A Gentle Introduction to Linear Algebra

A Gentle Introduction to Linear Algebra

Joe Fields
Southern Connecticut State University

September 2016

Edition: version 0.1

Website: GILA on GitHub

© 2016 Joe Fields

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

To Martha.

Acknowledgements

A big "Thank You!" to the students at SCSU who inspired this project and helped to make it a reality. I'd also like to thank my colleagues in the SCSU Math department whose encouragement gave me the strength to get started on this. This book was created using only open-source tools: MathbookXML, LaTeX, GIT, GIMP, XFig, OpenSCAD and Sage; thank you to all of the developers and the communities that selflessly donate their time and efforts to building and maintaining these projects. Finally, I'd like to gratefully acknowledge the support provided by my institution (and hence, by the taxpayers of Connecticut) – a one semester sabbatical leave during which I did the bulk of the design and writing.

Foreword

To Do: Get the most eminent mathematician - that you can rope into it - to write a foreword.

Preface

This is a free/open-source textbook that provides an introduction to Linear Algebra. There is a light emphasis on mathematical proof and a strong emphasis on applications, making this book appropriate for science and engineering majors and for mathematics majors who are taking Linear Algebra early in their college careers. Math majors who have already had an introduction to proofs course – my other open-source textbook "A Gentle Introduction to the Art of Mathematics" was written for such a course – may benefit from a more advanced treatment that features proof. A good choice would be "A First Course in Linear Algebra" (a.k.a. FCLA) by Robert Beezer, which is also a free, open-source project.

Contents

Chapter 1

Introduction

The subject we are about to study, Linear Algebra, sounds like it might have something to do with lines and doing algebra with them. This is true if you are willing to think metaphorically... It might be somewhat closer to the truth if we were to say that Linear Algebra is about learning to understand higher dimensions. We'll be surprisingly far along in our study of the topic before we can precisely define what "dimension" actually means, but we expect that you have some notion already: the Euclidean plane where we studied Geometry is two dimensional, the world we live in is three dimensional, Albert Einstein taught us to view the world not just as space — but as space-time — a four dimensional concept. We needn't jump off into super advanced physics (or science fiction for that matter) in order to understand higher dimensionality. Dimension, at least informally, just means the number of real numbers it takes to describe something. Locating a point in three-dimensional space requires three numbers — usually x, y and z. If we are keeping track of aircraft, knowing where they are in 3-space is certainly necessary, but it might also be a good idea to keep abreast of *which way they are going*! To truly understand an aircraft's state, one needs to have six numbers: x, y and z, but also the velocity components x', y' and z'. This make the state of an airplane 6-dimensional. Perhaps this is why air traffic controllers make the big bucks.

The 6-dimensionality of an aircraft's state may seem somewhat artificial. Aren't we really just dealing with two separate 3-dimensional entities?

In Economics there is a high-dimensional entity known as the Leontief Input-Output model. In this model the state of an Economic system is described by a large number of real quantities, one for each sector of the economy. In a 1965 Scientific American article Wassily Leontief (who won a Nobel prize for this work) described his model in terms of a "toy example" where the economy was divided into 82 sectors. Today one could easily develop a Leontief I/O model where the economy was divided up into a million sectors. Perhaps this is why Economists make even bigger bucks.

When we do Linear Algebra in two dimensions we are indeed talking about lines. One of the classic problems is to figure out whether two lines intersect and if so, where. This is a situation where our ability to visualize things in two dimensions can lead us straight to the answer. That is certainly not the case in a million (or even in six) dimensions. Fortunately, there are calculational techniques that work (and even work fairly quickly on a good computer) in just about any number of dimensions you may be interested in.

There are three different ways of looking at linear algebra problems: systems of linear equations, vector equations, and transformations. These three views actually represent the same underlying structure, just in different ways.

There are various situations where one of these three viewpoints is preferable, so it is a good idea to be able to switch back and forth between these representations.

In Section 1.1 we will look at the same (really easy) problem from each of these 3 perspectives.

1.1 Getting started

The first problem we're going to look at is fairly trivial. I bet you can solve this in your head:

> I'm thinking of two numbers x and y. Their sum is 42, and their difference is 6. What are they?

This word problem can be instantly translated into a pair of equations. Later, when we have more sophisticated problems there may be many more unknown quantities and there may be many more equations. Here we are dealing with a system of equations having 2 equations in 2 unknowns.

$$\begin{aligned} x + y &= 42 \\ x - y &= 6 \end{aligned}$$

This one is about as easy as a system of two equations in two variables can get. Actually, that's not quite true. The *easiest* form for a system of two equations in two unknowns is if they basically just are statements of the answer, like:

$$\begin{aligned} x &= 24 \\ y &= 18. \end{aligned}$$

Solving a system of equations just means (somehow) transforming it from something like the first form to something like this latter form.

There are a small number of simple procedures that we can apply to systems without effecting their solutions. We can use these operations to convert almost any system into one that looks like that latter form (each equation just states what the value of some variable is). We'll get around to the full story in section 1.2, but for now, notice that if we add the two original equations together (adding equations means adding left sides and adding right sides separately) we get something that only involves x. And of course, once we know one of the variables it isn't very hard to find the other.

For this example problem, finding the solution was very easy. There are more difficult systems where finding the solution by hand would be challenging so we are going to want to become familiar with some kind of computer tools for automating these things. In this book, we'll be using Sage, a free, open-source, computer algebra system developed by William Stein. Here is a sample of how Sage can be used to solve a system of equations:

```
x, y = var('x, y')
solve([x+y==42, x-y==6], x, y)
```

```
[[x == 24, y == 18]]
```

We glossed-over a small but important issue in the above. How do we know that our answer was the only answer? And for that matter, is it necessarily true that there must *be* an answer to some system of equations? These are what are known as existence and uniqueness questions: Does there exist an answer to our problem? (Existence.) And, if there *is* an answer, how do we know it is the only answer? (Uniqueness.) There are systems of equations where all of the possible behaviors are exhibited: no solutions, unique solutions and lots of solutions.

Exercise 1.1.1. Explain why the following system has no solutions at all.

$$2x - y = 7$$
$$2y - 4x = 8.$$

Hint. Put both equations into slope-intercept form ($y = mx + b$).

That was a linear algebra problem seen from the "systems of equations" perspective. We still need to look at the "vector equations" and "transformations" viewpoints. So next we'll look at a question of the vector flavor. We're going to think about playing chess, not on a board, but on the infinite x—y plane.

Consider the piece known as a bishop. If you're not familiar with chess, this is the piece that can move in the diagonal directions. Think of the bishop as having two moves that it can do (but it can do them any number of times). It can do a move we'll refer to as UR; move one unit in the x direction while simultaneously moving one unit in the y direction — by doing this multiple times the bishop can travel in the upper right direction. It also has a move that allows it to travel along the other diagonal — move one unit in the x direction while simultaneously moving negative one unit in the y direction. We'll call that move LR.

For those who are familiar with chess, you'll know that bishops are forever trapped on the same color square — one of your bishops is always on black and the other always on white. This means that some "bishop moving questions" won't have solutions — for example, a bishop sitting at the origin, $(0, 0)$, can never move to $(0, 5)$; those squares have opposite colors! To get around this limitation we're going to let our bishops make fractional moves. For instance if it starts at the origin and makes $1/2$ of the upper-right move then it will arrive at $(1/2, 1/2)$. Now, getting a little stranger, we're going to also allow our bishops to make negative moves. Maybe we should think of a negative move as "undoing" a regular move...

In any case negative moves allow us to move the bishop in the opposite directions along the diagonals. Finally, we may as well give our bishops the freedom to move *any amount* — that is, any real number can be used as a so-called scalar, shrinking or stretching either of the two basic moves. Got it? We can do things like $\pi \cdot UR$ and $\sqrt{2} \cdot LR$.

So, after all that setup, here's the question: If a bishop starts at $(0, 0)$, can it make some number of UR and LR moves and wind up at $(42, 6)$? If so, how many URs and how many LRs?

The things we've been calling UR and LR are *vectors*. If you ask someone from the physical sciences to define a vector they'll say "it's a thing that has both a magnitude and a direction". (Which is fine as far as it goes.) Meteorology provides some nice examples. A weather map often shows a lot of basic data about the conditions at various places — wind, temperature, barometric pressure and humidity are common. Of these, only the wind is a vector quantity, it needs to be specified with both a magnitude and a direction (e.g. 15 mph out of the Northeast), the others all just have magnitudes.

There is a different way of thinking about what a vector is, that is preferable in many circumstances. A vector is the difference between two positions. Let me put this another way: a vector gives you a set of *directions* to go from one point to another. (I mean "directions" in the sense of the things someone tells you if you ask "How do I get to the Kwik-E-Mart from here?")

If you are currently at the point $(3, 4)$ and you want to move to the point $(5, 12)$ you need to increase your x-coordinate by 2 units and you must increase your y-coordinate by 8 units. We just described the vector $\langle 2, 8\rangle$, the numbers 2 and 8 are known as the components of the vector. Note that this is different in a not-so-subtle way from the *point* $(2, 8)$. The point is stationary, the vector is there to describe a change. If you start at the origin and follow the directions specified by the vector $\langle 2, 8\rangle$ you will of course wind up at the point $(2, 8)$, but if you start at some other point, it's equally obvious that you won't!. Sometimes people will talk about "position vectors" in this sort of context — the position vector $\langle x, y\rangle$ goes from the origin to the point (x, y). Generally, it is preferable to keep the distinction between points and vectors clear. When you treat a vector as a position vector (i.e. think of it as a point) you are loosing something. Ordinarily a vector is free; it can be slid around from one point to another so long as its components aren't changed.

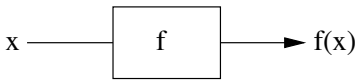
Here's how solving the vector variant of our problem might look in Sage:

```
x, y, u, v = var('x, y, u, v')
u = vector(QQ, [1, 1])
v = vector(QQ, [1,-1])
lhs = x*u+y*v
rhs = vector(QQ, [42,6])
solve([lhs[0]==rhs[0], lhs[1]==rhs[1]], x, y)
```

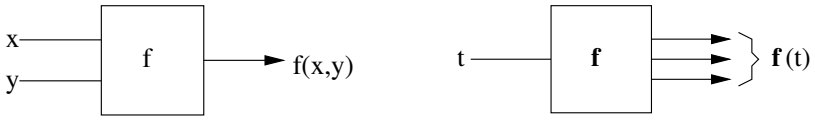
```
[[x == 24, y == 18]]
```

So, at this point we've looked at a simple linear algebra problem from the systems of equations perspective and from the vector equations perspective. The final perspective we want to illustrate is that of linear transformations.

Basically, a linear transformation is a function that takes vectors as inputs and spits out vectors as outputs. You're probably familiar with the following sort of diagram for functions.

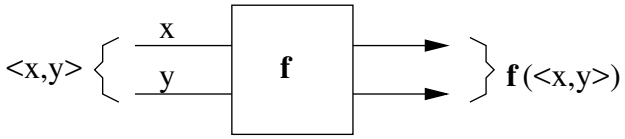


In Multivariable Calculus you may also encounter functions that are diagramed like so:



The first is a real-valued function of two variables — think of it as taking a vector as input and returning a scalar. The second is a vector-valued function of a single real variable. The mapping that gives temperature as a function of position on a metal plate is an example of the first sort. When we represent the position of a particle moving around in space (as a function of time) we are using the second sort.

Linear transformations are functions where there are vectors on both the input and the output side.



Moreover, linear transformations are *linear*, which means the components of the output are computed in a very simplistic way from the components of the inputs. The only things that are allowed are adding things up and multiplying by constants.

So let's give an example of a linear transformation. This will be a function that takes a vector $\langle x, y\rangle$ as input, and returns a vector $\langle u, v\rangle$ as output. We will compute u and v (the components of the output vector) from x and y (the components of the input vector by "adding things up and multiplying by constants":

$$u = x + y$$
$$v = x - y$$

By convention, people usually call a linear transformation T and use a notation that looks just like Euler notation for functions (because in fact, that's what it is!)

$$T(\langle x, y\rangle) = \langle u, v\rangle.$$

There are two kinds of problems one can ask: maybe you know the input vector and you'd like to find the output vector, or vice versa. When you've got the input it's very easy to find the output! You just plug in. The more interesting question is when it's vice versa, suppose you know that $\langle u, v\rangle = \langle 42, 6\rangle$ how can you arrive at the solution $\langle x, y\rangle = \langle 24, 18\rangle$? We'll be looking at this kind of thing in more depth in Section 1.4.

1.2 Systems of equations

In this section we'll look much more closely at the "systems of equations" approach to linear algebra.

First a few words about notation. When there are seventeen variables in a problem it becomes *really* awkward to use different letters for each variable. When there are a thousand variables it's impossible! We will follow the almost universal convention that the letter x will be used for the variables, with a subscript to identify which one. If we were to translate the problem from Section 1.1 into this notation it would become

$$x_1 + x_2 = 42$$
$$x_1 - x_2 = 6.$$

A *linear combination* of some set of numbers $\{x_1, x_2, \ldots, x_n\}$ is created by multiplying each of the x's by constants and then adding everything up. Of course if the constants are 1 or -1 (as in the previous example) we tend to forget that they're there!

Example 1.2.1. Consider $x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5$. This is a linear combination of the five variables $\{x_1, x_2, x_3, x_4, x_5\}$. The constants $(1, 2, 3, 4,$ and $5)$ are called the coefficients of the linear combination.

An equation is *linear* if it has the form of a linear combination set equal to some value on the right-hand side – or if it can be put into that form. For example

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 15$$

is a linear equation in five variables.
Also,

$$x_1 + 3x_3 = x_2 + x_4$$

is a linear equation (in four variables) because we can manipulate it into the form

$$x_1 - x_2 + 3x_3 - x_4 = 0.$$

Exercise 1.2.2. The linear equation

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 15$$

has a solution where all of the variables are set equal to 1. Are there others?

Hint. Try setting one of the variables to zero. That essentially eliminates that one and gives you a new equation with only four variables. Does the new equation have a solution?

A *system of equations* is just a collection of linear equations.

The notation for systems of equations gets a bit complicated when we try to write them in general (that is, without particular values given for the various constants involved). There are three sorts of things that need names in such a system: the variables, the coefficients of the variables, and the numbers on the right-hand sides. There is a convention that is fairly universal for the names and numbering of these elements. The variables are x's with subscripts, the right-hand sides are b's with subscripts, and the coefficients are a's with *two* subscripts (we need to indicate the equation that a given coefficient is in and also, which variable it is multiplying.

For example, here is how we would write the general form of a system of three equations in four unknowns:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \end{aligned}$$

Notice that the indices on the x's run from 1 to 4, the indices on the b's run from 1 to 3, and that there are a total of 12 coefficients.

Example 1.2.3 (Investments)**.** Suppose you have $\$10,000$ that you want to invest in the stock market. After some research you've found three companies that you think will be good investments. SolarCity Corp (SCTY) is trading at about \$20 per share. SunPower - Solar energy company (SPWR) is trading at about \$9 per share. First Trust Global Wind Energy (FAN) is at about \$12 per share. One equation you can immediately write down is

$$20x_1 + 9x_2 + 12x_3 = 10000,$$

where x_1 is the number of shares of SCTY we will buy, x_2 is the number of shares of SPWR, and x_3 is the number of shares of FAN.

If we said nothing further, we'd have just this one equation and there are many possible sets of values for the variables that satisfy it. Notice that there are two broad categories of companies represented in our stock picks – solar energy and wind power. Perhaps we'd be wise to split our investment between them based on some rational theory, for the sake of argument let's say that we've been advised to use a 60/40 split between solar and wind. What was previously a single equation is now two:

$$\begin{aligned} 20x_1 + 9x_2 &= 60000, \\ 12x_3 &= 40000. \end{aligned}$$

Notice that the second equation uniquely determines the value of x_3 but that the other variables still have a bit of freedom. (For instance, notice that we could set either x_1 or x_2 to 0, and the other variable's value would then be uniquely determined. Or, of course we could have some mixture where our $\$6,000$ is split up between the two companies. As it happens, these two companies are competitors and there is some probability that one will succeed and the other will fail. A wise investor tries to guess what that probability is and "hedge" their bets on the market. For the sake of argument let's say we think SCTY is three times more likely to come out the winner in this competition. You might be inclined to just buy only the SCTY stock, but that's not what a hedging strategy would indicate – you should mix your investments in a proportion that reflects the probabilities involved. As an equation in the x's we have

$$20x_1 = 3 \cdot 9x_2.$$

At this point we've obtained a system of 3 equations in 3 variables which, after manipulating the last one a little bit, looks like the following.

$$\begin{aligned} 20x_1 + 9x_2 &= 60000 \\ 12x_3 &= 40000 \\ 20x_1 - 27x_2 &= 0 \end{aligned}$$

It is usually a good idea to format your systems so that the variables in each equation line up in columns.

$$\begin{aligned} 20x_1 + \ 9x_2 \qquad &= 6000 \\ 12x_3 &= 4000 \\ 20x_1 - 27x_2 \qquad &= \quad 0 \end{aligned}$$

Solution. Now, let's go ahead and figure out what the values of the variables should be. In other words, how many shares of each stock should we purchase?

First, look at that middle equation. It isn't very complicated, indeed, it basically *tells* us the value of x_3 — we just need to divide both sides by 12 to

get that $x_3 = 333.\overline{3}$. Unfortunately, we can't buy fractions of a share of stock so we'll round to 333.

We're somewhat lucky in that the variable x_3 doesn't appear in the other equations, but *even if it did*, we could now substitute the value we just determined for it. Furthermore, at this point, we have no more use for that middle equation; we've used it up in finding the value of x_3. So now we've reduced our problem to a simpler system — one that consists of just two equations in the remaining two unknowns.

$$\begin{aligned} 20x_1 + \ 9x_2 &= 6000 \\ 20x_1 - 27x_2 &= \quad 0 \end{aligned}$$

If we subtract the first equation from the second we get

$$-36x_2 = -6000$$

and this tells us (just divide both sides by -36) the value of x_2.

What we've determined so far is that $x_3 = 333$ and $x_2 = 167$. By substituting those values into the very first equation we wrote down we'll be able to find the value of x_1.

After making those substitutions we get an equation that only has one variable:

$$20x_1 + 9 \cdot 167 + 12 \cdot 333 = 10000.$$

It's child's play to find the solution is $x_3 = 225$.

So in the end we should put in an order for 225 share of SCTY, 167 shares of SPWR and 333 shares of FAN. Notice that because of rounding we've come up one dollar short of our intended investment.

A bit more formalism is appropriate now. We'll start with some definitions.

Definition 1.2.4 (linear system)**.** A **linear system**, also known as a **system of linear equations** is a collection of m equations in n unknowns of the form

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

Note that the doubly-indexed quantities (a_{ij}) as well as the singly-indexed quantities (b_i) are real numbers and that the m variables are indicated by x's (with subscripts).

Remark 1.2.5. The use of variables with multiple indices in the above definition bears comment. First of all, note that we are trying to deal with the general situation where there is an unknown number of equations (m) in an unknown number of variables (n). Let's consider the b's first — these are the constants that appear on the right-hand sides of the equations, so there are m of them. The situation for the a's is more complicated. The a's are the coefficients, they are constant numbers that the variables are multiplied by, and there are two indices on each of them. The first index tells us which equation we are in. The second index matches with the subscript on the variable. For example a_{1423} would be the coefficient of x_{23} in the 14th equation in a system.

What does it mean to say we have found an "answer" to a system of equations? Essentially, it is this: we have found a set of values for the variables that "work" in all of the equations. Sometimes people say that this set of values "satisfies" the equations. To be completely clear, what is meant is that if one substitutes these values for the variables in the equations of the system, all of them (the equations) will be true. It is convenient to regard such a set of values as a vector. For example the solution we obtained in Example 1.2.3 would be regarded as the vector $\langle 225, 167, 333\rangle$.

Definition 1.2.6 (solution sets)**.** Given a system of m linear equations in n unknowns, the **solution set** of the system is the set of all vectors of length n that satisfy all m of the equations in the system.

Definition 1.2.7 (equivalent systems)**.** Two linear systems are called **equivalent** if and only if they have identical solution sets.

Remark 1.2.8. The equivalence of linear systems is an example of what is known as an equivalence relation. Equivalence relations are used in theoretical mathematics when we are trying to capture the notion that two things — while not *actually* equal — are similar enough that we can treat them as being sort of a junior version of equal...

For a relationship to earn the title "equivalence relation" it must have a short list of properties. These properties are certainly true of the ordinary equals sign:

reflexivity A relation is reflexive iff all elements are related to themselves.

symmetry A relation is symmetric iff whenever x and y are a pair of elements that are related, then y and x are also a pair that are related. (I.e. the order can always be reversed.)

transitivity Perhaps you've heard the phrase "Two things that are equal to a third must be equal to each other." That's the essence of transitivity.

There really is much more that we should say about equivalence relations in general and the consequences that ensue when we can show that some relation is an equivalence relation. We refer the interested reader to chapter 6 in GIAM. In the remainder of this book we are going to *see* how very useful the notion of equivalence of linear systems can be. Hopefully this will give you some indication of how useful equivalence relations in general can be!

One final word about equivalence relations (in general) and the equivalence of linear systems (in particular): It is customary, when introducing this notion, to ask students to come up with a proof that shows that some given relation (in this case, equivalence of linear systems) is indeed an equivalence relation. Such proofs are actually relatively straightforward, but *relax*, we're going to let you off the hook this time! Showing that equivalence of linear systems is an equivalence relation is actually too easy. What one needs to do is show that it has each of the three properties: reflexivity, symmetry and transitivity. Each of those is an almost immediate consequence of the way this equivalence is defined. We define two systems to be equivalent if and only if they have the same solution set. In other words, equivalence is *defined* in terms of set equality. Set equality is definitely an equivalence relation, so it has the three properties. Finally, the arguments that show that equivalence of linear systems has the three properties all have the same form: in order to show that the equivalence of linear systems has a property we use the fact that set equality has that property. This is called inheritance.

The general idea is this: there are lots and lots of different linear systems that are equivalent. They all have the same solution set. Some of these systems are in a nice form that allows us to see what the solution set is. Others are not. We need to transform the latter into the former!

More specifically, there are three operations that can be applied to linear systems which *do not have any effect on solution sets*. We can apply these three operations in any way we like! We'll just be transforming our linear system into a slightly different one that is equivalent to the original. Finally, you'll see that it is pretty easy to strategize a bit and transform difficult linear systems into the nice sort (where the solution set is very evident) using these three operations.

The three operations go by many names; we'll refer to them as Reordering, Scaling and Combining. In the next few paragraphs we'll discuss each of them in turn and explain why they don't have an effect on the solution set of a system.

Reordering means what it sounds like. The solution set is determined by checking whether a given solution vector satisfies *all* of the equations. It is pretty clear that the order that the equations are listed in is of little importance. In many treatments of linear algebra an operation called "swapping" is used instead — swapping two equations is a special (particularly simple) instance of reordering and any more general reordering can be accomplished by a succession of swaps.

Exercise 1.2.9 (permutations and swaps)**.** We have placed the letters A through F in sequence below — however they are not in the usual (alphabetic) order. Determine a sequence of swaps that will reorder them so that they *are* in alphabetic order.

$$DCABFE$$

Hint. There are many ways to proceed, but putting A then B then C *et cetera* where they belong using swaps is one possibility. What swap puts A in the first position?

Solution.

$DCABFE$	(given)
$ACDBFE$	swap D and A
$ABDCFE$	swap C and B
$ABCDFE$	swap D and C
$ABCDEF$	swap F and E

Scaling is another operation where it is fairly obvious that there will be no effect on solution sets. Scaling involves multiplying both sides of an equation by some non-zero constant. Very often that non-zero constant will be the reciprocal of the coefficient of one of the variables; scaling by such a constant is useful in solving for that variable. Perhaps it is clear that multiplying both sides of an equation by the same thing will have no impact on what values of the variables satisfy the equation... But why does the constant need to be non-zero? Multiplying both sides of *any* equation by 0 will produce a new equation that looks like $0 = 0$ which is certainly true! In fact, of course, that's what the problem is; if the equation was previously false for some vector of variable values (thus it served to exclude that vector from the solution set) it

will now be true. So vectors of variable values that previously were not in the solution set will now be in it — that's the sort of thing we are trying to avoid!

Combining (a.k.a replacement) is the most difficult of the three operations and as you might guess, it is also the most useful. Combining consists of adding a multiple of some other equation to a given one. Another way to think of this is that we replace some equation by *itself* plus a multiple of some other equation. This is probably why some people call this operation Replacement.

When we added the equations $x + y = 42$ and $x - y = 6$, obtaining the new equation $2x = 48$ back in Section 1.1 we were really doing a "Combining" operation. By the way, when we divided both sides of that new equation by 2 we were "Scaling."

We'll close this section by giving an example — using the three operations to find the solution of a small linear system.

Example 1.2.10 (A small linear system)**.** There is a unique solution to the following system of 3 equations in 3 unknowns. What is it?

$$\begin{aligned} x_1 + \ x_2 + x_3 &= 21 \\ 2x_1 - \ x_2 + x_3 &= 12 \\ x_1 + 3x_2 - x_3 &= 17 \end{aligned}$$

Solution. The first thing we'll do is a combining operation. We'll subtract twice the first equation from the second. It will be convenient to develop a shorthand for expressing these operations. This one could be written as $E_2 = E_2 - 2E_1$.

$$\begin{aligned} x_1 + \ x_2 + x_3 &= \ 21 \\ -3x_2 - x_3 &= -30 \\ x_1 + 3x_2 - x_3 &= \ 17 \end{aligned}$$

Next we'll do a similar combining operation to eliminate x_1 from the 3rd equation. This one would be expressed as $E_3 = E_3 - E_1$.

$$\begin{aligned} x_1 + \ x_2 + \ x_3 &= \ 21 \\ -3x_2 - \ x_3 &= -30 \\ +2x_2 - 2x_3 &= \ -4 \end{aligned}$$

You should take note that we have done something mildly clever in eliminating the occurrences of x_1 in the latter two equations. Now we can use them in further combination operations without fear that they will effect terms involving x_1.

For our next operation let's scale the last equation by $1/2$; this isn't strictly necessary but it makes things *look* a little simpler and since every coefficient in the 3rd equation is even we won't end up dealing with fractions ... $E_3 = \frac{1}{2}E_3$.

$$\begin{aligned} x_1 + \ x_2 + x_3 &= \ 21 \\ -3x_2 - x_3 &= -30 \\ x_2 - x_3 &= \ -2 \end{aligned}$$

We've just cleaned up the 3rd equation so that the first non-zero term in it (the one involving x_2) has a coefficient of 1. This makes equation 3 very useful as a tool for eliminating the variable x_2 from other equations, so next we'll do a reordering operation to move it a bit closer to the top of the heap.

$$\begin{aligned} x_1 + \ x_2 + x_3 &= \ 21 \\ x_2 - x_3 &= \ -2 \\ -3x_2 - x_3 &= -30 \end{aligned}$$

Next, let's use what is now equation 2 to eliminate x_2 from (the new) equation 3: $E_3 = E_3 + 3E_2$.

$$\begin{aligned} x_1 + x_2 + \ x_3 &= \ 21 \\ x_2 - \ x_3 &= -2 \\ -4x_3 &= -36 \end{aligned}$$

Finally, although (again) this isn't strictly necessary, let's scale the 3rd equation so that the coefficient of x_3 is 1... $E_3 = \frac{-1}{4}E_3$.

$$\begin{aligned} x_1 + x_2 + x_3 &= \ 21 \\ x_2 - x_3 &= -2 \\ x_3 &= \ 9 \end{aligned}$$

Wait! Should the last sentence really have started with the word "Finally"? It seems like the system is still pretty complicated. We certainly haven't achieved the simplest possible sort of linear system, but we *have* turned the original system into a type that is known as "triangular". Do you see why? This kind of system is very easy to solve by a process known as back-substitution. The 3rd equation tells you the exact value of the third variable ($x_3 = 9$), you can then substitute that value into the second equation to obtain $x_2 - 9 = -2$. So now we can easily see that $x_2 = 7$. Hmmm. Now we've got known values for x_2 and x_3 which we can substitute into the 1st equation to get

$$x_1 + 7 + 9 = 21$$

Okay. That's easy, $x_1 = 5$.

1.3 Vector equations

We have previously seen the idea of a linear combination of numbers. In this section we will look at forming linear combinations of vectors. The typical problem of the vector equations sort is: can we find the coefficients so that a linear combination of some set of vectors (with those coefficients) is equal to a given vector?

Recall that when we formed linear combinations of numbers we were allowed to "multiply by constants and add things up." So if we are planning to do the same thing with vectors we need to understand what it means to multiply a vector by a constant and what it means to add vectors.

We use the term *scalar* to refer to real numbers, *especially* when referring to the numbers that we multiply vectors by. Calling them "constants" is probably not the best plan; both a scalar and a vector can be *constant* — that just means they aren't changing. It's usually more important to distinguish the vectors from the scalars — which things have multiple components and which don't? When we think of vectors as "those things that have both a direction and a magnitude," the effect of multiplying by a scalar is to leave the direction unchanged, but change the magnitude by scaling it as the scalar indicates. If the scalar is less than 1, the magnitude of the vector will be reduced; if the scalar is greater than 1 it will be increased. Of course, if the scalar is negative the direction *will* be effected, but in a rather simplistic way: the vector ends up facing the opposite direction.

When we have an actual vector and a scalar we'd like to multiply it by, the operation we perform is almost the only thing it could be! Just multiply each of the components of the vector by the scalar.

Definition 1.3.1 (scalar-vector product)**.** If $\vec{v}$ is a vector having m components, $\vec{v} = \langle v_1, v_2, \ldots, v_m \rangle$ and s is a scalar, then the **scalar multiplication** of $\vec{v}$ by s is defined by

$$s\vec{v} = s\langle v_1, v_2, \ldots, v_m \rangle = \langle sv_1, sv_2, \ldots, sv_m \rangle$$

Remark 1.3.2. The scalar-vector product looks rather like a funny version of the distributive law!

The addition of vectors is best thought of in terms of "directions". Suppose the directions to got from my house to the Kwik-E-Mart are: "go 3 blocks north and 1 block east" (call that vector $\vec{v}$, we might write it's component form as $\vec{v} = \langle 1, 3 \rangle$). Suppose in addition that the directions to go from the Kwik-E-Mart to Moe's Tavern are "go 1 block north and 2 blocks west" (let's call this $\vec{w} = \langle -2, 1 \rangle$). The meaning of the vector sum is the vector that describes the change that would be effected if we follow one set of directions followed by the other — except we don't have to be slavish about it — we don't literally follow the first set of directions and then do the second. The sum is the set of directions that take us directly to Moe's without making a Kwik-E-Mart pit stop.

When we actually compute vector sums using the component forms of the vectors involved the computation is probably exactly what you would expect: just add up the corresponding components.

Definition 1.3.3 (vector addition)**.** If $\vec{v}$ and $\vec{w}$ are both vectors having m components,

$$\vec{v} = \langle v_1, v_2, \ldots, v_m \rangle$$

and

$$\vec{w} = \langle w_1, w_2, \ldots, w_m \rangle$$

then their **vector sum** is defined by

$$\vec{v} + \vec{w} = \langle v_1 + w_1, v_2 + w_2, \ldots, v_m + w_m \rangle.$$

Remark 1.3.4. The addition of vectors is also known as *componentwise* addition. It's worth pointing out that if two vectors have different numbers of components, adding them together generally doesn't make sense.

One last definition will be needed to work with vector equations. What does it mean for two vectors to be equal to one another? The answer is probably entirely obvious, but we'll include a formal definition here for completeness.

Definition 1.3.5 (vector equality)**.** If $\vec{v}$ and $\vec{w}$ are two vectors of length m having components

$$\vec{v} = \langle v_1, v_2, \ldots, v_m \rangle$$

and

$$\vec{w} = \langle w_1, w_2, \ldots, w_m \rangle$$

then we say $\vec{v}$ and $\vec{w}$ are **equal** and write $\vec{v} = \vec{w}$ if and only if for every i, $1 \leq i \leq m$, $v_i = w_i$.

Example 1.3.6 (a small vector problem)**.** Consider the following set of vectors: $\langle 1,1,0\rangle$, $\langle 1,1,1\rangle$ and $\langle 0,0,1\rangle$. Is it possible to find scalars x_1, x_2 and x_3 so that

$$x_1\langle 1,1,0\rangle + x_2\langle 1,1,1\rangle + x_3\langle 0,0,1\rangle = \langle 2,3,4\rangle$$

Solution. Let's modify the given problem by using the definitions of (first) scalar multiplication (and then) vector addition:

$$\langle x_1, x_1, 0\rangle + \langle x_2, x_2, x_2\rangle + \langle 0,0,x_3\rangle = \langle 2,3,4\rangle.$$

and then

$$\langle x_1 + x_2, x_1 + x_2, x_2 + x_3\rangle = \langle 2,3,4\rangle.$$

Now (surprise!) that final form — after we use the definition of vector equality — becomes a system of three equations in three unknowns.

$$\begin{aligned} x_1 + x_2 &= 2\\ x_1 + x_2 &= 3\\ x_1 + x_2 + x_3 &= 4 \end{aligned}$$

This system is different from the other systems we've seen so far. It doesn't have a solution. Its statement includes an impossibility; if x_1 and x_2 have a sum of 2 (from the first equation) how can they also have a sum of 3 (which is what the second equation asserts). So there simply *aren't* three numbers which can be used as the coefficients!

Let's make a tiny change to the previous problem. Sometimes small changes have large effects! We'll change the second component in the vector on the right-hand side to a 2.

Example 1.3.7 (a slightly tweaked vector problem)**.** Consider the following set of vectors: $\langle 1,1,0\rangle$, $\langle 1,1,1\rangle$ and $\langle 0,0,1\rangle$. Is it possible to find scalars x_1, x_2 and x_3 so that

$$x_1\langle 1,1,0\rangle + x_2\langle 1,1,1\rangle + x_3\langle 0,0,1\rangle = \langle 2,2,4\rangle$$

Solution. Notice that since the left-hand side vectors are all the same as before we can reuse our previous work. The final form of the vector equation is

$$\langle x_1 + x_2, x_1 + x_2, x_2 + x_3\rangle = \langle 2,2,4\rangle.$$

Now, as a system of equations, we have

$$\begin{aligned} x_1 + x_2 &= 2\\ x_1 + x_2 &= 2\\ x_1 + x_2 + x_3 &= 4 \end{aligned}$$

and the first two equations are identical — they no longer cause a contradiction. This system not only has a solution, it has *lots* of them!

When one equation is an exact duplicate of the other, is there really any reason to retain both copies in the system? Remember that we are mostly concerned with solution sets to linear systems. Either of the copies of this

equation will have the same effect on solution sets. For a given vector, they will both either say "Sure! it works for me, put it in the solution set" or "No way, that vector is *not* okay with me! It makes me false." So, from the perspective of solution sets, this system is really just a system of two equations in three unknowns.

$$\begin{aligned} x_1 + x_2 &= 2 \\ x_1 + x_2 + x_3 &= 4 \end{aligned}$$

By subtracting the first equation from the second we get a unique value for x_3 ($x_3 = 2$). But any pair of numbers that add up to 2 will work for x_1 and x_2. Not only is the solution not unique, the solution set for this system is infinite!

We can express the solution set of this system using set-builder notation and a parameter.

$$\{\langle 2 - t, t, 2\rangle \mid t \in \mathbb{R}\}$$

Notice how the parameter t allows the values of x_1 and x_2 to range over all possibilities that add up to 2? We essentially let x_2 have any value whatsoever (t can be any real number) and then we choose x_1 in such a way that the sum is 2. In a situation like this, x_2 is known as a *free variable*.

1.4 Transformations

A transformation is a function whose inputs and outputs are vectors. In order to discuss concepts like the range and domain of a transformation we'll need some terminology for *sets* of vectors. When we are considering the set of all possible vectors of some type it is known as a *vector space*. At first, we are going to be looking at the most basic and fundamental sorts of vector spaces — where the vectors are ordered tuples of real numbers — but be advised that later we will see that there are many other sorts of vectors!

Definition 1.4.1 (Real Euclidean spaces)**.** Given a positive integer n we define the **real Euclidean space of dimension** n (denoted $\mathbb{R}^n$) to be the set of all ordered n-tuples of real numbers.

$$\mathbb{R}^n = \{\langle v_1, v_2, \ldots, v_n\rangle \mid \forall i, 1 \leq i \leq n,\, v_i \in \mathbb{R}\}$$

Recall that the **domain** of a function is the set from which the inputs come. The set where the outputs may appear is known as the **codomain** of the function. The codomain must be contrasted with the **range** which is the set of outputs that actually *do* occur. We are going to be presuming a certain familiarity with the basic terminology used with functions. You can skip over the following list of (informal) definitions if you are already familiar.

domain The set of all inputs for a function. The domain is sometimes specified while defining the function, but if it isn't, the convention is to use the biggest possible set for the domain.

codomain The set where the outputs of a function lie.

range The set of outputs that actually occur. (The range is generally a subset of the codomain.)

image If an element, x, of the domain is given, we refer to $f(x)$ as the *image* of x.

pre-image If we have some y (an output) in mind, any x (an input) such that $f(x) = y$ is called a *pre-image* of y.

There is a bit of an asymmetry in the way we speak of the various sets that are related to a function. On the output side we have the codomain and the range. On the input side we have only the domain. There is no agreed upon name for a set that contains the domain, we simply insist that the function must be defined for every element of the domain (which basically sidesteps the issue). For the ordinary functions that one sees in calculus, the codomain is the real numbers; the range and domain are generally subsets of the real numbers. And so, the situation isn't terribly complex. When we are dealing with transformations things are harder. The domain and codomain of a transformation are generally real Euclidean spaces — potentially of different dimensions — so we will usually want to spell out what sorts of vectors are expected as inputs, what sorts of vectors will we see as outputs and only then do we get around to the heart of the matter: how do we compute the output from the input? We'll introduce the notation for a transformation via an example and then treat the general case.

Example 1.4.2 (an example transformation)**.** Let's look at a transformation that takes vectors of length 6 as inputs, and outputs vectors of length 3. We'll refer to the input vector as $\vec{x}$ and, as usual, its components will be x's with subscripts: $\vec{x} = \langle x_1, x_2, x_3, x_4, x_5, x_6 \rangle$. Similarly, the output will be $\vec{y} = \langle y_1, y_2, y_3 \rangle$. This is only an example so we'll just make up the rules that determine those output components from the input components, the point here is simply to demonstrate how one should write such a thing — which is as follows:

$$\begin{aligned} T : \mathbb{R}^6 &\longrightarrow \mathbb{R}^3 \\ T(\langle x_1, x_2, x_3, x_4, x_5, x_6 \rangle) &= \langle x_1, x_3, x_5 \rangle. \end{aligned}$$

So this transformation just picks out the odd-numbered components of $\vec{x}$ and puts them in $\vec{y}$.

The most important transformations for us in this context are the *linear* ones. In a linear transformation, the components of the output vector are computed from the components of the input vector by "multiplying by constants and adding everything up." Because of the simplistic way that the outputs are computed there is really nothing that can go wrong! With ordinary functions from $\mathbb{R}$ to $\mathbb{R}$ we usually look at the rule for computing the output and recognize certain values that must be eliminated from the domain — typically where one sees "division by zero" or "square root of a negative" errors. No such problem can arise with linear transformations, the domain will always be a real Euclidean space of some dimension. Similarly, the codomain will be a real Euclidean space; one whose dimension is simply the number of components in the output vectors. The dimensions of the domain and codomain are easy to think about — how many components do the input and output vectors have? The range of a linear transformation is slightly more complicated. The output vectors that actually occur will certainly be vectors having the number of components as specified by the codomain, but do all such vectors necessarily have to appear as outputs? In general, no.

The notation for a linear transformation first spells out the domain and codomain and then gives the rule(s) for computing the output. Thus the domain and codomain are known in advance; we need to do a little extra work to figure out the range.

Before proceeding further we'll give some formal definitions.

Definition 1.4.3 (Transformations)**.** Given positive integers m and n, a **transformation from $\mathbb{R}^m$ to $\mathbb{R}^n$** is a function, T, that takes vectors of length m as inputs and returns vectors of length n. We write

$$\begin{aligned} T : \mathbb{R}^m &\longrightarrow \mathbb{R}^n \\ T(\vec{x}) &= \vec{y}, \end{aligned}$$

where the components of the vector $\vec{y}$ will need to be specified in terms of the components of $\vec{x}$.

Definition 1.4.4 (Domain of a transformations)**.** The **domain** of a transformation, T is denoted by $\mathrm{Dom}(T)$ and is generally a subset of $\mathbb{R}^m$ (provided T is defined as above).

$$\mathrm{Dom}(T) \;=\; \{\vec{x} \in \mathbb{R}^m \;\mid\; T(\vec{x}) \text{ is defined}\}$$

Definition 1.4.5 (Co-domain of a transformations)**.** The **codomain** of a transformation, T is denoted by $\mathrm{Cod}(T)$ and is equal to $\mathbb{R}^n$ (provided T is defined as above).

Definition 1.4.6 (Linearity)**.** A transformation T is **linear** if and only if given any two elements $\vec{u}, \vec{v} \in \mathrm{Dom}(T)$ and any two real numbers α and β we have

$$T(\alpha\vec{u} + \beta\vec{v}) \;=\; \alpha T(\vec{u}) + \beta T(\vec{v}).$$

Linearity is a really important concept! We will be using the definition above over and over again. Let's try to nail down our understanding of this definition by translating it into ordinary language: A transformation is linear if and only if when you apply it to a linear combination of vectors, the result is equal to what you get if you form the same linear combination of the images of those vectors. More succinctly: "The image of a linear combination is the same linear combination of the images." My advice (seriously!) is to treat that last phrasing like a mantra — repeat it to yourself until you fully absorb the meaning and it becomes second nature to you.

Look back at the formal definition of linearity, and notice what it looks like symbolically: It appears as if the transformation T distributes over the sum and that the scalars can be moved to the outside of the T's. Sometimes an alternative definition of linearity is given which splits out these two issues. This is sometimes useful in formulating a proof that some transformation is linear (because it separates the argument into simpler parts).

Definition 1.4.7 (Linearity (alternate definition))**.** A transformation T is **linear** if and only if given any two elements $\vec{u}, \vec{v} \in \mathrm{Dom}(T)$ and any real number α, both of the following hold:

$$T(\vec{u} + \vec{v}) \;=\; T(\vec{u}) + T(\vec{v}),$$

and

$$T(\alpha\vec{u}) = \alpha T(\vec{u}).$$

Before we can go any further we have a small moral obligation to take care of. Since we've just presented two definitions for a concept we have a duty to verify that they actually define the same concept. If we state that two things are the same, that really aren't, we're making a **false equivalence**. One of the hallmarks of a good critical thinker is that they won't be taken in by false equivalences. So, what do you think? Are they definitely the same idea, or are there transformations that are linear by one definition but not by the other?

Theorem 1.4.8 (The two definitions of linearity are equivalent)**.** *Consider a given transformation T from $\mathbb{R}^m$ to $\mathbb{R}^n$. Let $\vec{u}$ and $\vec{v}$ be arbitrary vectors in $\mathbb{R}^m$, also let α and β be arbitrary real numbers. Then*

$$T(\alpha\vec{u}+\beta\vec{v}) \;=\; \alpha T(\vec{u})+\beta T(\vec{v})$$

if and only if

$$T(\vec{u}+\vec{v}) \;=\; T(\vec{u})+T(\vec{v}) \quad \textit{and} \quad T(\alpha\vec{u})=\alpha T(\vec{u})$$

Proof. ($\Rightarrow$) In this part of the proof we will be presuming the first statement (the definition of linearity given first) and showing that the second statement must be true.

Assume that T is a transformation and that for every pair of vectors $\vec{u}$ and $\vec{v}$, and every pair of real numbers α and β,

$$T(\alpha\vec{u}+\beta\vec{v}) \;=\; \alpha T(\vec{u})+\beta T(\vec{v}).$$

if we set $\alpha=\beta=1$ we get

$$T(\vec{u}+\vec{v}) \;=\; T(\vec{u})+T(\vec{v}).$$

Similarly, if we leave α arbitrary but set $\beta=0$ we get

$$T(\alpha\vec{u})=\alpha T(\vec{u}).$$

($\Leftarrow$) In this part of the proof we will be working in the reverse direction, so we assume that both

$$T(\vec{u}+\vec{v}) \;=\; T(\vec{u})+T(\vec{v}) \quad \text{and} \quad T(\alpha\vec{u})=\alpha T(\vec{u})$$

hold.

> It's important to realize that the hypotheses we are using above are generic statements. When we write $T(\alpha\vec{u})=\alpha T(\vec{u})$ the scalar α and the vector $\vec{u}$ are really beside the point. We are really asserting a general rule about how T interacts with scaled vectors — any other scalar times any other vector will work the same way. So for example, that hypothesis will also let us deduce that
>
> $$T(\beta\vec{v})=\beta T(\vec{v}).$$

Consider $T(\alpha\vec{u}+\beta\vec{v})$. Using our first hypothesis (the one that shows how T distributes over sums) we get

$$T(\alpha\vec{u}+\beta\vec{v}) \;=\; T(\alpha\vec{u})+T(\beta\vec{v})$$

. Using the second hypothesis (twice) we get

$$T(\alpha\vec{u})+T(\beta\vec{v}) \;=\; \alpha T(\vec{u})+\beta T(\vec{v}).$$

Finally, putting these pieces together we have

$$T(\alpha\vec{u}+\beta\vec{v}) \;=\; \alpha T(\vec{u})+\beta T(\vec{v})$$

which is the desired result. □

Definition 1.4.9 (Linear transformations)**.** Given positive integers m and n, a **linear transformation from $\mathbb{R}^m$ to $\mathbb{R}^n$** is a transformation T, that takes vectors of length m as inputs and returns vectors of length n and that is *linear*. We write

$$\begin{aligned} T : \mathbb{R}^m &\longrightarrow \mathbb{R}^n \\ T(\vec{x}) &= \vec{y}, \end{aligned}$$

where the components of the vector $\vec{y}$ will need to be specified in terms of the components of $\vec{x}$.

There is an interesting connection between our use of the word "linear" in talking about linear transformations and linear combinations. When a transformation is linear the functions that determine the output's components in terms of the input's components must *be* linear combinations. And *vice versa*, if the component functions are linear combinations then the transformation will be linear.

The content of the previous paragraph may not be surprising from a linguistic perspective; they wouldn't use the same word if the underlying concepts were really different, would they? From a mathematical perspective it's a bit less obvious. Indeed this is the sort of thing that mathematicians call a *theorem*. We'll state this theorem now, but we'll leave the proof to a later chapter.

Theorem 1.4.10 (coefficients of a linear transformation)**.** *Given a transformation $T : \mathbb{R}^m \longrightarrow \mathbb{R}^n$, T is linear if and only if, for all input vectors $\vec{x}$ the components of $T(\vec{x})$ can be expressed as particular linear combinations of the components of $\vec{x}$.*

In order to fully specify a linear transformation we need to give values for all of the constants that are used in the linear combinations where the y_i's are written in terms of the x_i's. For each of the n components of $\vec{y}$, we will need m numbers (as many as there are components in $\vec{x}$). In other words we must specify mn constants.

Definition 1.4.11 (components of a linear transformation)**.** Given mn real numbers, $a_{11}, \ldots a_{mn}$, we say they are the components of a linear transformation T,

$$\begin{aligned} T : \mathbb{R}^m &\longrightarrow \mathbb{R}^n \\ T(\vec{x}) &= \vec{y}, \end{aligned}$$

provided

$$\begin{aligned} y_1 &= a_{11}x_1 + \ldots + a_{1m}x_m \\ &\vdots \\ y_n &= a_{n1}x_1 + \ldots + a_{nm}x_m. \end{aligned}$$

1.5 Matrix notation

The three seemingly distinct viewpoints we've considered are unified by the concept of a **matrix**.

The word "matrix" is from Latin. The word entered the English language with a variety of meanings — in Latin it means *womb*. In mathematics, matrix (pl. matrices) always means a table containing numerical values. It is rather hard to guess how a word meaning "uterus" could get morphed into one meaning "table of numbers", but languages are funny that way...

Generally speaking, a table of numbers will have some arbitrary number of rows and of columns. There are some special cases that we'll need to talk about, but let's look at the general situation first. We'll use the variable m to refer to the number of rows in a matrix and the variable n to refer to the number of columns. We'll use upper-case letters (about 90

Example 1.5.1 (matrix notation)**.** Here are a couple of matrices:

$$A = \begin{bmatrix} 1 & 4 & 9 \\ 7 & \pi & 42 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -1 & 11 \\ -3 & e \end{bmatrix}$$

Notice how we are referring to the entire tables with the variables A and B? If we need to refer to the individual entries of a matrix we'll write things like $a_{23} = 42$ (the number in the 2nd row and 3rd column of A is 42), or $b_{12} = 11$ (the number in the 1st row, 2nd column of B is 11).

It's also fairly common to ignore this lower-case convention! That is, you may also see things like $A_{13} = 9$ and $B_{22} = e$.

Now to the special cases. When the number of columns is $n = 1$, the matrix is known as a **column vector**. When the number of rows is $m = 1$, the matrix is known as a **row vector**. There is clearly a choice to be made as to whether the things we have been referring to as (merely) "vectors" are going to be represented as column vectors, or as row vectors. Here's a surprising thing! Your Calculus teachers and I (up until now) have been lieing to you. When we wrote vectors as (for example) $\vec{v} = \langle 1, 2, 3 \rangle$, it was only for convenience. A row of numbers fits more easily on the page than a column does. For a variety of reasons it makes sense to treat vectors as columns of numbers, not rows.

There is an operation known as **transposition** that changes row vectors into column vectors and *vice versa*. The **transpose** of a matrix is indicated by a superscript T, the rows of the transposed matrix are the columns of the original matrix and its columns are the original matrix's rows. This idea (interchanging rows and columns) is surprisingly important and we'll be using it quite a bit in the future. For the moment let's just notice that it gives us a nice way to write a column vector — with the typographical advantage that the components appear in a row!

To summarize what the last few paragraphs have said: It is technically not right to write $\vec{v} = \langle 1, 2, 3 \rangle$, we should really write $\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$, but that takes up too much vertical space so instead we write $\vec{v} = [1\ 2\ 3]^T$. This may all seem like too high of a price to pay for accuracy, but it will pay future dividends if we start thinking now about rows and columns and how to switch between them.

If we only had row and column vectors to worry about we'd probably find some other way to distinguish them — maybe there'd be red vectors and blue vectors!

Note 1.5.2. In Physics (especially in the Tensor Analysis which is used in e.g. General Relativity) they distinguish between covariant and contravariant indices. An entity with a single contravariant index is a vector, if instead there is a single covariant index it is known as a co-vector. These concepts aren't

identical to row/column vectors, but nevertheless, contravariant vectors are usually written as columns and covariant vectors as rows.

By convention there is no need to refer to the entries of a row or column vector using double indices — one of them would always be 1 so we can omit it. When we have more general matrices, where m and n are both greater than 1, the roles of rows and columns are more evident and two indices will be necessary to refer to the entries.

One useful way to think about matrices is the following: When we write down a system of equations, a lot of the symbols that we write are redundant. If we eliminate all of the stuff that is utterly predictable we are left with a table of numbers — in other words, a matrix. So one way to think of matrices is that they are highly abbreviated ways of referring to a system of linear equations. In this scheme the rows of the matrix correspond to the individual equations in the system and the columns contain all the coefficients that multiply a given variable. A short example will probably help:

Example 1.5.3 (Converting a linear system to matrix form)**.** Consider the following system of 3 equations in 4 variables.

$$\begin{aligned} x_1 + \ x_2 \qquad + 3x_4 &= 101 \\ 2x_1 - \ x_2 + x_3 + \ x_4 &= 102 \\ 3x_2 - x_3 + 2x_4 &= 103 \end{aligned}$$

Now we'll take one step backwards before proceeding two steps forward. If a variable appears, but has no coefficient, that just means the coefficient is 1. If a variable doesn't appear at all, that means the coefficient is 0. Finally, if we see subtraction we can always replace it by addition (by putting a minus sign on the coefficient). So, let's re-express this system in a fully anal-retentive way...

$$\begin{aligned} 1x_1 + \ 1x_2 + \ 0x_3 + 3x_4 &= 101 \\ 2x_1 + -1x_2 + \ 1x_3 + 1x_4 &= 102 \\ 0x_1 + \ 3x_2 + -1x_3 + 2x_4 &= 103 \end{aligned}$$

Okay, so now the promised two steps forward. First, notice that in every equation in the system every variable is present and they all appear in ascending order. If we were only given the lists of coefficients we'd easily be able to reconstruct the equations. So, we're going to eject all of the plus signs and all of the variables with all of those subscripts. We just won't deign to write them down! Sometimes it's a good idea to imagine their presence but it certainly isn't necessary to. Also, the equals signs that separate the left- and right-hand sides of the equations always come before the very last number. There really isn't a lot of information conveyed by the appearance of those equals signs, but we usually keep a slight vestige of them around — a thin vertical line separates the last column of the matrix form from everything else. So, with no further ado, here is the matrix form of this system:

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 101 \\ 2 & -1 & 1 & 1 & 102 \\ 0 & 3 & -1 & 2 & 103 \end{array}\right]$$

In the previous example the final matrix we wrote is actually known as the **augmented matrix** of the system. Sometimes it is a good idea to separate out the part of the matrix that appears to the left of the thin vertical line. That part is known as the **coefficient matrix** of the system. This isn't just

pedantry! In many real-world applications we need to solve bunches of linear systems that all have the same coefficient matrix — so they only differ in the final column (a.k.a. the **augmented column**) of their augmented matrices. We can take advantage of such a situation, essentially solving all of the systems while only doing the work of solving the first one!

Matrix notation was probably invented purely out of laziness. When we use the Re-ordering, Scaling and Combining operations that we introduced in Section 1.2, we find ourselves having to re-copy the entire system over and over. By switching to matrix notation we get a considerable savings in effort. The operations that we originally developed to use on equations now become operations that one can apply to the rows of a matrix — a.k.a. row operations — which we will study in much greater depth in Section 2.4. Regardless of the origins of matrix notation, nowadays we don't think of matrices only in terms of being abbreviations for linear systems. They have taken on a life of their own!

There are two features of matrices that we'll explore in the remainder of this section. The first is that matrices may be thought of as "funny shaped" vectors. The second is that, under certain conditions, we can multiply matrices. If you've already studied multi-variable calculus (and perhaps even if you haven't) you'll have run into the dot product (a.k.a. scalar product) and the cross product (a.k.a. vector product) in $\mathbb{R}^3$. No matter what the dimension of the space, there is always a dot product. On the other hand, there is usually nothing analogous to the cross product — it depends on a very special coincidence, an odd fact about the space $\mathbb{R}^3$. The dot product is a way of multiplying vectors, but the product is *not* a vector. On the other hand, the cross product *does* result in a vector. Matrices (as "funny shaped" vectors) give us a way of multiplying vectors and getting other vectors.

The most important thing with vectors is that we need to be able to add them. The second most important thing is that we should know how to scale them.

If A and B are matrices, what would it mean to add them? As was the case with vectors, it doesn't make any sense to add them unless they are the same size. With vectors they needed to have the same number of components in order to even think about adding them. With matrices the restriction is even stronger; they need to have the same number of rows *and* of columns. Provided that that restriction is met, we just add the corresponding entries.

Definition 1.5.4 (matrix addition)**.** If A and B are both $m \times n$ matrices, their sum, $A + B$ is also an $m \times n$ matrix. For all integers i and j satisfying $1 \leq i \leq m$ and $1 \leq j \leq n$, the entry in the ith row and jth column of $A + B$ is $a_{ij} + b_{ij}$.

Scaling also works in much the same way as it did with vectors. If we multiply a scalar and a matrix, every entry of the matrix is multiplied by the scalar.

Definition 1.5.5 (matrix scaling)**.** If A is an $m \times n$ matrix, and s is a real number, the scalar product, sA is also an $m \times n$ matrix. For all integers i and j satisfying $1 \leq i \leq m$ and $1 \leq j \leq n$, the entry in the ith row and jth column of sA is $s \cdot a_{ij}$.

Example 1.5.6 (vector properties of matrices)**.** Let $A = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$ and

$B = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$. These matrices are both 2×2 so their sum is defined.

$$A + B = \begin{bmatrix} 1 & 0 \\ 1 & 5 \end{bmatrix}$$

Let's also provide an example of scaling. If we scale the matrix A by a factor of 3 we get

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

Exercise 1.5.7 (linear combinations of matrices)**.** Suppose that A and B are the following 2×3 matrices:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}$$

What is $5A - 2B$?

Solution.

$$5A - 2B = \begin{bmatrix} -1 & 0 & 1 \\ 12 & 13 & 14 \end{bmatrix}.$$

So that was nice! Once we know how to add matrices and how to multiply them by scalars, we can form linear combinations. Next we'll look at multiplying our funny shaped vectors...

The easiest example (and also a *very* instructive example) of multiplying vectors is the product of a row and a column vector. Provided they have the same number of entries, a row vector times a column vector produces a 1×1 matix — also known as a real number. You have almost certainly seen this before! The dot product of two vectors is actually a row/column matrix product. In fact, in many settings they will write $\vec{x}^T\vec{y}$ rather than $\vec{x} \cdot \vec{y}$ when referring to the dot product. As you move towards more advanced math the tendency will be to call this the "inner product" rather than the "dot product", one reason to make the change (other than it sounds more sophisticated) is that there is also an "outer product" of vectors which is what you get if you multiply a column times a row. As we'll see shortly, $\vec{x}^T\vec{y}$ and $\vec{x}\vec{y}^T$ are *extremely* different! Anyway, we need to do this row/column product as a component of the general matrix product computation so let's proceed to over-explain it by some huge factor...

If you've ever done the challenge where you rub your belly in a circular motion while simultaneously patting your head, then this shouldn't be too difficult. What you need to do is trace across the entries of a row with your left index finger, while simultaneously tracing down the entries of a column with your right index finger. As you encounter the entries you multiply them and keep a running tally of the sum of these products.

Example 1.5.8 (an inner product)**.** Suppose

$$\vec{x} = \begin{bmatrix} 3 \\ 1 \\ -2 \\ 5 \end{bmatrix} \quad \text{and} \quad \vec{y} = \begin{bmatrix} -1 \\ 6 \\ 4 \\ 7 \end{bmatrix}$$

then the inner product of these two vectors ($\vec{x}^T\vec{y}$) is the following row/column matrix computation:

$$\vec{x}^T\vec{y} \quad = \quad \begin{bmatrix} 3 & 1 & -2 & 5 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 6 \\ 4 \\ 7 \end{bmatrix} \quad = \quad 3\cdot(-1)+1\cdot 6+(-2)\cdot 4+5\cdot 7 \quad = \quad 30$$

Notice that if the vectors had different lengths (I mean "lengths" as in "number of entries") the process we've described wouldn't work out so good... One of your fingers would be out of entries before the other! This is our first example of an idea known as **conformability**. Suppose we have a row vector of length m (that is, a $1 \times m$ matrix) and a column vector of length n (in other words a $n \times 1$ matrix), then they are **conformable** if $m = n$ and if $m \neq n$ they are *not* conformable, in which case the matrix product can't be computed.

The general rule for computing matrix products involves doing this row/-column product multiple times. Suppose A is a $p \times q$ matrix and B is an $r \times s$ matrix. The product AB will be a $p \times s$ matrix, but it can only be computed if $q = r$. The entry in the p-th row and s-th column of the result is obtained using the p-th row of A and the s-th column of B. When you physically write the sizes of the multiplicands next to one another, the inner two numbers must match and the outer two numbers tell you the size of the result!

Definition 1.5.9 (matrix conformability)**.** Suppose A is a $p \times q$ matrix and B is an $r \times s$ matrix. If $q = r$ these matrices are **conformable** and the product AB can be computed.

Note 1.5.10. Conformability has a directionality. If A and B are conformable it is not necessarily the case that B and A are conformable. Matrices fail to obey the commutative law in a fairly spectacular way! It is *not* generally the case that $AB = BA$. Indeed, quite often it is *impossible* to compute the product BA, even given that it *is* possible to compute AB.

Definition 1.5.11 (matrix product)**.** Suppose we are given two matrices, A and B that are conformable for matrix multiplication, further, suppose that A is $m \times n$ and B is $n \times p$. The matrix product AB will be an $m \times p$ matrix. The entry in the i-th row and j-th column of AB is

$$AB_{ij} \;=\; \sum k = 1^n A_{ik} \cdot B_{kj}$$

Chapter 2

RREF

2.1 Triangular systems

Dummy text for introduction.

2.2 Echelon form

Dummy text for introduction.

2.3 RREF

Dummy text for introduction.

2.4 Row operations and Gaussian elimination

Dummy text for introduction.

2.5 Solving linear systems

Dummy text for introduction.

Chapter 3

Vectors

3.1 Vectors and scalars

Dummy text for introduction.

3.1.1 Vectors

3.1.2 Vector operations

3.1.3 Linear combinations

3.1.4 The Euclidean basis

3.1.5 parametric equations for lines and planes in 3-space

3.2 The matrix-vector product

Dummy text for introduction.

3.3 Homogeneous and non-homogeneous systems

Dummy text for introduction.

3.4 Matrix-matrix products

Dummy text for introduction.

3.4.1 Basic transformations

3.4.2 Compositions

3.4.3 The matrix of a composition

3.4.4 Matrix multiplication

3.5 Vector spaces - an introduction

Dummy text for introduction.

3.6 Dependence and independence

Dummy text for introduction.

3.7 Bases and dimension

Dummy text for introduction.

Chapter 4

Determinants

4.1 Torque, Area and Volume

The 2-d and 3-d cases.

4.2 Determinants by recursion

Using 2 by 2 determinants to form the 3 by 3 ones. Generalizing to 4 by 4. Generalizing to the n by n case and taking advantage of zeros.

4.3 Formal definition

Permutations and their signs. The determinant as a sum of signed products over all $n!$ permutations. Combinatorial explosion and why determinants are just theoretical tools except in low dimension.

Chapter 5

The spectral decomposition

5.1 Diagonal and diagonalizable systems

Dummy text for introduction.

5.2 Eigenvalues and eigenvectors

Dummy text for introduction.

5.3 Jordan form

Dummy text for introduction.

5.4 The Singular value decomposition

Dummy text for introduction.

Chapter 6

Algebraic structures

6.1 Groups, Rings and Fields

Dummy text for introduction.

6.1.1 Fields

6.1.2 The complex numbers

6.1.3 Rings

6.1.4 Groups

6.2 Modules

Dummy text for introduction.

6.3 Algebras

Dummy text for introduction.

6.3.1 Complex numbers

6.3.2 The cross product in 3-space

6.3.3 Quaternions

6.4 Inner product spaces

Dummy text for introduction.

Chapter 7

Abstract vector spaces

Truly abstract and concrete vector spaces that aren't $\mathbb{R}^n$.

7.1 Vector spaces

Enumerate the laundry list of properties that hold in $\mathbb{R}^n$ and pull together the definition of an abstract vector space. Examples in subsections.

7.1.1 Polynomials

7.1.2 Matrices

7.2 Infinite dimensional spaces

Dummy text for introduction.

7.2.1 Functions

Dummy text for introduction.

7.2.2 Time series

Dummy text for introduction.

7.3 Hilbert space

Dummy text for introduction.

7.4 Fourier analysis

Dummy text for introduction.