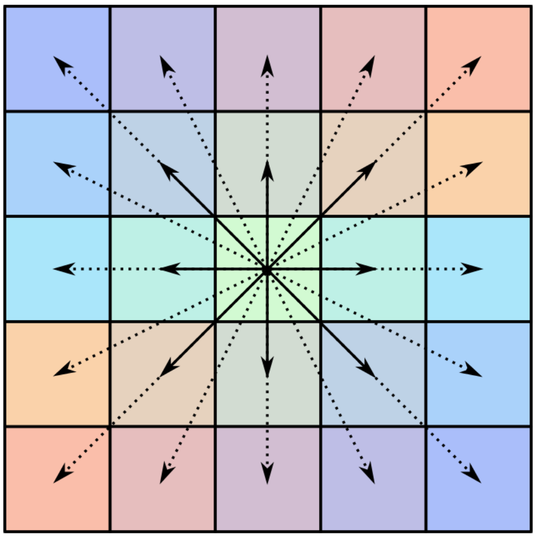
Gymnázium Evolution Jižní Město



Intro-ish To Linear Algebra

Madam Pekáč

December 26, 2024

Preface

This text covers selected topics from the curriculum of a typical undergraduate linear algebra course. Almost no pre-existing knowledge is strictly required save a superficial understanding of propositional logic and set theory. A reasonably good ability to manipulate algebraic expressions should prove advantageous, too.

Mathematics is an exact and rigorous language. Words and symbols have singular, precisely defined, meaning. Many students fail to grasp that intuition and imagination are paramount, but they serve as a *starting point*, with formal logical expression being the end. For example, an intuitive understanding of a *line* as an infinite flat 1D object is pretty much correct but not *formal.* It is indeed the formality of mathematics which puts many students off. Whereas high school mathematics is mostly algorithmic and non-argumentative, higher level maths tends to be the exact opposite – full of concepts and relations between those, which one is expected to be capable of grasping and formally describing. Owing to this, I wish this text would be a kind of synthesis of the formal and the conceptual. On one hand, rigorous definitions and proofs are given; on the other, illustrations, examples and applications serve as hopefully efficient conveyors of the former's geometric nature.

Linear algebra is a mathematical discipline which studies – as its name rightly suggests – the *linear.* Nevertheless, the word *linear* (as in 'line-like') is slightly misplaced. The correct term would perhaps be *flat* or, nigh equivalently, *not curved.* It isn't hard to imagine why curved objects (as in *geometric* objects, say) are more difficult to describe and manipulate than objects flat. For instance, the formula for the volume of a cube is just the product of the lengths of its sides. Contrast this with the volume of a still 'simple', yet curved, object – the ball. Its volume cannot even be *precisely* determined; its calculation involves approximating an irrational constant and the derivation of its formula is starkly unintuitive without basic knowledge of measure theory.

As such, linear algebra is a highly 'geometric' discipline and opportunities for visual interpretations abound. This is also a drawback in a certain sense. One should not dwell on visualisations alone as they tend to lead astray where imagination falls short. Symbolic representation of the geometry at hand is key.

The word *linear* however dons a broader sense in modern mathematics. It can be rephrased as reading, 'related by addition and multiplication by a scalar'. We trust kind readers have been acquainted with the notion of a *linear function.* A linear function is (rightly) called *linear* for it receives a number as input and outputs its *constant* multiple plus another *constant* number. Therefore, the output is in a *linear* relation to the input – it is multiplied by some fixed number and added to another. This understanding of the word is going to prove crucial already in the first chapter, where we study *linear systems.* Following are *vector spaces* and *linear maps*, concepts whose depth shall occupy the span of this text. Each chapter is further endowed with an *applications* section where I try to draw a simile between mathematics and common sense.

Contents

Chapter 1

Linear Systems

Linear systems are by definition sets of linear equations, that is, of equations which relate present variables in a *linear* way. It is important to understand what this means. Spelled out, an expression on either side of any of the equations is formed *solely* by

(1) multiplying the variables by a given number (**not another variable**),

(2) adding these multiples together.

Any such combination where variables are only allowed to be multiplied by a constant and added is called a *linear combination*. This term is extremely important and ubiquitous throughout the text; hence, it warrants an isolated definition.

Definition 1.0.1 (Linear combination)

Let $x_1, \dots, x_n$ with $n \in \mathbb{N}$ be variables. Their *linear combination* is any expression of the form

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n,$$

where $a_1, \dots, a_n$ are numbers.

Remark 1.0.2

In the definition above, we have deliberately not specified what type of *numbers* we mean. In the future, we shall work extensively with real and complex numbers as well as elements of other fields, which dear readers might not have even recognised as 'numbers' thus far. The only important concept in this regard is the clear distinction between a *number* (later *scalar*) and a *variable* (later *vector*).

Example 1.0.3

Consider the variables x, y and z. The expression

$$3x + 2y - 0.5z$$

is their linear combination whereas

$$5x + 3y - yz + 7z^2$$

is not.

To reiterate, a *linear system* is any set of equations featuring only linear combinations of variables; these equations are consequently called *linear* as well. A *solution* of a linear system is the set of all possible substitutions of numbers (in place of variables) which make the equations true.

It is clear that every linear equation can be rearranged to

$$a_1x_1 + \cdots + a_nx_n = c$$

for some variables $x_1, \ldots, x_n$ and numbers $a_1, \ldots, a_n, c$ by simple subtraction. This is how we shall define it, for simplicity.

Definition 1.0.4 (Linear equation)

Any equation of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = c, \tag{1.1}$$

where $x_1, \ldots, x_n$ are variables and $a_1, \ldots, a_n, c$ are numbers, is called *linear*. A *solution* of a linear equation is an n-tuple $(b_1, \ldots, b_n)$ of numbers such that under the substitutions $x_i \coloneqq b_i$, for $i \in \{1, \ldots, n\}$, the equation (1.1) is satisfied.

Example 1.0.5

The equation

$$3x_1 - 2x_2 + 4x_3 + x_4 = 5$$

is linear in variables x_1, x_2, x_3 and x_4. On the contrary,

$$3x_1x_2 - 4x_3^2 = 10$$

is **not** linear.

Definition 1.0.6 (Linear system)

Any set of linear equations in the given variables $x_1, \ldots, x_n$ is called a *linear system*. A *solution* of a linear system is an n-tuple $(b_1, \ldots, b_n)$ which solves every linear equation in the set.

Example 1.0.7

The set of equations

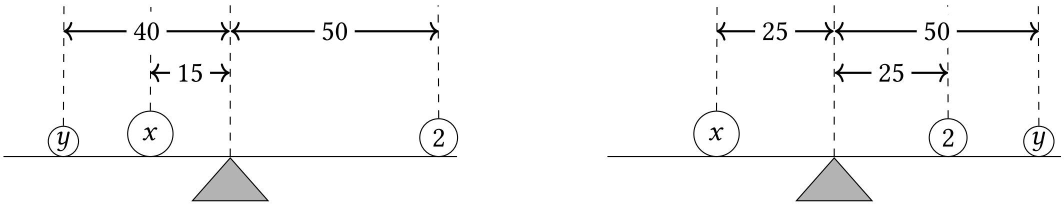
$$\begin{array}{rcrcrcr} 3x_1 & - & x_2 & + & 2x_3 & = & 1 \\ x_1 & & & - & x_3 & = & -1 \\ 2x_1 & - & 3x_2 & + & 3x_3 & = & 0 \end{array}$$

is a linear system whose solution is the triple $(0, 1, 1)$.

We proceed to discuss two trivial examples, which readers might have discussed in high school, naturally leading to linear systems. More sophisticated examples are presented in the applications section.

Example 1.0.8 (Static equations)

Suppose we have three objects – one with a mass of 2 and the other two with masses unknown. Experimentation produces these two balances.



For the weights to be in balance, the sums of *moments* on both sides of the scales must be identical one to another. A *moment* of an object is its distance from the centre of the scales times its mass. This condition yields a system of two linear equations

$$\begin{aligned} 15x + 40y &= 50 \cdot 2, \\ 25x &= 25 \cdot 2 + 50y. \end{aligned}$$

Or, after rearrangement (to stay true to our definition of linear equation),

$$\begin{aligned} 15x + 40y &= 50 \cdot 2, \\ 25x - 50y &= 25 \cdot 2. \end{aligned}$$

Example 1.0.9 (Chemical reactions)

Toluene, $\mathtt{C_7H_8}$, mixes (under right conditions) with nitric acid, $\mathtt{HNO_3}$, to produce trinitrotoluene (widely known as TNT), $\mathtt{C_7H_5O_6N_3}$, along with dihydrogen monoxide, $\mathtt{H_2O}$. If we want this chemical reaction to occur successfully, we must (among other things) ascertain we mix the constituents in the right proportion. In pseudo-chemical notation, the reaction to take place can be written as

$$x \cdot \mathtt{C_7H_8} + y \cdot \mathtt{HNO_3} \longrightarrow z \cdot \mathtt{C_7H_5O_6N_3} + w \cdot \mathtt{H_2O}.$$

Comparing the number of atoms of each element before the reaction and afterwards (which must remain identical owing to the conservation of energy) yields the system

$$\begin{aligned} \mathtt{H}:\ 8x + 1y &= 5z + 2w, \\ \mathtt{C}:\ 7x &= 7z, \\ \mathtt{N}:\ 1y &= 3z, \\ \mathtt{O}:\ 3y &= 6z + 1w. \end{aligned}$$

In the next section, we devise an algorithm to solve any system of linear equations.

1.1 Gauss-Jordan Elimination

Probably the most well-known algorithm for solving a linear system is the *Gauss-Jordan elimination.* As its name partially implies, its heart lies in the successive *elimination* of variables until only a single linear equation in one variable stands unsolved. This is done by applying different *transformations* to the initial system that are guaranteed not to alter the solution. We're going to solve a linear system first and describe the general method second.

Problem 1.1.1

Solve the linear system

$$\begin{array}{rcrcrcl} & & & & 3x_3 & = & 9 \\ x_1 & + & 5x_2 & - & 2x_3 & = & 2\,. \\ \frac{1}{3}x_1 & + & 2x_2 & & & = & 3 \end{array}$$

Solution. We aim to transform the system step by step to a form which allows us to (successively) eliminate all variables.

The first transformation entails a simple exchange of the first and third row.

`Swapped first and third row.`

$$\begin{array}{rcrcrcl} \frac{1}{3}x_1 & + & 2x_2 & & & = & 3 \\ x_1 & + & 5x_2 & - & 2x_3 & = & 2 \\ & & & & 3x_3 & = & 9 \end{array}$$

Next, we scale the first row by a factor of 3.

`Scaled the first row by 3.`

$$\xrightarrow{\cdot 3} \quad \begin{array}{rcrcrcl} x_1 & + & 6x_2 & & & = & 9 \\ x_1 & + & 5x_2 & - & 2x_3 & = & 2 \\ & & & & 3x_3 & = & 9 \end{array}$$

Finally, we subtract the first row from the second row. Said in a more foreshadowing manner, we add the (-1)-multiple of the first row to the second row.

`Subtracted the first row from the second.`

$$\cdot(-1) \quad \begin{array}{rcrcrcr} x_1 & + & 6x_2 & & & = & 9 \\ & - & x_2 & - & 2x_3 & = & -7 \\ & & & & 3x_3 & = & 9 \end{array}$$

These transformations have wrought the system into a state where it can be easily solved.

Indeed, we immediately see that the third equation implies $x_3 = 3$. Substituting into the second equation gives

$$-x_2 - 2 \cdot 3 = -7$$

whose solution is $x_2 = 1$. Finally, knowing the value of x_2, we can solve the first equation by another substitution. We get

$$x_1 + 6 \cdot 1 = 9,$$

thus $x_1 = 3$ and the triple $(3, 1, 3)$ is the *unique* solution of the system. ♠

Observant readers might have already identified the 'kinds' of transformations that were used in solving the linear system above. Nonetheless, we're about to spell them out.

The transformations that do not change the solution of a linear system include

(1) swapping two equations;

(2) scaling an equation by a non-zero constant;

(3) adding a multiple of an equation to *another* equation.

Note that transformations (2) and (3) come with sensible restrictions. Scaling an equation by 0 clearly changes the set of solutions of the system as it basically removes the equation entirely. Adding a multiple of an equation to *itself* suffers from the same problem; it might result in 'invalidating' the equation should the scaling factor be -1.

We know proceed to prove that transformations (1) - (3) truly do not alter the solutions of the initial system.

Theorem 1.1.2 (Gauss-Jordan)

The transformations (1) - (3) of a linear system outlined above do not change its solution set.

Proof. We will cover transformation (3) here. The proofs for transformations (1) and (2) are similar and thus left as an exercise.

Consider the linear system

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & + & a_{1,n}x_n & = & c_1 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & + & a_{2,n}x_n & = & c_2 \\ & & & & & & & \vdots & \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \cdots & + & a_{m,n}x_n & = & c_m \end{array}$$

of m equations in variables $x_1, \ldots, x_n$ and let $(b_1, \ldots, b_n)$ be one of its solutions. Choose a constant k and add the k-multiple of the i-th equation to the j-th equation for some indices $i \neq j \in \{1, \ldots, m\}$. Hence, the j-th equation of the system gets replaced by

$$(a_{j,1} + k \cdot a_{i,1})x_1 + (a_{j,2} + k \cdot a_{i,2})x_2 + \cdots + (a_{j,n} + k \cdot a_{i,n})x_n = c_j + k \cdot c_i,$$

which can be rearranged to

$$a_{j,1}x_1 + a_{j,2}x_2 + \cdots + a_{j,n}x_n + k \cdot (a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,n}x_n) = c_j + k \cdot c_i. \tag{1.2}$$

Since $(b_1, \ldots, b_n)$ is a solution of the original system, we know that

$$\begin{array}{ccccccccc} a_{i,1}b_1 & + & a_{i,2}b_2 & + & \cdots & + & a_{i,n}b_n & = & c_i \\ a_{j,1}b_1 & + & a_{j,2}b_2 & + & \cdots & + & a_{j,n}x_n & = & c_j \end{array}.$$

Substituting this into equation (1.2) gives

$$c_j + k \cdot c_i = c_j + k \cdot c_i,$$

hence $(b_1, \ldots, b_n)$ is also the solution of the transformed system, as required. ■

Exercise 1.1.3

Show that transformations (1) and (2) also don't change the set of solutions of the transformed linear system.

Definition 1.1.4 (Elementary operations)

The transformations (1) - (3) outlined above are called *elementary operations* or *row operations*.

As we've seen in problem 1.1.1, the application of transformations (1) - (3) has its purpose in preparing the system for a final back-substitution, where the values of all variables save the first in a row are known beforehand. A system which is 'ready' to be solved by back-substitution is said to be in *echelon form*.

Definition 1.1.5 (Echelon form)

In each row of a linear system, the first variable with a non-zero coefficient is called the row's *leading variable*.

A linear system is in *echelon form* (or *upper triangular form*) if the leading variable in each row is at least one column to the right of the leading variable in the row above and all rows filled with zeroes are at the bottom.

Example 1.1.6

The system

$$\begin{array}{rcrcrcr} x_1 & + & 6x_2 & & & = & 9 \\ & - & x_2 & - & 2x_3 & = & -7 \\ & & & & 3x_3 & = & 9 \end{array}$$

is in echelon form whereas

$$\begin{array}{rcrcrcr} 2x_1 & + & 3x_2 & - & x_3 & = & 9 \\ & & 3x_2 & - & 2x_3 & = & 2 \\ x_1 & & & - & x_3 & = & 0 \end{array}$$

is **not**.

For now, we shall employ intuition and a nibble of foresight to guide our transformation of a linear system into its echelon form. Later, we intend to present a precise algorithm (that computers also use) that achieves this.

Example 1.1.7

We're going to put the system

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & & & = & 0 \\ 2x_1 & - & x_2 & + & 3x_3 & = & 3 \\ x_1 & - & 2x_2 & - & x_3 & = & 3 \end{array}$$

into echelon form and solve it using back-substitution. We'll label the rows of the system by Roman letters and denote transformations accordingly. For example, adding a 3-multiple of row one to row three would be written symbolically as $3 \cdot \mathtt{I} + \mathtt{III}$.

First, we need to get rid of the variable x_1 in rows $\mathtt{II}$ and $\mathtt{III}$. This can be done by subtracting adequate multiples of row $\mathtt{I}$.

$$\begin{aligned} x_1 + x_2 &= 0 \\ 2x_1 - x_2 + 3x_3 &= 3 \\ x_1 - 2x_2 - x_3 &= 3 \end{aligned} \xmapsto[-\mathtt{I}+\mathtt{III}]{-2\mathtt{I}+\mathtt{II}} \begin{aligned} x_1 + x_2 &= 0 \\ -3x_2 + 3x_3 &= 3 \\ -3x_2 - x_3 &= 3 \end{aligned}$$

We continue by subtracting row $\mathtt{II}$ from row $\mathtt{III}$.

$$\begin{aligned} x_1 + x_2 &= 0 \\ -3x_2 + 3x_3 &= 3 \\ -3x_2 - x_3 &= 3 \end{aligned} \xmapsto{-\mathtt{II}+\mathtt{III}} \begin{aligned} x_1 + x_2 &= 0 \\ -3x_2 + 3x_3 &= 3 \\ -4x_3 &= 0 \end{aligned}$$

The system is now in echelon form. The equation in row $\mathtt{III}$ forces $x_3 = 0$. Substitution into row $\mathtt{II}$ immediately gives $x_2 = -1$ and one final substitution into row $\mathtt{I}$ yields $x_1 = 1$.

Hence, the solution of the system is the triple $(1, -1, 0)$.

Exercise 1.1.8

Using *Gauss-Jordan elimination* solve the systems from examples 1.0.8 and 1.0.9.

All the systems we've studied so far have had the same number of equations as variables. This of course need not be the case in general. Thankfully, Gauss-Jordan elimination can *always* be used to determine the solution set of a linear system. However, this set can also be empty or infinite in cases where the number of variables doesn't match the number of equations. The following examples illustrate this.

Example 1.1.9

This system has more equations than variables.

$$\begin{aligned} x_1 + 3x_2 &= 1 \\ 2x_1 + x_2 &= -3 \\ 2x_1 + 2x_2 &= -2 \end{aligned} \tag{1.3}$$

Before we put it into echelon form and solve it, let us ponder what the solution set may look like. Intuitively, a linear equation is basically a 'restraint' or 'condition' on the range of possible values the present variables may attain. If there are three equations restraining only two variables, then this restraint may be too harsh and lead to the system having no solution at all. The only case where solution *does* exist involves one of the equations being *redundant* – providing no additional condition. Algebraically, this happens if said equation is a linear combination of the other two.

To draw a 'real-life' simile, imagine the price of an apple being \$5/kg and that of bananas, \$1.5/kg. Saying that 3 kg of apples and 4 kg of bananas cost, say, \$30 is simply false because we can calculate (by the information ere provided) that this amount actually costs \$21. The third condition on the price of apples and bananas contradicted the previous two; just as a third equation in a linear system in two variables can contradict the first two equations. We tend to call such systems *overdetermined* and will in time dedicate a section to finding a 'good' approximation of their solution.

To solve the system (1.3), we transform it into echelon form. First, we subtract twice the first row from the other two.

$$\begin{aligned} x_1 + 3x_2 &= 1 \\ 2x_1 + x_2 &= -3 \\ 2x_1 + 2x_2 &= -2 \end{aligned} \quad \xmapsto[-2\mathtt{I}+\mathtt{III}]{-2\mathtt{I}+\mathtt{II}} \quad \begin{aligned} x_1 + 3x_2 &= 1 \\ -5x_2 &= -5 \\ -4x_2 &= -4 \end{aligned}$$

Finally, we add $(-4/5)$-times row $\mathtt{II}$ to row $\mathtt{III}$.

$$\begin{aligned} x_1 + 3x_2 &= 1 \\ -5x_2 &= -5 \\ -4x_2 &= -4 \end{aligned} \quad \xmapsto{-(4/5)\mathtt{II}+\mathtt{III}} \quad \begin{aligned} x_1 + 3x_2 &= 1 \\ -5x_2 &= -5 \\ 0 &= 0 \end{aligned}$$

Clearly, the third equation is *redundant* because it provides no condition binding the values of the variables. Back-substitution yields $x_2 = 1$ and $x_1 = -2$. As we've claimed (but not yet proven), row $\mathtt{III}$ is indeed a linear combination of rows $\mathtt{I}$ and $\mathtt{II}$. In this particular case, it holds that $(2/5)\mathtt{I} + (4/5)\mathtt{II} = \mathtt{III}$.

Example 1.1.10

Contrast this system with the system (1.3) from example 1.1.9.

$$\begin{aligned} x_1 + 3x_2 &= 1 \\ 2x_1 + x_2 &= -3 \\ 2x_1 + 2x_2 &= 0 \end{aligned}$$

In this case, the exact same row operations transform the system into

$$\begin{aligned} x_1 + 3x_2 &= 1 \\ -5x_2 &= -5\,, \\ 0 &= 2 \end{aligned}$$

which clearly has no solution. This is a case of one equation of the system contradicting the other two.

Naturally, the ambitious, purposeful and *overdetermined* systems have their disinterested and vagrant sisters – the *underdetermined* systems. We style such the systems that are short on the number of variables as compared to the number of equations. As an example, consider the system

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_2 + x_3 &= 0 \end{aligned}, \tag{1.4}$$

which already **is** in echelon form. In spite of that, the typical back-substitution method is (without needed alterations) rendered unusable by the presence of two variables in the last row.

The system (1.4) is *underdetermined* in the sense that not enough equations are present to pinpoint a **unique** solution. Quite the opposite, this system has infinitely many solutions that all depend on as many parameters as many equations are missing to bind the values of the variables completely – in this case, *one*.

In cases like these, one typically proceeds the following way: let one of the variables (say, x_3) in the last equation be *a parameter*. For the sake of clarity, we shall rename x_3 to t to highlight its updated social status. The system thus looks like this.

$$\begin{array}{rcrcrcl} x_1 & + & x_2 & + & t & = & 0 \\ & & x_2 & + & t & = & 0 \end{array}$$

Now, t is no longer a variable so the system is no longer underdetermined. We solve it briskly by setting $x_2 = -t$ and substituting this into the first equation to obtain $x_1 = 0$. Therefore, the solution to the system (1.4) is $(0, -t, t)$.

The dependence of the system's solution on a parameter naturally means that its solution set is infinite. Any choice for the value of t gives one particular solution – say $(0, -1, 1)$ or $(0, 0, 0)$.

We close the section off with a few exercises. The next section is dedicated to the geometric interpretation of linear systems.

Exercise 1.1.11

Use Gauss-Jordan elimination to solve the following system.

$$\begin{array}{rcrcrcl} x_1 & & & - & x_3 & = & 0 \\ 3x_1 & + & x_2 & & & = & 1 \\ -x_1 & + & x_2 & + & x_3 & = & 4 \end{array}$$

Exercise 1.1.12

Each of the following systems is in echelon form. Determine their number of solutions (without calculation).

$$\begin{array}{rcrcl} -3x_1 & + & 2x_2 & = & 0 \\ & - & 2x_2 & = & 0 \end{array} \qquad \begin{array}{rcrcl} 2x_1 & + & 2x_2 & = & 4 \\ & & x_2 & = & 1 \\ & & 0 & = & 4 \end{array} \qquad \begin{array}{rcrcl} 2x_1 & + & x_2 & = & 4 \end{array}$$

Exercise 1.1.13

Find the values of a, b and c that cause the graph of $f(x) = ax^2 + bx + c$ to pass through the points $(1, 2)$, $(-1, 6)$ and $(2, 3)$.

Exercise 1.1.14

Show that for all numbers a, b, c, d, j, k such that $ad - bc \neq 0$, the system

$$\begin{aligned} ax_1 + bx_2 &= j \\ cx_1 + dx_2 &= k \end{aligned}$$

has a *unique* solution.

1.2 Visualizing Linear Systems

In this, rather informal, section, we present a way to visualize linear systems in two and three variables and their solutions. Why two and three, you ask? The number of variables in a linear equation determines the *dimension* of the *geometric object* described by this equation. We shall soon provide the necessary definitions to make rigorous sense of the sentence previous. Intuitively, each variable represents a new 'direction' we're allowed to move in. Therefore, linear equations in two variables live in two-dimensional spaces and linear equations in three variables occupy three dimensions.

Nonetheless, the equations themselves (if non-trivial) never describe objects of the maximal possible dimension but of the dimension lower by one. This is because they establish a relationship between the variables – a relationship where one variable grows entirely dependent on the rest, essentially 'locking' a single direction of movement. Think of it like this: a linear equation in two variables is a sort of order, telling you that for every step forward you must also make (say) two steps to the right, thereby rendering you unable to ever walk in a direction different from the initial.

We proceed to show that the objects described by linear equations in two variables are *straight lines*. Said 'objects described' are formally the sets of points satisfying given equations. For instance, the object described by the equation $3x + 2y = 4$ is the set

$$L \coloneqq \{(x, y) \in \mathbb{R}^2 \mid 3x + 2y = 4\}.$$

Before we move on, we need establish an important fact. What is a *straight line* **exactly**? Wishing not to cheat and define straight line as the object described by a linear equation, we employ a more geometric approach to the definition. As we hope dear readers agree, a (one-dimensional) object is *straight* if moving along it requires 'keeping the initial direction', that is, always moving the same number of steps upward for a given number of steps rightward, or vice versa. In other words, the *ratio* between the number of steps upward and rightward must remain constant. We encourage kind readers to absorb that this particular property is what distinguishes *curved* objects from *straight* ones.

Figure 1.2 inspires the following definition.

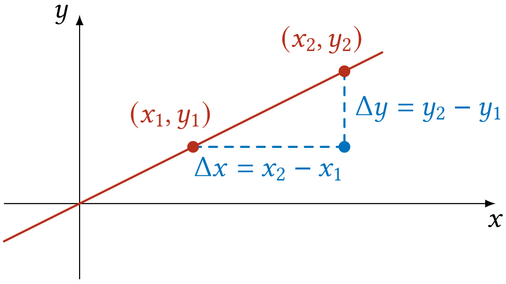


Figure 1.2: The 'definition' of straightness. The ratio $\Delta y/\Delta x$ must remain **constant**. It is habitually referred to as the *slope* of the line.

Definition 1.2.1 (Straight line)

An **infinite** subset $L \subseteq \mathbb{R}^2$ is called a *straight line* if for all triples of points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \in L$ it holds true that either

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_2}{x_3 - x_2}, \tag{1.5}$$

or $x_1 = x_2 = x_3$ (a vertical line).

We proceed to show that the all the points in the plane satisfying a linear equation form a straight line. This is exceedingly easy. Suppose we have three solutions (x_1, y_1), (x_2, y_2) and (x_3, y_3) satisfying the equation $ax + by = c$, where $a, b, c \in \mathbb{R}$ and at least one of a, b is not zero. In other words, we have $ax_i + by_i = c$ for $i \in \{1, 2, 3\}$.

We've had to exclude the case $a = b = 0$ because the set of solutions of the linear equation $0 = c$ is never a straight line. If $c \neq 0$, it is empty, and if $c = 0$, it equals $\mathbb{R}^2$.

Assume first that $b = 0$. Then, $x_i = c/a$ and so $x_1 = x_2 = x_3$. Hence, in this case, the set of solutions is indeed a straight line.

In case $b \neq 0$, we may rearrange

$$y_i = \frac{c - ax_i}{b}.$$

Plugging this into (1.5) gives

$$\frac{(c - ax_2) - (c - ax_1)}{b(x_2 - x_1)} = \frac{(c - ax_3) - (c - ax_1)}{b(x_3 - x_1)}. \tag{1.6}$$

Simple calculation yields

$$\frac{(c - ax_2) - (c - ax_1)}{b(x_2 - x_1)} = \frac{a(x_1 - x_2)}{b(x_2 - x_1)} = -\frac{a}{b}$$

and similarly for $(y_3 - y_1)/(x_3 - x_1)$. Hence, both sides of (1.6) equal $-a/b$ and the proof is complete.

1.2.1 Two-dimensional Linear Systems

We dedicate a subsection to the visualization of linear systems in two variables and their solutions. As already established, a linear equation in two variables represents a straight line. A solution to a linear system in two variables is a pair of real numbers (equivalently, a point in the real plane) which lies on every straight line determined by the equations of the system. Simply put, the solution of a linear system in two variables is the *intersection* of all objects described by its equations.

An ‘ideal’ linear system in two variables contains two linear equations describing distinct lines. One such system is

$$\begin{aligned} 2x - y &= 1 \\ x + y &= 2 \end{aligned}$$

with solution $(1, 1)$ and whose visual depiction is provided in figure 1.3.

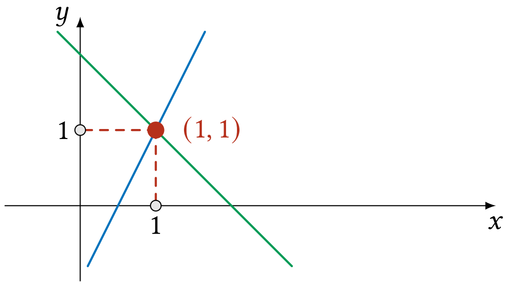


Figure 1.3: Well-determined linear system in two variables with solution $(1, 1)$.

An easily proven fact (which we shall eventually prove in greater generality) that follows immediately from the geometric view reads that a linear system in two variables with two *distinct* linear equations always has a solution – the intersection point of the corresponding lines.

A linear system in two variables can only be underdetermined should it feature just one non-trivial linear equation (or, equivalently, many identical linear equations). In this case, assuming the system consists of the single linear equation

$$ax + by = c,$$

its solution set is spanned by the points $(x, (c - ax)/b)$, for $x \in \mathbb{R}$, or $(c/a, y)$, for $y \in \mathbb{R}$, should $b = 0$. Geometrically, all points lying on the line determined by its sole equation solve the underdetermined linear system.

Overdetermined linear systems in two variables are considerably more interesting. There are four possible arrangements of three lines in the plane, they’re depicted in figure 1.4.

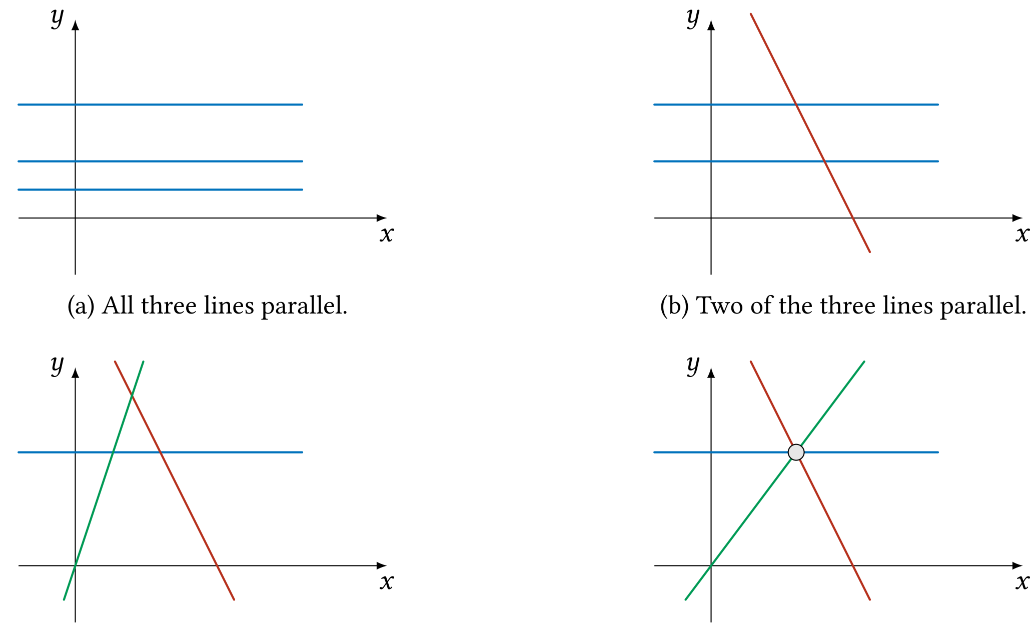
It is clear that in cases (a), (b) and (c) in figure 1.4, the linear system has no solution. In case (d), the system does have a solution but one of the lines is redundant – it can in fact (as we’ve claimed before) be written as a linear combination of the other two lines. By putting the linear system in question into echelon form, we can easily deduce which of the depicted cases emerged true.

Indeed, consider the system

$$\begin{aligned} x + y &= 2 \\ 2x + 2y &= 3\,. \\ -x - y &= 1 \end{aligned}$$

By subtracting `2I` from `II` and adding `I` to `III`, we put it into the following echelon form:

$$\begin{aligned} x + y &= 2 \\ 0 &= -1\,. \\ 0 &= 3 \end{aligned}$$



(a) All three lines parallel.

(b) Two of the three lines parallel.

(c) No lines parallel without a common intersection.

(d) No lines parallel with a common intersection.

Figure 1.4: All the possible arrangements of three lines in the real plane.

Since two of the three equations have no solutions, case (a) arises – the three lines are all parallel to one another.

As yet another example, we present in all its glory the system

$$\begin{array}{rcrcr} 2x & + & y & = & 5 \\ x & & & = & 2\,. \\ 3x & - & y & = & 0 \end{array}$$

By swapping $\mathtt{I}$ with $\mathtt{II}$, then subtracting $\mathtt{II} - 2\mathtt{I}$ and $\mathtt{III} - 3\mathtt{I}$, we get

$$\begin{array}{rcrcr} x & & & = & 2 \\ & & y & = & 1\,. \\ & - & y & = & -6 \end{array}$$

Each of the equations has a solution individually but the conditions bestowed on y by equations $\mathtt{II}$ and $\mathtt{III}$ are contradictory. Meaning, any pair of equations in the system can be satisfied simultaneously but all three equations can't. This is the situation depicted in figure 1.4, (c).

Exercise 1.2.2

Find examples of linear systems of three equations in two variables that correspond to parts (b) and (d) of figure 1.4.

1.2.2 Three-dimensional Linear Systems

Stepping up the game a little, we're taking a look at linear systems in three variables. Just as a linear equations in two variables are lines in the real plane, linear equations in three variables depict geometric objects called 'planes' in the three-dimensional real space, $\mathbb{R}^3$. They form the last class of linear systems that can be efficiently visualized; with linear systems in more variables being generally out of our perceptive reach.

Planes are the *straight* objects in three-dimensional kind of sense. They lock one direction of movement by making one variable wholly dependent on the other two. An illustration is provided in figure 1.5.

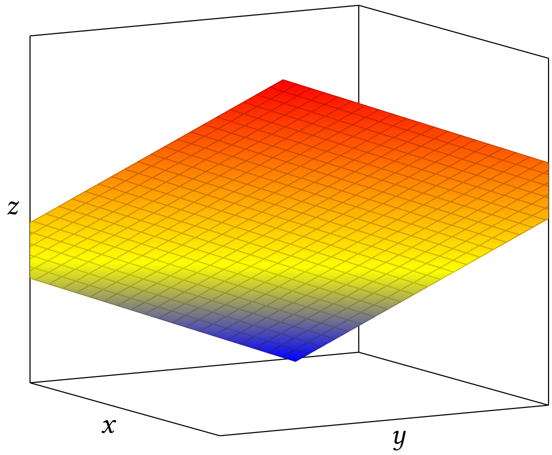


Figure 1.5: A plane defined by the equation $2x - y - 3z = -3$.

An *underdetermined* system in three variables can contain either one or two linear equations. In the former case, only one variable is dependent on the other two – we shall often call the independent variables by names such as *free variables* or *parameters*. Both these names signify that a substitution of any pair of real numbers in lieu of the two *free variables* yields a solution of the system.

For instance, the linear equation in figure 1.5 is effectively a linear system in three variables. We can choose any of the three variables to be dependent and leave the other two free, giving thus three different descriptions of *the same* solution set. The following equation (1.7) shows all of them with the chosen dependent variable written on the left in typewriter font.

$$\begin{aligned} \mathtt{x} &: \left\{\left(\tfrac{y+3z-3}{2}, y, z\right) \mid y, z \in \mathbb{R}\right\} \\ \mathtt{y} &: \{(x, 2x - 3z + 3, z) \mid x, z \in \mathbb{R}\} \\ \mathtt{z} &: \left\{\left(x, y, \tfrac{2x-y+3}{3}\right) \mid x, y \in \mathbb{R}\right\} \end{aligned} \tag{1.7}$$

Linear systems in three variables and two equations are also underdetermined. Geometrically, they correspond to arrangements of two planes in space. Those two planes can either be parallel – leading to the system having no solution – or not – intersecting in a straight line describable as a set of triples with exactly one free variable. In a case similar to two-dimensional linear systems, putting the system in question into echelon form *can* reveal (albeit not always) its geometric nature.

Consider the system

$$\begin{aligned} x - y + 2z &= 2 \\ 2x - 2y + 4z &= 9 \end{aligned}. \tag{1.8}$$

Subtracting $\mathtt{II} - 2 \cdot \mathtt{I}$ produces

$$\begin{aligned} x - y + 2z &= 2 \\ 0 &= 5 \text{'} \end{aligned}$$

clearly a system with no solution. Subsequently, the two corresponding planes are parallel to each other. See them depicted in figure 1.6.

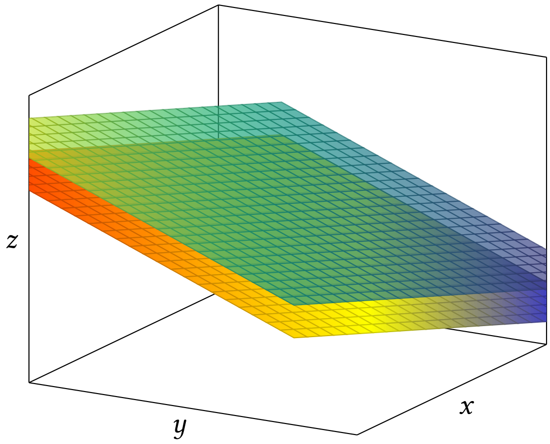


Figure 1.6: The two parallel planes from the system (1.8).

A system of two non-parallel planes is presented below.

$$\begin{aligned} x - y + 2z &= 2 \\ 2x + 3y - z &= -1 \end{aligned} \tag{1.9}$$

By subtracting, once again, $\mathtt{II} - 2 \cdot \mathtt{I}$, we put into the following echelon form.

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

The algorithm of Gauss-Jordan elimination limits our choice of parameters to the ones left in the last row. We are hence to set either y or z loose while caging the latter. Custom dictates to label as parameters all variables but the first of the last row, making z the victor. The rest is just back-substitution. We calculate $y = z - 1$ and substitute into the first equation to receive

$$x - (z - 1) + 2z = 2, \quad \text{hence} \quad x = 1 - z.$$

It follows that *one possible* description of the solution set of the system (1.9) is

$$\{(1 - z, z - 1, z) \mid z \in \mathbb{R}\}.$$

See it depicted in figure 1.7.

Reaching the apex of 'ideal' linear systems in three variables and three equations, we stop to ponder the number of arrangements of three planes in three-dimensional space. There are two obvious ones:

(1) All three planes are parallel to each other.

(2) Only two planes are parallel to each other.

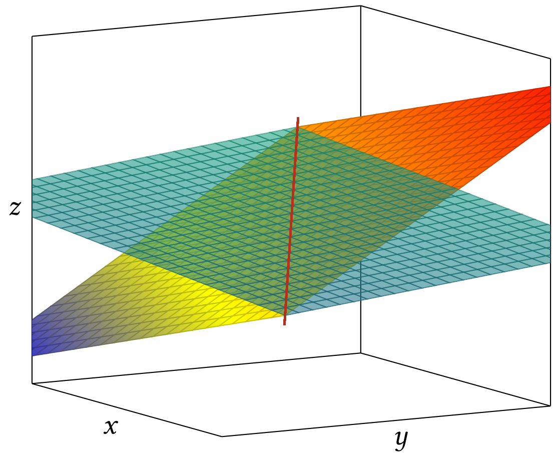


Figure 1.7: The two non-parallel planes from the system (1.9) and their intersection.

Corresponding to (1), resp. (2), is a linear system with two equations, resp. one equation, with no solution.

As an example, consider the system

$$\begin{aligned} -x + 2y - z &= 4 \\ x - 6y + z &= 1 \,. \\ x - 2y + z &= 3 \end{aligned} \tag{1.10}$$

Its echelon form looks like this:

$$\begin{aligned} -x + \quad 2y - z &= 4 \\ -4y \qquad &= 5 \,. \\ 0 &= 7 \end{aligned}$$

Clearly, the third equation has no solution while the second does. This fact alone, alas, carries not the full picture. To successfully determine that this system corresponds to case (2) above, one need additionally take note of the fact that the left side of row `III` of (1.10) is a (-1)-multiple of row `I`, meaning the two planes in question are parallel.

The system (1.10) is shown in figure 1.8.

Finally, there are three other possible arrangements of three planes in space:

(3) non-parallel planes that fail to have a common intersection (the so-often-called ‘tent' configuration);

(4) non-parallel planes that meet in a single point;

(5) non-parallel planes that meet in a single line.

The echelon form of a linear system is not enough to distinguish case (2) from case (3). For instance, the echelon form of the system

$$\begin{aligned} x + 2y - z &= -1 \\ 2x - 3y + 2z &= 4 \\ -x + 5y - 3z &= 0 \end{aligned} \tag{1.11}$$

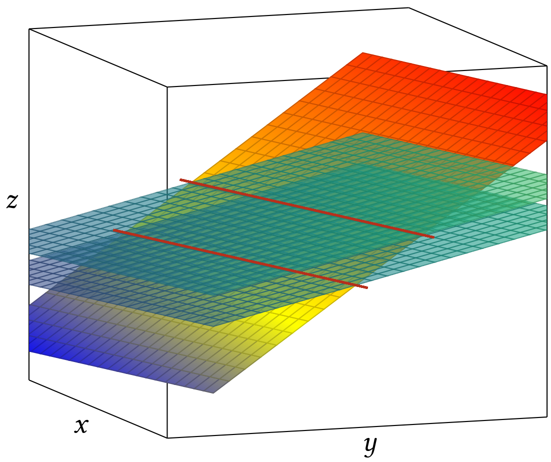


Figure 1.8: Depiction of the system (1.10).

can easily be computed to be

$$\begin{aligned} x + 2y - z &= -1 \\ -7y + 4z &= 6\,. \\ 0 &= 5 \end{aligned}$$

Notice the likeness to the echelon form of the previously studied system (1.10). One equation without solution, two solvable. Its visual representation is to be found in

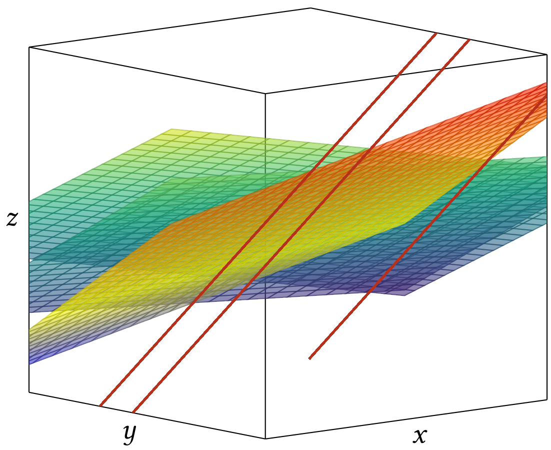


Figure 1.9: Example of the 'tent' arrangement of planes in (1.11).

Without delving into visualisations of linear systems in three variables and more than three equations – which do not actually bring anything new to the table – we conclude the section with a few exercises.

Exercise 1.2.3

Draw the following linear systems.

$$\begin{aligned} 2x + y &= 1 \\ 3x + 2y &= 3 \end{aligned} \qquad \begin{aligned} -x + y &= 2 \\ 2x - 2y &= 5 \end{aligned} \qquad \begin{aligned} -x - y &= 1 \\ 3x + 2y &= 0 \end{aligned}$$

Exercise 1.2.4

Without depicting them visually, determine the arrangement of planes corresponding to the linear system below.

$$\begin{aligned} 2x - y + z &= 3 \\ x - 3y + 4z &= 1 \\ x + 2y - 3z &= 2 \end{aligned}$$

Exercise 1.2.5

Find linear systems in three variables and three equations corresponding to cases (1), (4) and (5) in the text above.

1.3 Describing Solution Sets of Linear Systems

In section 1.2, we studied specific (simple) classes of linear systems and touched upon a few important concepts, including, but not limited to, *parameters*, *free variables*, *underdetermined* and *overdetermined* systems.

We continue down this road and bring a general description of solution sets of linear systems. Before we formulate the result we shall endeavour to prove in this section, we introduce a few pieces of notation which are going to allow us to manipulate linear systems more efficiently. Do note that behind these mere 'pieces of notation' there lies hidden a much deeper geometric meaning, to be uncovered in later chapters.

Definition 1.3.1 (Matrix)

An $m \times n$ *matrix* is an array of numbers with m rows and n columns. The numbers are then called *entries* of the matrix.

Matrices allow us to write linear systems in a much more succinct manner. For example, the system

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

can be written using a matrix like this:

$$\left(\begin{array}{cc|c} -1 & 1 & 2 \\ 2 & -2 & 5 \end{array}\right),$$

abusing the fact that the same variables are piled in a single column and each row is a single linear equation. The bar on the right side simply serves to divide left sides of the equations from right ones.

Matrices make (amongst other things) Gauss-Jordan elimination easier to perform and keep track of its progress. The matrix of the eliminated system looks like this

$$\left(\begin{array}{cc|c} -1 & 1 & 2 \\ 0 & 0 & 9 \end{array}\right)$$

and has been reached by the row operation $\mathtt{II} + 2\mathtt{I}$.

Certain matrices are special (for reasons soon to be revealed) and we call them *vectors*.

Definition 1.3.2 (Vector)

A *column vector* is an $n \times 1$ matrix (that is, matrix with a single column) and a *row vector* is a $1 \times n$ matrix (a matrix with a single row). As column vectors are the 'default', we call them simply *vectors*.

There exists an obvious bijection between tuples $(v_1, \ldots, v_n)$ and column vectors $\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$. Consequently, we say that a vector $\mathbf{v}$ with entries $v_1, \ldots, v_n$ *solves* a linear equation

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = c$$

if the tuple $(v_1, \ldots, v_n)$ does.

The addition of vectors and their multiplication by a number are defined naturally.

Definition 1.3.3 (Adding vectors)

Given vectors

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \quad \text{and} \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix},$$

their *sum* is defined as the vector

$$\mathbf{u} + \mathbf{v} \coloneqq \begin{pmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{pmatrix}.$$

Definition 1.3.4 (Multiplying vector by a number)

Given a vector

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

and a number c, the *scalar c-multiple of $\mathbf{v}$* is the vector

$$c\mathbf{v} \coloneqq \begin{pmatrix} cv_1 \\ cv_2 \\ \vdots \\ cv_n \end{pmatrix}.$$

The multiplying number c is often referred to as a *scalar*.

We further need to discuss the concept of *free variables* and *parameters*.

In the previous section, we described the solution set of the system (1.7) using three different ways. In each case, two of the variables were independent and the third was their linear combination. We style the two independent variables, *parameters*. Vaguely said, a *parameter* is a variable on the value whereof other variables depend.

The question arises: 'Which variables to choose as *parameters*?' The answer descends: 'Why, of course, my child, choose the *free variables*!' After the process of Gauss-Jordan elimination, a preceding row always has more variables present than its neighbour downstairs. Occasionally, the number of additional variables is larger than one. It is clear that in such cases, back-substitution cannot determine the values of those additional variables exactly (as it leads to a linear equation in more than one variable). All save one of those variables are to be chosen as *parameters* and serve the noble purpose of describing the value of the last variable standing. Custom dictates that all but the leftmost variable in such a row are labelled *free* and the leftmost variable called a *pivot*. In light of this, the heavenly answer can be decrypted – the *free* variables shall serve as *parameters* and the value of the *pivot* be written as a linear combination of free variables.

To understand explicitly the preceding paragraph, consider the eliminated system

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 3 & 1 \\ 0 & 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Row `II` has two more variables than row `III` and row `I` also wins by two variables over row `II`. As the holy text states, the variable x_4 of the fourth column is *free*, whereas x_3 is a *pivot*. Therefore, x_4 now serves as a parameter and from row `II` we get the relation

$$x_3 = -x_4 + 4.$$

Row `I` brings in a new free variable – x_2 – and a new pivot – x_1. Using the fact that x_3, the pivot from row `II`, is already expressed as a linear combination of free variables, we substitute into row `I` to get

$$x_1 + 2x_2 - (-x_4 + 4) + 3x_4 = 1.$$

A tiny bit of cheap computation yields

$$x_1 = -2x_2 - 4x_4 + 5.$$

Thereby, all the pivots of the system are expressed as linear combinations of free variables. The set of solutions of this system can be described as the set of quadruples $(-2x_2 - 4x_4 + 5, x_2, -x_4 + 4, x_4)$.

Visualisation of the concepts of pivots and free variables is provided in figure 1.10.

Using vectors, the solution set of the currently studied system can be expressed quite elegantly. First, the quadruple $(-2x_2 - 4x_4 + 5, x_2, -x_4 + 4, x_4)$ corresponds to the column vector

$$\begin{pmatrix} -2x_2 - 4x_4 + 5 \\ x_2 \\ -x_4 + 4 \\ x_4 \end{pmatrix}.$$

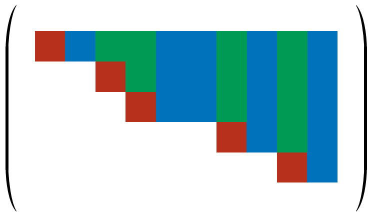


Figure 1.10: Visual depiction of an eliminated matrix. Red variables are pivots, blue ones are free and green ones are pivots from lower rows.

This vector can be further broken down into three vectors, two for the free variables and one for the constants. Explicitly,

$$\begin{pmatrix} -2x_2 - 4x_4 + 5 \\ x_2 \\ -x_4 + 4 \\ x_4 \end{pmatrix} = x_2 \begin{pmatrix} -2 \\ 1 \\ 0 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} -4 \\ 0 \\ -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 5 \\ 0 \\ 4 \\ 0 \end{pmatrix}.$$

Take note that the last vector is a *particular* solution of the system obtained by setting $x_2 = x_4 = 0$. Adding random multiples of the vectors

$$\begin{pmatrix} -2 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -4 \\ 0 \\ -1 \\ 1 \end{pmatrix}$$

to this particular solution generates more solutions of the system.

Let's make another example, shall we? In this eliminated system of two equations in three variables,

$$\left(\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 0 & 2 & 2 \end{array}\right),$$

the variables x_1 and x_3 are pivots and x_2 is free. Judging from the previous example, we should be able to express its solution as $\mathbf{u} + x_2\mathbf{v}$ where $\mathbf{u}$ and $\mathbf{v}$ are vectors and, furthermore, $\mathbf{u}$ is some particular solution of the system at hand.

Indeed, choosing x_2 to be a parameter, back-substitution yields $x_3 = 1$ and $x_1 = 2 - x_2 + x_3 = -x_2 + 3$. Hence, every vector of the shape

$$\begin{pmatrix} -x_2 + 3 \\ x_2 \\ 1 \end{pmatrix} = x_2 \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

solves the system.

We're now equipped to formulate a result about the 'shape' of a linear system's solution set with a rather far-reaching importance.

Theorem 1.3.5 (Solution set of a linear system)

The solution set of every linear system can be written in the form

$$\{\mathbf{u} + t_1\mathbf{v_1} + t_2\mathbf{v_2} + \ldots + t_l\mathbf{v_l}\},$$

where $\mathbf{u}$ *is a particular solution,* $\mathbf{v_1}, \ldots, \mathbf{v_l}$ *are vectors and* $t_1, \ldots, t_l$ *are parameters corresponding to the free variables of the eliminated system.*

Before the proof, we formulate an immediate corollary.

Corollary 1.3.6 (Number of solutions of a linear system)

Every linear system has zero, one or infinitely many solutions.

Proof. Referring to the form of the solution set of a linear system from theorem 1.3.5, we distinguish three cases:

(1) The vector $\mathbf{u}$ doesn't exist, therefore the system has *no solution.*

(2) The vector $\mathbf{u}$ exists and there are no free variables (only pivots) in the eliminated system. In this case, the solution is *unique.*

(3) The vector $\mathbf{u}$ exists and there is at least one free variable to be found in the eliminated system. In this case, the substitution of any number in place of the free variables generates a solution. Hence, there are *infinitely many.* ■

On our way to the proof of theorem 1.3.5, we make a preparatory step. We call a linear system *homogeneous* if the right side of its every equation is 0. Concretely, a *homogeneous* linear system assumes the form

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & 0 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & 0 \\ & & & & & & & \vdots & \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & 0. \end{array}$$

Notice that this system always has at least one solution, namely the vector $\mathbf{0}$ – the vector whose every entry is 0. We shall first prove the following proposition.

Proposition 1.3.7 (Solution set of a homogeneous linear system)

The solution set of a homogeneous *linear system can be written in the form*

$$\{t_1\mathbf{v_1} + t_2\mathbf{v_2} + \ldots + t_l\mathbf{v_l}\},$$

where $\mathbf{v_1}, \ldots, \mathbf{v_l}$ *are vectors and* $t_1, \ldots, t_l$ *are parameters corresponding to the free variables of the eliminated system.*

Proof. We consider a homogeneous linear system as above:

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \dots & + & a_{1,n}x_n & = & 0 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \dots & + & a_{2,n}x_n & = & 0 \\ & & & & & & & \vdots & \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \dots & + & a_{m,n}x_n & = & 0. \end{array} \tag{1.12}$$

Firstly, in the light of theorem 1.1.2, we may assume that the system has been reduced to echelon form. We shall prove that every pivot can be written as a linear combination of free variables by induction on the number k of rows (counting from the bottom) already substituted into. This approach basically mimics and formalises the traditional back-substitution process.

Without loss of generality, we may also assume that no rows full of zeroes are left at the bottom of the system, as those can be ignored. Hence, the last row of the eliminated linear system looks like this:

$$a_{m,j}x_j + a_{m,j+1}x_{j+1} + \dots + a_{m,n}x_n = 0$$

for adequate $1 \leq j \leq n$ and $a_{m,j} \neq 0$. Here, x_j is the pivot and $x_{j+1}, \dots, x_n$ are free. This gives the expression

$$x_j = -\frac{1}{a_{m,j}}(a_{m,j+1}x_{j+1} + \dots + a_{m,n}x_n)$$

of the pivot x_j as a linear combination of the free variables $x_{j+1}, \dots, x_n$. So, the result holds for $k = 0$.

Now, supposing all pivots in the last k rows of the system (1.12) have been expressed as linear combinations of free variables, we write the pivot of the $(m-k)$-th row (or $(k+1)$-st from the bottom) also as a linear combination of free variables. Again, there exists some smallest $1 \leq i \leq n$ such that $a_{m-k,i} \neq 0$. The $(m-k)$-th row is thus

$$a_{m-k,i}x_i + a_{m-k,i+1}x_{i+1} + \dots + a_{m-k,n}x_n = 0.$$

Performing an analogous computation gives

$$x_i = -\frac{1}{a_{m-k}}(a_{m-k,i+1}x_{i+1} + \dots + a_{m-k,n}x_n). \tag{1.13}$$

All the variables found on the right side of (1.13) are either free or pivots from lower rows. However, by the induction hypothesis, all pivots from lower rows have already been expressed as linear combinations of free variables. Simple substitution now yields an expression of x_i as a linear combination of free variables. With l denoting the number of free variables of the eliminated system and with the solution vector having been split into a sum of scalar multiples of free variables, the result is proven. ■

Remark 1.3.8

By proposition 1.3.7 above, a *homogeneous linear system* has either one or infinitely many solutions since the n-tuple $(0, 0, \dots, 0)$ always solves it.

Example 1.3.9

The echelon form of the homogeneous linear system

$$\left(\begin{array}{ccc|c} 1 & 4 & 2 & 0 \\ -2 & -3 & 1 & 0 \\ 3 & 7 & 1 & 0 \end{array}\right)$$

is equal to

$$\left(\begin{array}{ccc|c} 1 & 4 & 2 & 0 \\ 0 & 5 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Hence, the variables x_1 and x_2 are *pivots* and x_3 is *free*. Back-substitution gives $x_2 = -x_3$ and $x_1 = 2x_3$. The solution set of this system is thus given by all the vectors

$$\begin{pmatrix} 2x_3 \\ -x_3 \\ x_3 \end{pmatrix} = x_3 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}.$$

In the notation of proposition 1.3.7, we'd have

$$t_1 = x_3 \quad \text{and} \quad \mathbf{v}_1 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}.$$

We've reached the climax of the section – the proof of theorem 1.3.5. Armed with proposition 1.3.7, it behoves us to merely work a link between the solution set of a linear system and its corresponding homogeneous system.

Proof (of theorem 1.3.5). Let

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \dots & + & a_{1,n}x_n & = & c_1 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \dots & + & a_{2,n}x_n & = & c_2 \\ & & & & & & & \vdots & \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \dots & + & a_{m,n}x_n & = & c_m. \end{array} \tag{1.14}$$

be the linear system in question. We proceed to show that its every solution is of the form $\mathbf{u}+\mathbf{h}$, where $\mathbf{u}$ is a particular solution and $\mathbf{h}$ is a solution of the corresponding homogeneous linear system (as in (1.12)), and that, contrariwise, for every solution $\mathbf{h}$ of the homogeneous linear system, the vector $\mathbf{u}+\mathbf{h}$ solves the system (1.14), assuming $\mathbf{u}$ does.

Let's start with the former. Denote by $\mathbf{u}$ any fixed solution of system (1.14). We want to show that any other solution $\mathbf{v}$ of the same system can be written as $\mathbf{v} = \mathbf{u} + \mathbf{h}$ where $\mathbf{h}$ solves the corresponding homogeneous linear system (1.12). For this, it is clearly enough to show that $\mathbf{h} \coloneqq \mathbf{v} - \mathbf{u}$ solves the system (1.12). Substituting $x_i \coloneqq h_i = v_i - u_i$ into the left side of the j-th equation of (1.12) yields

$$a_{j,1}(v_1 - u_1) + a_{j,2}(v_2 - u_2) + \dots + a_{j,n}(v_n - u_n),$$

which can be broken into

$$(a_{j,1}v_1 + \dots + a_{j,n}v_n) - (a_{j,1}u_1 + \dots + a_{j,n}u_n).$$

As both **u** and **v** solve (1.14), we know that

$$a_{j,1}v_1 + \ldots + a_{j,n}v_n = c_j = a_{j,1}u_1 + \ldots + a_{j,n}u_n,$$

and thus

$$(a_{j,1}v_1 + \ldots + a_{j,n}v_n) - (a_{j,1}u_1 + \ldots + a_{j,n}u_n) = c_j - c_j = 0.$$

This is true for all $1 \le j \le m$, hence the result.

As for the inverse inclusion, we must show that $\mathbf{u}+\mathbf{h}$ where $\mathbf{h}$ is an arbitrary solution of (1.12) also solves (1.14), assuming that **u** solves it. This time, we substitute $x_i \coloneqq u_i + h_i$ into the left side of the j-th equation of (1.14) and get

$$a_{j,1}(u_1 + h_1) + \ldots + a_{j,n}(u_n + h_n) = (a_{j,1}u_1 + \ldots + a_{j,n}u_n) + (a_{j,1}h_1 + \ldots + a_{j,n}h_n). \quad (1.15)$$

We know that

$$\begin{aligned} a_{j,1}u_1 + \ldots + a_{j,n}u_n &= c_j, \\ a_{j,1}h_1 + \ldots + a_{j,n}h_n &= 0. \end{aligned}$$

Thus, the expression (1.15) equals $c_j + 0 = c_j$ and $\mathbf{u} + \mathbf{h}$ solves the j-th equations of (1.14). Again, this being true for all $1 \le j \le m$ proves this inclusion and with it, the theorem. ■

Example 1.3.10

We change the right side of the system from example 1.3.9 to produce

$$\left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ -2 & -3 & 1 & 0 \\ 3 & 7 & 1 & 5 \end{array}\right),$$

and after elimination:

$$\left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ 0 & 5 & 5 & 10 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Typical back-substitution yields $x_2 = 2 - x_3$ and $x_1 = -3 + 2x_3$. The solution can thus be written as

$$\begin{pmatrix} -3 \\ 2 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix},$$

where $\mathbf{u} = \begin{pmatrix} -3 \\ 2 \\ 0 \end{pmatrix}$ is a particular solution of the system and $\mathbf{h} = x_3 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$ is for any x_3 a solution of the corresponding homogeneous linear system from example 1.3.9.

Exercise 1.3.11

Solve each of the systems below using matrix notation. Write the solution in the form of theorem 1.3.5.

$$\begin{aligned} 3x + 6y &= 18 \\ x + 2y &= 6 \end{aligned} \qquad \begin{aligned} x + y &= 1 \\ x - y &= -1 \end{aligned} \qquad \begin{aligned} x_1 + 2x_2 - x_3 \phantom{{}+x_4} &= 3 \\ 2x_1 + x_2 \phantom{{}-x_3} + x_4 &= 4 \\ x_1 - x_2 + x_3 + x_4 &= 1 \end{aligned}$$

Exercise 1.3.12

Show that any five points in the plane $\mathbb{R}^2$ lie on a common *conic section*, that is, they all satisfy an equation of the form

$$ax^2 + by^2 + cxy + dx + ey + f = 0$$

for some $a, \dots, f \in \mathbb{R}$.

Exercise 1.3.13

Prove that if $\mathbf{s}$ and $\mathbf{t}$ are solutions of a homogeneous linear system, then so are

(1) $\mathbf{s} + \mathbf{t}$,

(2) $3\mathbf{s}$,

(3) $k\mathbf{s} + m\mathbf{t}$ for any numbers k, m.

What is wrong with the following argument: 'These three show that if a homogeneous system has one solution, then it has many solutions – any multiple of a solution is another solution, and any sum of solutions is also a solution – so there are no homogeneous linear systems with exactly one solution.'?

1.4 Applications

In this final section of chapter 1, we focus on some 'real-world' applications of linear systems and, more generally, on methods of solving linear systems using computers.

The software we shall employ toward this end styles SageMath. It's a free open-source mathematics software capable of numeric and symbolic manipulation of objects from various fields of mathematics, linear algebra included. It can be installed on most operating systems following the official guide.

SageMath is essentially a terminal-based software and out of the box offers no graphical user interface. Upon launch, the user is greeted by a screen similar to this one:

```
┌────────────────────────────────────────────────────────────────────┐
│ SageMath version 10.4, Release Date: 2024-07-19                    │
│ Using Python 3.12.8. Type "help()" for help.                       │
└────────────────────────────────────────────────────────────────────┘
sage:
```

SageMath is mainly built upon C and Python and is *interpreted*, meaning every piece of code is immediately run without a need for compilation.

Before we focus on applications of linear systems in fields like *economics* and *physics*, we need to learn to solve them using SageMath. By far the simplest way to encode linear systems is using matrix notation. SageMath features the `Matrix` class which hosts a plethora of methods for matrix manipulation we are going to make great use of in time.

Example 1.0.8 contains the system

$$\left(\begin{array}{cc|c} 15 & 40 & 100 \\ 25 & -50 & 50 \end{array}\right).$$

Let us solve it using SageMath. The **Matrix** class expects a matrix to be defined as a list of rows which are themselves lists of elements. In addition, we may specify the number set wherein the elements lie. For example,

```
sage: A = Matrix(ZZ, [
....:     [15, 40],
....:     [25, -50],
....: ])
```

creates the matrix

$$\begin{pmatrix} 15 & 40 \\ 25 & -50 \end{pmatrix}$$

with entries in $\mathbb{Z}$, the integers. This has a caveat. When we tell SageMath our matrix contains entries *exclusively* in $\mathbb{Z}$, it will fulfil our wish with utmost conscientiousness. This means that `A` can never contain anything but integers. A problem might emerge should we wish to put it into echelon form for example. Gauss-Jordan elimination of the matrix `A` would clearly require subtracting $(25/15)$-multiple of row **I** from row **II**. Assuming the entries of `A` are solely integers, such an operation is not permitted. The **Matrix** class has an in-built method for Gauss-Jordan elimination. Let us try to use it.

```
sage: A.echelon_form()
[  5 130]
[  0 350]
```

The result is somewhat unexpected. Thankfully or unfortunately, SageMath is clever enough to know that simply following the algorithm of the Gauss-Jordan elimination does not yield an integer matrix. So, it instead follows the algorithm and then multiplies the matrix by the least common multiple of the denominators of all entries in order to yield an integer matrix. Beware however, that trying to solve linear systems whose solutions are rational with integer matrices might result in an error. To stay in the clear, we instead use the real numbers throughout the calculation. Not specifying the number set would lead to SageMath 'guessing' it based on the values of the entries – which are all integral.

We thus rewrite our matrix `A` like this:

```
sage: A = Matrix(RR, [
....:     [15, 40],
....:     [25, -50],
....: ])
```

We will also create a **vector** (a **Matrix** with a single row basically) of the right hand side of the studied system.

```
sage: b = vector(RR, [100, 50])
```

The **Matrix** method for solving a system with a given **vector** of right hand side is called **solve_right**. Using it gives

```
sage: A.solve_right(b)
(4.00000000000000, 1.00000000000000)
```

Since we explicitly required SageMath solve the system over the real numbers, it returned the solution as a pair of decimals rounded based on a default precision parameter. We would instead prefer to write the solution as $(4, 1)$. Should we wish to record the solution as a pair of fractions or integers instead, we would need to define `A` and `b` over $\mathbb{Q}$.

```
sage: A = Matrix(QQ, [
....:     [15, 40],
....:     [25, -50],
....: ])
sage: b = vector(QQ, [100, 50])
sage: A.solve_right(b)
(4, 1)
```

1.4.1 Numerical Stability

Numerical stability (of a linear system) refers to one of its computational qualities – the quality described often as ‘small change in input causes a small change in output’. As real numbers are represented in computer memory with a given precision (more or less the number of decimal places), deviations in input data small enough to go unnoticed may cause issues. We shall highlight two of said ‘issues’ (and possible countermeasures) in this subsection.

Consider the system

$$\begin{aligned} 2x + y &= 3 \\ 2x + y &= 3 \end{aligned} \tag{1.16}$$

with infinitely many solutions of the form $((3 - y)/2, y)$. Now, altering the system slightly

$$\begin{aligned} 2x + {} & y = 3 \\ 2.000000002x + {} & 1.000000001y = 3.000000003 \end{aligned}$$

yields a system with exactly one solution – $(1, 1)$. We see that immediately but a computer with limited precision might regard this altered system exactly the same way as the previous one. Should we draw the system, we would basically see just one line given that the size of the angle between the lines corresponding to the two equations is negligible.

Systems where two or more equations are indistinguishable with low enough precision are typically called *ill-conditioned.* In this case, there is not much that can be done to alleviate the problem. See for yourself.

```
sage: A = Matrix(RR, [
....:     [2, 1],
....:     [2 + 2*10**-18, 1 + 10**-18],
....: ])
sage: b = vector(RR, [3, 3 + 3*10**-18])
sage: A.solve_right(b)
(1.50000000000000, 0.000000000000000)
```

The solution given by SageMath is clearly wrong because of the tiny deviation in input data. It instead computed the solution to the system (1.16) and substituted $y = 0$, which is default behaviour.

Next, we take a look at the system

$$\begin{aligned} \tfrac{1}{1000}x + y &= 1 \\ x - y &= 0 \end{aligned}$$

with unique solution $(1000/1001, 1000/1001)$. Here, depending on the order of the equations, computers can arrive at a wrong solution. In the first step of Gauss-Jordan elimination, we subtract a 1000-multiple of row **I** from row **II**, obtaining

$$\begin{aligned} \tfrac{1}{1000}x + \quad\; y &= \quad 1 \\ -1001y &= -1000. \end{aligned} \tag{1.17}$$

Even if we are working with enough precision to represent thousandths of integers, the result of the computation

$$y = \frac{-1000}{-1001}$$

may easily be rounded to 1 due to the way computers perform division. As three decimal places are hardly enough to push modern computers to their limits, see the following example instead.

```
sage: a = -1 * 10**18
sage: b = -1 * 10**18 - 1
sage: numerical_approx(a / b)
1.00000000000000
```

The `numerical_approx` function tells SageMath to represent a/b as a real number, otherwise it would have stored it as a fraction.

Should we now begin the process of back-substitution in the system (1.17), we would inevitably get a wrong solution. If the second equation yields (with low precision) that $y = 1$, then from the first equation, we get $x = 0$. This is a *completely* different solution from the exact one. The difference between $(0, 1)$ and $(1000/1001, 1000/1001)$ might not seem too high but imagine x and y represented *percentages* for example. Then, instead of both x and y being nearly 100%, x gets smashed down all the way to 0%.

Perhaps a little surprisingly, this problem can be *thoroughly* solved by simply changing the order of the equations. If we had instead used Gauss-Jordan elimination to solve the system

$$\begin{aligned} x - y &= 0 \\ \tfrac{1}{1000}x + y &= 1, \end{aligned}$$

we wouldn't have run into any issues. Indeed, the first step here entails subtracting $(1/1000)$-multiple of row **I** from row **II**. This yields

$$\begin{aligned} x - \quad y &= 0 \\ \tfrac{1001}{1000}y &= 1. \end{aligned}$$

This time, even if $1001/1000$ *does* get rounded to one, the exact solution will still be reached with sufficient degree of accuracy. Supposing the second equation is evaluated to be true if $y = 1$, the first equation then gives $x = 1$. Clearly, the number $1000/1001$ is much closer to 1 than it is to 0.

All in all, there exist cases where additional steps performed during Gauss-Jordan elimination greatly increase the accuracy of the approximation of potential solutions of a linear system. One very simple and statistically effective method is to always swap the row which is to be used for elimination of other rows with the row with highest (in absolute value) pivot coefficient. The reason this works is that computers are, vaguely speaking, prone to rounding numbers that *are not* close to 0. This method is exactly what we employed here, by the way. Instead of solving

$$\begin{aligned} \tfrac{1}{1000}x + y &= 1 \\ x - y &= 0 \end{aligned}$$

we swapped row `I` with row `II` as row `II` has a 1000-times larger coefficient of the variable x than row `II`. In the next section, we intend to show how Gauss-Jordan elimination can be coded in SageMath while also including the aforementioned 'accuracy-improving' step.

> **Exercise 1.4.1**
>
> Devise a linear system the accuracy of the solution whereof suffers from insufficient precision but falls into neither of the two categories described.

1.4.2 Gauss-Jordan Elimination Revisited

Here, we provide a fully algorithmic description of the Gauss-Jordan elimination algorithm discussed in section 1.1 and also one possible way of encoding it in SageMath. Here goes nothing.

Algorithm 1: Gauss-Jordan Elimination.

input : An $n \times m$ matrix $A = (a_{i,j})_{i=1,j=1}^{n,m}$ with real entries.
output: The matrix A in echelon form.

```
/* Row to be used for elimination of other rows.                    */
r ← 1;
/* Traverse the columns.                                            */
for c ∈ {1, ..., m} do
    /* Find the row (below r) with maximal value in column c. Denote by
       b the row with the maximal currently known value.            */
    b ← r;
    /* Traverse the rows below r.                                   */
    for i ∈ {r + 1, ..., n} do
        if |a_{i,c}| > |a_{b,c}| then
            /* Found a row with higher value in column c. Replace b with
               i.                                                   */
            b ← i;
    /* If a_{b,c} = 0, then move to next column since this column is full
       of zeroes.                                                   */
    if a_{b,c} = 0 then
        continue;
    swap rows with indices r and b;
    /* Eliminate variables in column c in all rows below r.         */
    for i ∈ {r + 1, ..., n} do
        for j ∈ {c, ..., m} do
            a_{i,j} ← a_{i,j} − (a_{i,c} / a_{r,c}) a_{r,j};
    /* Row r now contains the pivot in column c so it will remain the
       same for the rest of the algorithm. Move to the next row.    */
    r ← r + 1;
return A;
```

Example 1.4.2

Let's put the matrix

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

into echelon form using algorithm 1. At first, we have $r = 1$ and $c = 1$. Going through rows 2 and 3, we see that the number with the highest value in column 1 lies in row 3. Hence, we first swap row 1 with row 3.

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

Now begins the process of elimination. Since $r = 1$, the index i exhausts the set $\{2, 3\}$. For $i = 2$, we calculate $a_{i,c}/a_{r,c} = a_{2,1}/a_{1,1} = -1/4$. We thus subtract $(-1/4)$-multiple of row 1 from row 2.

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 0 & \frac{3}{2} & \frac{3}{2} \\ 2 & 0 & 1 \end{pmatrix}$$

Next, we set $i := 3$ and calculate $a_{i,c}/a_{r,c} = 1/2$; we then subtract $(1/2)$-multiple of row 1 from row 3.

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 0 & \frac{3}{2} & \frac{3}{2} \\ 0 & -1 & 0 \end{pmatrix}$$

Since all rows in column c below r have been eliminated, we move to the next row by setting $r := 2$. We also move to the next column via $c := 2$.

Now, the number with the largest absolute value in column c and all rows below (and including) r already lies in row r, so no swap is needed. We perform the elimination of row 3 by calculating $a_{3,c}/a_{r,c} = a_{3,2}/a_{2,2} = -2/3$ and subtracting the $(-2/3)$-multiple of row 2 from row 3.

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 0 & \frac{3}{2} & \frac{3}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

We move to the next row via $r := 3$ and the next column via $c := 3$. No further elimination takes place because there are no rows below row 3. The matrix A has been put into echelon form.

Before finally marching on, we present a way of implementing algorithm 1 in SageMath. This implementation almost aligns with the implementation of the algorithm in Python. Let us be responsible adults and break it into functions.

The function for finding the row with the highest pivot coefficient in the current row might look like this:

```
sage: def find_best_row(A: Matrix, cur_row: int, cur_col: int):
....:     best_row = cur_row
....:
....:     for row_below in range(cur_row + 1, len(A.rows())):
....:         if abs(A[row_below][cur_col]) > abs(A[best_row][cur_col]):
```

```
....:             best_row = row_below
....:
....:     return best_row
```

We also implement the function to eliminate the first non-zero element in all rows below a given row.

```
sage: def eliminate_rows(A: Matrix, cur_row: int, cur_col: int):
....:     for row_below in range(cur_row + 1, len(A.rows())):
....:         scalar = A[row_below][cur_col] / A[cur_row][cur_col]
....:         A[row_below] = A[row_below] - scalar * A[cur_row]
```

These are all the functions we need to cleanly implement Gauss-Jordan elimination in SageMath.

```
sage: def eliminate(A: Matrix):
....:     cur_row = 0
....:
....:     for cur_col in range(len(A.columns())):
....:         best_row = find_best_row(A, cur_row, cur_col)
....:         A[cur_row], A[best_row] = A[best_row], A[cur_row]
....:
....:         if A[cur_row][cur_col] == 0:
....:             continue
....:
....:         eliminate_rows(A, cur_row, cur_col)
....:         cur_row += 1
```

And, to wrap things up, a short application on the matrix from example 1.4.2.

```
sage: A = Matrix(QQ, [
....:     [2, 0, 1],
....:     [-1, 1, 1],
....:     [4, 2, 2],
....: ])
sage: eliminate(A)
sage: A
[  4   2   2]
[  0 3/2 3/2]
[  0   0   1]
```

1.4.3 Input-Output Analysis

A place where linear systems naturally flourish is *economics*. Put briefly, economy is a network of mutually influenced industries. An important observation is that this 'influence' is mostly of *linear* nature. We take as an example the *steel* and *automobile* industries. Both of these industries use its own output and the other industry's output to optimize production. The steel industry might use steel to produce factories, and use cars for the transport of goods between them. Similarly, the automobile industry uses its own cars to transports its other cars and uses steel to produce them in the first place. In economics, we're typically interested in predicting the future value of an industry. However, in cases like these, it isn't intuitively evident how the total value of steel used by external actors (meaning not the steel or automobile industries) would influence the system, for example.

Suppose we accumulated the following data:

	used by steel	used by auto	used by others	total
value of steel (in billions of $)	6.90	1.28	10.51	18.69
value of auto (in billions of $)	2.24	4.40	7.63	14.27

Table 1.1: The annual summary of the value of steel and automobile industries.

Based on this data, how ought we to attempt to predict the total values of steel and automobile industries based on shifting external demand? First and foremost, why do we care primarily about external demand? The answer is simple. As long as external demand stays stable, it is improbable that the automobile industry would suddenly produce more cars or that the steel industry more steel. It is indeed mostly individual customers and other affiliated industries which cause a change in production.

Suppose that the value of steel and automobile industries used externally in the next year shifts by d_s and d_a, respectively. How does this affect their total value? To answer this, we need observe that the steel and automobile industries form a linear system. Under the premise that the steel industry uses the same *fraction* of its own output and the automobile industry also uses the same fraction of the steel industry output as this year, we can predict its value next year (which we denote s) to equal

$$s = (6.90/18.69)s + (1.28/14.27)a + (10.51 + d_s).$$

This formula essentially says the obvious:

$$\begin{aligned}\text{next year's value of steel} &= \text{next year's value of steel used by steel}\\ &+ \text{next year's value of steel used by auto}\\ &+ \text{next year's value of steel used by others.}\end{aligned}$$

We are just predicting the next year values based on this year's ones while keeping the ratios of output distribution stable.

Similarly, the equation for the predicted next year's total automobile industry value (denoted a) is

$$a = (2.24/18.69)s + (4.40/14.27)a + (7.63 + d_a).$$

Both of these linear equations put together form the linear system

$$\begin{aligned} s &= (6.90/18.69)s + (1.28/14.27)a + (10.51 + d_s)\\ a &= (2.24/18.69)s + (4.40/14.27)a + (7.63 + d_a)\end{aligned}$$

An easy computation and rearrangement gives

$$\begin{aligned} (11.79/18.69)s - (1.28/14.27)a &= (10.51 + d_s)\\ -(2.24/18.69)s + (9.87/14.27)a &= (7.63 + d_a)\end{aligned}$$

As we did many a time already, we collect the equations into a matrix A and a vector b like so:

$$A = \begin{pmatrix} 11.79/18.69 & -1.28/14.27 \\ -2.24/18.69 & 9.87/14.27 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 10.51 + d_s \\ 7.63 + d_a \end{pmatrix}.$$

Fortunately, SageMath has in-built support for variables. We can thus let it solve the system for us and represent the solution in terms of variables d_s and d_a.

```
sage: var('ds da')
(ds, da)
sage: A = Matrix([
....:     [11.79/18.69, -1.28/14.27],
....:     [-2.24/18.69, 9.87/14.27],
....: ])
sage: b = vector([10.51 + ds, 7.63 + da])
sage: sol = A.solve_right(b)
(0.210776906804487*da + 1.62528755481273*ds + 18.6900000000000,
1.48231851778104*da + 0.281627945702251*ds + 14.2700000000000)
```

We can now easily get a solution for *concrete* values of d_s and d_a by using SageMath's symbolic substitution capabilities. For example, if we expect the external output value of automobile industry will rise by $d_a = 0.05$ and the external output value of steel will fall by 0.10, that is $d_s = -0.10$, we can calculate the predicted future total values of the industries by setting

```
sage: sol(da=0.05,ds=-0.10)
(18.5380100898590, 14.3159531313188)
```

In this case, we predict the total value of the steel industry to fall by about \$0.15 billion and the value of the automobile industry to rise by roughly \$0.045 billion.

Exercise 1.4.3

Predict next year's total productions of each of the three sectors of the hypothetical economy shown in table 1.2.

value of / **used by**	**farm**	**rail**	**shipping**	**others**	**total**
farm	25	50	100		800
rail	25	50	50		300
shipping	15	10	0		500

Table 1.2: The output data of a hypothetical economy.

if next year's external demands are as stated.

(a) 625 for farm, 200 for rail, 475 for shipping,

(b) 650 for farm, 150 for rail, 450 for shipping.

Can you solve the system with data presented in (a) and (b) simultaneously by making the given external demands into parameters?

1.4.4 Electric Networks

The final presented application comes from engineering. In figure 1.11, you can see a simplified version of a car's electric network.

A designer of this network must be able to answer questions similar to: 'How much electricity flows when both the hi-beam headlights and the brake lights are on?' Even very sophisticated electric networks can be analysed using Kirchhoff's laws and the theory of linear systems.

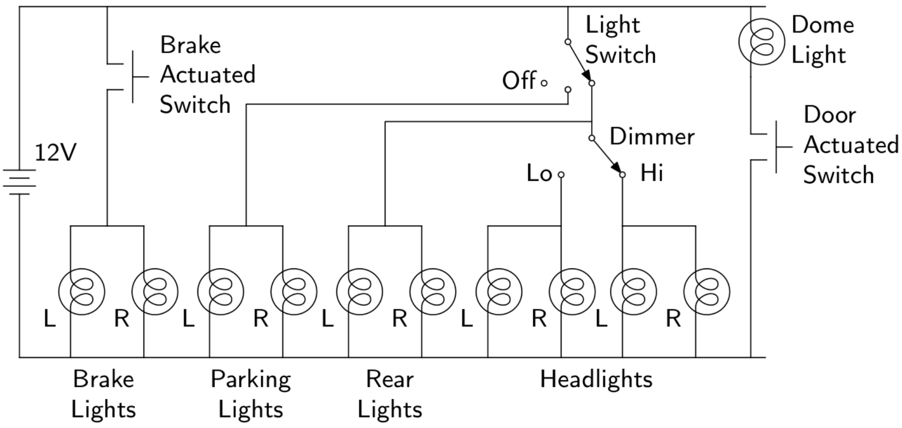


Figure 1.11: An excerpt from a car's electric network.

The intuitive explanation of electric circuits (which suffices for our purposes) tells that there are three interconnected forces at play – voltage (U), resistance (R) and current (I). At any point of the circuit, these are tied by the formula $U = RI$. The battery serves as a capacitor; it provides *voltage* – or electric potential – to the circuit, making electricity flow as long as there is a path. The moment a path is formed (we say the circuit is closed), the battery creates a force through the circuit – the *current*. Finally, some components of the circuit act as *resistors*, effectively limiting the amount of voltage that is 'available' to the subsequent components of the circuit. This limiting factor is the *resistance* of the component and is often proportional to the force provided by the battery. We can think of the resistors causing *voltage drops* throughout the current whilst the battery provides a *voltage rise*.

To interpret electric networks (basically meshes of electric circuits) as linear systems, two physical laws are needed – *Kirchhoff's Current Law* and *Kirchhoff's Voltage Law*. The former states that at any point in the network, the flow in equals the flow out. The latter then that around any circuit in the network, the total voltage rise equals the total voltage drop.

Let us start with a simple network consisting of a single circuit.

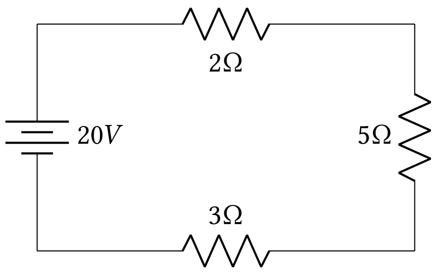


Figure 1.12: An electric circuit with a battery and three resistors.

The component represented by ⏚ is the battery and ⩘ depicts a resistor. We measure voltage provided by the battery in *volts* (V) and the resistance of the other components in *ohms* (Ω).

Since this network features only a single closed circuit, the current – measured in *amperes* (A) – is consistent throughout by Kirchhoff's Current Law. By Kirchhoff's Voltage Law, the total voltage rise (which is $20V$) equals the total voltage drop. In this circuit, there are three voltage drops, each equal to the resistance of the component times the current flowing through it. This gives us a linear system consisting of the single equation

$$20 = 2I + 5I + 3I$$

wherefrom we infer that $I = 2A$; the current around the circuit equals 2 amperes.

An example of a network leading to a more elaborate linear system requires connecting the resistors *in parallel* which automatically creates more circuits in the network.

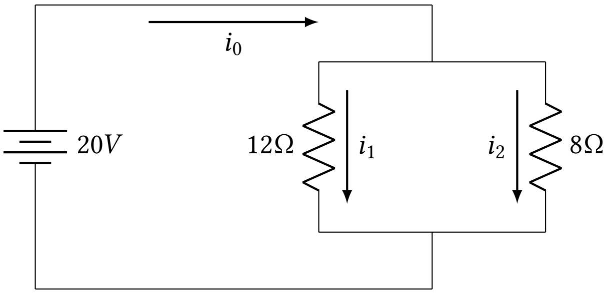
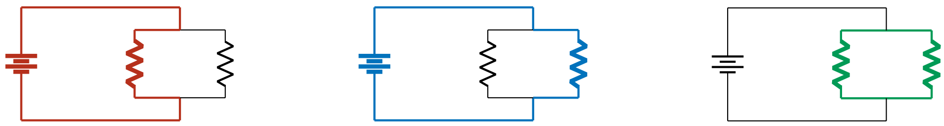


Figure 1.13: An electric network with resistors connected in parallel.

It might not look like it but the network in figure 1.13 actually hosts three circuits depicted in figure 1.14. Each of those circuits obeys Kirchhoff's Voltage Law. Spelt out for the first circuit, it



(a) First circuit. (b) Second circuit. (c) Third circuit.

Figure 1.14: The three circuits of an electric network.

says the total voltage rise of $20V$ must equal the total voltage drop of 12Ω times the current flowing through this circuit, which we labelled i_1. Similarly, the voltage rise in the second circuit is $20V$ and equals $8i_2$. Finally, the voltage rise in circuit three is $0V$ and equals the voltage drop through the first resistor plus the voltage drop through the second resistor. The only caveat here is the choice of orientation of the current. The current flowing through the first resistor must do so in direction opposite to the second resistor as the circuit forms a closed oriented loop. This means that one of the currents (for instance i_2) must be given a negative sign, signifying a direction of flow opposite to the one in circuit two. This gives a total voltage drop in the third circuit as $12i_1 - 8i_2$.

Finally, there are two points in the network where the flow splits. Applying Kirchhoff's Current Law thus awards two more equations: $i_0 = i_1 + i_2$ and $i_1 + i_2 = i_0$. All in all, we ended up with a linear system of five equations.

$$\begin{array}{rcrcrcr} & & 12i_1 & & & = & 20 \\ & & & & 8i_2 & = & 20 \\ & & 12i_1 & - & 8i_2 & = & 0 \\ i_0 & - & i_1 & - & i_2 & = & 0 \\ -i_0 & + & i_1 & + & i_2 & = & 0 \end{array} \tag{1.18}$$

Clearly, there are redundant equations in the system (1.18). This just goes to show that redundancy arises in practice and the problem of determining which equations are redundant is generally not entirely trivial; we shall discuss it later in the book.

In this case, of course, the first two equations already give us equalities $i_1 = \frac{5}{3}A$ and $i_2 = \frac{5}{2}A$. Finally, the fourth equation (or the fifth for that matter) ensures that $i_0 = \frac{25}{6}A$. Hence, the total current through the entire network is $\frac{25}{6}A$.

The final example to discuss is the so-called Wheatstone Bridge. There is *a lot* of circuits in this

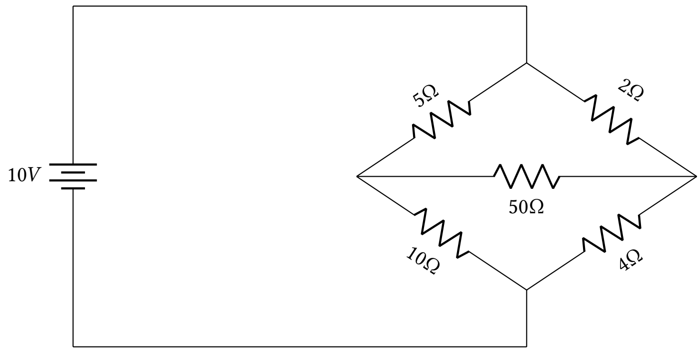


Figure 1.15: The Wheatstone Bridge network.

network. We first choose an arbitrary orientation of the currents through each of the branches as in figure 1.16.

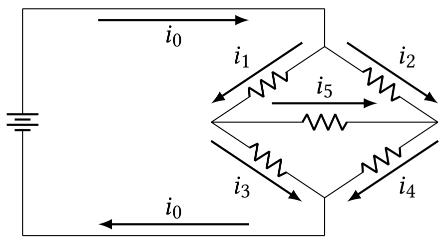


Figure 1.16: A choice of current orientation in the Wheatstone Bridge network.

We can't yet be sure how many or which equations we will need to calculate i_0 – the total current. We definitely need at least 6 given the number of variables. Kirchhoff's Current Law yields many equations but we (based mostly on intuition) pick these three:

$$\begin{aligned} i_0 &= i_1 + i_2 \\ i_3 + i_4 &= i_0 \\ i_2 + i_5 &= i_4 \end{aligned}$$

We've chosen these particular equations in a way that makes every variable appear at least once. Using Kirchhoff's Voltage Law on the inner, outer and the upper 'triangle-shaped' circuit gives respectively:

$$\begin{aligned} 10 &= 5i_1 + 10i_3 \\ 10 &= 2i_2 + 4i_4 \\ 0 &= 5i_1 + 50i_5 - 2i_2 \end{aligned}$$

Again, we have chosen these equations in order to make the resistance of every component appear at least once. Having collected the six equations into a linear system, we pray that we get a unique solution.

$$\begin{array}{rcrcrcrcrcrcr}
i_0 & - & i_1 & - & i_2 & & & & & & & = & 0 \\
-i_0 & & & & & + & i_3 & + & i_4 & & & = & 0 \\
& & & & i_2 & & & - & i_4 & + & i_5 & = & 0 \\
& & 5i_1 & & & + & 10i_3 & & & & & = & 10 \\
& & & & 2i_2 & & & + & 4i_4 & & & = & 10 \\
& & 5i_1 & - & 2i_2 & & & & & + & 50i_5 & = & 0
\end{array}$$

And... yes! We do. As SageMath confirms.

```
sage: A = Matrix(QQ, [
....:     [1, -1, -1, 0, 0, 0],
....:     [-1, 0, 0, 1, 1, 0],
....:     [0, 0, 1, 0, -1, 1],
....:     [0, 5, 0, 10, 0, 0],
....:     [0, 0, 2, 0, 4, 0],
....:     [0, 5, -2, 0, 0, 50],
....: ])
sage: b = vector(QQ, [0, 0, 0, 10, 10, 0])
sage: A.solve_right(b)
(7/3, 2/3, 5/3, 2/3, 5/3, 0)
```

A somewhat surprising fact about this solution is the equality $i_5 = 0$, meaning no electricity flows through the corresponding component.

Exercise 1.4.4

Figure 1.17 depicts an electric network.

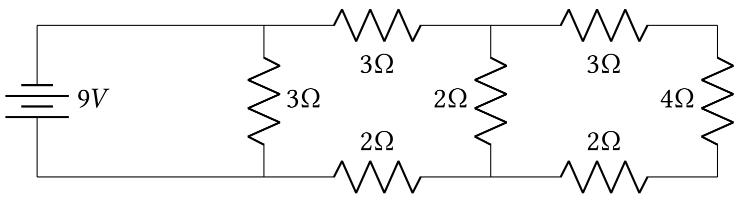


Figure 1.17: An electric network with 7 resistors.

Calculate the current in each branch of the network.