.** Two matrices  $A$  and  $B$  are called **row equivalent** if each can be obtained from the other by a sequence of elementary row operations.

Row operations and row equivalence are useful because of the following fact:

**Theorem 2.1.** *Suppose  $M$  and  $N$  are the augmented matrices of two system of linear equations. If  $M$  is row equivalent to  $N$  then the two systems have the same solution set.*

We'll prove this theorem later (Chapter 3), and but for now let's explore its consequences. It means that if we start with a system of equations, applying elementary row operations to the augmented matrix will give us another system of equations **with the same solution set**. The aim of the game is to obtain a system of equations which is easier to solve. For example, consider the system:

First we convert this into augmented matrix form (noting that the "missing"  $x$  terms are really  $0x$  and become 0s in the matrix):

---

<sup>5</sup>There is an obvious "dual" idea of **elementary column operations**, written  $c_i \rightarrow \lambda c_i$ ,  $c_i \rightarrow c_i + \lambda c_j$  and  $c_i \leftrightarrow c_j$ .

Now we swap rows 1 and 2 ( $r_1 \leftrightarrow r_2$ ), to get

Next let's scale rows 1 and 2 to make the first entry in each 1 ( $r_1 \rightarrow -r_1$ ,  $r_2 \rightarrow \frac{1}{10}r_2$ ):

Next we subtract 10 times row 2 from row 3 ( $r_3 \rightarrow r_3 - 10r_2$ ), giving:

Now we convert the augmented matrix back to a system of equations:

It turns out that these equations are easier to solve than the original ones. Equation [3] tells us straight away that  $z = 3$ . Now substituting this into equation [2] we get  $y - \frac{3}{10} \times 3 = -\frac{19}{10}$ , in other words,  $y = -1$ . Finally, substituting both of these into equation [1] gives  $x - 4 \times (-1) + 3 = 9$ , that is,  $x = 2$ .

So we have found the solutions to equations [1], [2] and [3]. But these were obtained from our original system of equations by elementary row operations, so (by Theorem 2.1) they are also the solutions to the original system of equations. (**Check this for yourself!**)

This strategy forms the basis of a very general algorithm, called **Gaussian<sup>6</sup> Elimination**, which we shall see in Section 2.7.

## 2.6 Row Echelon Matrices

**Remark.** Notice how we converted the matrix into a roughly “triangular” form, with all the entries towards the “bottom left” being 0. It was this property which made the new equations easy to solve. The following definitions will make precise what we mean by “triangular” here.

**Pivots.** The **pivot** (or **leading entry**) of a row in an augmented matrix is the position of the leftmost non-0 entry to the left of the bar. (If all entries left of the bar are 0, we call the row a **zero row** and it has no pivot.)

**Example.** In the matrix  $A$  on the right

- the pivot of the first row is the 5;
- the pivot of the second row is the 3;
- the third row is a zero row (it has no pivot)

$$A = \left( \begin{array}{ccc|c} 0 & 5 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{array} \right)$$

**Row Echelon Matrices.** An augmented matrix is called a **row echelon matrix** if

---

<sup>6</sup>After Carl Friedrich Gauß (pronounced to rhyme with “house”), 1777–1855.

- (1) any zero rows come at the bottom; and
- (2) the pivot of each of the other rows is **strictly to the right of the pivot of the row above**; and
- (3) all the pivots are 1.

A row echelon matrix is a **reduced row echelon matrix** if in addition:

- (4) each pivot is the only non-zero entry in its column.

**Example.** The matrix  $A$  above is **not** a row echelon matrix, because the second row pivot is not to the right of the first row pivot. The matrix  $B$  on the right **is** a row echelon matrix, but it is **not** a **reduced** row echelon matrix because the 2 in the first row is in the same column as the second row pivot.

$$B = \left( \begin{array}{cccc|c} 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{array} \right)$$

## 2.7 Gaussian Elimination

**Gaussian Elimination** is a simple algorithm which **uses elementary row operations to reduce the augmented matrix to row echelon form**:

- (1) By swapping rows if necessary, make sure that **no non-0 entries (in any row) are to the left of the first row pivot**.
- (2) Scale the first row so that the pivot is 1.
- (3) Add multiples of the first row to each of the rows below, so as to make all entries below the first row pivot become 0. (Since there were no non-0 entries left of the first row pivot, this ensures all pivots **below** the first row are strictly to the right of the pivot **in** the first row.)
- (4) Now ignore the first row, and repeat the entire process with the second row. (This will ensure that all pivots **below** the second row are to the right of the pivot **in** the second row, which in turn is to the right of the pivot in the first row.)
- (5) Keep going like this (with the third row, and so on) until either we have done all the rows, or all the remaining rows are zero.
- (6) At this point the matrix is in row echelon form.

**Exercise.** Go back to the example in Section 2.5, and compare what we did with the procedure above.

## 2.8 Another Example

**Example.** Let's solve the system of equations:

$$\begin{array}{rcrcrcrcrcr} 2x & + & y & - & z & = & -7 \\ 6x & & & - & z & = & -10 \\ -4x & + & y & + & 7z & = & 31 \end{array}$$

**Solution.** First we express the system as an augmented matrix:

Notice (Step 1) that there are no non-0 entries to the left of the pivot in the first row (there can't be, since the pivot is in the first column!). So we do not need to swap rows to ensure this is the case.

Next (Step 2) we scale to make the pivot in the first row 1, that is,  $r_1 \rightarrow \frac{1}{2}r_1$ , which gives

Next (Step 3) we seek to remove the entry in row 2 which is below the row 1 pivot. We can do this with the operation  $r_2 \rightarrow r_2 - 6r_1$ , giving the matrix:

Similarly, we remove the entry in row 3 below the row 1 pivot by  $r_3 \rightarrow r_3 + 4r_1$ :

Now we ignore the first row, and repeat the whole process with the remaining rows. Notice that there are no pivots to the left of the row 2 leading entry. (The one in row 1 doesn't count, since we are ignoring row 1!). Now we scale to make the second row pivot into 1, with  $r_2 \rightarrow -\frac{1}{3}r_2$ , giving:

Then remove the entry in row 3 which is below the row 2 pivot, with  $r_3 \rightarrow r_3 - 3r_2$ , giving:

Finally, we repeat the process again, this time ignoring rows 1 and 2. All we have to do now is scale row 3 to make the pivot 1, with  $r_3 \rightarrow \frac{1}{7}r_3$ :

Our matrix is now in row echelon form, so we convert it back to a system of equations:

These matrices can easily be solved by “backtracking” through them. Specifically:

- equation [3] says explicitly that  $z = 4$ ;
- substituting into equation [2] gives  $y - \frac{2}{3} \times 4 = -\frac{11}{3}$ , so  $y = -1$ ;
- substituting into equation [1] gives  $x + \frac{1}{2} \times (-1) - \frac{1}{2} \times 4 = -\frac{7}{2}$ , so  $x = -1$ .

Thus, the solution is  $x = -1$ ,  $y = -1$  and  $z = 4$ . (You should **check this** by substituting the values back into the original equations!)

**Remark.** Notice how the row echelon form of the matrix facilitated the “backtracking” procedure for solving equations [1], [2] and [3].

**Exercise.** Use Gaussian elimination to solve the following systems of equations:

$$(i) \quad \begin{array}{rcl} 4x + y & = & 9 \\ 2x - 3y & = & 1 \end{array} \qquad (ii) \quad \begin{array}{rcl} 2x - 4y & = & 12 \\ -x + 2y & = & -5 \end{array}$$

## 2.9 Gauss-Jordan Elimination

Gaussian elimination lets us transform any matrix into a row echelon matrix. Sometimes it is helpful to go further and obtain a **reduced** row echelon matrix. To do this we use a slight variation called **Gauss-Jordan<sup>7</sup> Elimination**:

- First use Gaussian elimination to compute a row echelon form.
- Add multiples of the 2nd row to the row above to remove any non-0 entries above the 2nd row pivot.
- Add multiples of the 3rd row to the rows above to remove any non-0 entries above the 3rd row pivot.
- Continue down the rows, adding multiples of row  $k$  to the rows above to eliminate any non-0 entries above the  $k$ th row pivot.

**Theorem 2.2.** *Let  $A$  be a matrix. Then  $A$  is row equivalent to a **unique** reduced row echelon matrix (called the **reduced row echelon form of the matrix**).*

*Proof.* It should be clear that applying the Gauss-Jordan elimination algorithm to  $A$  will give a reduced row echelon matrix row equivalent to  $A$ . The proof of uniqueness is more difficult, and we omit it.  $\square$

**Important Consequence.** We can check if two matrices are row equivalent, by applying Gauss-Jordan elimination to compute their reduced row echelon forms, then checking if these are equal. (If the reduced row echelon forms are equal then clearly the original matrices are equivalent. Conversely, if the matrices are equivalent then by Theorem 2.2, they must have the same reduced row echelon form.)

**Exercise.** Check if  $\begin{pmatrix} 1 & 3 & 6 \\ 0 & 1 & 2 \\ 3 & 7 & 14 \end{pmatrix}$  and  $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 5 & 10 \\ 1 & 1 & 2 \end{pmatrix}$  are row equivalent.

## 2.10 Elimination with More Variables than Equations

What if, after applying Gaussian elimination and converting back to equations, there are more variables than equations? This can happen either because the system we started with was like this, or because elimination gives us rows of zeros in the matrices: these correspond to the equation “ $0 = 0$ ” which obviously can’t be used to find the value of any variables.

Suppose there are  $k$  equations and  $n$  variables, where  $n > k$ . We can still apply Gaussian elimination to find a row echelon form. But when we convert back to equations and “backtrack” to find solutions, we will sometimes still encounter an equation with more than one unknown variable.

If this happens, there are **infinitely many** solutions. We can describe all the solutions by introducing **parameters** to replace  $n - k$  of the variables.

---

<sup>7</sup>After the German Wilhelm Jordan (1842–1899) and **not** the more famous French mathematician Camille Jordan (1838–1922). The correct pronunciation is therefore something like “YOR-dan” (not zhor-DAN!) but most people give up and go with the English pronunciation!

**Example.** Consider the equations.

$$\begin{array}{rclcrcl} -x & + & 2y & + & z & = & 2 \\ 3x & - & 5y & - & 2z & = & 10 \\ x & - & y & & & = & 14 \end{array}$$

The augmented matrix is:

$$\left( \begin{array}{ccc|c} -1 & 2 & 1 & 2 \\ 3 & -5 & -2 & 10 \\ 1 & -1 & 0 & 14 \end{array} \right) \text{ and reducing to row echelon form we get } \left( \begin{array}{ccc|c} 1 & -2 & -1 & -2 \\ 0 & 1 & 1 & 16 \\ 0 & 0 & 0 & 0 \end{array} \right).$$

which gives the equations

$$\begin{array}{rclcrcl} x & - & 2y & - & z & = & -2 & [1] \\ & & y & + & z & = & 16 & [2] \\ & & & & 0 & = & 0 & [3] \end{array}$$

Equation [3] is always satisfied but clearly useless for finding the values of the variables, so we can ignore it. Equation [2] has two unknown variables ( $y$  and  $z$ ) so we introduce a parameter  $\lambda$  to stand for one of them, say  $z = \lambda$ . Now solving we get  $y = 16 - \lambda$ . Substituting into [1] gives  $x - 2(16 - \lambda) - \lambda = -2$ , or  $x = 30 - \lambda$ . So the solutions are:

$$x = 30 - \lambda, \quad y = 16 - \lambda, \quad z = \lambda$$

for  $\lambda \in \mathbb{R}$ .

**Remark.** When we say these are the solutions, we mean that substituting in different values of  $\lambda \in \mathbb{R}$  will give all the solutions to the equations. For example,  $\lambda = 1$  would give  $x = 29$ ,  $y = 15$  and  $z = 1$ , so this is one solution to the system of equations (**check this!**). Or then again,  $\lambda = 0$  gives  $x = 30$ ,  $y = 16$  and  $z = 0$ , so this is another possible solution.

**Exercise.** Use Gaussian elimination to solve

$$\begin{array}{rcl} 2x & - & 4y & = & 10 \\ -x & + & 2y & = & -5 \end{array}$$

## 2.11 Homogeneous Systems of Equations

A system of linear equations is called **homogeneous** if the constant terms are all 0. For example:

$$\begin{array}{rclcrcl} x & + & 7y & - & 4z & = & 0 \\ 2x & + & 4y & - & z & = & 0 \\ 3x & + & y & + & 2z & = & 0 \end{array}$$

An obvious observation about homogeneous systems is that they **always have at least one solution**, given by setting all the variables equal to 0. This is called the **trivial solution**. Geometrically, this means that the solution space to a homogeneous system always contains the origin.

**Forward Pointer.** Solution sets to homogeneous systems — that is, intersections of hyperplanes through the origin — are very special and important subsets in  $\mathbb{R}^n$ , called **vector subspaces**. More on this later (Chapter 5).

When it comes to solving them, homogeneous systems of equations are treated just like non-homogeneous ones. The only difference is that nothing interesting ever happens on the RHS of the augmented matrix, as the following exercise will convince you!

**Exercise.** Find solutions to the system of equations above.

**Remark.** Notice how, in your solution, everything stays “homogeneous” throughout. All the augmented matrices in the elimination process, and all the resulting equations, have only 0’s on the RHS.

## 2.12 Computation in the Real World

In describing methods to solve equations (Gaussian and Gauss-Jordan elimination) we have implicitly assumed two things:

- that the equations are precisely known; and
- that we can perform the necessary arithmetic with perfect accuracy.

But in reality, these assumptions are often not true. If we want to solve a system of equations because it models a real-world problem, then the coefficients in the equations may have been obtained by measurement, in which case their values are subject to experimental error. And even if the coefficients are known precisely, real-world computers don't actually store and process real numbers: they can't, because there is an (uncountably) infinite set of real numbers but even the most powerful digital computer has a finite memory. Instead they work with **approximations** to real numbers: typical is **floating point arithmetic** in which numbers are rounded to a certain number of significant figures. At every stage of the computation numbers have to be rounded to the nearest available approximation, introducing slight inaccuracies even if the original data was perfect.

If the starting data or the arithmetic are imperfect then it clearly isn't realistic to expect perfect solutions. But some methods have the useful property that data which is **almost** right will produce solutions which are **almost** right<sup>8</sup>. Other methods do not have this property at all: they may work perfectly in theory, but the tiniest error in the starting data gets magnified as the computation progresses, and leads to a wildly inaccurate answer. Gaussian elimination, in the simple form we have described, is not very good in this respect, but it can be made better with a slightly more sophisticated approach (called **partial pivoting**) which takes account of the relative size of the matrix entries.

Further consideration of these practical issues is beyond the scope of this course, but you can learn about them in later years, starting with the second year option **MATH20602 Numerical Analysis I**.

---

<sup>8</sup>Or the subtly different property that the solutions produced are exact solutions to **almost the right problems**. Of course this is an oversimplification; for an accurate explanation see the first few sections of Chapter 1 in the book *Accuracy and Stability of Numerical Algorithms (2nd edition)* by N. J. Higham, and in particular the distinction between forward error and backward error. (But this is not examinable in this course.)