EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 0

Welcome to 16A!

If you're like many students, you might find the official catalog title of this course "Designing Information Devices and Systems" a bit confusing. Why are "information devices and systems" important?

Let's look at an example. In the 17th century, the only way to see inside of the human body was to actually *look* inside of the human body via surgery. Thankfully, today non-invasive imaging tools such as computed tomography (CT) scanners, X-ray machines, and ultrasound devices exist that can provide much of this information. In other words, we have *devices*, such as imagers, that provide *information*, such as a visualization of the human body. Often, these devices don't work alone — they are part of a larger *system* that uses a combination of both physical sensors and signal processing techniques. Today, these "information devices and systems" are everywhere. For instance, smartphones take user input from touchscreens, microphones, accelerometers, and GPS to let you search the web, share pictures, communicate with others, and tell you how to get to Safeway.

The goal of this class is not just to teach you the basics of linear algebra or the basics of circuits, but to help show you how those concepts weave together to enable the complex devices we use today. Doing this will require helping you discover both basic circuits and linear algebra concepts. Without circuits to interface with the real world, these devices wouldn't be very useful, and without processing techniques, we wouldn't be able to extract meaningful information from circuit measurements. What is less obvious to an outside observer is that the ways of thinking required to understand both sides are actually similar, and this manifests in the mathematical language that is used. You might have heard buzz or hype around the term "machine learning" before taking this class — it turns out that too is another branch of the same tree. Our goal in 16A is to make concepts you will learn about have a direct real-world application (and, in many cases, a multitude of applications). In order to put these concepts together to build a working system, we need to understand design thinking.

0.1 What this course is like and how you can succeed

Before going more into the actual subject matter of this course, it is useful to step back and reflect on the nature of this course at a higher level. The EECS 16AB series is fundamentally about teaching you how to think like an engineer while making you stronger and more mathematically mature. The primary mathematical vehicle to do this is linear algebra (and related concepts) with the goal of you making these tools your own through your own hard work. This is done by showing you how you could have come up with all the definitions yourself, as well as all the key proofs/derivations, and actually having you do many of these yourself —including deriving things and working through ideas seemingly not explicitly discussed in lecture/discussion, in ways that exercise the ways of thinking that are demonstrated and taught to you in lecture and discussion.

While mathematical curiosity is going to be fostered, the style is largely what is called "use-inspired" (from Louis Pasteur), where we show how each of the ideas emerge organically from engineering contexts, while being useful beyond the specific contexts that first motivated them. **Our promise[1] to you is that you will always know why you are learning whatever it is that we are teaching you.**

0.1.1 Why is this course this way?

[Note to students: feel free to skip this section and come back to read it whenever you find yourself wondering why things are the way that they are.]

This course is likely to feel very different to you. To understand why, let us step back for a moment and consider the role of education itself. Simply put, the goal of the educational system is to help students grow and become productive members of society. Our society is the inheritor of many thousands of years of human civilization, and today we take approximately 24 years of formal and informal education to get you to take your rightful place on the front lines of society and then eventually as leaders who can then hand our civilization to the generations that will come after you.

Obviously, it is not possible to have everyone know everything. At some point, the system starts specializing more and more so that it is possible for students to realistically grow towards some kind of mastery of something. Undergraduate education is one of the points at which people start to specialize, and if you are in this course, you suspect that Engineering (broadly understood) might be something that you want to be involved with more deeply.

If you are a first-year undergraduate student, you are basically starting year 14 of this educational process. Let us think of someone who has just graduated with a PhD (or just finished professional training)[2] as being at the end of year 24 of this process. It can be hard to think about what you might be like ten years into the future. One way to get some intuition is to think about the past. Go back ten years in your education to elementary school. Hopefully you can see that that person, although they had a lot of potential, clearly had a lot to learn and a lot more mental growth ahead of them. It should also be clear that the difference between you now and you in elementary school is not merely that you know more facts now — you can think differently now. Once you have begun to grasp the gap between you now and your elementary school self, it is useful to understand yourself as having a lot to learn in the next 10 years.

EECS 16AB are designed to help you transition to the kind of thinking that you are going to need as a professional. Essentially, for the majority of your working lives, you will be constantly encountering situations that are, to varying degrees, unknown to you. You will have to learn in-situ and there will not be a teacher there to point out what you need to do. There will not be a website or book that will tell you what you need to do. You will find yourself exposed to ideas in one context, and will have to actually weave them into what you do in a different context. For many of you, this might be different from what has been the dominant experience in your education before coming to Berkeley.

Young children are sometimes compared to sponges when it comes to education — they soak things up rapidly. Consequently, uncritical thinking and learning a vast amount of patterns often defines early education. This is also what is most useful for taking standardized exams like the SAT. However, to succeed as

[1] If you ever find yourself losing track of this, just ask. Usually the lectures, discussions, labs, notes, and problems will try to make it clear. But if it isn't clear to you, ask. The course staff will help you.

[2] Of course, not everyone is going to pursue a formal graduate degree. However, the first few years of working in Industry typically involve a lot of informal training/mentorship and picking up skills as well as ways of thinking — this plays a similar role as graduate school in terms of bringing people up to speed.

an engineer, you will have to take more ownership of your mental models. You may have to change what you think of as understanding itself. In the past, it might have been possible to solve problems by looking in the textbook or online for a very similar problem or example, and then figuring out how to apply the same approach to the problem you were facing. This approach needs to be expanded on in your undergraduate career. We need you to be able to approach new problems systematically, without necessarily having any easy pattern to match to or formulas to apply.

In Sanskrit poetics, the swan/*hamsa* bird is said to be imbued with the ability to drink milk that has been mixed into mud. This is a classical image used to convey the idea of being able to discriminate between what is important and what is not. As a student, you will be learning to discriminate as well. However, you need to be aware that some of your previous instincts could lead you astray. For example, some you might be used to skimming a mathematical text or listening to a lecture and taking care to note down key definitions or key formulas and examples of their use. In the context of EECS 16AB (and most of the later courses that you will take), this is almost always not very helpful and represents leaving much of the figurative milk behind. What is really important to understand is how to come up with the definitions and how to derive/discover the relevant formulas. The examples we share often represent questions that allow you to discover the underlying ideas.

EECS 16AB are structured in a way that tries to help you achieve the required transition in your attitude and ways of thinking. Of course, there can be no true learning without sincere effort. And at times, frustration is a natural accompaniment to effort. You've experienced this in video games, sports, the arts, and everything else. You might find yourself thinking "This problem is unfair! The lecture and discussion did not cover this type of problem!" If you feel this way, take a deep breath and remember that it is alright if you do not know how to solve a problem right away. You just need to be able to read carefully, start working, and then be able to keep going, even if you make mistakes along the way and have to backtrack. After you succeed in doing this many times, approaching the unknown will get less scary. There is no shortcut to achieving true confidence.

If you succeed in internalizing the "how to think" lessons in EECS 16AB, we hope they will help you tremendously in all your future endeavors. The specific technical skills and concepts taught here are also incredibly powerful, and will serve as a base for your future education.

0.1.2 How you can succeed in this course

The short answer for how to succeed is don't fall behind, and catch up if you do. Attend lecture, pay attention and actively take your own notes, and participate in discussion. Read the official course notes, ask questions if things are unclear. Accept the fact that you might be confused when you first encounter any given HW problem — but don't let that stop you from trying. Overcoming what you do not understand and your own fears is a big part of the learning process. Work with others and ask for help from course staff. This is a course in which truly understanding all the labs, homeworks, and discussions is the key to success on exams and beyond.

0.2 System design

Returning to the material of the course, as engineers, we want to solve meaningful problems. Usually, this requires building a system that can **sense** something about the real world, **compute** something useful, and then **communicate** relevant information or **act** on this information to change the environment. In

16A, we will mostly focus on the first two problems — sensing and computing — in the context of three applications that you'll investigate in lab: imaging, touch sensing, and positioning. 16B builds on the knowledge developed in 16A to explore how we can use the information we compute to interact with the environment.

In general, information devices and systems require (1) a real-world quantity to measure, (2) a means of translating this quantity to the electrical domain,[3] and (3) some computation to extract information from the electrical readings. We can represent this flow graphically for the three applications studied in this course:

- **Module 1**: In an imaging system, our goal is to get a picture of something physical. In lab, you'll be using a component called a photodiode to convert visible light into a measurable electrical quantity, current. Modern cameras have millions of these photodiodes — one for each pixel — but in lab, we'll explore how to use linear algebra techniques to build an image with only *one* photodiode.

- **Module 2**: The goal of a touchscreen is to determine the position of one or more fingers from a user. In the second module of this class, you'll be exploring a few different ways that touchscreens can be built. In one approach, you will use a circuit called a resistive voltage divider to translate the position of a touch into a voltage. In the second approach, we'll use the fact that the presence or absence of human touch can influence an electrical quantity called capacitance, which is a principle used in most touchscreens that you interact with today.

- **Module 3**: In a positioning system, the goal is to find your location in a broader physical space. In lab, we will use a microphone to detect sounds from three speakers at fixed locations — translating information to the electrical domain — and then use optimization techniques to compute the microphone's location from these readings. In real-world GPS systems, these speakers are replaced by satellites orbiting the earth at known positions.

[3] Why do we care about electrical signals? First, pretty much any physical quantity that we care about can be translated into a measurable electrical quantity. Second, most processing techniques can be implemented efficiently with computers, which operate on electrical signals.

0.3 Dealing with complexity

Modern information devices and systems often integrate many sub-systems which are quite complex. For instance, a single smartphone typically has all three of the sub-systems we will study in this class — a camera, touchscreen, and GPS system — in addition to many other features (perhaps a fingerprint reader, voice control, etc.). While the focus of 16A will be on understanding the basics of these sub-systems, approaching even these simpler problems requires **multi-step thinking**.

In previous classes, you may have had homeworks that involved only plugging numbers into an equation to find your answer. In this class, problems will typically require multiple steps — finding an appropriate model for a real-world system, representing it with the right equations, solving it, interpreting the solution, and perhaps using the results to do more analysis. Our motivation for doing this is to give you practice thinking in a way that can be applied to complex real-world problems. It may be difficult, but in the end you'll learn from it, and the course staff is here to help you along the way.

0.4 From models to design

The first part of 16A focuses on building the conceptual tools of linear algebra to start modeling the real world. The second module introduces the next step — going from models to design. There is a difference between analysis of a system (what happens) to design thinking (what do I build so that I can have a system that does what I want). Here the abstraction we use is that of electrical circuits — the nice thing is that they are easily modeled by simple linear equations (that we can solve) and they are fundamental to our ability to build computational devices. Building a full machine learning pipeline, for example, requires us to be able to take data from the real world, analyze it, model the world, create designs that can have the impact we want, and then implement it. Through the 16A-16B series we hope to give you the opportunity to experience this full stack through the cumulative experience of the labs. Each of the labs gives you this in small part but the two classes together really help put this together.

A key example is the touchscreen, but there are other examples too. Here are some more reasons why circuit design is an essential part of this course.

- **Applications**: Beyond the fact that all computer processing runs on integrated circuits, circuits shape the way we interact with electronic devices to help give them value — to actually interface with the real world, any device is going to require circuits. Moreover, it is difficult to truly create something new without a full understanding of how an actual engineering system is designed at both the hardware and software level. For instance, some companies like Apple practice "vertical integration," where engineers work on nearly all aspects of a system, from the mechanical design of a product to the software down to the electrical hardware implementation. Without at least a foundational understanding of circuits, it would be difficult to create an innovative system, which is becoming increasingly relevant in industry with the growth of "internet of things" (IoT) based devices in recent years.
- **Understanding design primitives**: Building any type of complex system — be it software, mechanical, electrical, something else, or any combination of these — requires breaking it into a set of simpler sub-systems, which might consist of even more sub-systems. Introducing different layers of abstraction in this manner is an important tool for managing complexity, and circuit design is one of the most powerful ways of introducing you to this process. Just as software programs are typically built

from many subroutines constructed from a set of primitive data types (strings, integers, floating-point numbers, lists, etc.), an electrical circuit typically consists of sub-circuits built from simple primitive building blocks (resistors, capacitors, voltage/current sources) that place constraints on each other.

- **Translation to other fields**: Most circuits can be expressed as a system of equations that can be solved using linear algebraic techniques, making circuit analysis just one example of how the linear algebra concepts you'll learn about in this class can be used. In fact, this mathematical approach to circuit analysis ties directly to other fields of engineering, which can generally be characterized as the study of "effort" and "flow" variables — one variable that applies a force ("effort"), and another variable that moves in response to this ("flow"). What does that mean? In electrical circuit analysis, the "effort" variable is voltage, and the "flow" variable is current. In mechanical engineering, the "effort" variable could be a physical force on an object, and the corresponding "flow" variable would be the velocity of this object. In chemical engineering, "effort" is a chemical potential, while "flow" is molar flow, the amount of a substance that passes into a fixed area over time. While you'll be learning about these ideas in the context of circuits specifically in 16A, they are relevant to many other applications. Even circuit components like resistors and capacitors have mechanical analogues — resistance behaves like a mechanical damping force such as friction, while capacitance is like the compliance of a spring.[4]

- **Relevance to neural networks**: Because circuit analysis translates to a wide range of fields, we can model many physical systems as electrical circuits, often gaining insight about the system. You may have heard of neural networks, an important machine learning tool that can be used to "learn" tasks such as image and voice recognition from examples instead of explicit programming. Neural networks are modeled after biological neural networks, which are fundamentally circuits operating on electrical signals within a brain:

Biological Neuron

Circuit Model

Artificial Neuron Model

In general, studying circuits provides you with the conceptual and mathematical tools needed to analyze such networks. More broadly, circuit concepts are relevant to understanding network analysis and signal flows in systems, which can be applied to areas ranging from transportation analysis to social network analysis.

[4]There's a whole Wikipedia article on the matter if you want more details: `https://en.wikipedia.org/wiki/Mechanical-electrical_analogies`

- **Interdisciplinary thinking**: Our last example with neural networks brings up an important point: being exposed to a broad range of disciplines can help you come up with new ideas. In the case of neural networks, biological systems inspired a key modern machine learning tool. Understanding concepts outside of your field gives you a broader range of ideas to pull from when trying to develop innovative solutions.

0.5 Additional Resources

Throughout these notes you will find references to additional resources on the topics covered in this class. The following two textbooks will be referred to in the "Additional Resources" boxes in some of the upcoming notes.

- **Introduction to Linear Algebra by Gilbert Strang, 5th Ed.** (referred to as *Strang* in these notes)
- **Linear Algebra by Lipschutz, Seymour and Lipson, Marc, Schaum's Outlines, 5th Ed.** (referred to as *Schaum's* in these notes)

Acknowledgements

The reality is that these notes are the combined effort of many people. In particular, credit for the overall vision of the Linear-Algebraic side of the course and how to present these ideas belongs largely to Gireeja Ranade. The overall aesthetic[5] of 16A owes a great deal to both Gireeja Ranade and Elad Alon, the main instructors for the pilot Sp15 offering and designers of the 16AB course sequence. The other instructors in the pilot, namely Babak Ayazifar, Claire Tomlin, and Vivek Subramaniam also had considerable influence on the ideas here. Furthermore, the faculty who helped put together the ideas for the labs, like Miki Lustig and Laura Waller (imaging) and Tom Courtade (locationing), and Michel Maharbiz (spirit), helped shape the overall cadence of the course. The undergrads who worked tirelessly over the summer (2015) after the pilot shaped these notes considerably, with particular credit going to Chenyang Yuan, Rachel Zhang, and Siddharth Karamcheti.

Much work was put into the notes over the years, with credit to: Chenyang Yuan, Anant Sahai, Jennifer Shih, Elad Alon, Babak Ayazifar, Dominic Labanowski, Sajjad Moazeni, Nick Sutardja, Angie Wang, Amy Whitcombe, Vladimir Stojanovic, Grace Kuo, Rahul Arya, and Gireeja Ranade. A major new and radically different freshman course doesn't just happen — lots of people in the EECS department worked together for years to make all this a reality.

[5]Proper credit here also acknowledges the strong cultural influence of the faculty team that made EECS 70 and shaped it into a required course: Alistair Sinclair, Satish Rao, David Tse, Umesh Vazirani, Christos Papadimitriou, David Wagner, and Anant Sahai, along with pioneering TA's like Thomas Vidick, all helped make and polish the notes for 70. The 16AB+70 sequence all follows the aesthetic that mathematical ideas should have practical punchlines. 16AB emphasizes the rooting of mathematical intuition in physical examples as well as virtual ones.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 1A

Overview

In this note, we will introduce the fundamental objects of **vectors** and **matrices**, and discuss how to use them to represent systems of linear equations. We will further introduce the concept of **linearity**, and discuss how its formal definition relates to our intuitive understanding of "linear" functions. To illustrate all of these concepts, we will discuss the real-world technique of tomographic imaging.

1.1 What is Linear Algebra?

- Linear algebra is the study of linear functions and linear equations, typically via their representation using vectors and matrices.
- A lot of objects in EECS can be treated as vectors and studied with linear algebra.
- Linearity is a good first-order approximation to the complicated real world.
- There exist good fast algorithms to do many of these manipulations in computers.
- Linear algebra concepts are an important tool for modeling the real world.

As you will see in the homeworks and labs, these concepts can be used to do many interesting things in real-world-relevant application scenarios. In the previous note, we introduced the idea that all information devices and systems (1) take some piece of information from the real world, (2) convert it to the electrical domain for measurement, and then (3) process these electrical signals. Because so many efficient algorithms exist that perform linear algebraic manipulations with computers, linear algebra is often a crucial component of this processing step.

1.2 Application: Tomography

Throughout this course, we will motivate the introduction of concepts by considering a real-world application - this is the first one!

Tomography allows us to "see inside" a solid object, such as the human body or even the earth, by taking measurements with a penetrating wave, such as X-rays. CT scans in medical imaging are perhaps the most famous such example — in fact, CT stands for "computed tomography."

Let's look at a specific toy example, using tomography to help with a (fairly unlikely!) real-world scenario.

A grocery store employee just had a truck load of bottles given to him. Each bottle is either empty, contains milk, or contains juice, and the bottles are packaged in boxes, with each box containing 9 bottles in a 3×3 grid. Inside a single box, it might look something like this:

If we choose symbols such that M represents milk, J represents juice, and O represents an empty bottle, we can represent the stack of bottles shown above as follows:

$$\begin{matrix} M & J & O \\ M & J & O \\ M & O & J \end{matrix} \tag{1}$$

Imagine that our grocer cannot see directly into the box, but needs to determine its contents using a light source and light sensor. How can we help him do this?

Let the light source emit light with a certain known intensity. As the light passes through a bottle, its intensity diminishes by an amount that depends on the contents of the bottle - milk absorbs 3 units of light, juice absorbs 2 units of light and an empty bottle absorbs 1 unit of light. The box itself does not affect the intensity of the light. After the light emitted exits the box, we can use our sensor to measure the final intensity, and so determine the amount of light absorbed by each bottle.

Thus, if we shine light in a straight line through some bottles within the box, we can determine the total amount of light absorbed by the bottles as the sum of the light absorbed by each bottle. For instance, in our specific example, shining a light from left to right would look like this, with each row observed to absorb 6 total units of light:

In order to deal with this more generally, let's assign variables to the amount of light absorbed by each bottle:

This means that x_{11} would be the amount of light the top left bottle absorbs, x_{21} would be the amount of light the top middle bottle absorbs, and so forth. Shining the light from left to right for our specific example gives the following equations:

$$x_{11} + x_{21} + x_{31} = 6 \tag{2}$$
$$x_{12} + x_{22} + x_{32} = 6 \tag{3}$$
$$x_{13} + x_{23} + x_{33} = 6 \tag{4}$$

Similarly, we could consider shining a light from bottom to top:

Which would give the following equations:

$$x_{13} + x_{12} + x_{11} = 9 \tag{5}$$
$$x_{23} + x_{22} + x_{21} = 5 \tag{6}$$
$$x_{33} + x_{32} + x_{31} = 4 \tag{7}$$

Thus, we now know how to determine our observations given the contents of the box. But can we do the reverse? That is to say, given the amounts of light absorbed by each row and column of bottles, can we reconstruct the box's original contents?

From our above observations, one possible solution for the values of x_{ij} (corresponding to the actual configuration of bottles) is

$$\begin{matrix} 3 & 2 & 1 \\ 3 & 2 & 1 \\ 3 & 1 & 2 \end{matrix} \tag{8}$$

However, the following solution also works:

$$\begin{matrix} 3 & 2 & 1 \\ 3 & 1 & 2 \\ 3 & 2 & 1, \end{matrix} \tag{9}$$

which corresponds to a different configuration of bottles within the box. In other words, our observations are not sufficient to **uniquely** identify the configuration of bottles within the box. This is a problem!

Usually, if we can't identify an object from a set of observations, we make more observations. To get these additional observations, we could shine light at different angles through the box.

This brings up some very natural questions: Would shining light through the diagonals of the box provide us with enough information? If not, how many different directions do we need to shine light through before we are certain of the configuration of bottles? Do some measurements provide us with more information than others do? What happens as we vary the number of bottles in our box?

In Module 1 of this course, we will develop the tools to answer all of these questions.

1.3 What is a Linear Equation?

1.3.1 Intuition

In our bottle-sorting tomography example, we represented each measurement in a row or column as an equation. The collection of equations is an example of a **system of linear equations**, which summarizes the known relationships between the variables we want to solve for (x_{11}, x_{12}, x_{13}, etc.) and our measurements.

But what makes an equation "linear"? Essentially, a linear equation is one where each variable has degree 1. For instance, for unknowns x and y, the equation

$$5x + 6y = 7 \tag{10}$$

is made up of the two coefficients 5 and 6 multiplied by the two unknowns x and y respectively, summed together, that are then set equal to the scalar constant 7.

In contrast, the equation

$$y^2 = 5 \tag{11}$$

is not a linear equation, since we multiply our unknown y with itself, rather than a constant scalar.

Observe that equations such as

$$8x = 4y,$$

or

$$8x - 4y = 0,$$

or

$$2x - y = 0$$

are all examples of linear equations. Notice that the scalar constant is allowed to be 0.

1.3.2 Linear Functions

Let's try to formalize this intuition a little. First, we will define the concept of a **linear function**. At this point, we can define a linear function as simply a function f of one scalar argument with the property that, for arbitrary scalars α and x,

$$f(\alpha x) = \alpha \cdot f(x).$$

In other words, if the input to a linear function is multiplied by some scalar α, the output of the function will be multiplied by α as well.

To gain a better understanding of this definition, we'll consider a few examples. First, consider the function

$$f_1(x) = x^2.$$

Plugging in the values $x = 1$ and $x = 2$, we see that

$$f_1(1) = 1 \text{ and } f_1(2) = 4.$$

So when x is scaled by a factor of 2 (from $x = 1$ to $x = 2$), $f_1(x)$ scales by a factor of 4, which is not the same factor. Due to this counterexample, we have shown that x^2 is not a linear function.

Now, let's look at the function

$$f_2(x) = 5x.$$

Plugging in the same values for x as before, we see that

$$f_2(1) = 5 \text{ and } f_2(2) = 10.$$

Here, when x was scaled by a factor of 2 (from 1 to 2), $f_2(x)$ scaled by a factor of 2 as well (from 5 to 10)! This suggests that $f_2(x)$ *might* be a linear function.

Let's try to quickly prove it. For arbitrary scalars α and x, we see from the definition of $f_2(x)$ that

$$f_2(\alpha x) = 5 \cdot (\alpha x) = \alpha \cdot (5x) = \alpha \cdot f_2(x),$$

so $f_2(x)$ is indeed a linear function.

More generally, a similar proof to the above can be done to show that *all* functions of the form $f(x) = kx$ are linear - in fact, we can even show that *any* linear function with a single scalar input and output can be written in the form $f(x) = kx$, for some constant k! Let's try to prove this!

Let $f(x)$ be an arbitrary linear function. Observe that, by the definition of linear functions,

$$f(x) = f(x \cdot 1) = x \cdot f(1).$$

Clearly, $f(1)$ does not depend on x, and so is a scalar constant. Let this scalar constant be $k = f(1)$. Thus, we have shown that there exists a k such that

$$f(x) = kx,$$

no matter what $f(x)$ we choose, so long as it is a linear function with scalar inputs and outputs.

1.3.3 Affine Functions

But what about functions like

$$f_3(x) = 2x + 1?$$

Plotting this function, we see that it is a line. But it doesn't seem to fit into the form $f(x) = kx$, so is it linear? A simple check, if we're ever unsure about the behavior of a function, is to plug in some simple input values and see how the output behaves. Let's do that here, for $x = 1$ and $x = 2$. We see that

$$f_3(1) = 3 \text{ and } f_3(2) = 5,$$

so doubling the input value from 1 to 2 changes the output by a factor of $5/3$. Thus, this function is not linear, *even though* it describes the equation of a line.

Nevertheless, it still seems to have some "linear" properties, compared to an expression like x^2. More precisely, observe that it can be written as a sum of a linear function of x and a scalar constant. We define **affine functions** to be the set of functions that can be written as a sum of a linear function and a scalar constant, so though $f_3(x)$ is not *linear*, it is still *affine*.

Notice that the definition of affine functions includes all linear functions as a subset (by setting the scalar constant to 0), so every linear function is also affine, though not vice-versa.

These definitions mean that while all functions describing a line can be shown to be affine, not all of them are linear. This has the unfortunate consequence that, in informal conversation, *affine* functions may be called *linear*, since both describe a line. This usage, though common, is **wrong**.

1.3.4 Linear Equations

So what's a linear equation, then? Formally, a linear equation with the unknown scalars $x_1, x_2, \ldots, x_n$ is an equation where each side is a sum of scalar-valued linear functions of each of the unknowns plus a scalar constant.

What does this mean? Expressed algebraically, we obtain the most general form of a linear equation, where the f_i and g_i are each linear functions with a single scalar input and output, and b_f and b_g are two scalar constants:

$$f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) + b_f = g_1(x_1) + g_2(x_2) + \ldots + g_n(x_n) + b_g.$$

Now, recall that linear functions with a single scalar input and output can be expressed in a very particular form - we know that we can write $f_i(x) = a_i \cdot x$, and $g_i(x) = a_i' \cdot x$, where all the a_i and a_i' are scalar constants. Substituting, we find that the general form of a linear equation can be rewritten as

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n + b_f = a_1'x_1 + a_2'x_2 + \ldots + a_n'x_n + b_g.$$

Let's take a closer look at part of this equation - specifically, the expression

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n.$$

Notice that this expression can be thought of as a "weighted sum" of the x_i, where the weights are the scalar constants a_i. When the weights do not depend on any of the terms (such as when the weights are constants), we call the weighted sum a **linear combination** of said terms. So the above expression is typically referred to as a *linear combination* of the x_i.

Thus, we can simplify our definition of a linear equation, to the following: A linear equation is one that equates two linear combinations of the unknowns plus a constant term.

1.4 Vectors and Matrices

We will now introduce some new notation that will help us deal with systems of linear equations in a more compact form.

1.4.1 Vectors

Definition 1.1 (Vector):

A vector is an ordered list of numbers. Suppose we have a collection of n real numbers: $x_1, x_2, \cdots, x_n$. This collection can be written as a single point in an n-dimensional space, denoted as:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{12}$$

We call $\vec{x}$ a **vector**. Because $\vec{x}$ contains n real numbers, we can use the $\in$ ("in" — i.e., is a member of) symbol to write $\vec{x} \in \mathbb{R}^n$ ($\mathbb{R}$ represents the set of real numbers). If the elements of $\vec{x}$ were complex numbers, we would write $\vec{x} \in \mathbb{C}^n$. Each x_i (for i between 1 and n) is called a **component**, or **element**, of the vector. The **size** of a vector is the number of components it contains (so the example vector is of size n, and the example below is of size two).

Example 1.1 (Vector of size two):

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

In the above example, $\vec{x}$ is a vector with two components. Because the components are both real numbers, $\vec{x} \in \mathbb{R}^2$. We can represent the vector graphically on a 2-D plane, using the first element, x_1, to denote the horizontal position of the vector and the second element, x_2, to denote its vertical position:

Additional Resources For more on vectors, read pages 1-6 of *Strang* and try Problem Set 1.1. *Extra: Try reading the portions on linear combinations which generate a "space."*

Read more on vectors in *Schuam's* on pages 1-3 and try Problems 1.1 to 1.6.

1.4.2 Matrices

Definition 1.2 (Matrix): A **matrix** is a rectangular array of numbers, written as:

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} \tag{13}$$

Each A_{ij} (where i is the row index and j is the column index) is a **component**, or **element** of the matrix A.

Example 1.2 (4×3 **Matrix**):

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 8 & 12 \end{bmatrix}$$

In the example above, A has $m = 4$ rows and $n = 3$ columns (a 4×3 matrix).

1.5 Representing Linear Systems using Matrices

1.5.1 Augmented Matrices

First, we will use what we call an **augmented matrix** to represent the coefficients of a linear system of equations, like the system we saw earlier when experimenting with tomography:

$$x_{11} + x_{21} + x_{31} = 6 \tag{14}$$
$$x_{12} + x_{22} + x_{32} = 6 \tag{15}$$
$$x_{13} + x_{23} + x_{33} = 6 \tag{16}$$
$$x_{13} + x_{12} + x_{11} = 9 \tag{17}$$
$$x_{23} + x_{22} + x_{21} = 5 \tag{18}$$
$$x_{33} + x_{32} + x_{31} = 4. \tag{19}$$

Recall that the first three equations of the above system came from measuring the light absorbed through each row of bottles, while the second three equations came from measurements of the columns.

First, we will rewrite our system slightly to introduce some additional structure. Specifically, notice that each equation involves only a subset of our unknowns: for instance, the first equation only includes x_{11}, x_{21}, and x_{31}, and not the other six unknowns in the problem. We can make sure all nine unknowns are included in each equation by introducing linear terms with a zero coefficient, as follows:

$$1 \times x_{11} + 0 \times x_{12} + 0 \times x_{13} + 1 \times x_{21} + 0 \times x_{22} + 0 \times x_{23} + 1 \times x_{31} + 0 \times x_{32} + 0 \times x_{33} = 6 \tag{20}$$
$$0 \times x_{11} + 1 \times x_{12} + 0 \times x_{13} + 0 \times x_{21} + 1 \times x_{22} + 0 \times x_{23} + 0 \times x_{31} + 1 \times x_{32} + 0 \times x_{33} = 6 \tag{21}$$
$$0 \times x_{11} + 0 \times x_{12} + 1 \times x_{13} + 0 \times x_{21} + 0 \times x_{22} + 1 \times x_{23} + 0 \times x_{31} + 0 \times x_{32} + 1 \times x_{33} = 6 \tag{22}$$
$$1 \times x_{11} + 1 \times x_{12} + 1 \times x_{13} + 0 \times x_{21} + 0 \times x_{22} + 0 \times x_{23} + 0 \times x_{31} + 0 \times x_{32} + 0 \times x_{33} = 9 \tag{23}$$
$$0 \times x_{11} + 0 \times x_{12} + 0 \times x_{13} + 1 \times x_{21} + 1 \times x_{22} + 1 \times x_{23} + 0 \times x_{31} + 0 \times x_{32} + 0 \times x_{33} = 5 \tag{24}$$
$$0 \times x_{11} + 0 \times x_{12} + 0 \times x_{13} + 0 \times x_{21} + 0 \times x_{22} + 0 \times x_{23} + 1 \times x_{31} + 1 \times x_{32} + 1 \times x_{33} = 4 \tag{25}$$

So far, we've made some fairly elementary transformations to our system of linear equations. But what's the point? While our resultant system is both longer and harder to read than our original system, it is structured. This will allow us to easily use a standard algorithm to process and solve the system.

A natural next step is to put the coefficients into what we call an "augmented matrix" as below. Each row of the augmented matrix represents one equation. Each column corresponds to the coefficients for a given variable across all equations - i.e. the first column corresponds to x_{11}, the second to x_{12}, and so on:

$$\left[\begin{array}{ccccccccc|c} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 6 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 4 \end{array}\right] \tag{26}$$

Typically, when representing a system of linear equations as a single matrix, it is conventional to draw a line before the last column as done above, since that column contains the constants, not the coefficients, of our system:

This representation of a linear system is known as the **augmented matrix representation**. An interesting thing to notice about this representation is that the symbols corresponding to our unknowns have vanished entirely!

1.5.2 Matrix-Vector Form

Intuitively, we can imagine that augmented matrices convey the "underlying" linear system corresponding to our problem, without involving the actual variable names that we chose. Still, we might want to preserve the variable names, since without them we are losing some information about our problem. To do so, we

can write our above system of equations in the following form:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \\ 6 \\ 9 \\ 5 \\ 4 \end{bmatrix}.$$

What's going on here? Compared to the augmented matrix form, notice that the constant terms have not been included into the main matrix (known as the *coefficient matrix*), but instead have been placed in a vector on the right-hand-side of an equality. In addition, observe that the unknowns have been placed in a vector that is right-multiplied with the coefficient matrix on the left-hand-side of the equality. Often, the letter A is used to represent the coefficient matrix, $\vec{x}$ for the vector of unknowns, and $\vec{b}$ for the vector of coefficients. After these substitutions, the above equation becomes

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ \vec{x} &= \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{bmatrix} \\ \vec{b} &= \begin{bmatrix} 6 \\ 6 \\ 6 \\ 9 \\ 5 \\ 4 \end{bmatrix} \\ \implies \quad A\vec{x} &= \vec{b}. \end{aligned}$$

The above representation is fundamentally an alternative shorthand for systems of linear equations that preserves the variable names of the original system.

1.6 Practice Problems

These practice problems are also available in an interactive form on the course website, along with their solutions.

1. Is $x + 2y = 4z$ linear?
2. Is $\sin x - 2 = 6$ linear?
3. Is $\sum_{i=1}^{50} i \cdot x - e^{-3}y = \sin \frac{\pi}{3}$ linear?
4. Write $\begin{cases} 2x - 3y = 1 \\ 3x + y = -2 \end{cases}$ in matrix form.

   (a) $\begin{bmatrix} 2 & -3 & 3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

   (b) $\begin{bmatrix} 2 & 3 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

   (c) $\begin{bmatrix} 2 & -3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

   (d) $\begin{bmatrix} 2 & -3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 1B

Overview

In the previous note, we considered how to represent systems of linear equations. Now, we will develop a systematic algorithm, known as **Gaussian Elimination**, to solve arbitrary systems of linear equations, or determine that no solutions exist.

1.1 Example

We may already be able to solve linear systems in a so-called "ad-hoc" manner - manipulating equations according to our intuition until we get a solution or have convinced ourselves that no solution exists. Let's remind ourselves of how we could do that.

Consider the linear system with unknowns x and y:

$$\begin{aligned} 5x+6y &= 40 \\ 8x+9y &= 61. \end{aligned}$$

To solve it, one way would be to use the first equation to express y in terms of x, and then substitute into the second to obtain a single linear equation involving only x. Rearranging the first equation, we find that

$$\begin{aligned} & 5x+6y = 40 \\ \implies \quad & 6y = 40-5x \\ \implies \quad & y = \frac{20}{3}-\frac{5}{6}x. \end{aligned}$$

Now, substituting into the second equation, we obtain

$$\begin{aligned} & 8x+9y = 61 \\ \implies \quad & 8x+9\left(\frac{20}{3}-\frac{5}{6}x\right) = 61 \\ \implies \quad & \frac{1}{2}x+60 = 61 \\ \implies \quad & \frac{1}{2}x = 1 \\ \implies \quad & x = 2. \end{aligned}$$

Finally, substituting this value for x into our equation for y in terms of x, we find

$$\begin{aligned} y &= \frac{20}{3} - \frac{5}{6}x \\ &= \frac{20}{3} - \frac{5}{6} \cdot 2 \\ &= 5. \end{aligned}$$

This ad-hoc, substitution-based approach worked well for this small system. However, as we start to consider larger systems, potentially with thousands or even millions of variables, manually solving them becomes infeasible.

We need to develop a systematic approach to solve linear systems that generalizes well as the numbers of equations and variables increases. After developing this approach, we should be able to prove (at least to a certain level of rigor) that our approach is guaranteed to work whenever a solution to a linear system exists.

1.2 Gaussian Elimination

Gaussian elimination is an **algorithm** (a sequence of programmatic steps) that accomplishes this task. Specifically, it can be used to solve any arbitrarily large system of linear equations, or decide that no solution exists. Gaussian elimination isn't the *only* algorithm that does this (for instance, we could try writing an algorithm that formalizes the substitution-based approach that we tried above), but it's pretty good!

Side note: Gaussian elimination was named after Carl Friedrich Gauss, a German mathematician from the 18th century. Despite being its namesake, he did not invent it, though he contributed to its development. As it turns out, Gaussian elimination was initially developed in China over 2000 years ago!

In Europe, Gaussian elimination was refined over the course of 200 years by mathematicians including Newton, Rolle, and Gauss. To quote Newton,

> *And you are to know, that by each Æquation one unknown Quantity may be taken away, and consequently, when there are as many Æquations and unknown Quantities, all at length may be reduc'd into one, in which there shall be only one Quantity unknown.*

To read about how it evolved, check out `https://www.ams.org/notices/201106/rtx110600782p.pdf`!

1.2.1 Example

Before we look at a precise formulation of Gaussian elimination, let's look at an example of how it works, to build our intuition. Specifically, let's try solving the following example using Gaussian elimination:

$$\begin{aligned} 5x+6y+z &= 43 \quad (1) \\ 8x+9y+2z &= 67 \quad (2) \\ x+y+4z &= 19. \quad (3) \end{aligned}$$

Intuitively, the basic idea behind Gaussian elimination is to use each equation to *eliminate* one variable from all the subsequent equations, until we end up with an equation with just one unknown, which we can directly solve (we'll make this intuition more rigorous in a moment). Let's try using the first equation to eliminate one of our unknowns - say, x. First, we **multiply** the first equation by $1/5$, in order to remove the coefficient of 5 in front of x. This is called normalizing. Multiplying every term in the equation doesn't change the solution, so is allowed. Thus, we obtain

$$x+\frac{6}{5}y+\frac{1}{5}z=\frac{43}{5}. \quad (4)$$

Now, we will try to eliminate the variable x in (Eq. 2) and (Eq. 3). One way would be to write x in terms of y and substitute, like we did earlier. Instead, however, Gaussian elimination requires us to **add multiples** of (Eq. 4) from (Eq. 2) and (Eq. 3), in order to accomplish the same goal (we'll see why we're currently trying to avoid substitution in a moment, when we make this process more rigorous).

Let's try eliminating x from (Eq. 2) first. What multiple of (Eq. 4) would be best to use? Well, since (Eq. 4) has an x term, and we'd like it to cancel out with the $8x$ term from (Eq. 2), it makes sense to multiply (Eq. 3) by -8 and add it from (Eq. 2) to produce (Eq. 5):

$$\begin{aligned} & (8x+9y+2z)-8\cdot\left(x+\frac{6}{5}y+\frac{1}{5}z\right)=67-8\cdot\frac{43}{5} \\ \implies \quad & \frac{-3}{5}y+\frac{2}{5}z=-\frac{9}{5} \qquad (5) \end{aligned}$$

Similarly, it would make sense to subtract (Eq. 4) from (Eq. 3) (equivalently, to multiply (Eq. 4) by -1 and add it to (Eq. 3)), since both (Eq. 3) and (Eq. 4) have an x term with a coefficient of unity. This produces

$$\begin{aligned} & (x+y+4z)-\left(x+\frac{6}{5}y+\frac{1}{5}z\right)=19-\frac{43}{5} \\ \implies \quad & -\frac{1}{5}y+\frac{19}{5}z=\frac{52}{5}. \qquad (6) \end{aligned}$$

Now, observe that (Eq. 5) and (Eq. 6) together form a linear system with just two unknowns: y and z, since x has been eliminated from the latter two equations. Now, let's see if we can repeat this process, using (Eq. 5) to eliminate y from (Eq. 6). First, as before, we should simplify things by scaling (Eq. 5) to remove the coefficient of y. Multiplying (5) by $-5/3$, we obtain

$$y-\frac{2}{3}z=3. \quad (7)$$

Again, we'd like to add some scalar multiple of (Eq. 7) from (Eq. 6), in order to eliminate y from (Eq. 6). Since (Eq. 6) has a $(-1/5)y$ term, it makes sense to multiply (Eq. 7) by $1/5$ and add it to (Eq. 6), which produces

$$\begin{aligned} -\frac{1}{5}y + \frac{19}{5}z + \frac{1}{5}\left(y - \frac{2}{3}z\right) &= \frac{52}{5} + \frac{1}{5}\cdot 3 \\ \implies \qquad \frac{11}{3}z &= 11. \end{aligned} \tag{8}$$

Now, (Eq. 8) is a simple linear equation in only one variable, which we know how to immediately solve. Awesome! Rearranging a little, we see that

$$z = 3.$$

When we did our initial example above, after solving for x, we substituted it into an equation for y in terms of x. Can we do something similar here? Specifically, do we have an equation for x or y in terms of just z?

Notice that the equation we used to eliminate y (Eq. 7) had no variables "before" y, since they were eliminated, and doesn't even have a coefficient for y. Pulling all terms except for y onto the right-hand-side of the equality, we obtain

$$y = 3 + \frac{2}{3}z.$$

Perfect! Substituting in $z = 3$, we find that

$$y = 3 + \frac{2}{3}\cdot 3 = 5.$$

Can we do the same thing again? Well, when we eliminated x using (Eq. 4), we again needed an equation with a unit coefficient for x. Pulling all the terms in (Eq. 4) except for x to the right-hand-side, we obtain

$$x = \frac{43}{5} - \frac{6}{5}y - \frac{1}{5}z.$$

Again, this is looking pretty good. Substituting in our known values for y and z, we find that

$$x = \frac{43}{5} - \frac{6}{5}\cdot 5 - \frac{1}{5}z = 2,$$

so we've solved for all of our unknowns using Gaussian elimination. Awesome!

1.2.2 Steps of Gaussian Elimination

Let's take a moment to reflect on the approach we just used.

- First, we selected an equation involving x (possibly with some coefficient) and scaled it to make the x coefficient unity.
- Then, we added multiples of this equation from all the other equations to eliminate x, producing a system with one fewer unknowns and one fewer equations.
- We then repeated the first two steps until we arrived at an equation with exactly one unknown, which we could solve directly.

- Finally, we substituted the known value of the final unknown into a previous equation to recover the last two unknowns, and continued substituting until we recovered all of our unknowns!

These are the key steps of Gaussian elimination. The first three steps are known as **row reduction**, and the final step is known as **back-substitution**.

Now that we know how to perform the steps of Gaussian elimination for systems where a solution is known to exist, it is important to ask ourselves *why* these steps work. In particular, even if we have some intuition for why they work in cases when a system of equations has a unique solution, we'd like to show that these steps remain valid even when working with a system with zero or infinitely many solutions.

1.2.3 Operations

The key idea behind Gaussian elimination is that of "invertible" operations. As we manipulate our equations, we want to preserve their set of solutions. In particular, we neither want to *introduce* new solutions in the process, nor to *remove* potential solutions. To do so, we use three operations that we are certain will *never* change the solution set of a system, and then apply these operations repeatedly in order to solve a linear system. By only applying these operations, we can be confident that our approach will never yield a wrong answer, since the solution set of our system is preserved throughout.

These operations, which we have just seen, are as follows:

1. **Multiplying an equation by a *nonzero* scalar constant.** For instance, if we have the equation

$$2a + 3b = 4,$$

we can multiply it by the nonzero scalar -2 to obtain

$$-4a - 6b = -8.$$

Why does this operation preserve all the solutions to a system? Well, any particular solution that satisfies the first equation still satisfies the second equation, so this operation has not removed any potential solutions.

But does it introduce a new solution? Consider any particular solution to the second equation. Notice that we can multiply the second equation by the *reciprocal* of our original nonzero scalar multiplier, to obtain the first equation. Thus, this particular solution will also satisfy the first equation. In other words, no solution exists that satisfies the second equation, but not the first. Consequently, the second equation is not only *implied by*, but also *implies* the first equation.

When each of two equations imply the other, we say that they are *equivalent*, since replacing one with the other does not change their solution set. Notice that, to obtain equivalence, we had to restrict our multiplication to one by a *nonzero* scalar, not an arbitrary scalar. Otherwise, we would not be able to obtain the first equation from the second (since the reciprocal of zero is undefined), so we would only obtain a one-way implication, not two-way equivalence.

2. **Adding a scalar constant multiple of one equation to another.** For instance, if we have the equa-

tions

$$\begin{aligned} 5a+6b &= 7 \qquad (1) \\ 8a+9b &= 10, \quad (2) \end{aligned}$$

we can multiply the second equation by the scalar 3 and add it to the first, to obtain the new system

$$\begin{aligned} 29a+33b &= 37 \qquad (3) \\ 8a+9b &= 10, \quad (2) \end{aligned}$$

Any solution to the first system will also be a solution to the second, since the first system of equations implies the second. But is the reverse true? Well, observe that equation (1) can be recovered by taking equation (3) and subtracting our scalar (in this case, 3) multiplied by equation (2). In other words, our second system is, not only implied by, but also implies the first system, so it does not introduce any new solutions. Thus, replacing the first system with the second does not change the solution set of our linear system, so this operation is valid.

3. **Swapping two equations.** We have not yet seen when we need to swap two equations (though we will in Example 1.3), but it is clear that the solution set of a linear system of equations does not depend on the order of equations! Therefore, this final operation is clearly valid.

Now that we have developed these three operations, we can repeatedly use them in a structured manner to solve arbitrary systems of linear equations, as will be illustrated in the following examples:

Example 1.1 (System of 2 equations): Consider the following system of two equations with two variables:

$$\begin{array}{rcrcrl} x & - & 2y & = & 1 & \quad (1) \\ 2x & + & y & = & 7 & \quad (2) \end{array}$$

We would like to find an explicit formula for x and y, but the presence of both x and y in each of the equations prevents this. If we can eliminate a variable from one of the equations, we can get an explicit formula for the remaining variable. To eliminate x from (Eq. 2), we can subtract 2 times (Eq. 1) from (Eq. 2) to obtain a new equation, (Eq. $2'$):

$$\begin{array}{rcrcrl} 2x & + & y & = & 7 & \\ -2\times(x & - & 2y & = & 1) & \\ & & \Downarrow & & & \\ 2x & + & y & = & 7 & \\ -2x & + & 4y & = & -2 & \\ & & \Downarrow & & & \\ & & 5y & = & 5 & \quad (2') \end{array}$$

Scaling (Eq. 1) by the amount that x is scaled in (Eq. 2) allows us to cancel the x term. As a result, we can replace (Eq. 2) with (Eq. $2'$) to rewrite our system of equations as:

$$\begin{array}{rrcrcrl} & x & - & 2y & = & 1 & \quad (1) \\ \text{(Eq. 2)} - 2\times\text{(Eq. 1)}: & & & 5y & = & 5 & \quad (2') \end{array}$$

From here, we can divide both sides of (Eq. $2'$) by 5 to see that $y = 1$. We will call this (Eq. $2''$). Next, we would like to solve for x. It would be natural to proceed by substituting $y = 1$ into (Eq. 1) to solve directly

for x, and doing this will certainly give you the correct result. However, our goal is to find an *algorithm* for solving systems of equations. This means that we would like to be able to repeat the same sequence of operations over and over again to come to the solution. Recall that to cancel x in (Eq. 2), we:

1. **Scaled** (Eq. 1) by a factor of 2.
2. **Subtracted** (Eq. 1) from (Eq. 2).

To solve for x, we would like to eliminate y from (Eq. 1) using a similar process of scaling and subtracting. Because y is scaled by a factor of -2 in (Eq. 1), we can scale (Eq. $2''$) by -2 and subtract it from (Eq. 1) to cancel the y term. Doing so gives:

$$\begin{array}{rrcll} \text{(Eq. 1)} + 2 \times \text{(Eq. 2}''\text{):} & x & & = & 3 & (1') \\ \text{(Eq. 2}'\text{)} \,/5\text{:} & & y & = & 1 & (2'') \end{array}$$

Soon we will generalize this technique so that it can be extended to any number of equations. Right now we will use an example with 3 equations to help build intuition.

Example 1.2 (System of 3 equations): Suppose we would like to solve the following system of 3 equations:

$$\begin{array}{rcrcrcrl} x & - & y & + & 2z & = & 1 & (1) \\ 2x & + & y & + & z & = & 8 & (2) \\ -4x & + & 5y & & & = & 7 & (3) \end{array}$$

As in the 2 equation case, our first step is to eliminate x from all but one equation by adding or subtracting scaled versions of the first equation from the remaining equations. Because x is scaled by 2 and -4 in (Eq. 2) and (Eq. 3) (respectively), we can multiply (Eq. 1) by these factors and subtract it from the corresponding equations:

$$\begin{array}{rrcrcrcrl} & x & - & y & + & 2z & = & 1 & (1) \\ \text{(Eq. 2)} - 2\times \text{(Eq. 1):} & & & 3y & - & 3z & = & 6 & (2') \\ \text{(Eq. 3)} + 4\times \text{(Eq. 1):} & & & y & + & 8z & = & 11 & (3') \end{array}$$

Next, we would like to eliminate y from (Eq. $3'$). First, we can divide (Eq. $2'$) by 3 such that y is scaled by 1:

$$\begin{array}{rrcrcrcrl} & x & - & y & + & 2z & = & 1 & (1) \\ \text{(Eq. 2}'\text{)} \,/3\text{:} & & & y & - & z & = & 2 & (2'') \\ & & & y & + & 8z & = & 11 & (3') \end{array}$$

Now, since y is also scaled by 1 in (Eq. $3'$), we can subtract (Eq. $2''$) from (Eq. $3'$) to get a formula[1] with only z:

$$\begin{array}{rrcrcrcrl} & x & - & y & + & 2z & = & 1 & (1) \\ & & & y & - & z & = & 2 & (2'') \\ \text{(Eq. 3}'\text{)} - \text{(Eq. 2}''\text{):} & & & & & 9z & = & 9 & (3'') \end{array}$$

[1] At this point we have made a decision in our algorithm to eliminate y from (Eq. $3'$) but *not* (Eq. 1). The motivation for this might not be completely evident now, but approaching it this way can be more computationally efficient for certain systems of linear equations — typically if the system has an infinite number of solutions or no solutions.

Dividing (Eq. $3''$) by 9 gives an explicit formula for z:

$$\begin{array}{rrcrcrcrl} & x & - & y & + & 2z & = & 1 & (1) \\ & & & y & - & z & = & 2 & (2'') \\ \text{(Eq. } 3'')/9: & & & & & z & = & 1 & (3''') \end{array}$$

At this point, we can see that our system of equations has a "triangular" structure — all three variables are contained in (Eq. 1), two are in (Eq. $2''$), and only z remains in (Eq. $3'''$). If we look back to the previous example with 2 equations, we obtained a similar result after eliminating x from (Eq. 2):

System of 2 Equations

$$\begin{array}{rcrcr} x & - & 2y & = & 1 \\ & & 5y & = & 5 \end{array}$$

System of 3 Equations

$$\begin{array}{rcrcrcr} x & - & y & + & 2z & = & 1 \\ & & y & - & z & = & 2 \\ & & & & z & = & 1 \end{array}$$

This similarity is not coincidental, but a direct result of the way in which we successively eliminate variables in our algorithm moving left to right. To understand why it is useful to have our system of equations in this format, we will now proceed to solve for the remaining variables in this 3-equation example. First, we would like to eliminate z from (Eq. 1) and (Eq. $2''$). As usual, we can accomplish this by scaling (Eq. $3'''$) by the amount z is scaled in (Eq. 1) and (Eq. $2''$) and subtracting this from these equations:

$$\begin{array}{rrcrcrcrl} \text{(Eq. 1)} - 2\times \text{(Eq. } 3'''): & x & - & y & & & = & -1 & (1') \\ \text{(Eq. } 2'') + \text{(Eq. } 3'''): & & & y & & & = & 3 & (2''') \\ & & & & & z & = & 1 & (3''') \end{array}$$

Finally, by adding (Eq. $2'''$) to (Eq. $1'$), we can find the solution:

$$\begin{array}{rrcrcrcrl} \text{(Eq. } 1') + \text{(Eq. } 2'''): & x & & & & & = & 2 & (1'') \\ & & & y & & & = & 3 & (2''') \\ & & & & & z & = & 1 & (3''') \end{array}$$

After obtaining an explicit equation for z using a repetitive process of scaling and subtraction, we were able to obtain an explicit equation for y, and then x, using this same process — this time propagating equations upwards instead of downwards.

So far, the two operations we've previously encountered seem to be sufficient to solve every system of equations we've encountered. Are there any other operations we might need to perform in addition to scaling and adding/subtracting equations?

Example 1.3 (System of 3 equations): Suppose we would like to solve the following system of 3 equations:

$$\begin{array}{rcrcrcrl} & & 2y & + & z & = & 1 & (1) \\ 2x & + & 6y & + & 4z & = & 10 & (2) \\ x & - & 3y & + & 3z & = & 14 & (3) \end{array}$$

As in the 2 equation case, our first step is to eliminate x from all but one equation. Since x is the first variable to be eliminated, we want the equation containing it to be at the top. However, the first equation does not contain x. To solve this problem, we **swap** the first two equations. Clearly, swapping two equations does not

change the system's solution set, so we will obtain the equivalent linear system:

$$\begin{array}{rrcrcrcrl} \text{(Eq. 2):} & 2x & + & 6y & + & 4z & = & 10 & (1') \\ \text{(Eq. 1):} & & & 2y & + & z & = & 1 & (2') \\ & x & - & 3y & + & 3z & = & 14 & (3) \end{array}$$

Now we can proceed as usual, dividing the first equation by 2, and then subtracting this from (Eq. $3'$) to eliminate x:

$$\begin{array}{rrcrcrcrl} \text{(Eq. 1}')\ /2: & x & + & 3y & + & 2z & = & 5 & (1'') \\ & & & 2y & + & z & = & 1 & (2') \\ \text{(Eq. 3)} - \text{(Eq. 1}''): & & - & 6y & + & z & = & 9 & (3') \end{array}$$

Now there is only one equation containing x. Of the remaining two equations, we want only one of them to contain y, so we can divide (Eq. $2'$) by 2 and then add 6 times (Eq. $2''$) to (Eq. $3'$):

$$\begin{array}{rrcrcrcrl} & x & + & 3y & + & 2z & = & 5 & (1'') \\ \text{(Eq. 2}')\ /2: & & & y & + & \frac{1}{2}z & = & \frac{1}{2} & (2'') \\ \text{(Eq. 3}') + 6\times \text{(Eq. 2}''): & & & & & 4z & = & 12 & (3'') \end{array}$$

Now, we have the "triangular" structure from the previous examples. To proceed, we can divide the last equation by 4 to solve for $z = 3$, and use this to eliminate z from the remaining equations:

$$\begin{array}{rrcrcrcrl} \text{(Eq. 1}'') - 2\times \text{(Eq. 3}'''): & x & + & 3y & & & = & -1 & (1''') \\ \text{(Eq. 2}'') - \frac{1}{2}\times \text{(Eq. 3}'''): & & & y & & & = & -1 & (2''') \\ \text{(Eq. 3}'')\ /4: & & & & & z & = & 3 & (3''') \end{array}$$

Finally, we can subtract 3 times (Eq. $2'''$) from (Eq. $1'''$) to solve for x:

$$\begin{array}{rrcrcrcrl} \text{(Eq. 1}''') - 3\times \text{(Eq. 2}'''): & x & & & & & = & 2 & (1'''') \\ & & & y & & & = & -1 & (2''') \\ & & & & & z & = & 3 & (3''') \end{array}$$

1.2.4 Gaussian Elimination with Matrices

Now, let's try to apply our previous approach for solving linear equations on the matrix representation of a linear system, to make notation simpler.

Let's look at the system we saw in the previous note as an augmented matrix:

$$\begin{bmatrix} 5x & + & 3y & = & 5 \\ -4x & + & y & = & 2 \end{bmatrix} \qquad \left[\begin{array}{cc|c} 5 & 3 & 5 \\ -4 & 1 & 2 \end{array}\right]$$

In the examples we have seen, there are three basic operations that we can perform to a system of equations, that we know will preserve the solution set of the associated system of linear equations. Let's see how they work when applied to the augmented matrix representation of a system of linear equations:

1. Multiplying a row by a scalar. For example, we can multiply the first row by 2:

$$\begin{bmatrix} 10x & + & 6y & = & 10 \\ -4x & + & y & = & 2 \end{bmatrix} \qquad \left[\begin{array}{cc|c} 10 & 6 & 10 \\ -4 & 1 & 2 \end{array}\right]$$

2. Swapping rows. For example, we swap the 2 rows:

$$\begin{bmatrix} -4x & + & y & = & 2 \\ 5x & + & 3y & = & 5 \end{bmatrix} \qquad \left[\begin{array}{cc|c} -4 & 1 & 2 \\ 5 & 3 & 5 \end{array}\right]$$

3. Adding a scalar multiple of a row to another row. For example, we can modify the second row by adding 2 times the first row to the second:

$$\begin{bmatrix} 5x & + & 3y & = & 5 \\ 6x & + & 7y & = & 12 \end{bmatrix} \qquad \left[\begin{array}{cc|c} 5 & 3 & 5 \\ 6 & 7 & 12 \end{array}\right]$$

Our procedure so far has been to successively eliminate variables using the above steps. A bit more precisely, if we number the variables 1 through n in the order they appear from left to right, to begin Gaussian elimination we eliminate a variable i with the following steps, beginning with $i = 1$ and ending when $i = n$:

1. Swap rows if needed so that an equation containing variable i is contained in row i (in the augmented matrix, this means column i and row i should be nonzero).
2. Divide row i by the coefficient of variable i in this row such that the ith row and column of the augmented matrix is 1.
3. For rows $j = i + 1$ to n, subtract row i times the entry in row j and column i to cancel variable i.

So far, the above steps eliminating variables from left to right (operating on equations from top to bottom) proceeded until we found a "triangular" system of linear equations with an explicit equation at the bottom, which we could then propagate upwards to solve for the remaining variables.

This "triangular form" is known as **row echelon form**. More precisely, a matrix is in row echelon form when the following criteria are met:

- All nonzero rows are above all zero rows.
- The leading coefficient of a non-zero row is always to the right of the leading coefficient of the row above it.

Some textbooks will require a third property to be true for row echelon form:

- The leading coefficient of every non-zero row (which we call the *pivot*, and say is in the *pivot position*) is 1.

In addition to row echelon form, there is also **reduced row echelon form**. This requires that: In addition, after the upwards propagation of variables in step (3), we will obtain a matrix with the following properties, in addition to the two mentioned above:

- The matrix is in row echelon form.
- The leading coefficient of every non-zero row (which we call the *pivot*, and say is in the *pivot position*) is 1.
- Each *column* with an element that is in the pivot position of some row has 0s everywhere else.

When all of the above properties are met, we say a matrix is in **reduced row echelon form**, sometimes abbreviated (especially in programming) as `rref`.

Note that, depending on the source, row echelon form is sometimes defined to also require all the leading coefficients of non-zero rows to be normalized to be 1 - however, this is not a requirement in our definition.

We are introducing the terminology of echelon form just for consistency with textbooks. However, such jargon and definitions are not emphasized in this class; what is most important is that you understand the fundamental principles.

For illustrative purposes, the following represents a matrix in row echelon form

$$\left[\begin{array}{cccc|c} 1 & * & * & * & * \\ 0 & 1 & * & * & * \\ 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{1}$$

where $*$ denotes that any value may be entered in that location.

Step 2 corresponds to the “back-substitution" of variables performed in the previous examples. At the conclusion of Step 2, we are left with a matrix satisfying the following two properties:

- The matrix is in row echelon form.
- Each leading entry of a nonzero row is the only nonzero entry in its column.

Such a matrix is said to be in **reduced row echelon form**, sometimes abbreviated (especially in programming) as `rref`. As another illustration, the following represents a matrix in reduced row echelon form

$$\left[\begin{array}{cccc|c} 1 & 0 & * & 0 & * \\ 0 & 1 & * & 0 & * \\ 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{2}$$

By construction, the Gaussian elimination algorithm always results in a matrix that is in reduced row echelon form. Once an augmented matrix is reduced to reduced row echelon form, variables corresponding to

columns containing leading entries are called **basic variables**, and the remaining variables are called **free variables**. For example, if we consider the augmented matrix

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 0 & 3 \\ 0 & 1 & 4 & 0 & 5 \\ 0 & 0 & 0 & 1 & -8 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{3}$$

then it corresponds to the system of equations

$$\begin{array}{ccccccc} x_1 & & & +2x_3 & & & = 3 \\ & & x_2 & +4x_3 & & & = 5 \\ & & & & & x_4 & = -8. \end{array} \tag{4}$$

In this system of equations, x_1, x_2, x_4 are all basic variables, and x_3 is a free variable. As will be discussed shortly, the distinction between basic and free variables allows us to characterize all solutions to the system of linear equations (if any exist!).

Why does Gaussian elimination always result in a matrix in reduced row echelon form? And will this result *always* allow us to determine a single explicit solution for any system of equations? The next few examples explore what might happen after these steps are applied. On the left hand side, we will show the system of equations, and on the right hand side, we show the corresponding augmented matrix.

1.2.4.1 Gaussian Elimination Examples

Example 1.4 (Equations with exactly one solution):

$$\begin{bmatrix} 2x & + & 4y & + & 2z & = & 8 \\ x & + & y & + & z & = & 6 \\ x & - & y & - & z & = & 4 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 2 & 4 & 2 & 8 \\ 1 & 1 & 1 & 6 \\ 1 & -1 & -1 & 4 \end{array}\right]$$

First, divide row 1 by 2, the scaling factor on x in the first equation.

$$\begin{bmatrix} x & + & 2y & + & z & = & 4 \\ x & + & y & + & z & = & 6 \\ x & - & y & - & z & = & 4 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 1 & 1 & 1 & 6 \\ 1 & -1 & -1 & 4 \end{array}\right]$$

To eliminate x from the two remaining equations, subtract row 1 from row 2 and 3.

$$\begin{bmatrix} x & + & 2y & + & z & = & 4 \\ & - & y & & & = & 2 \\ & - & 3y & - & 2z & = & 0 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & -1 & 0 & 2 \\ 0 & -3 & -2 & 0 \end{array}\right]$$

To ensure y is scaled by 1 in the second equation, multiply row 2 by -1. Then, to eliminate y from the final equation, subtract -3 times row 2 from row 3.

$$\begin{bmatrix} x & + & 2y & + & z & = & 4 \\ & & y & & & = & -2 \\ & & & - & 2z & = & -6 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & -2 & -6 \end{array}\right]$$

To scale z by 1 in the final equation, divide row 3 by -2.

$$\begin{bmatrix} x & + & 2y & + & z & = & 4 \\ & & y & & & = & -2 \\ & & & & z & = & 3 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

Notice that our matrix is now in *row echelon* form, since the leading coefficient of each nonzero row is to the right of the leading coefficient of the row above it. However, it is not yet in *reduced row echelon form*, since the second and third columns, which each contain an element in the pivot position of a row, have a nonzero element in another row.

Continuing, we then subtract row 3 from row 1 to eliminate z from the first equation.

$$\begin{bmatrix} x & + & 2y & & = & 1 \\ & & y & & = & -2 \\ & & & z & = & 3 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

Finally, subtract 2 times row 2 from row 1 to obtain an explicit equation for all variables.

$$\begin{bmatrix} x & & & = & 5 \\ & y & & = & -2 \\ & & z & = & 3 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

Observe that our matrix is now in *reduced row echelon form*, since in addition to still being in row echelon form, each column with an element in pivot position (which, in this case, are all the columns) has only one nonzero element, which equals 1 and is in the pivot position of a row.

This system of equations has a unique solution — x, y, and z can take on only *one* value in order for each equation to be true.

Example 1.5 (Equations with an infinite number of solutions):

$$\begin{bmatrix} x & + & y & + & 2z & = & 2 \\ & & y & + & z & = & 0 \\ 2x & + & y & + & 3z & = & 4 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 3 & 4 \end{array}\right]$$

To eliminate x from the third equation, subtract 2 times row 1 from row 3.

$$\begin{bmatrix} x & + & y & + & 2z & = & 2 \\ & & y & + & z & = & 0 \\ & - & y & - & z & = & 0 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 \end{array}\right]$$

To eliminate y from the third equation, add row 2 to row 3.

$$\begin{bmatrix} x & + & y & + & 2z & = & 2 \\ & & y & + & z & = & 0 \\ & & & & 0 & = & 0 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

At this point, the third equation no longer contains z so we cannot "eliminate" it. We can, however, proceed

by eliminating y from the first equation. To do this, subtract row 2 from row 1.

$$\begin{bmatrix} x & & + & z & = & 2 \\ & y & + & z & = & 0 \\ & & & 0 & = & 0 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

We are now in reduced row echelon form, with basic variables x and y, and z a free variable. For any choice of z, setting $x = 2 - z$ and $y = -z$ yields a solution to this system of equations. This explains the *free variable* terminology; the values of free variables can be chosen arbitrarily, and then the resulting basic variables can be (uniquely) solved for to yield a solution to the system of equations.

This is the best we can do. Notice that the third equation is redundant (it is simply $0 = 0$), so we are left with two equations but three unknown variables. One solution is $x = 1, z = 1, y = -1$. Another possible solution would be $x = 2, z = 0, y = 0$. In fact, this system of equations has an infinite number of solutions — we could choose any value for z, set y to be $-z$ and x to be $2 - z$ and the two equations would still be true.

Despite the lack of a solution, notice that we have still placed this matrix into row echelon form, since the first two nonzero rows are both above the zero row, and the leading coefficient of the second row is to the right of the leading coefficient of the first row. Indeed, it is in fact even in *reduced* row echelon form, since the two elements in pivot position are both 1 and the first two columns, containing these two elements, contain no other elements.

Notice that the third column has two nonzero elements. Doesn't this violate the requirements of reduced row echelon form? No, since the third column does not contain either of the two entries in pivot position, so it is permitted to have more than one nonzero elements.

A key takeaway from this example is that placing a matrix in reduced row echelon form does not imply that it has a unique solution! However, it makes finding the *set* of possible solutions a lot easier, as will be discussed in future notes.

In a later note, we will discuss this situation in more detail.

Example 1.6 (Equations with no solution):

$$\begin{bmatrix} x & + & 4y & + & 2z & = & 2 \\ x & + & 2y & + & 8z & = & 0 \\ x & + & 3y & + & 5z & = & 3 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 1 & 2 & 8 & 0 \\ 1 & 3 & 5 & 3 \end{array}\right]$$

To eliminate x from all but the first equation, subtract row 1 from row 2 and row 3.

$$\begin{bmatrix} x & + & 4y & + & 2z & = & 2 \\ & - & 2y & + & 6z & = & -2 \\ & - & y & + & 3z & = & 1 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 0 & -2 & 6 & -2 \\ 0 & -1 & 3 & 1 \end{array}\right]$$

To make 1 the leading coefficient in row 2, divide row 2 by -2.

$$\begin{bmatrix} x & + & 4y & + & 2z & = & 2 \\ & & y & - & 3z & = & 1 \\ & - & y & + & 3z & = & 1 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 0 & 1 & -3 & 1 \\ 0 & -1 & 3 & 1 \end{array}\right]$$

To eliminate y from the final equation, add row 2 to row 3.

$$\begin{bmatrix} x & + & 4y & + & 2z & = & 2 \\ & & y & - & 3z & = & 1 \\ & & & & 0 & = & 2 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 0 & 1 & -3 & 1 \\ 0 & 0 & 0 & 2 \end{array}\right]$$

Now the third equation gives a contradiction, $0 = 2$. No choice of x, y, and z will change the rules of mathematics such that $0 = 2$, so there is no solution to this system of equations. If these were experimentally measured results, this contradiction might indicate that our modeling assumptions are incorrect or that there is noise in our measurements. In fact, this situation comes up frequently in real experiments, and later in this course we'll investigate techniques for dealing with noisy measurements.

Since we ran into a contradiction, we stopped Gaussian elimination early, so the matrix is not yet in reduced row echelon form. We can see this by inspecting the second column, which contains the element in the pivot position of the second row, but also contains a nonzero element in the first row! Still, we proceeded far enough in Gaussian elimination to put the matrix in row echelon form - can you see why?

1.2.4.2 Algorithm Stopping Point

Based on the previous examples, we have seen that running Gaussian elimination does *not* guarantee that we will be able to find a solution to the system of equations. However, running the algorithm will tell us whether or not there is **one**, **zero**, or **infinitely many** solutions.

If a single solution exists, we will have an explicit equation for each variable. From the augmented matrix perspective, this means that the portion of the matrix corresponding to the coefficient weights will have 1's on the diagonal and 0's everywhere else, as in this example for a system of three equations with three unknowns (the first three columns are the coefficient weights):

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

If we think of Gaussian elimination as a way to rewrite our system of m equations with n variables as a set of explicit equations for each variable, intuitively there must be at least one equation for each variable ($m \geq n$) for a solution to exist. What happens if $m > n$? If the system of equations is consistent, the extra rows of the final augmented matrix should be all zeros — running Gaussian elimination will set the variable coefficients in these rows to zero, so the corresponding result entry should also be zero if a solution exists.

Now we can generalize this strategy to an arbitrary number of equations.

1. For a system of m equations and n variables ($m \geq n$), the first n rows of the augmented matrix have a triangular structure — specifically, the leftmost nonzero entry in row i is a 1 and appears in column i for $i = 1$ to n. If $m > n$, exactly $(m - n)$ rows are all-zero, and all correspond to the equation $0 = 0$. With 4 equations and 3 unknowns, this could be an augmented matrix such as

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

This means that the system of equations has a **unique solution**. Notice that there may still be zero rows after the nonzero rows, but so long as the zero rows have a zero constant entry, they are consistent and so do not pose an issue. We can solve for one unknown by scaling the final row appropriately and eliminating it from every other equation. Repeat this until every equation has one unknown left and the system of equations is solved.

2. There are effectively fewer non-zero rows in the augmented matrix than there are variables, and any rows with all-zero variable coefficients also have a zero result, corresponding to the equation $0 = 0$. This could be an augmented matrix such as

$$\left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \text{ or } \left[\begin{array}{ccc|c} 1 & 1 & 3 & 2 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

Notice that this situation can occur even if there are no zero rows. If it occurs, there are fewer equations than unknowns and the system of linear equations is underdetermined. There are an **infinite number of solutions**.

3. There is a row in the augmented matrix with all-zero variable coefficients but a nonzero result, corresponding to the equation $0 = a$ where $a \neq 0$. This could be an augmented matrix such as

$$\left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 0 & 1 & -3 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{array}\right] \text{ or } \left[\begin{array}{cc|c} 1 & 4 & 2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right] \text{ or } \left[\begin{array}{ccc|c} 1 & 4 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{array}\right]$$

This means that the system of linear equations is inconsistent and there are **no solutions**. Be aware that this scenario can occur *regardless of the shape of the matrix*, and automatically means that no solutions exist, since there exists an inconsistency in the given system of equations.

1.2.4.3 Formal Algorithm

So far, we have walked through in detail how to implement Gaussian elimination by hand. However, this quickly becomes impractical for large systems of linear equations — realistically, this algorithm will be implemented using a software program instead. Given the wide range of programming languages, each with unique syntax conventions, algorithms are often represented instead as **pseudocode**, a detailed English language description of an algorithm. While there is no single format for pseudocode, it should be general enough that it is free of syntactic dependencies but specific enough that it can be translated to actual code nearly automatically if you are familiar with the proper syntax.

Below is one possible pseudocode description of the Gaussian elimination algorithm. Note that we specify the input to the algorithm ("data") and the expected outcome ("result") at the top. We include explicit iterative statements ("for"), which execute the indented steps for each value of the parameters in the description, and conditional ("if," "if/else") statements, which (as the name implies) execute the indented code only if the conditions listed are met. Also note the "gets" symbol ($\longleftarrow$): $a \longleftarrow b$ means that a "gets" the value of b. In many programming languages, this is implemented as an equals sign, but the directed arrow notation makes it completely clear which variable takes on the value of the other variable.

```
Data: Augmented matrix A ∈ ℝ^{m×(n+1)}, for a system of m equations with n variables
Result: Reduced form of augmented matrix
# Forward elimination procedure:
for each variable index i from 1 to n do
    if entry in row i, column i of A is 0 then
        if all entries in column i and row > i of A are 0 then
            proceed to next variable index;
        else
            find j, the smallest row index > i of A for which entry in column i ≠ 0 ;
            # The following rows implement the "swap" operation:
            old_row_j ⟵ row j of A;
            row j of A ⟵ row i of A;
            row i of A ⟵ old_row_j;
        end
    end
    divide row i of A by entry in row i, column i of A;
    for each row index k from i + 1 to m do
        scaled_row_i ⟵ row i of A times entry in row k, column i of A;
        row k of A ⟵ row k of A − scaled_row_i;
    end
end
# Back substitution procedure:
for each variable index u from n − 1 to 1 do
    if entry in row u, column u of A ≠ 0 then
        for each row v from u − 1 to 1 do
            scaled_row_u ⟵ row u of A times entry in row v, column u of A;
            row v of A ⟵ row v of A − scaled_row_u;
        end
    end
end
```

Algorithm 1: The Gaussian elimination algorithm.

1.2.5 Tomography Revisited

How does what we have learned so far relate back to our tomography example, way back at the start of this note? We know that because our grocer's measurements come from a specific box with a particular assortment of milk, juice, and empty bottles, there must be one underlying solution, but insufficient measurements could give us a system of equations with an infinite number of solutions. So, how many measurements do we need?

Initially, we thought about shining a light vertically and horizontally through the box, giving six total equations because there are three rows and three columns per box. However, there are nine bottles to identify, and therefore nine variables, so we will need nine equations. Based on what you have learned about Gaussian elimination, you now understand that we need at least three more measurements — likely taken diagonally — in order to properly identify the bottles. In coming notes, we will discuss in further detail how you can tell whether or not the nine measurements you choose will allow you to find the solution.

1.3 Practice Problems

These practice problems are also available in an interactive form on the course website, along with their solutions.

1. True or False: A system of 3 equations with 2 free variables has solutions along a line.
2. How many solutions does the system of equations have if, after performing Gaussian elimination, the row reduced form of the augmented matrix is $\left[\begin{array}{ccc|c} 3 & -1 & 2 & 1 \\ 0 & 0 & 2 & 1 \end{array}\right]$?
   (a) One solution
   (b) Infinite solutions
   (c) No solutions
3. How many solutions does the system of equations have if, after performing Gaussian elimination, the row reduced form of the augmented matrix is $\left[\begin{array}{ccc|c} 3 & -1 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right]$?
   (a) One solution
   (b) Infinite solutions
   (c) No solutions
4. True or False: A system of equations with more equations than unknowns will always have either infinite solutions or no solutions.
5. Perform Gaussian elimination on the following set of equations to find x, y, and z. Remember to convert it to matrix form!
$$\begin{cases} 10x - 6y + 2z = 2 \\ 3x + 2y = 10 \\ -5x + 3y - z = 1 \end{cases}$$
   (a) $x = 2, y = 7, z = 4$
   (b) $x = 0, y = 3, z = 16$
   (c) Infinite solutions
   (d) No solutions
6. Solve this system of equations: $\begin{cases} 2x + y + 3z = 1 \\ x - y + 4z = 2 \\ x + 8y + z = 1 \end{cases}$
7. Solve this system of equations: $\begin{cases} 2x - 16y + 4z = -8 \\ x + 12y + 4z = 6 \\ x + 8y - 2z = 4 \end{cases}$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 2A

This note covers the following topics:

- **Solutions to Systems of Linear Equations:** How do we know when a system of linear equations has a solution? If it does have a solution, how do we solve it?
- **Vectors:** The notion of a vector is introduced, examples are given, and various mathematical operations on vectors are discussed.
- **Matrices:** The notion of a matrix is introduced, examples are given, and various mathematical operations on matrices are discussed.
- **Matrix-vector Form for Systems of Linear Equations:** We'll see that systems of linear equations can be expressed algebraically using matrices and vectors. This is the starting point for harnessing the full power of linear algebra to solve and understand systems of linear equations.

2.1 Solutions to Systems of Linear Equations

In the previous note, we introduced systems of linear equations and discussed their representation in terms of augmented matrices. We introduced Gaussian elimination, which is a systematic method for solving systems of equations. In terms of the augmented matrix representation, Gaussian elimination proceeds by performing a sequence of elementary row operations until the augmented matrix is in reduced row echelon form. Once this point is reached, it was generally possible to determine whether the given system of equations had a solution or not by inspection of the augmented matrix. The aim of the present section is to make this last step more precise.

To start, we make two important definitions. We say that a system of linear equations possessing one or more solutions is **consistent**. Conversely, if the system of linear equations does not have a solution, we say that it is **inconsistent**. This terminology makes sense when we consider real-life examples of systems of linear equations. In particular, for the tomography example, inconsistency would imply that the set of measurements could not have occurred given our model (hence, the measurements would be "inconsistent" with the model). On the other hand, if the system of equations was consistent, it would imply that the measurements we obtained were "consistent" with the model proposed, and the solution to the system of equations would correspond to a possible configuration of beverages in the crate. While it might seem that inconsistency would not be encountered in real life, it often is. This can be a consequence of our modeling assumptions being imperfect, or it could also be the result of "noise" in our measurements. We will discuss how to deal with inconsistent systems of equations later in this course.

2.1.1 Determining Consistency of Systems of Linear Equations

We state the following useful fact as a Theorem, to highlight it as an important result to keep in mind. In particular, it provides a simple way to determine consistency of a system of linear equations by inspecting the corresponding augmented matrix representation in reduced row echelon form.

Theorem 2.1: A system of linear equations is inconsistent if and only if the corresponding augmented matrix in reduced row echelon form contains the row $[0,0,\ldots,0|1]$.

Why should this be true? Well, the interpretation of such a row is the linear equation $0 = 1$, which is absurd! Since we can never make this true for any assignment of values to the unknowns, the system of equations under consideration does not have a solution (it is inconsistent). In fact, we can make the following similar observation: If the augmented matrix corresponding to a system of linear equations contains a row of the form $[0,0,\ldots,0|b]$ for $b \neq 0$, then the system of linear equations must be inconsistent. This observation can allow us to terminate the Gaussian elimination algorithm early, since we can immediately determine inconsistency without fully reducing the augmented matrix to reduced row echelon form.

On the other hand, if the row $[0,0,\ldots,0|1]$ does not appear in the augmented matrix in reduced row echelon form, then the leading entry of every non-zero row corresponds to a variable (more specifically, a basic variable by definition). By definition of reduced row echelon form, each leading entry is the only non-zero entry in its column. These two observations can be concisely summarized in terms of variables and linear equations: The equations in the linear system (corresponding to the augmented matrix in reduced row echelon form) are in one-to-one correspondence with the basic variables. Hence, the set of solutions to the system of linear equations can be obtained as follows: set the free variables to whatever you like, and solve for the unique values of the basic variables. In particular, if a consistent system of linear equations has at least one free variable, then it has infinitely many solutions. On the other hand, if a consistent system of linear equations has no free variables, then it has a unique solution.

Let's take a few examples to illustrate the ideas discussed above. In each example, we consider a system of equations that has already been transformed to reduced row echelon form (e.g., by running Gaussian elimination). In each case, we determine consistency and solutions, provided they exist.

Example 2.1 (Inconsistent System of Linear Equations):

$$\begin{bmatrix} x & + & & & 2z & = & 2 \\ & & y & - & 3z & = & 1 \\ & & & & 0 & = & 1 \end{bmatrix} \rightarrow \quad \left[\begin{array}{ccc|c} 1 & 0 & 2 & 2 \\ 0 & 1 & -3 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right].$$

The augmented matrix contains the row $[0,0,0|1]$, and therefore the system of equations is inconsistent (i.e., it does not have any solution).

Example 2.2 (Consistent System of Linear Equations with Unique Solution):

$$\begin{bmatrix} x & & & = & 2 \\ & y & & = & 3 \\ & & z & = & 1 \\ & & & & \end{bmatrix} \rightarrow \quad \left[\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

The system is consistent since the augmented matrix does not contain the row $[0,0,0|1]$. There are no free variables, and the system has the unique solution $x = -2$, $y = 3$ and $z = 1$.

Example 2.3 (Consistent System of Linear Equations with Infinitely Many Solutions):

$$\begin{bmatrix} v & + & 2w & & - & 2y & & = & 3 \\ & & & x & + & 3y & & = & -1 \\ & & & & & & z & = & 8 \end{bmatrix} \rightarrow \quad \left[\begin{array}{ccccc|c} 1 & 2 & 0 & -2 & 0 & 3 \\ 0 & 0 & 1 & 3 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 8 \end{array}\right].$$

The system is consistent since the augmented matrix does not contain the row $[0,0,0,0,0|1]$. The variables v,x,z are basic variables, and w,y are free variables. For any choice of $w,y \in \mathbb{R}$, the following choices of v,x,z yield a solution:

$$\begin{aligned} v &= 3-2w+2y \\ x &= -1-3y \\ z &= 8. \end{aligned}$$

Important Note: To determine consistency, it is a good idea to first transform the augmented matrix to reduced row echelon form (e.g., by Gaussian elimination). Indeed, if the augmented matrix is *not* in reduced row echelon form and does *not* contain the row $[0,\ldots,0|1]$, we can *not* guarantee that the system of equations is consistent.

2.2 Vectors and Matrices

In the previous note, we mentioned that linear algebra is the study of linear functions and linear equations, usually in terms of vectors and matrices. We now formally introduce the concepts of vectors and matrices, and discuss how to perform associated computations such as addition and multiplication. As a consequence, we will find that matrices and vectors can be used to algebraically express systems of linear equations. In future notes, we will consider additional properties of vectors and matrices and see how these can help us understand and analyze real-world systems.

2.3 Introduction to Vectors

What is a vector? It is an ordered list of numbers. More specifically, if we are given an ordered collection of n real numbers such as $x_1, x_2, \cdots, x_n$, we can represent this collection as a single point in an n-dimensional real space, $\mathbb{R}^n$, denoted as a **vector** $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Each x_i (for i between 1 and n) is called a **component**, or **element**, of the vector. The elements of a vector do not have to be real — they could be complex-valued, meaning $\vec{x} \in \mathbb{C}^n$ — but the majority of vectors you will encounter in 16A will be real. The **dimension** (or **size**) of a vector is the number of components it contains. Two vectors $\vec{x}$ and $\vec{y}$ are said to be **equal**, denoted $\vec{x} = \vec{y}$, if they have the same dimension and $x_i = y_i$ for all i.

Remark. Depending on the text, vectors may be written simply as x, or they might be written as letters in boldface $\mathbf{x}$, or they might be denoted with a small arrow on top $\vec{x}$. In these lecture notes we will use the latter arrow notation, and the dimension of the vector will be given (for example, $\vec{x} \in \mathbb{R}^3$ means that $\vec{x}$ contains three real numbers).

Example 2.4 (Vector of size two):

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

In the above example, $\vec{x}$ is a vector with two components. Because the components are both real numbers, $\vec{x} \in \mathbb{R}^2$. We can represent the vector graphically on a 2-D plane, using the first element, x_1, to denote the horizontal position of the vector and the second element, x_2, to denote its vertical position:

Additional Resources For more on vectors, read pages 1-6 of *Strang* and try Problem Set 1.1 in *Strang*. *Extra: Try reading the portions on linear combinations which generate a "space."*

Read more on vectors in *Schuam's* on pages 1-3 and try Problems 1.1 to 1.6.

Example 2.5 (3D Vector): Assume we have $x_1 = 2$, $x_2 = 3$, and $x_3 = 2.5$, which can be represented as the vector $\vec{x} = \begin{bmatrix} 2 \\ 3 \\ 2.5 \end{bmatrix}$. This represents a point in 3-D space ($\mathbb{R}^3$), which can be drawn as an arrow from the origin to this point in space:

As we shall see shortly, vectors can be used to help write systems of equations compactly. However, they are also useful for representing a multitude of things — anything that can be represented as an ordered list of numbers can be expressed as a vector. For instance, in the tomography example, we can write a vector to represent the amount of light absorbed by each bottle in a row or column.

2.3.1 More Examples of Vectors

Example 2.6 (Position vs. time): $\vec{x} \in \mathbb{R}^n$ can represent samples of a quantity at n time points. Imagine a car moving along a line. Its position at time $t_1, t_2, \cdots, t_n$ can be represented with a vector:

$$\vec{x} = \begin{bmatrix} x_{t_1} \\ x_{t_2} \\ \vdots \\ x_{t_n} \end{bmatrix}$$

Here, x_{t_i} represents the position of the car at time t_i.

Example 2.7 (Quadrotor state): Vectors can be used to represent the *state* of a system, which is the minimum information you need to completely characterize a system at a given point in time, without any need for more information about the past of the system. State is a powerful concept because it lets us separate the past from the future. The state completely captures the present — and the past can only affect the future through the present. We will revisit the concept of state in EECS 16B, and you will see this in your homework.

As an example, consider modeling the dynamics of a quadrotor. A quadrotor is a type of helicopter that has four rotors. The state of a quadrotor at a particular time can be summarized by its: 3D position (x, y, z), angular position (roll, pitch, yaw), velocity ($\dot{x}$, $\dot{y}$, $\dot{z}$) and angular velocity ($\dot{roll}$, $\dot{pitch}$, $\dot{yaw}$), which can be represented as a vector $\vec{q} \in \mathbb{R}^{12}$, as illustrated in Figure 1.

$$\vec{q} = \begin{bmatrix} x \\ y \\ z \\ \text{roll} \\ \text{pitch} \\ \text{yaw} \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\text{roll}} \\ \dot{\text{pitch}} \\ \dot{\text{yaw}} \end{bmatrix}$$

Figure 1: A quadrotor drone (left) and the vector containing the information needed to determine its state (right). The 3D position of the quadrotor is captured in its x, y, and z coordinates, while its angular position is measured as its roll, pitch, and yaw. A dot above a variable implies taking a derivative with respect to time, giving the velocity of each component (for instance, $\dot{x} = \frac{dx}{dt}$ is the quadrotor's velocity in the x direction).

Example 2.8 (Electrical circuit quantities): Later in this course, we will learn about *current* and *voltage*, which are physical quantities present in electrical circuits. The current and voltage vary at different points in the electrical circuit, and we can represent these quantities as elements in a vector where the size of the vector will depend on the complexity of the circuit. We'll get into this more in Module 2!

Example 2.9 (Vectors representing functions): Consider a simple game in which points are earned or lost by rolling a 6-sided die. If you roll a 5 or lower, you earn the number of points that you roll, but if you roll a 6, you earn -6 points. We could define a function that maps the die roll outcome, d, to the number of points earned by that roll, $f(d)$:

$$f(d) = \begin{cases} d & \text{if } d \leq 5 \\ -d & \text{if } d = 6 \end{cases}$$

We could also capture this information in a vector that summarizes all possible values of $f(d)$:

$$\begin{bmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ -6 \end{bmatrix}$$

Because the input domain of the function $f(d)$ is restricted to the integers 1 through 6, it is easy to see how it can be represented as a vector. It turns out that we can build a vector representation of any function with a numerical output — even functions that operate on an input domain of infinite size, such as all real numbers or all integers. Such functions can be represented by an *infinite-dimensional* vector. But for the most part, "vectors" in 16A will be finite-dimensional.

Example 2.10 (Color): This is relevant to the lab we will do! The vector

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \vec{x} \in \mathbb{R}^3$$

can represent color with its components giving red, green, and blue (RGB) intensity values. Typically, the entries of $\vec{x}$ will fall in a range that reflects the range of the color (often 0 to 255)[1]. However, the human perception of color is quite subtle and involves psychophysics as well as ideas that are related to sampling that we will talk about in EECS 16B.

2.3.2 Special Vectors

Definition 2.1 (Zero Vector): A **zero vector** is a vector with all the components equal to zero, usually just represented as $\vec{0}$. You can usually tell the size of the zero vector $\vec{0}$ from the context: if $\vec{x} \in \mathbb{R}^n$ is added to $\vec{0}$, then $\vec{0}$ must also be in $\mathbb{R}^n$.

Definition 2.2 (Standard Unit Vector): A **standard unit vector** is a vector with all components equal to 0 except for one element, which is equal to 1. A standard unit vector where the ith position is equal to 1 is written as $\vec{e}_i$. We can denote the 3 standard unit vectors in $\mathbb{R}^3$ as:

$$\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \vec{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

When talking about standard unit vectors in the context of states, we might also use the word "pure" to refer to such states. This is because they only have one non-zero component in them. Other states are mixtures of pure states.

2.4 Vector Operations

2.4.1 Vector Addition

Two vectors of the same size and in the same space (e.g. complex numbers, real numbers, etc.) can be added together by adding their corresponding components. For example, we can add two vectors in $\mathbb{R}^3$:

$$\begin{bmatrix} -1 \\ 3.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} -1+2 \\ 3.5-1 \\ 0+3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2.5 \\ 3 \end{bmatrix}$$

Vector addition can be performed graphically as well. In $\mathbb{R}^n$, you place the tail of the first vector at the origin, and then place the tail of the second vector at the first vector's head. The vector from the origin to the head of the second vector is the resulting vector sum. The following illustration shows how this can be done for two vectors, $\vec{x}$ and $\vec{y}$, in $\mathbb{R}^2$:

[1] Why 0 to 255? We will cover this more in 16B, but generally speaking, the 256 integers in this range can be represented easily as an 8-bit binary number because $256 = 2^8$.

Figure 2: Adding two vectors in $\mathbb{R}^2$. We can see that $\vec{x}+\vec{y}=\vec{y}+\vec{x}$, so vector addition is commutative.

Properties of Vector Addition: Many of the properties of addition you are already familiar with when adding individual numbers hold for vector addition as well. For three vectors $\vec{x},\vec{y},\vec{z}\in\mathbb{R}^n$ (and $\vec{0}\in\mathbb{R}^n$), the following properties hold:

$$\begin{aligned}
\textbf{Commutativity} \text{ (shown in Figure 2)}: &\quad \vec{x}+\vec{y}=\vec{y}+\vec{x}\\
\textbf{Associativity}: &\quad (\vec{x}+\vec{y})+\vec{z}=\vec{x}+(\vec{y}+\vec{z})\\
\textbf{Additive identity}: &\quad \vec{x}+\vec{0}=\vec{x}\\
\textbf{Additive inverse}: &\quad \vec{x}+(-\vec{x})=\vec{0}
\end{aligned}$$

2.4.2 Scalar Multiplication

We can multiply a vector by a number, called a **scalar**.[2]

Definition 2.3 (Scalar): A **scalar** is a number. In mathematics and physics, scalars can be used to describe magnitude or used to scale things (e.g. cut every element of a vector in half by multiplying by 0.5, or flip the signs of each element in a vector by multiplying by -1).

To perform scalar multiplication, just multiply each of the components of the vector by the scalar:

$$-3\times\begin{bmatrix}-1\\3.5\\0\end{bmatrix}=\begin{bmatrix}3\\-10.5\\0\end{bmatrix}$$

In general, for a scalar α and vector $\vec{x}$, this looks like

$$\alpha\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}=\begin{bmatrix}\alpha x_1\\\alpha x_2\\\vdots\\\alpha x_n\end{bmatrix}$$

Example 2.11 (Negative Vector): To obtain a negative vector $-\vec{x}$, where $\vec{x}\in\mathbb{R}^n$, we can just multiply the vector $\vec{x}$ by the scalar -1. In other words, $-\vec{x}=-1\times\vec{x}$.

Example 2.12 (Zero Vector): We can obtain the zero vector by multiplying any vector by 0: $0\vec{x}=\vec{0}$

[2]Note that this scalar must be in the same space as the vector (e.g. also in the reals, $\mathbb{R}$, or in the complex numbers, $\mathbb{C}$). Since we'll only be working over the reals in this class, you don't have to worry about this detail.

$$\vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, 2\vec{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, -2\vec{x} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

Properties of Scalar Multiplication: Just as multiplying a scalar by a scalar is associative ($(xy)z = x(yz)$), distributive ($x(y+z) = xy + xz$), and the multiplicative identity holds (multiplying any number by 1 returns that original number), multiplying a scalar by a vector has the same properties:

$$\begin{aligned} \textbf{Associativity}: &\quad (\alpha\beta)\vec{x} = \alpha(\beta\vec{x}) \\ \textbf{Distributivity}: &\quad (\alpha+\beta)\vec{x} = \alpha\vec{x} + \beta\vec{x} \\ \textbf{Multiplicative identity}: &\quad 1\vec{x} = \vec{x} \end{aligned}$$

2.4.3 Vector Transpose

The transpose of a vector (or a matrix) is a crucial piece of notation for vector and matrix calculations. We write the transpose of a vector $\vec{x}$ as $\vec{x}^T$. If $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, then $\vec{x}^T = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$. The transpose of a **column vector** (a vertical vector) is called a **row vector** (a horizontal vector).The transpose of a row vector is a column vector. Consequently, the transpose of the transpose of a vector recovers the original vector.

It is important to recognize that, though the transpose of a vector contains the same elements as the original vector, it is still a different vector! That is to say, for any vector $\vec{x}$ (with at least two components), $\vec{x}^T \neq \vec{x}$!

2.4.4 Vector-Vector Multiplication

While the operations we have discussed thus far may be largely consistent with your intuition from working with scalars, vector-vector multiplication is not quite as straightforward. By convention, a row vector can only be multiplied by a column vector (and vice versa). Moreover, vector-vector multiplication is not commutative, so $\vec{y}^T\vec{x} \neq \vec{x}\vec{y}^T$. $\vec{x}$ and $\vec{y}$ are both column vectors. In fact, while $\vec{y}^T\vec{x}$ is only defined when both vectors have the same number of elements, $\vec{x}\vec{y}^T$ is defined for vectors of any size.

Multiplying a column vector on the right by a row vector on the left is called computing the **inner product** (also called the **dot product**) of two vectors,[3] which we will discuss further in later notes. An alternative notation for the inner product of two vectors is $\langle \vec{x}, \vec{y} \rangle$ - we will see a lot more of this notation in Module 3 of the course. Below is an illustration of $\vec{y}^T\vec{x}$:

[3] These terms are not identical mathematically, but will give the same result for the real vectors we will be working with in 16A.

$$\vec{y}^T\vec{x} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = y_1x_1 + y_2x_2 + \cdots + y_nx_n$$

Fundamentally, $\vec{y}^T\vec{x}$ is a sum of products — each element in $\vec{y}$ is multiplied by the corresponding element in $\vec{x}$, and all of these products are added together, resulting in a scalar. This form of vector-vector multiplication thus maps a pair of vectors in $\mathbb{R}^n$ to a single scalar in $\mathbb{R}$. Notice that the inner product as we have defined it is *commutative*, since $\vec{y}^T\vec{x} = \vec{x}^T\vec{y}$. As it turns out, this ceases to be true when we start working with complex numbers, as will be seen in EECS 16B.

Multiplying a row vector on the right by a column vector on the left, on the other hand, generates a **matrix**, which we will define formally in the next section. In particular, the product of $\vec{x} \in \mathbb{R}^n$ and $\vec{y}^T \in \mathbb{R}^m$ is given by:

$$\vec{x}\,\vec{y}^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_m \\ x_2y_1 & x_2y_2 & \cdots & x_2y_m \\ \vdots & \vdots & \vdots & \vdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_m \end{bmatrix}$$

The entry in row i and column j of $\vec{x}\,\vec{y}^T$ is the product of x_i and y_j. To distinguish it from the inner product, this type of vector-vector multiplication is often called the **outer product**. This is perhaps best understood as a particular form of matrix-matrix multiplication, which we will discuss more in the next section.

2.5 Introduction to Matrices

Definition 2.4 (Matrix): A **matrix** is a rectangular array of numbers, written as:

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} \tag{1}$$

Each A_{ij} (where i is the row index and j is the column index) is a **component**, or **element** of the matrix A.

We will generally consider the case of *real* matrices (meaning that the elements are real numbers), in which case we write $A \in \mathbb{R}^{m\times n}$ to denote that A is an $m \times n$ matrix (i.e., it has m rows and n columns).

Remark. Matrices are often represented by capital letters (e.g. A), sometimes in boldface (e.g. $\mathbf{A}$). In these notes, the notations are used interchangeably. Subscripts are typically used to specify an element of a matrix, with the first subscript corresponding the row index and the second corresponding to the column index.

A matrix is said to be **square** if $m = n$ (that is, if the number of rows and number of columns are equal). Just as we could compute the **transpose** of a vector by transforming rows into columns, we can compute the

transpose of a matrix, A^T, where the rows of A^T are the (transposed) columns of A:

$$A^T = \begin{bmatrix} A_{11} & \cdots & A_{m1} \\ \vdots & \ddots & \vdots \\ A_{1n} & \cdots & A_{nm} \end{bmatrix}$$

Mathematically, A^T is the $n \times m$ matrix given by $(A^T)_{ij} = A_{ji}$. A square matrix is said to be **symmetric** if $A = A^T$, which means that $A_{ij} = A_{ji}$ for all i and j.

2.5.1 Examples of Matrices

There are many ways matrices can be useful for representing objects; we will provide a few examples below, beginning with an array of numbers.

Example 2.13 (4×3 **Matrix**):

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 8 & 12 \end{bmatrix}$$

In order to describe the dimensions of the above matrix, we would say $A \in \mathbb{R}^{4\times 3}$, or that A has $m = 4$ rows and $n = 3$ columns, or that A is a 4×3 matrix.

Example 2.14 (Image): A grayscale image of $m \times n$ pixels may be regarded as a matrix with m rows and n columns, with entries corresponding to the grayscale levels at the pixel location. This matrix in $\mathbb{R}^{m\times n}$ could also be represented by a vector of length mn (and therefore, a vector in $\mathbb{R}^{mn}$) by placing all pixel readings in a single list, as illustrated explained in the figure below. The choice of representation is somewhat arbitrary, as long as we agree on what each representation signifies. This brings up an interesting philosophical point:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \Rightarrow \begin{bmatrix} x_{11} \\ \vdots \\ x_{1n} \\ x_{21} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{m1} \\ \vdots \\ x_{mn} \end{bmatrix}$$

Why not represent an image as a matrix? As we will see in the rest of this note, we can perform any linear operation on a vector using matrix-vector multiplication. So even though an image looks like a matrix, it is mathematically easier to represent it as a vector.

What about a color image? The image could be stored in a vector in $\mathbb{R}^{3mn}$, because each of the mn pixels has values for red, green, and blue.

Example 2.15 (Illumination pattern): The imaging lab, in which individual pixels are illuminated by a projector, is very similar to the tomography example, in which individual bottles are illuminated depending on the angle of light being applied to the box. The pixel illumination pattern can similarly be represented as a square matrix of 1's and 0's, with a row for each photodiode measurement and a column for each pixel in the image. If a particular pixel is illuminated during a particular measurement, the matrix contains a 1

at the appropriate row and column index; otherwise, it contains a 0. In the tomography example, we set up a system of linear equations and built a matrix of coefficient weights corresponding to which bottle was included within a particular measurement.

2.5.2 Special Matrices

Definition 2.5 (Zero Matrix): Similar to a zero vector, a zero matrix is a matrix with all components equal to zero. It is usually just represented as 0, where its size is implied from context. Here is an example of a 2×2 zero matrix:

$$0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Definition 2.6 (Identity Matrix): An identity matrix, often written as I, is a square matrix whose diagonal elements are 1 and whose off-diagonal elements are all 0. Here is an example of the 3×3 identity matrix:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that the n column vectors (and the transpose of the row vectors) that make up an $n \times n$ identity matrix are the unit vectors in $\mathbb{R}^n$. The identity matrix is useful because multiplying it with a vector $\vec{x}$ will leave the vector unchanged - in other words, $I\vec{x} = \vec{x}$. We will prove this fact later in this note.

In fact, we will see that multiplying a square matrix by an identity matrix of the same size will yield the same initial matrix: $AI = IA = A$ (we will discuss matrix-matrix multiplication further in the next note).

2.5.3 Column vs Row Perspective

The interpretation of rows and columns from the system-of-linear-equations perspective of doing experiments. This highlights that both the rows and columns of a matrix have importance. Each row of the matrix represents a particular experiment that took one particular measurement. For a given row, the coefficient entries represent how much the corresponding state variable affects the outcome of this *particular* experiment. In contrast, the columns represent the influence of a particular state variable on the *collection* of experiments taken together. These perspectives come in handy in interpreting matrix multiplication.

2.6 Matrix Operations

2.6.1 Matrix Addition

Two matrices of the same size can be added together by adding their corresponding components. For instance, we can add two matrices A and B (both in $\mathbb{R}^{m\times n}$) as follows:

$$A+B=\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}+\begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \vdots & \vdots & & \vdots \\ B_{m1} & B_{m2} & \dots & B_{mn} \end{bmatrix}=\begin{bmatrix} A_{11}+B_{11} & A_{12}+B_{12} & \dots & A_{1n}+B_{1n} \\ A_{21}+B_{21} & A_{22}+B_{22} & \dots & A_{2n}+B_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1}+B_{m1} & A_{m2}+B_{m2} & \dots & A_{mn}+B_{mn} \end{bmatrix}$$

Here is a specific example of adding two 3×2 matrices:

$$\begin{bmatrix} -1 & 3 \\ 3.5 & 2 \\ 0 & -0.1 \end{bmatrix}+\begin{bmatrix} 2 & -1 \\ -1 & -2 \\ 3 & 0.1 \end{bmatrix}=\begin{bmatrix} 1 & 2 \\ 2.5 & 0 \\ 3 & 0 \end{bmatrix}$$

Note that if two matrices are not the same size, we cannot add them. Intuitively, this is because there is no obvious way to pair up each of their entries.

Properties of Matrix Addition: Just as many of the properties of scalar addition hold for vector addition, many of the same properties hold for matrix addition as well. For three matrices A, B, and C of the same size (and a zero matrix of the same size), the following properties are true:

$$\begin{aligned} \textbf{Commutativity}: \quad A+B &= B+A \\ \textbf{Associativity}: \quad (A+B)+C &= A+(B+C) \\ \textbf{Additive identity}: \quad A+0 &= A \\ \textbf{Additive inverse}: \quad A+(-A) &= 0 \end{aligned}$$

2.6.2 Scalar-Matrix Multiplication

As with scalar-vector multiplication, multiplying a matrix by a scalar requires multiplying each component of the matrix by that scalar. For instance, to multiply a scalar α by a matrix $A\in\mathbb{R}^{m\times n}$, we do the following:

$$\alpha A=\alpha\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}=\begin{bmatrix} \alpha A_{11} & \alpha A_{12} & \dots & \alpha A_{1n} \\ \alpha A_{21} & \alpha A_{22} & \dots & \alpha A_{2n} \\ \vdots & \vdots & & \vdots \\ \alpha A_{m1} & \alpha A_{m2} & \dots & \alpha A_{mn} \end{bmatrix}$$

Here is a specific example for a 3×2 matrix:

$$(-3)\begin{bmatrix} -1 & 3 \\ 3.5 & 2 \\ 0 & -0.1 \end{bmatrix}=\begin{bmatrix} 3 & -9 \\ -10.5 & -6 \\ 0 & 0.3 \end{bmatrix}$$

Example 2.16 (Negative Matrix): To obtain a negative matrix $-A$, we can just multiply the original matrix A by the scalar -1. In other words, $-A = -1 \times A$.

Example 2.17 (Zero Matrix): We can obtain the zero matrix by multiplying any matrix by 0: $0 \times A = 0$

Properties of Scalar-Matrix Multiplication: Scalar-matrix multiplication has many of the same properties as scalar-scalar multiplication and scalar-vector multiplication. Specifically, for scalars α, β and matrices $A, B \in \mathbb{R}^{m \times n}$, the following properties hold:

$$\begin{aligned}
\textbf{Associativity}: \quad (\alpha\beta) \times A &= (\alpha) \times (\beta A) \\
\textbf{Distributivity}: \quad (\alpha + \beta) \times A &= \alpha A + \beta A \text{ and } \alpha(A+B) = \alpha A + \alpha B \\
\textbf{Multiplicative identity}: \quad 1 \times A &= A
\end{aligned}$$

2.6.3 Matrix-Vector Multiplication

A matrix $A \in \mathbb{R}^{m \times n}$ and a vector $\vec{x} \in \mathbb{R}^n$ can be multiplied together to create a new vector of length m as follows:

$$A\vec{x} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix}$$

Each entry of the resulting vector is the inner product of the corresponding row of A with $\vec{x}$. We perform the multiplication by doing inner product calculations on each row, one at a time:

Mathematically speaking, if we say $A\vec{x} = \vec{b}$, with $A \in \mathbb{R}^{m \times n}$, $\vec{x} \in \mathbb{R}^n$, and $\vec{b} \in \mathbb{R}^m$, element i of $\vec{b}$ can be

calculated as follows:

$$b_i = \sum_{j=1}^{n} A_{ij} x_j$$

We can also see that the matrix-vector product can be rewritten as a sum of vectors, which can be expressed as a linear combination of the columns of A:

$$\begin{aligned} A\vec{x} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix} &= \begin{bmatrix} A_{11}x_1 \\ A_{21}x_1 \\ \vdots \\ A_{m1}x_1 \end{bmatrix} + \begin{bmatrix} A_{12}x_2 \\ A_{22}x_2 \\ \vdots \\ A_{m2}x_2 \end{bmatrix} + \cdots + \begin{bmatrix} A_{1n}x_n \\ A_{2n}x_n \\ \vdots \\ A_{mn}x_n \end{bmatrix} \\ &= x_1 \begin{bmatrix} A_{11} \\ A_{21} \\ \vdots \\ A_{m1} \end{bmatrix} + x_2 \begin{bmatrix} A_{12} \\ A_{22} \\ \vdots \\ A_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} A_{1n} \\ A_{2n} \\ \vdots \\ A_{mn} \end{bmatrix} \end{aligned}$$

How can this be interpreted? Consider the following example:

$$\begin{bmatrix} -1 & 3 \\ 3.5 & 2 \\ 0 & -0.1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ -0.1 \end{bmatrix}$$

Multiplying a matrix by a standard unit vector effectively selects a single column of the matrix — in this case, the second column. Matrix-vector multiplication is a way to compute a linear combination of the columns in the matrix, where the weights are given by the vector that is being multiplied.

A particular example of matrix-vector multiplication occurs when the matrix we're using is the identity matrix. Earlier, we mentioned that $I\vec{x} = \vec{x}$ for all vectors $\vec{x}$. Now, let's try to prove this. Assume that $\vec{x}$ is an n-dimensional column vector, so we can write

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T.$$

Now, by the above definition of matrix-vector multiplication, we see that

$$I\vec{x} = \begin{bmatrix} \sum_{j=1}^{n} I_{1j} x_j \\ \sum_{j=1}^{n} I_{2j} x_j \\ \cdots \\ \sum_{j=1}^{n} I_{nj} x_j \end{bmatrix}$$

However, recall that the identity matrix has 0s everywhere except along its main diagonal, where it has 1s. In other words, we can express its elements as

$$\begin{aligned} I_{ij} &= 0 \quad \text{if } i \neq j \\ I_{ij} &= 1 \quad \text{if } i = j. \end{aligned}$$

Thus, we can write a single term of $I\vec{x}$ as

$$\begin{aligned}(I\vec{x})_i &= \sum_{j=1}^{n} I_{ij}x_j \\ &= I_{i1}x_1 + I_{i2}x_2 + \ldots + I_{in}x_n \\ &= 0\cdot x_1 + 0\cdot x_2 + \ldots + 0\cdot x_{i-1} + 1\cdot x_i + 0\cdot x_{i+1} + \ldots + 0\cdot x_n \\ &= \vec{x}_i.\end{aligned}$$

If $(I\vec{x})_i = x_i$ for all i, then

$$I\vec{x} = \vec{x},$$

since if all the components of two vectors are equal, then the two vectors are themselves equal. So we have now proven that multiplying the identity matrix with a vector leaves the vector unchanged, as we expected.

2.7 Matrix-Vector Form for Systems of Linear Equations

Now, let us finally return to systems of linear equations. Consider a system of m linear equations with n unknowns $x_1, \ldots, x_n$. As we've seen, such a system of equations can be written in the general form

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

where the a_{ij} and b_i are all real-valued constants.

We previously introduced the augmented matrix notation

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{array}\right],$$

which was useful as a bookkeeping device for systematically determining solutions (e.g., using Gaussian elimination).

Now, using matrix-vector multiplication, observe that we can also write this system in **matrix-vector form**:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

If we denote the coefficient matrix by A, the variable vector by $\vec{x}$ and the vector of constant terms by $\vec{b}$ then

the system of equations can be concisely written as

$$A\vec{x} = \vec{b}.$$

Unsurprisingly, this is called the **matrix-vector form** (or **matrix-vector representation**) of the system of linear equations Eq. (Eq. 2).

Now that we understand how to multiply matrices with vectors, we can revisit this representation of systems of equations, to demonstrate that it is algebraically equivalent to the original system.

To do so, we can evaluate the matrix-vector multiplication on the left-hand-side, to obtain

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Notice that the left-hand-sides of our original system of equations have suddenly appeared as components of the vector on the left-hand-side! Since equating two vectors is the same as equating their corresponding coefficients, we can rewrite the above vector equation as a set of n scalar equations:

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

which are exactly the equations in our linear system.

Thus, we have seen that matrix-vector form is not merely an alternate *notation* for a linear system of equations, but in fact is *algebraically equivalent* to the linear system, since we can use the definition of matrix-vector multiplication to go from one to another.

Additional Resources For more on matrix-vector multiplication, read *Strang* page 59. For additional practice with these ideas, try Problem Set 2.3 in *Strang*.

2.8 Practice Problems

1. Is $(\vec{x}+\vec{y})\mathbf{A}$ equivalent to $\mathbf{A}(\vec{x}+\vec{y})$ where $\vec{x}$ and $\vec{y}$ are column vectors in $\mathbb{R}^n$ and $\mathbf{A}$ is an $n \times n$ matrix?
2. True or False: If $\mathbf{A}^2 = \mathbf{0}$, where $\mathbf{0}$ is the zero matrix, then $\mathbf{A} = \mathbf{0}$.
3. Let $\mathbf{A}$ be a 4×4 transformation matrix. You know that $\mathbf{A}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$. If possible, find the second column of $\mathbf{A}$.

4. Multiply $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ with $\begin{bmatrix} 7 \\ 8 \end{bmatrix}$ using both the element-wise and linear combination interpretations of matrix-vector multiplication, and verify that you get the same answer for this particular example.

5. The effects on the current state, $\vec{x}$, of a fighter jet performing a Barrel Roll, an Immelman, or a Nose-Dive can be described by the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. The jet's final state after a Nose-Dive, followed by a Barrel Roll, followed by an Immelman, can be described by the expression:

   (a) $\mathbf{CAB}\vec{x}$

   (b) $\mathbf{BAC}\vec{x}$

   (c) $\mathbf{CBA}\vec{x}$

   (d) $\mathbf{ABC}\vec{x}$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 2B

2.1 Matrix-Matrix Multiplication (Transformation of Spaces)

Matrix-matrix multiplication is another powerful tool for modeling linear systems, which we will use extensively. As an example, two matrices A and B in $\mathbb{R}^{2\times 2}$ can be multiplied as follows:

$$\underset{\text{A}}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{\text{B}}{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}} = \underset{\text{AB}}{\begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}}$$

Computationally, matrix-matrix multiplication involves multiplying each row vector in A with each column vector in B, starting from the top row of matrix A and leftmost column of matrix B. Effectively, the left matrix is multiplied by each column vector in the second matrix to produce a new column of AB. Why columns and not rows? That's just convention. But this does lead to an important point about the dimensions of matrix-matrix multiplication.

To left-multiply a matrix B by another matrix A, the number of columns in A must equal the number of rows in B. Otherwise, the product $A \times B$ cannot be calculated. Moreover, if A is an $m \times n$ matrix and B is $n \times p$, the product $A \times B$ will have dimensions $m \times p$. A visual illustration of this can be seen here, where the left matrix is broken up into m row vectors and the right matrix is represented as p column vectors:

$$\begin{bmatrix} ---- \vec{r}_1^T ---- \\ ---- \vec{r}_2^T ---- \\ \vdots \\ ---- \vec{r}_m^T ---- \end{bmatrix} \begin{bmatrix} | & | & & | \\ | & | & & | \\ | & | & & | \\ \vec{c}_1 & \vec{c}_2 & \cdots & \vec{c}_p \\ | & | & & | \\ | & | & & | \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \vec{r}_1^T\vec{c}_1 & \vec{r}_1^T\vec{c}_2 & \cdots & \vec{r}_1^T\vec{c}_p \\ \vec{r}_2^T\vec{c}_1 & \vec{r}_2^T\vec{c}_2 & \cdots & \vec{r}_2^T\vec{c}_p \\ \vdots & \vdots & & \vdots \\ \vec{r}_m^T\vec{c}_1 & \vec{r}_m^T\vec{c}_2 & \cdots & \vec{r}_m^T\vec{c}_p \end{bmatrix}$$

In order for the inner product $\vec{r}_i^T\vec{c}_j$ to be defined, each row vector ($\vec{r}_i^T$) must have the same number of entries as each column vector ($\vec{c}_j$). As a result, matrix-matrix multiplication is typically not commutative — $A \times B$ does not necessarily equal $B \times A$. In fact, it's not guaranteed that both quantities can even be calculated unless the number of rows in A equals the number of columns in B **and** the number of rows in B equals the number of columns in A.

To illustrate this, consider the following example of taking the product of two 2×2 matrices.

Example 2.1 (Matrix Multiplication is Not Commutative):

Matrix multiplication does not commute - that is to say, there exist matrices A and B such that $AB \neq BA$.

Here is an example to verify that assertion:

$$\begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} (2)(1)+(4)(3) & (2)(2)+(4)(4) \\ (3)(1)+(1)(3) & (3)(2)+(1)(4) \end{bmatrix} = \begin{bmatrix} 14 & 20 \\ 6 & 10 \end{bmatrix}$$

Now we multiply them in the other order, and see if we get the same answer:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} (1)(2)+(2)(3) & (1)(4)+(2)(1) \\ (3)(2)+(4)(3) & (3)(4)+(4)(1) \end{bmatrix} = \begin{bmatrix} 8 & 6 \\ 18 & 16 \end{bmatrix}$$

As expected, we did not end up with the same result as we did before. Having produced a counterexample, we have therefore proven that matrix multiplication is not generally commutative.

Be aware, however, that there still might (and indeed do!) exist pairs of matrices whose product *is* commutative. All we have shown here is that *not all* pairs of matrices produce the same product when multiplied in the opposite order.

Example 2.2 (Matrix Multiplication is Associative): Having seen above that matrix multiplication is not commutative, we might start asking questions about associativity, as well. In particular, is it true that given three matrices A, B, and C, that $(AB)C = A(BC)$? Put differently, does the *grouping* of matrices in a product matter, if the order is kept the same?

As it turns out, this is true. Unfortunately, a general proof of associativity is tedious so we will only consider an example.

$$\begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix} \begin{bmatrix} 11 & 12 \\ 13 & 14 \end{bmatrix}.$$

Let's first multiply the first two matrices together, before multiplying their product with the third:

$$\begin{aligned} \left(\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix} \right) \begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix} &= \begin{bmatrix} (1)(3)+(0)(1) & (1)(2)+(0)(4) \\ (2)(3)+(3)(1) & (2)(2)+(3)(4) \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 2 \\ 9 & 16 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} (3)(2)+(2)(0) & (3)(4)+(2)(1) \\ (9)(2)+(16)(0) & (9)(4)+(16)14) \end{bmatrix} \\ &= \begin{bmatrix} 6 & 14 \\ 18 & 52 \end{bmatrix}. \end{aligned}$$

Then, let's try multiplying the last two matrices together first, before multiplying the first matrix with that

product:

$$\begin{aligned}
\begin{bmatrix}1 & 0\\ 2 & 3\end{bmatrix}\left(\begin{bmatrix}3 & 2\\ 1 & 4\end{bmatrix}\begin{bmatrix}2 & 4\\ 0 & 1\end{bmatrix}\right) &= \begin{bmatrix}1 & 0\\ 2 & 3\end{bmatrix}\begin{bmatrix}(3)(2)+(2)(0) & (3)(4)+(2)(1)\\ (1)(2)+(4)(0) & (1)(4)+(14)(1)\end{bmatrix}\\
&= \begin{bmatrix}1 & 0\\ 2 & 3\end{bmatrix}\begin{bmatrix}6 & 14\\ 2 & 8\end{bmatrix}\\
&= \begin{bmatrix}(1)(16)+(0)(2) & (1)(14)+(0)(8)\\ (2)(16)+(3)(2) & (2)(14)+(3)(8)\end{bmatrix}\\
&= \begin{bmatrix}6 & 14\\ 18 & 52\end{bmatrix},
\end{aligned}$$

which is the same as what we got before!

The fact that three fairly arbitrary matrices exhibit associativity when being multiplied should be a strong hint that matrix multiplication is probably associative - however, it is important to understand that *this is not a proof* of the associativity of matrix multiplication. To prove that matrix multiplication is associative, we'd have to show that *any* triplet of matrices can be multiplied in either order without changing the final answer - showing that it works for particular examples is not sufficient.

Example 2.3 (Matrices as Functions): In a single-variable situation, we might have a function f that takes in a number x and outputs a number $f(x)$. If we want functions of multiple variables, we can use vectors. The input $\vec{x}$ is now a list of variables. The output is another list of numbers. If f is **linear**, then it acts on a list of variables by multiplying them by scalars and adding them together. In this case, we can represent f as a matrix. Therefore, matrices can also be called **linear maps** or **linear transformations**.

In algebra, we learned how to manipulate functions of one variable. Linear algebra teaches us how to manipulate linear functions of multiple variables. In a later note, we will further explore how matrix-matrix multiplication applies to linear transformations.

Additional Resources For more on matrix-matrix multiplication, read *Strang* pages 61-62, and try Problem Set 2.3.

In *Schuam's*, read pages 30-33 and try Problems 2.4 to 2.11, 2.39 to 2.40, 2.42, 2.44 - 2.49, 2.12 to 2.16, 2.41, 2.43, and 2.72. *Extra: Understand Polynomials in Matrices.*

2.2 Linear Transformations

2.2.1 A Natural Generalization

In an earlier note, we looked at linear functions over the reals - specifically, we defined a scalar function $f(x)$ to be linear if, for any scalar k,

$$f(kx) = k \cdot f(x).$$

We will now work to generalize this definition to functions acting on vectors. The most natural generalization would simply be to replace x with $\vec{x}$ in the above definition - in other words, we might define a function

f to be linear if and only

$$f(k\vec{x}) = kf(\vec{x}).$$

This property is called homogeneity, but isn't quite sufficient. Why? Well, in the scalar ("one-dimensional") case, knowing $f(x_0)$ for a single nonzero scalar x_0 was sufficient to define $f(x)$ over all the reals, since we could write

$$x = (x/x_0)x_0 \implies f(x) = (x/x_0)f(x_0).$$

Can we do something similar now, working over vectors? Imagine working in two-dimensional space, where the span of two vectors $\vec{x}_0$ and $\vec{x}_1$ is $\mathbb{R}^2$. By definition, we know any vector $\vec{x} \in \mathbb{R}^2$ can be expressed as a linear combination

$$\vec{x} = \alpha\vec{x}_0 + \beta\vec{x}_1$$

of the two vectors whose span we are considering. Thus, a natural analog of our result over scalars would be to say that

$$f(\vec{x}) = f(\alpha\vec{x}_0 + \beta\vec{x}_1) = \alpha f(\vec{x}_0) + \beta f(\vec{x}_1).$$

More generally, given the output of a linear function for a given set of vectors, we'd like to be able to evaluate the function at any point in the span of the given set of vectors. Unfortunately, our proposed definition of linearity doesn't let us do this. Why? Consider the following function $f : \mathbb{R}^2 \to \mathbb{R}$:

$$f(\vec{x}) = f\left(\begin{bmatrix} x_0 \\ x_1 \end{bmatrix}\right) = \left\{ \begin{array}{ll} 2x_0, & \text{for } x_0 = x_1 \\ x_1, & \text{for } x_0 = -x_1 \\ 0, & \text{otherwise} \end{array} \right\}.$$

The graph of this function consists of two lines intersecting at the origin: one for $f(\vec{x}) = 2\vec{x}_0$ when the first condition is satisfied and one for $f(\vec{x}) = \vec{x}_1$. The function is equal to 0 at all other points. Some inspection of $f(\vec{x})$ will show that $f(k\vec{x}) = kf(\vec{x})$ for all $\vec{x}$. But observe that while

$$\begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

$$f\left(\begin{bmatrix} 2 \\ 0 \end{bmatrix}\right) = 0 \neq 2 - 1 = f\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) + f\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right),$$

so our desired generalization doesn't hold. The two vectors, $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ satisfy the first and second condition respectively, meaning they lie in different regions of the function. Therefore, when the function is evaluated at $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$, it does not match the sum of the function evaluated at the two vectors individually. Clearly, if we want our generalized result to hold, we need to strengthen our definition of linearity.

2.2.2 Additivity

One way to do so is to introduce one further requirement, known as **additivity** - specifically, that

$$f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$$

for all $\vec{x}$ and $\vec{y}$. Observe now that, by applying additivity as well as our previous requirement (**homogeneity**), we can directly show that

$$\begin{aligned} f(\alpha\vec{x}_0 + \beta\vec{x}_1) &= f(\alpha\vec{x}_0) + f(\beta\vec{x}_1) \\ &= \alpha f(\vec{x}_0) + \beta f(\vec{x}_1), \end{aligned}$$

as desired! As it turns out, these two requirements are all that are needed to generalize linear functions to act over vectors, where they are known as **linear transformations**. One interesting thing to note is that additivity also holds for scalar linear functions, and can be derived from homogeneity - it's only when working with vectors that additivity starts to give us something new.[1]

2.2.3 Matrices as Linear Transformations

So far, we've established the requirements that a linear transformation must satisfy. But what *is* a linear transformation, really? As it turns out, multiplying a matrix with a column vector is a linear transformation - specifically, the function

$$f_A(\vec{x}) = A\vec{x}$$

is a linear transformation for any matrix A. The matrix acts like a function performed on the vector to transform it to a new vector. Typically, we simplify this statement by stating that the matrix A *itself* is a linear transformation, with the matrix used to represent the transformation f_A.

But why is this true? To check if a function is a linear transformation, we simply need to verify that it satisfies the requirements of homogeneity and additivity. Observe that, by the rules of matrix-vector multiplication,

$$\begin{aligned} f_A(\vec{x}+\vec{y}) = A(\vec{x}+\vec{y}) = A\vec{x}+A\vec{y} = f_A(\vec{x}) + f_A(\vec{y}) \\ f_A(k\vec{x}) = A(k\vec{x}) = k(A\vec{x}) = kf_A(k\vec{x}), \end{aligned}$$

where $\vec{x}$ and $\vec{y}$ are arbitrary vectors, A is a matrix with the appropriate dimensions, and k is an arbitrary real scalar, so both additivity and homogeneity are satisfied by matrix multiplication. Thus, matrix multiplication is a linear transformation, as we claimed earlier.

One final piece of jargon remains to be introduced - when a linear transformation yields vectors of the same dimension as its input (i.e. if $f(\vec{x})$ has the same dimension as $\vec{x}$) then it is sometimes called a **linear operator**.

2.3 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Multiply $\begin{bmatrix} 1 & 5 & 0 \\ 10 & 3 & 7 \\ 6 & 4 & 11 \end{bmatrix}$ with $\begin{bmatrix} 2 & 12 & 3 \\ 1 & 8 & 0 \\ 9 & 1 & 2 \end{bmatrix}$. What is the first row of the resulting matrix?

[1] A good question to ask at this point would be: does additivity imply homogeneity? As it turns out, the answer is no - try to produce a function that satisfies additivity but not homogeneity! There are many examples when working over the field of complex numbers, but it is *much* harder to do so when working over the reals, like we do here.

(a) $\begin{bmatrix} 16 & 52 & 5 \end{bmatrix}$
(b) $\begin{bmatrix} 3 & 40 & 7 \end{bmatrix}$
(c) $\begin{bmatrix} 7 & 52 & 3 \end{bmatrix}$
(d) $\begin{bmatrix} 14 & 13 & 2 \end{bmatrix}$

2. Matrix Multiplication

Consider the following matrices:

$$\mathbf{A_1} = \begin{bmatrix} 1 & 4 \end{bmatrix} \quad \mathbf{B_1} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 9 & 5 & 7 \\ 4 & 3 & 2 & 2 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 5 & 5 & 8 \\ 6 & 1 & 2 \\ 4 & 1 & 7 \\ 3 & 2 & 2 \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 5 & 3 & 4 \\ 1 & 8 & 2 \\ 2 & 3 & 5 \end{bmatrix}$$

For each matrix multiplication problem, if the product exists, find the product by hand. Otherwise, explain why the product does not exist.

(a) $\mathbf{A_1B_1}$
(b) $\mathbf{AB}$
(c) $\mathbf{BA}$
(d) $\mathbf{AC}$
(e) $\mathbf{DC}$
(f) $\mathbf{CD}$ (Write down the dimensions of the product if it exists. For practice, you can compute the product on your own)
(g) $\mathbf{EF}$ (Practice on your own)
(h) $\mathbf{FE}$ (Practice on your own)

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 3

3.1 Linear Dependence

Recall the tomography example from Note 1, in which we tried to determine the composition of a box of bottles by shining light at different angles and measuring light absorption. The Gaussian elimination algorithm implied that we needed to take at least 9 measurements to properly identify the 9 bottles in a box so that we had at least one equation per variable. However, will taking *any* 9 measurements guarantee that we can find a solution? Answering this question requires an understanding of linear dependence. In this note, we will define linear dependence (and independence), and take a look at what it implies for systems of linear equations.

3.1.1 What is Linear Dependence?

Linear dependence is used to characterize the "redundancy" of information in real world applications. If two vectors are linearly dependent, then they do not both provide new information. We will give two equivalent definitions of linear dependence.

Definition 3.1 (Linear Dependence (I)): A set of vectors $\{\vec{v}_1, \ldots, \vec{v}_n\}$ is linearly dependent if there exist scalars $\alpha_1, \ldots, \alpha_n$ such that $\alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n = \vec{0}$ and not all α_i's are equal to zero.

Definition 3.2 (Linear Dependence (II)): A set of vectors $\{\vec{v}_1, \ldots, \vec{v}_n\}$ is linearly dependent if there exist scalars $\alpha_1, \ldots, \alpha_n$ and an index i such that $\vec{v}_i = \sum_{j \neq i} \alpha_j \vec{v}_j$.[1] In words, a set of vectors is linearly dependent if one of the vectors could be written as a linear combination of the rest of the vectors.

Why did we introduce two equivalent definitions? They could be useful in different settings. For example, it is often easier mathematically to show linear dependence with definition (I). Can you see why? If we would like to prove linear dependence with definition (II), we need to first choose a vector $\vec{v}_i$ and show that it is a linear combination of the other vectors. However, with definition (I), we don't need to try to "single out" a vector to get started with the proof. We can blindly write down the equation $\alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n = \vec{0}$ and begin our proof from there. On the other hand, definition (II) gives us a more intuitive way to talk about redundancy. If a vector can be constructed from the rest of the vectors, then this vector does not contribute any information that is not already captured by the other vectors.

Now we will show that the two definitions are equivalent. This is the first formal proof in the course! We will walk you through it.[2] First, we ask the question, "What does it mean when we say two definitions are

[1] In case you are unfamiliar with this notation, the $\sum$ symbol is simply shorthand for addition. For instance, $\alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n$ can be written as $\sum_{i=1}^{n} \alpha_i\vec{v}_i$ or $\sum_i \alpha_i\vec{v}_i$, which is a sum over all possible i values. In this instance, $\sum_{j \neq i} \alpha_j\vec{v}_j$ is the sum over all $\alpha_j\vec{v}_j$ excluding the $\alpha_i\vec{v}_i$ term, which can also be calculated as $\alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n - \alpha_i\vec{v}_i$.

[2] Note 4 provides a more in-depth treatment on how to approach proofs.

equivalent?" It means that when the condition in definition (I) holds, the condition in definition (II) must hold as well. And when the condition in definition (II) holds, the condition in definition (I) must also hold. So there are two directions that we have to show:

(i) To see how definition (II) implies definition (I), we start from the condition in definition (II) — suppose there exist scalars $\alpha_1, \ldots, \alpha_n$ and an index i such that $\vec{v}_i = \sum_{j \neq i} \alpha_j \vec{v}_j$. We want to somehow transform this equation into the form that appears in definition (I). How can we achieve that? We can move $\vec{v}_i$ to the right:

$$\vec{0} = -1 \times \vec{v}_i + \sum_{j \neq i} \alpha_j \vec{v}_j. \tag{1}$$

Now setting $\alpha_i = -1$, we have

$$\vec{0} = \alpha_i \times \vec{v}_i + \sum_{j \neq i} \alpha_j \vec{v}_j = \sum_j \alpha_j \vec{v}_j. \tag{2}$$

Since $\alpha_i = -1$, at least one of the α_j terms is not zero, and the condition in definition (I) is satisfied.

(ii) Now let's show the reverse — that definition (I) implies definition (II). Suppose the condition in definition (I) is true. Then there exist scalars $\alpha_1, \ldots, \alpha_n$ such that $\alpha_1 \vec{v}_1 + \cdots + \alpha_n \vec{v}_n = 0$ and not all α_i's are equal to zero. Since at least one of the α_i's is nonzero, let's assume that α_1 is nonzero since we can always reorder terms in the summation because addition is commutative. Now how do we get the equation into the form identical to that in definition (II)? Observe that if we move $\alpha_1 \vec{v}_1$ to the opposite side of equation and divide both sides by α_1, we have

$$\vec{v}_1 = \sum_{j \neq 1} \left(\frac{\alpha_j}{\alpha_1} \right) \vec{v}_j. \tag{3}$$

We see that this is identical to the second definition. In our proof, we made the assumption that $\alpha_1 \neq 0$. However, notice that we could as well have supposed that $\alpha_2 \neq 0$, $\alpha_3 \neq 0$, or any index i so that $\alpha_i \neq 0$. The convention is to set the first index, in this case 1, to be nonzero. In mathematical texts, we typically write *"Without loss of generality (W.L.O.G.), we let $\alpha_1 \neq 0$."*

Now that we have introduced the notion of linear dependence, what does it mean to be linearly *independent*?

3.1.2 Linear Independence

If a set of vectors is linearly independent, then they are not redundant - each vector provides some new information.

Definition 3.3 (Linear Independence): A set of vectors is linearly independent if it is not linearly dependent. More specifically, from the first definition of linear dependence we can deduce that a set of vectors $\{\vec{v}_1, \ldots, \vec{v}_n\}$ is linearly independent if $\alpha_1 \vec{v}_1 + \cdots + \alpha_n \vec{v}_n = \vec{0}$ implies $\alpha_1 = \cdots = \alpha_n = 0$.

Let's see some simple examples of linear dependence and linear independence.

Example 3.1 (Linear dependence of 2 vectors): Consider vectors $\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\vec{b} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$. These vectors

are linearly dependent because we can write $\vec{b}$ as a scaled version of $\vec{a}$:

$$\vec{b} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 2 \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2 \times \vec{a}.$$

Example 3.2 (Linear independence of 2 vectors): Consider vectors $\vec{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\vec{b} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$. We will show that the two vectors are linearly independent. Consider scalars α_1, α_2 such that $\alpha_1\vec{a} + \alpha_2\vec{b} = \vec{0}$. We can write this vector equation as a system of linear equations:

$$\begin{aligned}
&\alpha_1\vec{a} + \alpha_2\vec{b} = \vec{0} \\
\Rightarrow &\alpha_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\Rightarrow &\begin{bmatrix} 2\alpha_1 \\ \alpha_1 \end{bmatrix} + \begin{bmatrix} \alpha_2 \\ 5\alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\Rightarrow &\begin{bmatrix} 2\alpha_1 + \alpha_2 \\ \alpha_1 + 5\alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\Rightarrow &\begin{cases} 2\alpha_1 + \alpha_2 = 0 \\ \alpha_1 + 5\alpha_2 = 0 \end{cases}
\end{aligned}$$

Solving this system of linear equations with Gaussian elimination yields a unique solution, $\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Therefore, by definition, $\vec{a}$ and $\vec{b}$ are linearly independent.

Additional Resources For more on the definition of linear dependence, read *Strang* pages 164-167 or read *Schuam's* pages 121-124. For additional practice with these ideas, try *Schuam's* Problems 4.17 to 4.22, and 4.89 to 4.96.

3.1.3 Linear Dependence and Systems of Linear Equations

Previously, we saw that a system of linear equations can have zero solutions, a unique solution, or infinitely many solutions. Is there a way to tell what kind of solution a system of linear equations has without running Gaussian elimination or explicitly solving for the solution? Yes! Recall that a system of linear equations can be written in matrix-vector form as $A\vec{x} = \vec{b}$, where A is a matrix of variable coefficients, $\vec{x}$ is a vector of variables, and $\vec{b}$ is a vector of values that these weighted sums must equal. We will show that just looking at the columns or rows of the matrix A can help tell us about the solutions to $A\vec{x} = \vec{b}$.

Theorem 3.1: **If the system of linear equations $A\vec{x} = \vec{b}$ has an infinite number of solutions, then the columns of A (which can be thought of as vectors) are linearly dependent.**

Let's see why this is the case: If the system has infinite number of solutions, it must have at least two distinct solutions. Let's call them $\vec{x}_1$ and $\vec{x}_2$. (Note that $\vec{x}_1, \vec{x}_2$ are full vectors, not vector elements.) Then $\vec{x}_1$

and $\vec{x_2}$ must satisfy

$$A\vec{x_1} = \vec{b} \tag{4}$$

$$A\vec{x_2} = \vec{b}. \tag{5}$$

Subtracting the first equation from the second equation, we have $A(\vec{x_2} - \vec{x_1}) = \vec{0}$. Let $\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \vec{x_2} - \vec{x_1}$.

Because $\vec{x_1}$ and $\vec{x_2}$ are distinct, not all α_i's are equal to zero. Let the columns of A be $\vec{a_1}, \ldots, \vec{a_n}$. Then, $A\vec{\alpha} = \sum_{i=1}^{n} \alpha_i \vec{a_i} = \vec{0}$. By definition, the columns of A are linearly dependent.

Note that in this proof, we used the property of matrix multiplication that $A\vec{\alpha} = \sum_{i=1}^{n} \alpha_i \vec{a_i}$. We scale each column and add them together. In other words, matrix-vector multiplication is a linear combination of columns. This property is often a useful way to think about matrix multiplication. The following example might help:

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_1 a_1 + \alpha_2 b_1 + \alpha_3 c_1 \\ \alpha_1 a_2 + \alpha_2 b_2 + \alpha_3 c_2 \\ \alpha_1 a_3 + \alpha_2 b_3 + \alpha_3 c_3 \end{bmatrix} = \alpha_1 \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} + \alpha_2 \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \alpha_3 \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Theorem 3.2: **If the columns of A in the system of linear equations $A\vec{x} = b$ are linearly dependent, then the system does not have a unique solution.**

Let's walk through this proof step by step: we'll start by assuming we have a matrix A with linearly dependent columns, and then we will show that this means that the system does not have a unique solution.

Since we are interested in the columns of A, let's start by explicitly defining the columns of A as vectors $\vec{a}_i$:

$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \ldots & \vec{a}_n \\ | & | & & | \end{bmatrix},$$

We've defined A to have linearly dependent columns, so by the definition of linear dependence, there exist scalars $\alpha_1, \ldots, \alpha_n$ such that $\alpha_1 \vec{a}_1 + \ldots + \alpha_n \vec{a}_n = \vec{0}$ where not all of the α_i's are zero. We can put these α_i's in a vector

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

and by the definition of matrix-vector multiplication, we can compactly write the expression above:

$$A\vec{\alpha} = \vec{0}$$

where $\vec{\alpha} \neq \vec{0}$.

Recall that we are trying to show that the system of equations $A\vec{x} = \vec{b}$ does not have a unique solution. We know that systems of equations can have either zero, one, or infinite solutions. If our system of equations has zero solutions, then it cannot have a unique solution, so we don't need to consider this case. Now let's

consider the case where we have at least one solution, $\vec{x}$:

$$\begin{aligned} A\vec{x} &= \vec{b} \\ A\vec{x} + \vec{0} &= \vec{b} \text{ and substitute in } \vec{0} = A\vec{\alpha} \text{ from above giving:} \\ A\vec{x} + A\vec{\alpha} &= \vec{b} \\ A(\vec{x} + \vec{\alpha}) &= b \end{aligned}$$

Therefore, $\vec{x} + \vec{\alpha}$ is also a solution to the system of equations! Since both $\vec{x}$ and $\vec{x} + \vec{\alpha}$ are solutions, and $\vec{\alpha} \neq \vec{0}$, the system has more than one solution. We've now proven the theorem.

Note that we can add any multiple of $\vec{\alpha}$ to $\vec{x}$ and it will still be a solution – therefore, if there is at least one solution to the system and the columns of A are linearly dependent, then there are infinite solutions. Let's think about why this makes sense intuitively. In an experiment, each column in matrix A represents the influence of each variable x_i on the measurements. If the columns are linearly dependent, this means that some of the variables influence the measurement in the same way, and therefore cannot be disambiguated. The next example illustrates this idea.

Example 3.3 (S): uppose we have a black and white image with two pixels. We cannot directly see the shade of each pixel, but we can measure how much light the two pixels absorb in total. Can we figure out the shade of each pixel? Let's model this as a system of linear equations. Suppose pixel 1 absorbs x_1 units of light and pixel 2 absorbs x_2 units of light. Our measurement indicates that total amount of light absorbed by the image are 10 units of light. Then we could write down the equation,

$$x_1 + x_2 = 10. \tag{6}$$

Written in matrix form, we have

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10 \end{bmatrix}. \tag{7}$$

We see that the columns are $\begin{bmatrix} 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \end{bmatrix}$. The total amount of light absorbed is influenced by 1 unit of x_1 and 1 unit of x_2. However, we cannot pin down the exact influence by x_1 and x_2 because if pixel 1 absorbs c units less, we can just have pixel 2 absorb c units more. This is connected with the fact that the two columns are linearly dependent — if one pixel absorbs less, it is possible to find a way such that the other pixel absorbs more to make up for the loss (the column of that pixel can be written as a linear combination of the columns of the other pixels).

This result has important implications to the design of engineering experiments. Often times, we can't directly measure the values of the variables we're interested in. However, we can measure the total weighted contribution of each variable. The hope is that we can fully recover each variable by taking several of such measurements. Now we can ask: "What is the minimum number of measurements we need to fully recover the solution?" and "How do we design our experiment so that we can fully recover our solution with the minimum number of measurements?" Consider the tomography example. We are confident that we can figure out the configuration of the stack when the columns of the lighting pattern matrix A in $A\vec{x} = \vec{b}$ are linearly independent. On the other hand, if the columns of the lighting pattern matrix are linearly dependent, we know that we don't yet have enough information to figure out the configuration. Checking whether the columns are linearly independent gives us a way to validate whether we've effectively designed our experiment.

Note that we have two perspectives: in the matrix, each row represents a measurement, while each column corresponds to a variable. Therefore, if the columns are linearly dependent, then we have at least one redundant variable. From the perspective of rows, linear dependency tells us that we have one or more redundant measurements. We will revisit this idea in a later note when we talk about rank.

3.2 Span

Let's introduce **span**, a closely related concept to linear dependence that will be used throughout this course.

Definition 3.4 (Span): The span of a set of vectors $\{v_1, \ldots, v_n\}$ is the set of *all* linear combinations of $\{v_1, \ldots, v_n\}$. We can write this mathematically as

$$span(v_1, \ldots, v_n) = \left\{ \sum_{i=1}^{n} \alpha_i \vec{v}_i \mid \alpha_i \in \mathbb{R} \right\}$$

This represents all of the possible points in n-dimensional space that we can reach by taking linear combinations of the vectors in our set. For example, the span of two non-zero, non-parallel vectors is a 2D plane.

We can now rephrase our second definition of linear dependence: A set of vectors is linearly dependent if any one of the vectors is in the *span* of the remaining vectors.

We should also be aware of the following annoying bit of jargon: that when given a matrix A, the *span*, *range*, and *column space* of A all refer to the span of the columns of A!

Additional Resources For more on linear span, read *Schaum's* pages 119-121. For additional practice with these ideas, try Problems 4.13 to 4.16, 4.66, 4.69, and 4.83 to 4.88.

Example 3.4 (Span Practice): Let's see how to solve problems involving the span of a set of vectors. Consider the three vectors:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ -2 \end{bmatrix}.$$

How can we compute and express their span?

First, let's try to gain some intuition for these vectors, by plotting them on a set of axes:

Intuitively, it seems like linear combinations of these three vectors can reach any point on the plane. Let's see if we can justify this rigorously. Consider an arbitrary point $\begin{bmatrix} a \\ b \end{bmatrix}$ on the plane. We'd like to see if we can write this point as a linear combination of our three vectors - in other words, we'd like to show that no matter what a and b we pick, we can choose scalars c_1, c_2, and c_3 such that

$$c_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + c_2 \begin{bmatrix} -3 \\ 1 \end{bmatrix} + c_3 \begin{bmatrix} -2 \\ -2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}.$$

Applying the rules of vector algebra that we presented in the previous note to simplify the summation on the left-hand-side, this equation is equivalent to

$$\begin{bmatrix} (1)c_1 + (-3)c_2 + (-2)c_3 \\ (2)c_1 + (1)c_2 + (-2)c_3 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Observe that the above equation is essentially a system of linear equations with three unknowns - c_1, c_2, and c_3. Writing it in the standard "$Ax = b$" form, we obtain

$$\begin{bmatrix} 1 & -3 & -2 \\ 2 & 1 & -2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}.$$

Recall our original goal - to show that no matter how a and b are chosen, we can solve for constants c_1, c_2, and c_3 such that the above equation is satisfied. But now that we have written it in the above form, we know how to solve it - Gaussian elimination! Remember that since we are trying to solve for the c_i as functions of a and b, a and b should not be treated as unknowns for the purpose of Gaussian elimination, but rather as arbitrarily chosen constants.

Observe that the pivot of the first row is already 1, so we can immediately eliminate the coefficient of c_1 in the second row by subtracting twice the first row from the second, to obtain

$$\begin{bmatrix} 1 & -3 & -2 \\ 0 & 7 & 2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} a \\ b - 2a \end{bmatrix}.$$

Now, we should scale the second row by a factor of $1/7$ in order to get a 1 in the coefficient for c_2 in the second row, so we obtain

$$\begin{bmatrix} 1 & -3 & -2 \\ 0 & 1 & 2/7 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} a \\ (1/7)b - (2/7)a \end{bmatrix}.$$

Observe that we have now placed our matrix of coefficients in row echelon form, so we first determine whether a solution exists. We see here that there are two nonzero rows, no zero rows equating to a nonzero constant, and three unknowns. Thus, we have a consistent system with fewer equations than unknowns, so there are an *infinite number* of solutions, no matter how we choose a and b.

In other words, no matter how a and b are chosen, there is not just one, but an *infinite number* of ways to compute a linear combination of our three vectors to reach $\begin{bmatrix} a \\ b \end{bmatrix}$. Thus, we have shown that our three vectors span the entirety of two-dimensional space. Expressed mathematically, we have:

$$\text{span}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ -2 \end{bmatrix}\right) = \mathbb{R}^2.$$

3.3 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Are the vectors $\vec{a} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ and $\vec{b} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ linearly independent?

2. Are the vectors $\vec{a} = \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}, \vec{b} = \begin{bmatrix} 1 \\ 6 \\ 2 \end{bmatrix}, \vec{c} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ linearly independent?

3. Suppose for some matrix $\mathbf{A}$, $\mathbf{A}\vec{x}_1 = \vec{b}$ and $\mathbf{A}\vec{x}_2 = \vec{b}$, where $\vec{x}_1 \neq \vec{x}_2$. Are the columns of $\mathbf{A}$ linearly independent?

4. Is $\vec{v} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$ in the *span* $\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix}\right)$?

5. Which of the following are equivalent to $\text{span}\{\vec{v}_1, \vec{v}_2\}$?

   (a) $\text{span}\{\vec{v}_1, \vec{v}_1 + \vec{v}_2\}$
   (b) $\text{span}\{\vec{v}_1\}$
   (c) $\text{span}\{\vec{v}_1, \vec{v}_2 - \alpha\vec{v}_1\}$
   (d) (a) and (b)
   (e) (a) and (c)

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 4

Mathematical Thinking & Derivation

For many students EECS 16A might be the first time you are asked to "prove" an idea or a concept. This note tries to explain the main ideas behind proofs and give you some tips on how to approach them.

A mathematical proof provides a means for guaranteeing that a statement is true. So what is a proof? A proof is a finite sequence of steps, called logical deductions, which establishes the truth of a desired statement. In particular, the power of a proof lies in the fact that using limited (finite) means, we can guarantee the truth of a statement that may have infinitely many cases.

More specifically, a proof is typically structured as follows. Recall that there are certain statements, called axioms or postulates, that we accept without proof (we have to start somewhere). Starting from these axioms, a proof consists of a sequence of logical deductions: Simple steps that apply the rules of logic. This results in a sequence of statements where each successive statement is necessarily true if the previous statements were true. This property is enforced by the rules of logic: Each statement follows from the previous statements. These rules of logic are a formal distillation of laws that were thought to underlie human thinking.

In this note, we are going to guide you through the process of developing proofs with a few examples. In particular, we aim to demonstrate the thought process of turning the problem statement into mathematical form and deriving successive mathematically rigorous statements that lead to the desired result.

When we encounter a proof problem, we generally try to understand the problem by asking the following questions:

- "What are the things we can assume based on the problem statement?"
- "What is it that we would like to show?"

The answer to the first question gives us the conditions that we are working under and the answer to the second question gives us a clear picture of our goal. Then, we ask the question:

- "How can we utilize what we know to get to what we would like to show under the specified conditions?"

To write a proof, the following steps are useful in guiding your thought process and helping you understand the fundamental ideas of the problem.

1. **Carefully read the statement.** Check to see what the direction of the implication is, are you being asked to assume P is true and prove Q is true ($P \implies Q$)? Or are you being asked to assume Q is true and prove P is true?

2. **Write out what you know from the statement of the theorem.** Say you are asked to prove something like "If P is true, then Q is true." Here, "P is true" is what is given to you/what is known. If the theorem statement has this written out in word, write it out in mathematical notation. Be explicit. Try to simplify any complex notation or jargon.

3. **Write out what you want to prove.** In the example above, what you want to prove is "Q" is true. Again, this might be given to you in words, make sure you translate it into equations. Simplify. Make sure you write this on the paper; even though it seems trivial, it can often help. If there are different ways of writing this – write out both. For example, if you want to prove a set of vectors is linearly dependent, you have two definitions of linear dependance you can work with. Write out both.

4. **Observe the two statements you just wrote down for what is known and what you want to prove. Find similarities.** Take note of these. It might help to also write them down. How might one of the expressions you wrote be made to look like the other?

5. **Try out a simple example.** Now that you have written out what is given and what you want to prove, try thinking about a simpler version. For example, if you are asked to prove something for an $n \times n$ matrix, try writing out the same statements for a 2×2 matrix. Check again for similarities between what is known and what you want to prove. See if you can prove the theorem for the simpler case of 2×2.

   In the process of trying out an example, you may notice a pattern in your working that extends to the general proof. Don't be afraid to try several examples, if just one isn't giving you enough intuition. Trying out examples is also a great way of ensuring that you understand what you want to prove in concrete terms, rather than just as an abstract claim.

6. **Manipulate both sides of the claim. JUSTIFY each step.** After coming up with an algebraic representation of the desired claim, try to manipulate both what you are *given* and what you are *trying* to prove, to see if you can simplify the desired claim. Often, it helps to get rid of complex notation as part of this process - for instance, if you see a summation expressed using $\sum$ notation, it might help to write it out explicitly to get a better understanding of what exactly is being summed.

   Still, at this stage it is important to ensure that your manipulations are valid - it's no use making an amazing simplification if it turns out to be wrong! It can't hurt to break down complex steps into multiple smaller ones, and always ask yourself what conditions are needed for each step to be true. For instance, if you are dividing two quantities, then you should ensure that you are not dividing by zero!

7. **Different approaches.** There are different ways to prove a statement. You might encounter some of the following types of proofs.

   - *Direct proofs.* This style of proof directly shows what is to be proven from what is known by doing a series of mathematical and logical steps.
   - *Constructive proofs.* Essentially, when you are asked to prove that a certain object "exists", you can prove the statement by explicitly constructing the object.
   - *Proof by contradiction.* Another common technique when asked to prove a claim is to try and show that it is impossible for the claim to *not* be true. This technique tends to be useful when the negation of the desired claim is easier to express algebraically than the claim itself.

   No one approach is better or more powerful than any other approach — there is no hierarchy. When you are thinking about proofs try out different approaches to see which one might work.

Solving proof problems can be similar to solving design problems, which is one of the focuses of this course. In a proof, there is something we want to show, and in a design problem, there are specifications we want our design to meet. Both cases are open ended, and we frequently have to integrate many ideas and explore several different possibilities to reach a solution.

Now, we'll look at two examples: one of a constructive proof, the other a proof by contradiction.

Example 4.1 (Constructive Proof): Prove that $\text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right\} = \mathbb{R}^2$

First, let's figure out exactly what the question is asking us to prove. The span of two vectors is a set of all vectors that can be formed as a linear combination of those two vectors. $\mathbb{R}^2$ is the set of all (Cartesian) vectors with two real components.

We want to show that these two sets are equal, meaning that no element can be in one set but not the other. Thus, we need to show

1. That every vector in the given span is inside $\mathbb{R}^2$, and that
2. Every vector in $\mathbb{R}^2$ is in the given span.

Let's prove each of these statements in order. First, how do we show that every vector in $\text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right\}$ is contained within $\mathbb{R}^2$? Let's express what we are given algebraically. Consider an arbitrary vector $\vec{u} \in \text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right\}$. By the definition of span and linear combinations, we know that we can write

$$\vec{u} = \alpha \begin{bmatrix}1\\1\end{bmatrix} + \beta \begin{bmatrix}1\\-1\end{bmatrix}$$

for some real scalar coefficients α and β. By the rules of vector arithmetic, we can multiply in our scalar coefficients and simplify to obtain

$$\vec{u} = \begin{bmatrix}\alpha+\beta\\ \alpha-\beta\end{bmatrix}.$$

Why is this vector in $\mathbb{R}^2$? Well, since α and β are both real numbers, their sums and differences are both also real. And $\vec{u}$ clearly has exactly two components. Thus, $\vec{u} \in \mathbb{R}^2$, for arbitrary real scalars α and β. Since $\vec{u}$ could have been *any* vector in $\text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right\}$, we have shown that *any* vector in the given span is inside $\mathbb{R}^2$, which is the first thing we wanted to show.

But we're not done yet! All we've done is shown that $\text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}\right\} \subseteq \mathbb{R}^2$ - there might still exist a vector outside of the given span that is still within $\mathbb{R}^2$! Let's now try to show that *every* vector in $\mathbb{R}^2$ lies in the given span, by writing any vector in $\mathbb{R}^2$ as a linear combination of the two given vectors.

We'll start with an arbitrary vector $\vec{u} \in \mathbb{R}^2$. By definition, $\vec{u}$ has two real components, so it can be written as

$$\vec{u} = \begin{bmatrix}u_1\\u_2\end{bmatrix}$$

where u_1 and u_2 are both real scalars. We want to know if $\vec{u}$ can be written as a linear combination of the two given vectors, and so must ask whether we can find real scalars α and β (that will depend on u_1 and u_2) such that

$$\alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}?$$

Manipulating the above equation, we obtain the linear system

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

We want to determine whether there will always exist real solutions α and β to the above linear system, no matter what u_1 and u_2 are. And we know how to do this - Gaussian elimination!

Rewriting the above as an augmented matrix and performing row operations without comment, we obtain

$$\begin{aligned} & \left[\begin{array}{cc|c} 1 & 1 & u_1 \\ 1 & -1 & u_2 \end{array}\right] \\ \iff & \left[\begin{array}{cc|c} 1 & 1 & u_1 \\ 0 & -2 & u_2 - u_1 \end{array}\right] \\ \iff & \left[\begin{array}{cc|c} 1 & 1 & u_1 \\ 0 & 1 & \frac{u_1-u_2}{2} \end{array}\right] \\ \iff & \left[\begin{array}{cc|c} 1 & 0 & \frac{u_1+u_2}{2} \\ 0 & 1 & \frac{u_1-u_2}{2} \end{array}\right]. \end{aligned}$$

Rewriting our end result in matrix-vector form once again, we obtain

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{u_1+u_2}{2} \\ \frac{u_1-u_2}{2} \end{bmatrix},$$

which can be expanded and rearranged as

$$\begin{aligned} \alpha &= \frac{u_1+u_2}{2} \\ \beta &= \frac{u_1-u_2}{2}. \end{aligned}$$

What have we shown? Since we proved in an earlier note that Gaussian elimination cannot introduce spurious solutions, we've shown that for all u_1 and u_2, we can produce coefficients α and β such that

$$\alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \vec{u}.$$

So all $\vec{u} \in \mathbb{R}^2$ can be written as a linear combination of our two vectors! Thus,

$$\text{span}\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \supseteq \mathbb{R}^2,$$

which when combined with our earlier result proves that the two sets are equal, as desired! This completes our proof.

Notice that the second half of this proof was a "constructive proof" since, in order to show that the coefficients α and β existed for all $\vec{u}$, we were actually able to construct them explicitly!

Example 4.2 (Proof by contradiction): If a theorem statement asks you to prove that something does not exist (e.g. the inverse of a matrix does not exist, or a unique solution does not exist etc.) it is often useful to consider a proof by contradiction. Assume that a unique solution exists, and then show that there is some other contradiction.

Here we repeat the theorem from the previous note to illustrate this.

Theorem 4.1: **If the columns of A in the system of linear equations $A\vec{x} = b$ are linearly dependent, then the system does not have a unique solution.**

Let's walk through this proof step by step: we'll start by assuming we have a matrix A with linearly dependent columns, and then we will show that this means that the system does not have a unique solution.

Since we are interested in the columns of A, let's start by explicitly defining the columns of A:

$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \dots & \vec{a}_n \\ | & | & & | \end{bmatrix},$$

Now, first, let us write out what is known to us.

By the definition of linear dependence, there exist scalars $\alpha_1, \dots, \alpha_n$ such that $\alpha_1\vec{a}_1 + \dots + \alpha_n\vec{a}_n = \vec{0}$ where not all of the α_i's are zero. We can put these α_i's in a vector

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

and by the definition of matrix-vector multiplication, we can compactly write the expression above:

$$A\vec{\alpha} = \vec{0}$$

where $\vec{\alpha} \neq \vec{0}$.

What do we want to prove?

We are trying to show that the system of equations $A\vec{x} = \vec{b}$ does not have a unique solution.

We know that systems of equations can have either zero, one, or infinite solutions. If our system of equations has zero solutions, then it cannot have a unique solution, so we don't need to consider this case.

Now, let us assume we have a unique solution and arrive at a contradiction.

Now let's consider the case where we have at least one solution, $\vec{x}_0$:

$$\begin{aligned} A\vec{x}_0 &= \vec{b} \\ A\vec{x}_0 + \vec{0} &= \vec{b} \\ A\vec{x}_0 + A\vec{\alpha} &= \vec{b} \\ A\left(\vec{x}_0 + \vec{\alpha}\right) &= b \end{aligned}$$

Therefore, $\vec{x}_0 + \vec{\alpha}$ is also a solution to the system of equations! To complete our proof, we now must check that these two solutions are in fact not equal to each other.

Since $\vec{\alpha} \neq \vec{0}$, we know that $\vec{x}_0 \neq \vec{x}_0 + \vec{\alpha}$. Therefore, our system has two distinct solutions. We've now proven the theorem.

Note that we can add any multiple of $\vec{\alpha}$ to $\vec{x}$ and it will still be a solution – therefore, if there is at least one solution to the system and the columns of A are linearly dependent, then there are infinite solutions.

Let's illustrate the ideas above on the following examples of proof problems:

Example 4.3 (Explicitly constructing a linear combination): 1. Let $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ be a set of linearly dependent vectors in $\mathbb{R}^n$. Take any matrix $A \in \mathbb{R}^{m \times n}$. Prove that the set of vectors $\{A\vec{v}_1, A\vec{v}_2, \ldots, A\vec{v}_n\}$ is linearly dependent.

Proof: (1) What do we know? Based on the problem statement, we know that $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ is a set of linearly dependent vectors. How do we translate this into mathematical form? Recall one of the two definitions of linear dependence we introduced in the previous note – the set of vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ is linearly dependent if there exists an index i and scalars α_j's such that

$$\vec{v}_i = \sum_{j \neq i} \alpha_j \vec{v}_j. \tag{1}$$

We rewrite this by removing the sigma notation:

$$\vec{v}_i = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \cdots + \alpha_n v_n. \tag{2}$$

(2) What would we like to show? We would like to show that the set of vectors $\{A\vec{v}_1, A\vec{v}_2, \ldots, A\vec{v}_n\}$ is linearly dependent. Again using the definition of linear dependence, we can translate it into a mathematical statement – we would like to show that there exist index k and scalars β_l's such that

$$A\vec{v}_k = \sum_{l \neq k} \beta_l \left(A\vec{v}_l\right). \tag{3}$$

(3) Now, how do we use what we know mathematically from (1) to prove the mathematical statement in (2)? We somehow would like to get vectors of the form $A\vec{v}$. How could we do that? Let's multiply both sides of equation (1) by the matrix A:

$$A\vec{v}_i = A\left(\sum_{j\neq i}\alpha_j\vec{v}_j\right). \tag{4}$$

By distributivity of matrix-vector multiplication, we know that

$$A\left(\sum_{j\neq i}\alpha_j\vec{v}_j\right) = \sum_{j\neq i}A\left(\alpha_j\vec{v}_j\right) = \sum_{j\neq i}\alpha_j\left(A\vec{v}_j\right). \tag{5}$$

Now, we have that

$$A\vec{v}_i = \sum_{j\neq i}\alpha_j\left(A\vec{v}_j\right), \tag{6}$$

which is in exactly the mathematical form we would like to show in equation (3). So what are the values of the β's we should choose in equation (3)? We have $\beta_l = \alpha_l$ for all l.

Hence, we have completed our proof. Therefore, we have explicitly found a linear combination of the columns of matrix A that gives another column of the matrix.

Example 4.4 (Direct proof): This proof does not follow any particular style of proof — it directly deduces what is to be proved from what is known. **$\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_1+\vec{v}_2$ are all solutions to the system of linear equation $A\vec{x}=\vec{b}$. Prove that $\vec{b}$ must be the zero vector.**
Proof: What does it mean for $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_1+\vec{v}_2$ to be the solutions to $A\vec{x}=\vec{b}$? It means these vectors must satisfy the following equations:

$$A\vec{v}_1 = \vec{b} \tag{7}$$

$$A\vec{v}_2 = \vec{b} \tag{8}$$

$$A\left(\vec{v}_1+\vec{v}_2\right) = \vec{b} \tag{9}$$

Notice that using distributivity of matrix-vector multiplication, equation (8) can be rewritten as

$$A\vec{v}_1 + A\vec{v}_2 = \vec{b}. \tag{10}$$

Now from equation (6) and (7), we can substitute $A\vec{v}_1$ and $A\vec{v}_2$ with the vector $\vec{b}$, which leads us to

$$\vec{b}+\vec{b} = \vec{b}. \tag{11}$$

Subtracting $\vec{b}$ from both sides of the equation above, we have

$$\vec{b} = \vec{0}. \tag{12}$$

Hence $\vec{b}$ is the zero vector, as desired.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 5

5.1 Water Reservoirs and Pumps

One application we will use to visualize matrix-vector multiplication is water reservoirs and water pumps. We are presenting this example because it is vital, as an engineer, to understand the ideas that we are talking about in an intuitive way; this intuition often comes from having a series of examples. After all, we define mathematical operations the way that we do because these definitions are useful; they don't come out of nowhere. The act of doing mathematics and engineering is often about making definitions and seeing where they lead us, while checking the consistency of these definitions with what we are trying to model in the real world.

For this example, we will have three water reservoirs, A, B, C. Let's say the initial amounts of water they respectively hold are A_0, B_0, C_0. Next, say we have a system of pumps connecting the reservoirs that move certain amounts of water between the reservoirs every day. We can represent the reservoirs as the following vector, where each element describes how much water is currently in that reservoir:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Then, we can represent the system of pumps as a matrix:

$$\begin{bmatrix} P_{A\to A} & P_{B\to A} & P_{C\to A} \\ P_{A\to B} & P_{B\to B} & P_{C\to B} \\ P_{A\to C} & P_{B\to C} & P_{C\to C} \end{bmatrix}$$

Each element $P_{i\to j}$ represents the fraction of water in reservoir i that gets pumped into reservoir j the next day. The matrix acts on the vector just as the pumps act on the reservoirs, performing a transformation — multiplying a vector representing the distribution of water in one day by the pump matrix will give a vector with the distribution of water the next day. We call this matrix a **state transition matrix**. This example can also extend to matrix-matrix multiplication. Both this pumps and reservoirs example and a similar example (PageRank — how search engines can use link information to figure out which pages are important) will show further applications of linear algebra.

5.1.1 Basic Pump

The most basic pump system will move all water from one reservoir into another. Pictorially, we can show this as follows (blue circles are the reservoirs and arrows represent how the pumps move the water):

Given this diagram, we can say that $P_{A\to B} = 1, P_{B\to C} = 1, P_{C\to A} = 1$. Therefore, the corresponding matrix-vector multiplication is:

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix},$$

which will describe the new water levels in the reservoirs after one day has passed.

Each time the pumps act on the reservoirs, all of the water in reservoir A flows into reservoir B. All of the water in reservoir B flows into reservoir C. All of the water in reservoir C flows into reservoir A. If A, B, and C all start with the same amount of water, then the pumps acting on the reservoirs would not change the amount of water in each reservoir. As an example, let the amount of water in each reservoir initially be A_0, B_0, C_0. Then we can calculate the amount of water in each reservoir after activating the pumps once (A_1, B_1, C_1) as follows:

$$\begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix} = \begin{bmatrix} C_0 \\ A_0 \\ B_0 \end{bmatrix}.$$

5.1.2 Identity Matrix Pump

What happens when your pump system can be represented as the identity matrix? What does that mean? If the initial amounts of water in the reservoirs are represented by the vector $\begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix}$, and the identity matrix represents how the pumps move the water, after one activation of the pumps, nothing changes!

$$\begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix} = \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix}$$

Here's how that would look pictorially:

5.1.3 Drain

Another special matrix is the zero matrix:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In terms of the reservoirs, it would be some sort of drain (e.g. an evil monster evaporated all of the water in the reservoirs). The zero matrix acting on the reservoirs results in zero water left in each reservoir.

This does not obey water conservation — the total amount of water after the pump matrix is applied will not equal the initial total — but can still be represented as a matrix.

5.1.4 Conservation of Water

Now let's look at what happens when pumps move different amounts of water from each reservoir into other reservoirs. Specifically, let's work with this diagram:

These pumps can be described with this matrix:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix}$$

Each element of the matrix still represents a pump and indicates how much water is moved where. The first row indicates how much of each reservoir contributes to reservoir A when the pumps are activated. The second row does the same for reservoir B, and the third row is for reservoir C. For example, the upper left element $\frac{1}{2}$ tells us that half of what is in reservoir A will stay in reservoir A. Similarly, the $\frac{1}{2}$ on the middle left tells us that the other half of what was in reservoir A will flow into reservoir B when the pumps turn on. As we can see, each column of the matrix sums to one. This means the water is conserved (none is mysteriously lost or gained). The water will either stay in the original reservoir or move to a different one. This is useful to describe water moving between pools with no evaporation, but it is not something that is going to hold in all useful applications of matrices.

After activating the pumps once, how do we know how much water is in each reservoir? That can be calculated with a matrix-vector product, just as we saw with the simpler pump models:

$$\begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot A_0 + \frac{1}{4} \cdot B_0 + \frac{1}{3} \cdot C_0 \\ \frac{1}{2} \cdot A_0 + 0 \cdot B_0 + \frac{1}{3} \cdot C_0 \\ 0 \cdot A_0 + \frac{3}{4} \cdot B_0 + \frac{1}{3} \cdot C_0 \end{bmatrix}$$

Matrix Multiplication Examples

Now let's look at how matrix-matrix multiplication can be applied to water reservoirs and pumps.

5.1.5 Twin Cities

In this scenario, we have two cities that each have three reservoirs. The pump systems in the cities are identical. Let's start with the basic pump system.

(a) City 1 (b) City 2

The pump system can still be represented as a matrix, like before:

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

However, now the pump system acts on two sets of reservoirs instead of just one. Can our pump matrix act on two vectors representing the reservoirs instead of just one? We can combine the two vectors representing the water reservoirs in each city into a single matrix, with a column for each reservoir vector as follows:

$$\begin{bmatrix} A & R \\ B & S \\ C & T \end{bmatrix}$$

Now, let's use the pump matrix to find the water distribution in both cities in a single calculation. In City 1, the reservoirs initially have water amounts A_0, B_0, C_0. In City 2, the reservoirs initially have water amounts R_0, S_0, T_0. Once the pumps act on the reservoirs, the amount of water in each reservoir can be found through matrix-matrix multiplication:

$$\begin{bmatrix} A_1 & R_1 \\ B_1 & S_1 \\ C_1 & T_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} A_0 & R_0 \\ B_0 & S_0 \\ C_0 & T_0 \end{bmatrix} = \begin{bmatrix} C_0 & T_0 \\ A_0 & R_0 \\ B_0 & S_0 \end{bmatrix}$$

If the cities have identical but more complicated pumps (such as the conservation pumps in the previous example), finding out how the reservoirs change is the same process. All that would be different is the "pump system" matrix.

What happens if you have the same pump-reservoir system in k cities? To find out how the pumps act on the reservoirs, you can still use matrix-matrix multiplication. One matrix describes the pumps, while the other describes the reservoirs. There would be k columns in the reservoir matrix, because each column is a vector that represents the reservoirs of a certain city.

In this case, we see a matrix acting on another matrix to transform multiple vectors the same way. Because of this, we can also see why the dimensions of the matrix have certain restrictions. The number of columns in the pumps matrix must match the number of rows in the reservoir matrix. The pumps matrix acts on each column of the reservoir matrix to produce a new column for the resulting matrix that describes amount of water for that city's reservoirs.

5.1.6 Activate Pumps Once... And Then Once More

Now, imagine we have one system of pumps for one city with three reservoirs. How can we calculate the amount of water in each reservoir after activating the pumps twice? From matrix-vector multiplication, we know how to find the amount after one activation. If matrix A represents the pumps and $\vec{v}_0$ represents the initial reservoir vector, then $\vec{v}_1 = A \cdot \vec{v}_0$ will tell us how much water is in each reservoir after one activation. Then $\vec{v}_2 = A \cdot \vec{v}_1$ will tell us how much water is in each reservoir after the second activation.

But from the reservoirs' standpoints, how they got from $\vec{v}_0$ to $\vec{v}_2$ does not matter. For all they know, it could have been some other system of pumps (matrix B) that acted on the initial reservoir vector ($\vec{v}_0$) that resulted in $\vec{v}_2$. This means that one set of pumps acting twice on the reservoirs is equivalent to *another* matrix acting on the reservoirs:

$$\begin{aligned} A\vec{v}_1 &= \vec{v}_2 \\ A(A \cdot \vec{v}_0) &= B \cdot \vec{v}_0 \\ (A \cdot A)\vec{v}_0 &= B \cdot \vec{v}_0 \\ A^2 &= B \end{aligned}$$

As an example, let's take the pump system from the conservation example in section 5.1.4:

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix}, \vec{v}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Let's calculate $\vec{v}_2$:

$$\vec{v}_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{13}{12} \\ \frac{5}{6} \\ \frac{13}{12} \end{bmatrix}$$

$$\vec{v}_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} \frac{13}{12} \\ \frac{5}{6} \\ \frac{13}{12} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{65}{72} \\ \frac{71}{72} \end{bmatrix}$$

For comparison:

$$B = A^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{3}{8} & \frac{3}{8} & \frac{13}{36} \\ \frac{1}{4} & \frac{3}{8} & \frac{5}{18} \\ \frac{3}{8} & \frac{1}{4} & \frac{13}{36} \end{bmatrix}$$

$$B \cdot \vec{v}_0 = \begin{bmatrix} \frac{3}{8} & \frac{3}{8} & \frac{13}{36} \\ \frac{1}{4} & \frac{3}{8} & \frac{5}{18} \\ \frac{3}{8} & \frac{1}{4} & \frac{13}{36} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{65}{72} \\ \frac{71}{72} \end{bmatrix}$$

From this example, we can see that matrix-matrix multiplication results in an equivalent matrix. Pump system A acting twice on the reservoirs is the same as pump system B acting once on the reservoirs.

5.1.7 A Multitude of Pumps

Another example of matrix-matrix multiplication with these pumps and reservoirs is when two (or more) *different* sets of pumps act sequentially on a city's reservoirs. From the previous example, we know that a matrix multiplied by another matrix is equivalent to another matrix. That principle can be applied here. So if we have Pump System A act on the reservoirs ($\vec{v}_0$) and then Pump System B act on the reservoirs, it is the same as if some other Pump System C acted on the reservoirs:

$$\begin{aligned} B \cdot (A \cdot \vec{v}_0) &= C \cdot \vec{v}_0 \\ (B \cdot A)\vec{v}_0 &= C \cdot \vec{v}_0 \\ B \cdot A &= C \end{aligned}$$

5.1.8 Continuous vs. Discrete Pumps

In all of the examples above, we've assumed that the pumps act instantaneously. That is, each time the pumps transfer water, first we calculate how much water will be moved based on the initial water levels in each reservoir. Then, all of the water is moved instantaneously. (At any given time, water entering reservoir A is not used in calculating how much water leaves reservoir A.) We can repeat this process (every day, minute, hour, etc), but each time all the water moves instantaneously. We call this process a *discrete time system* because water is transferred only at a discrete times.

While some physical properties happen in discrete time, others happen in *continuous time* – in other words, not instantaneously. We can describe continuous time systems with *differential equations*, which you will learn more about in EE 16B. But even without these techniques, we can still approximate the solution to a continuous time system by modeling it as a discrete time system where we take very small steps in time. Here, small is relative to how long the process takes: if it takes a minute for the pumps to transfer water, we could calculate the new water levels every second. If we are trying to model how light travels in space, we might need to calculate a new time step every femtosecond!

5.2 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Let the state transition matrix $\begin{bmatrix} 0.5 & 0.3 & 0 \\ 0 & 0.5 & 1 \\ 0.4 & 0.2 & 0 \end{bmatrix}$ represent people moving between three cities. If $\vec{x}[0] = \begin{bmatrix} 100 \\ 200 \\ 100 \end{bmatrix}$, find $\vec{x}[1]$.

2. Let the state transition matrix $\begin{bmatrix} 0.5 & 0.3 & 0 \\ 0 & 0.5 & 1 \\ 0.4 & 0.2 & 0 \end{bmatrix}$ represent people moving between three cities. Do people stay within these three cities?

3. Let the state transition matrix $\begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.5 \\ 0.6 & 0.15 & 0 & 0.2 \\ 0.3 & 0.5 & 0.3 & 0.2 \\ 0 & 0.25 & 0.3 & 0.1 \end{bmatrix}$ represent the transfer of water between different buckets. The amount of water in each bucket a, b, c, and d at time n is $\begin{bmatrix} 3 \\ 4 \\ 19 \\ 1 \end{bmatrix}$. How much water is in bucket c at time $n+1$?

4. Consider the web traffic among the search engines given below. Write the state transition matrix for this system assuming that the state vector is $\vec{x}[t] = \begin{bmatrix} x_{\text{Google}}[t] \\ x_{\text{Yahoo}}[t] \\ x_{\text{Bing}}[t] \end{bmatrix}$.

5. Is the web traffic system modeled in the previous question conservative, i.e., is the number of web surfers in the system constant?

6. If a column adds up to a number larger than 1, what does this imply about the corresponding node?

   (a) People are leaving the system at that node.
   (b) People are entering the system at that node.
   (c) The node can exist in a conservative system.
   (d) The node has been wrongly modeled in the system.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 6

6.1 Introduction: Matrix Inversion

In the last note, we considered a system of pumps and reservoirs where the water in each reservoir is represented as a vector and the pumps, represented as a matrix, act on the reservoirs to move water into a new state. If we know the current state of the reservoirs, $\vec{v}[t]$, and we know the state transition matrix describing the pumps, A, we can find the water at the next time step through matrix-vector multiplication:

$$\vec{v}[t+1] = A\vec{v}[t]$$

However, suppose we'd like to find the water in the reservoirs at a *previous* timestep, $\vec{v}[t-1]$. Is there a state transition matrix B, that can take us backwards in time?

$$\vec{v}[t-1] = B\vec{v}[t]$$

Yes, the matrix that "undoes" the effects of A is its inverse! In this note, we'll define matrix inverses, introduce some properties, and investigate when matrix inverses exist (and when they don't).

6.1.1 Definition and properties of matrix inverses

Definition 6.1 (Inverse): A square matrix A is said to be invertible if there exists a matrix B such that

$$AB = BA = I. \tag{1}$$

where I is the identity matrix. In this case, we call the matrix B the inverse of the matrix A, which we denote as A^{-1}.

Example 6.1 (Matrix inverse): Consider the 2×2 matrix $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$. The inverse of A is $A^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$. We can verify that the following holds

$$AA^{-1} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2}$$

$$A^{-1}A = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3}$$

Example 6.2 (Using Inverses to Find Unique Solutions): Previously, when we were trying to find solutions to $A\vec{x} = \vec{b}$ we used Gaussian Elimination to solve the system. However, if the system has a unique

solution, we can find $\vec{x}$ using the inverse. We start by left multiplying the inverse A^{-1} to both sides of our equation:

$$A^{-1}A\vec{x} = A^{-1}\vec{b} \tag{4}$$
$$I\vec{x} = A^{-1}\vec{b} \tag{5}$$
$$\vec{x} = A^{-1}\vec{b} \tag{6}$$
$$\tag{7}$$

Now consider the system $A\vec{x} = \vec{b}$ where $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ and $\vec{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. We can use the inverse $A^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$ to solve for $\vec{x}$.

$$\vec{x} = A^{-1}\vec{b} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We mentioned that this method is only applicable for unique solutions. You will see in examples below that for other solution cases you will not be able to find an inverse.

Let's show an important property of matrix inverses:

Theorem 6.1: If A is an invertible matrix, then its inverse must be unique.

Proof. Suppose B_1 and B_2 are both inverses of the matrix A. Then we have

$$AB_1 = B_1A = I \tag{8}$$
$$AB_2 = B_2A = I \tag{9}$$

Now take the equation

$$AB_1 = I. \tag{10}$$

Multiplying both sides of the equation by B_2 from the left, we have

$$B_2(AB_1) = B_2I = B_2. \tag{11}$$

Notice that by associativity of matrix multiplication, the left hand side of the equation above becomes

$$B_2(AB_1) = (B_2A)B_1 = IB_1 = B_1. \tag{12}$$

Hence we have

$$B_1 = B_2. \tag{13}$$

We see that B_1 and B_2 must be equal, so the inverse of any invertible matrix is unique. □

Another important property of inverses is that the "left" inverse and the "right" inverse are equal to each other. In particular

Theorem 6.2: If $QP = I$ and $RQ = I$, then $P = R$. The matrix P can be thought of as the "right" inverse of Q and the matrix R can be thought of as the "left" inverse of Q.

Proof. We start the proof by noticing that we know two things $QP = I$ and $RQ = I$. To move ahead, we can try setting $QP = RQ$, but we cannot proceed from here, since the multiplication by Q is on different sides. So instead we take the equation $QP = I$ and multiply both sides on the left by R. This gives

$$R(QP) = R(I) = R. \tag{14}$$

Now, using the associative property of matrix multiplication we have that

$$R(QP) = (RQ)P = IP = P. \tag{15}$$

Here we used $RQ = I$.

Combining 14 and 15 we have that $R = P$, and we are done. □

In discussion, you will see a few more useful properties of matrix inverses.

Some of the next natural questions to ask are:

- How do we know whether or not a matrix is invertible?
- If a matrix is invertible, how do we go about finding its inverse?

It turns out Gaussian elimination could help us answer these questions!

6.1.2 Finding Inverses With Gaussian Elimination

A square matrix M and its inverse M^{-1} will always satisfy the following conditions $MM^{-1} = I$ and $M^{-1}M = I$, where I is the identity matrix. Let's take an example with

$$M = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \text{ and we are trying to find } M^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

We want to find the values of b_{ij} such that the equation $MM^{-1} = I$ would be satisfied.

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Just as we did for solving equations of the form $A\vec{x} = \vec{b}$, we can use Gaussian Elimination to solve for the unkown variables, even though they are in a matrix instead of a vector. To do so, we start by writing the above as an **augmented matrix**, which joins the left and right numerical matrices together and hides the variable matrix, as shown below.

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right]$$

Now, to find the inverse matrix M^{-1} using Gaussian elimination, we have to transform the left numerical matrix (left half of the augmented matrix) to the identity matrix, then the resulting right half of the augmented matrix becomes our solution. In equation form $MM^{-1} = I$, we are transforming M and I simultaneously using row operations so that the equation becomes $IM^{-1} = A$, where A is the resulting numerical matrix

from the Gaussian elimination. Since M^{-1} is multiplied by the identity matrix I, the resulting numerical matrix A must equal to M^{-1}, and we have the values for the elements in our inverse matrix. We will now do the actual computation below:

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right] \Rightarrow R_2 - 2R_1 \Rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & -1 & -2 & 1 \end{array}\right] \Rightarrow -1(R_2) \Rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & -1 \end{array}\right]$$

$$\Rightarrow R_1 - R_2 \Rightarrow \left[\begin{array}{cc|cc} 1 & 0 & -1 & 1 \\ 0 & 1 & 2 & -1 \end{array}\right]$$

M^{-1} is the right half of the augmented matrix. Therefore $M^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$. More generally, for any $n \times n$ matrix M, we can perform Gaussian elimination on the augmented matrix. Note that the matrix I_n is the identity matrix in $\mathbb{R}^{nxn}$.

$$\left[\begin{array}{c|c} M & I_n \end{array}\right]$$

If at termination of Gaussian elimination, we end up with an identity matrix on the left, then the matrix on the right is the inverse of the matrix M.

$$\left[\begin{array}{c|c} I_n & M^{-1} \end{array}\right]$$

If we don't end up with an identity matrix on the left after running Gaussian elimination (i.e. due to a row of zeros), we know that the matrix is not invertible. Therefore, an inverse can only be found in a unique solution case. Knowing if a matrix is invertible can tell us about the rows/columns of a matrix, and knowing about the rows/columns can tell us if a matrix is invertible - let's look at how.

Additional Resources For more on matrix inverses, read *Strang* pages 83-85 and try Problem Set 2.5; or read *Schuam's* pages 33-34 and try Problems 2.17 to 2.20, 2.54, 2.55, 2.57, and 2.58.

6.2 Connecting invertibility with matrix rows and columns

First let's consider how the rows of the matrix relate to invertibility.

Example 6.3 (Invertibility Intuition – Rows): Suppose we have a black and white image with two pixels. We cannot directly see the shade of each pixel, but we can measure *linear combinations* of the light the two pixels absorb. Define the light absorbed by the two pixels as x_1 and x_2. We take the following two measurements, y_1, y_2:

$$\begin{aligned} y_1 &= x_1 + x_2 \\ y_2 &= 2x_1 + 2x_2 \end{aligned}$$

Written in matrix form:

$$\vec{y} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \vec{x} \tag{16}$$

Can we invert our measurement matrix to solve for the shade of each pixel? No, because our second measurement y_2 does not provide any new information, since it is linearly dependent with the first measurement. (i.e. one row is a multiple of the other or $\begin{bmatrix}2 & 2\end{bmatrix} = 2\begin{bmatrix}1 & 1\end{bmatrix}$) As a result, there is no unique solution, which is required for inversion. Hence, that we need linearly independent rows to have an invertible matrix.

We won't prove this rigorously, but we can extend this intuition by examining the Gaussian elimination method for finding matrix inverses: If we run Gaussian elimination on a matrix M and do not end up with the identity matrix, this means that there is no unique solution, and therefore the matrix is not invertible. If we don't get the identity matrix, we will have a row of zeros, which indicates that the rows of M are linearly dependent.

Now let's look from the column perspective.

Consider A as an operator on any vector $\vec{x} \in \mathbb{R}^n$. What does it mean for A to have an inverse? It suggests that we can find a matrix that "undoes" the effect of matrix A operating on any vector $\vec{x} \in \mathbb{R}^n$. What property should A have in order for this to be possible? A should map each distinct vector to a distinct vector in $\mathbb{R}^n$. This is to say that the operator A should not map multiple input vectors to the same output vector, since that would make it impossible to figure out which input vector resulted in that output vector, so the system would not be invertible. Mathematically, we say, $A\vec{x}_1 \neq A\vec{x}_2$ for input vectors $\vec{x}_1, \vec{x}_2$ such that $\vec{x}_1 \neq \vec{x}_2$.

Consider this example:

Example 6.4 (Invertibility Intuition – Columns):
The matrix $A = \begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}$ is not invertible because it can map two distinct vectors into the same vector:

$$\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}1\\ 3\end{bmatrix} = \begin{bmatrix}4\\ 0\end{bmatrix} \tag{17}$$

$$\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}2\\ 2\end{bmatrix} = \begin{bmatrix}4\\ 0\end{bmatrix}. \tag{18}$$

We cannot recover the input vector uniquely after it is operated on by A because there are multiple possible input vectors that could have resulted in the same output vector. This is connected with the fact that the columns are linearly dependent – different weighted combinations of columns could generate the same vector.

We can generalize these observations to obtain a couple of theorems. Let's look at some proofs:

Theorem 6.3: **If a matrix A is invertible, there exists a unique solution to the equation $\mathbf{A}x = \vec{b}$ for all possible vectors $\vec{b}$.**

To prove this, we need to prove two statements:

1. That there exists *at least one* solution to the equation $\mathbf{A}x = \vec{b}$, and that
2. There exists *no more than one* solution to the equation $\mathbf{A}x = \vec{b}$.

For both of the above statements, $\vec{b}$ can be any vector in $\mathbb{R}^n$. Let's prove the first statement first. Imagine we are given a vector $\vec{b}$. Consider the candidate solution $\vec{x} = \mathbf{A}^{-1}\vec{b}$. Observe that

$$\mathbf{A}\vec{x} = \mathbf{A}(\mathbf{A}^{-1}\vec{b}) = (\mathbf{A}\mathbf{A}^{-1})\vec{b} = \vec{b}.$$

Thus, our candidate solution satisfies the equation $\mathbf{A}\vec{x} = \vec{b}$, so there exists at least one solution to that equation!

Now, let's show the second statement - that no more than one solution to the equation $\mathbf{A}\vec{x} = \vec{b}$ can exist. Consider a particular solution $\vec{x}$, so $\mathbf{A}\vec{x} = \vec{b}$. Pre-multiplying both sides of this equation by $\mathbf{A}^{-1}$, we obtain

$$\mathbf{A}^{-1}(\mathbf{A}\vec{x}) = \mathbf{A}^{-1}\vec{b} \implies x = \mathbf{A}^{-1}\vec{b},$$

so if $\vec{x}$ exists, it must be the particular vector $\mathbf{A}^{-1}\vec{b}$. In other words, there exists at most one solution to the equation $\mathbf{A}x = \vec{b}$, so we have proven the second statement.

Now, let's look at another (related) theorem that also seems to be suggested by our observations.

Theorem 6.4: **If a matrix A is invertible, its columns are linearly independent.**

Let's prove this theorem. We know that the statement "the columns of $\mathbf{A}$ are linearly independent" is equivalent to the statement "$\mathbf{A}\vec{x} = \vec{0}$ only when $\vec{x} = \vec{0}$." This fact follows from the definition of linear independence: by definition, if $\vec{v}_1, ..., \vec{v}_n$ are linearly independent, then $\sum_{i=1}^{n} x_i\vec{v}_i$ is only $\vec{0}$ when $x_i = 0$. Using the column perspective of matrix multiplication (covered in Note 3), $\mathbf{A}\vec{x} = \sum_{i=1}^{n} x_i\vec{v}_i$ where $\vec{v}_i$ is the ith column of $\mathbf{A}$. Therefore, $\mathbf{A}\vec{x} = \vec{0}$ only when all $x_i = 0$.

Therefore, we can rephrase what we're trying to prove as

$$\mathbf{A}^{-1} \text{ exists} \implies (\mathbf{A}\vec{x} = \vec{0} \text{ only when } \vec{x} = \vec{0})$$

To prove this, assume that $\mathbf{A}$ is invertible. Let $\vec{v}$ be some vector such that $\mathbf{A}\vec{v} = \vec{0}$:

$$\begin{aligned} \mathbf{A}\vec{v} &= \vec{0} \longleftarrow \text{left-multiply by } \mathbf{A}^{-1} \\ \mathbf{A}^{-1}\mathbf{A}\vec{v} &= \mathbf{I}\vec{v} = \vec{0} \\ \vec{v} &= \vec{0} \end{aligned}$$

Hooray! We've successfully proven this theorem!

As it turns out, we can actually strengthen both of these results, to obtain equivalence, rather than just implication.

Theorem 6.5: The following statements are all equivalent for a square matrix $\mathbf{A}$:

1. $\mathbf{A}$ is invertible
2. The equation $\mathbf{A}\vec{x} = \vec{b}$ has a unique solution for any $\vec{b}$
3. $\mathbf{A}$ has linearly independent columns
4. $\mathbf{A}$ has a trivial nullspace
5. the determinant of $\mathbf{A} \neq 0$.

We will discuss the last two points in future notes.

We have shown that

- $\mathbf{A}$ is invertible $\Longrightarrow$ the equation $\mathbf{A}\vec{x} = \vec{b}$ has a unique solution for any $\vec{b}$.
- $\mathbf{A}$ is invertible $\Longrightarrow$ $\mathbf{A}$ has linearly independent columns
- $\mathbf{A}$ is invertible $\Longrightarrow$ $\mathbf{A}$ has a trivial nullspace.

We have not yet shown implications in the other direction, and have not introduced the definition for a determinant. We will define a determinant in the coming notes. Even though we have not yet shown that these statements are equivalent, i.e. the implications go both ways, you may use them as tools to help your understanding and proving subsequent results. The full proofs of these will be covered in EECS 16B.

6.3 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Find the inverse of $\begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & -1 \\ 0 & -1 & 0 \end{bmatrix}$.
2. Find the inverse of $\begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & -1 \\ 0 & -1 & -1 \end{bmatrix}$.
3. Suppose $\mathbf{A} = \mathbf{BC}$, where $\mathbf{B}$ is a 4×2 matrix and $\mathbf{C}$ is a 2×4 matrix. Is $\mathbf{A}$ invertible?
   (a) Yes, $\mathbf{A}$ is invertible.
   (b) No, $\mathbf{A}$ is not invertible.
   (c) Depends on $\mathbf{C}$ only.
   (d) Depends on $\mathbf{B}$ and $\mathbf{C}$.
4. Let the matrix $\mathbf{A}$ be the state transition matrix for some system. Given some state after n steps $\vec{x}[n]$, can we always find $\vec{x}[n+1]$?
   (a) Yes, we simply apply the matrix $\mathbf{A}$ on $\vec{x}[n]$.
   (b) No, we need to know the intial state $\vec{x}[0]$.
   (c) No, we don't have enough information about the system.
5. Let the matrix $\mathbf{A}$ be the state transition matrix for some system. Given some state after n steps $\vec{x}[n]$, can we always find $\vec{x}[n-1]$?
   (a) Yes, we can use Gaussian elimination to find the initial state.
   (b) Yes, we simply apply the matrix $\mathbf{A}^{-1}$ on $\vec{x}[n]$.
   (c) No, we don't know whether the matrix $\mathbf{A}$ is invertible.

6. Suppose that the state transition matrix for a system is given by $\begin{bmatrix} 1 & 0.5 \\ 0 & 0 \end{bmatrix}$. Given some state after n steps $\vec{x}[n]$, can we find $\vec{x}[n-1]$?

7. True or False: The inverse of a diagonal matrix, where all of the diagonal entries are non-zero, is another diagonal matrix.

8. True or False: If $\mathbf{A}^n = \mathbf{0}$, where $\mathbf{0}$ is the zero matrix, for some $n \in \mathbb{R}$, then $\mathbf{A}$ is not invertible.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 7

7.1 Vector Spaces

A **vector space** $\mathbb{V}$ is a set of vectors and two operators that satisfy the following properties:

- Vector Addition
  - Associative: $\vec{u}+(\vec{v}+\vec{w})=(\vec{u}+\vec{v})+\vec{w}$ for any $\vec{v},\vec{u},\vec{w}\in\mathbb{V}$.
  - Commutative: $\vec{u}+\vec{v}=\vec{v}+\vec{u}$ for any $\vec{v},\vec{u}\in\mathbb{V}$.
  - Additive Identity: There exists an additive identity $\vec{0}\in\mathbb{V}$ such that $\vec{v}+\vec{0}=\vec{v}$ for any $\vec{v}\in\mathbb{V}$.
  - Additive Inverse: For any $\vec{v}\in\mathbb{V}$, there exists $-\vec{v}\in\mathbb{V}$ such that $\vec{v}+(-\vec{v})=\vec{0}$. We call $-\vec{v}$ the additive inverse of $\vec{v}$.
  - Closure under vector addition: For any two vectors $\vec{v},\vec{u}\in\mathbb{V}$, their sum $\vec{v}+\vec{u}$ must also be in $\mathbb{V}$.
- Scalar Multiplication
  - Associative: $\alpha(\beta\vec{v})=(\alpha\beta)\vec{v}$ for any $\vec{v}\in\mathbb{V}$, $\alpha,\beta\in\mathbb{R}$.
  - Multiplicative Identity: There exists $1\in\mathbb{R}$ where $1\cdot\vec{v}=\vec{v}$ for any $\vec{v}\in\mathbb{V}$. We call 1 the multiplicative identity.
  - Distributive in vector addition: $\alpha(\vec{u}+\vec{v})=\alpha\vec{u}+\alpha\vec{v}$ for any $\alpha\in\mathbb{R}$ and $\vec{u},\vec{v}\in\mathbb{V}$.
  - Distributive in scalar addition: $(\alpha+\beta)\vec{v}=\alpha\vec{v}+\beta\vec{v}$ for any $\alpha,\beta\in\mathbb{R}$ and $\vec{v}\in\mathbb{V}$.
  - Closure under scalar multiplication: For any vector $\vec{v}\in\mathbb{V}$ and scalar $\alpha\in\mathbb{R}$, the product $\alpha\vec{v}$ must also be in $\mathbb{V}$.

You have already seen vector spaces before! For example, $\mathbb{R}^n$ is the vector space of all n-dimensional vectors. With the definitions of vector addition and scalar multiplication defined in the previous notes you could show that it satisfies all the properties above. In fact, the set of all matrices the same size is also a vector space $\mathbb{R}^{n\times m}$ since it fulfills all of the properties above as well – but in this class we will generally only deal with vector spaces containing vectors in $\mathbb{R}^n$.

Additional Resources For more on vector spaces, read *Strang* pages 123 - 125 and try Problem Set 3.1.

In *Schaum's*, read pages 112-114 and try problems 4.1, 4.2, and 4.71 to 4.76. *Extra: Read and Understand Polynomial Spaces, Spaces of Arbitrary "Field."*

7.1.1 Bases

We can use a series of vectors to define a vector space. We call this set of vectors a **basis**, which we define formally below:

Definition 7.1 (Basis):

Given a vector space $\mathbb{V}$, a set of vectors $\{\vec{v}_1, \vec{v}_2, ..., \vec{v}_n\}$ is a **basis** of the vector space if it satisfies the following two properties:

- $\vec{v}_1, \vec{v}_2, ..., \vec{v}_n$ are linearly independent vectors
- For any vector $\vec{v} \in \mathbb{V}$, there exist scalars $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{R}$ such that $\vec{v} = \alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + ... + \alpha_n\vec{v}_n$.

Intuitively, a basis of a vector space is the *minimum* set of vectors needed to represent all vectors in the vector space. If a set of vectors is linearly dependent and "spans" the vector space, it is still not a basis because we can remove at least one vector from the set and the resulting set will still span the vector space.

The next natural question to ask is: Given a vector space, is the basis unique? Intuitively, it is not because multiplying one of the vectors in a given basis by a nonzero scalar will not affect the linear independence or span of the vectors. We could alternatively construct another basis by replacing one of the vectors with the sum of itself and any other vector in the set.

To illustrate this mathematically, suppose $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ is a basis for the vector space we are considering. Then

$$\{\alpha\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\} \tag{1}$$

where $\alpha \neq 0$ is also a basis because, just as we've seen in Gaussian elimination row operations, multiplying a row by a nonzero constant does not change the linear independence or dependence of the rows. We can generalize this to say that multiplying a vector by a nonzero scalar also does not change the linear independence of the set of vectors. In addition, we know that

$$span(\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}) = span(\{\alpha\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}). \tag{2}$$

because any vector in $span(\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\})$ can be created as a linear combination of the set $\{\alpha\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ by dividing the scale factor on $\vec{v}_1$ by α. We can use a similar argument to show that $\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\}$ is also a basis for the same vector space.

Example 7.1 (Vector space $\mathbb{R}^3$): Let's try to find a basis for the vector space $\mathbb{R}^3$. We want to find a set of vectors that can represent any vector of the form $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ where $a, b, c \in \mathbb{R}$. One basis could be the set of standard unit vectors:

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

The set of vectors is linearly independent and we can represent any vector $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ in the vector space using the three vectors:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3}$$

Alternatively, we could show that

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\}$$

is a basis for the vector space.

Now that we have defined bases, we can define the dimension of a vector space.

Definition 7.2 (Dimension): The dimension of a vector space is the number of basis vectors.

Since each basis vector can be scaled by one coefficient, the dimension of a space as the fewest number of parameters needed to describe an element or member of that space. The dimension can also be thought of as the degrees of freedom of your space – that is, the number of parameters that can be varied when describing a member of that space.

Example 7.2 (Dimension of $(\mathbb{R}^3, \mathbb{R})$): Previously, we identified a basis

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

for the vector space $(\mathbb{R}^3, \mathbb{R})$. The basis consists of three vectors, so the dimension of the vector space is three.

Note that a vector space can have many bases, but each basis must have the same number of vectors. We will not prove this rigorously, but let's illustrate our arguments. Suppose a basis for the vector space we're considering has n vectors. This means that the minimum number of vectors we can use to represent all vectors in the vector space is n, because the vectors in the basis would not be linearly independent if the vector space could be represented with fewer vectors. Then we can show that any set with less than n vectors cannot be a basis because it does not have enough vectors to span the vector space — there would be some vectors in the vector space that cannot be expressed as a linear combination of the vectors in the set. In addition, we can show that any set with more than n vectors must be linearly dependent and therefore cannot be a basis. Combining the two arguments, we have that any other set of vectors that forms a basis for the vector space must have exactly n vectors.

We introduced quite a few terms in this lecture note, and we'll see how we can connect these with our understanding of matrices in the next lecture note!

Additional Resources For more on bases, read *Strang* pages 167 - 171 and try Problem Set 3.4. *Extra: Read Sections on Matrix and Function Space.*

In *Schaum's*, read pages 124-126 and pages 127-129. Try Problems 4.24 to 4.28, 4.97 to 4.103, and 4.33 to 4.40.

7.2 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. True or False: $\left\{ \begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 5 \\ 2 \end{bmatrix} \right\}$ spans $\mathbb{R}^2$.

2. True or False: $\left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} -3 \\ 6 \\ 5 \end{bmatrix} \right\}$ is a basis for $\mathbb{R}^3$.

3. The following vectors span $\mathbb{R}^3$:
   $\vec{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \vec{x}_2 = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}, \vec{x}_3 = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}, \vec{x}_4 = \begin{bmatrix} 2 \\ 7 \\ 4 \end{bmatrix}, \vec{x}_5 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$
   Which vectors of this set form a basis for $\mathbb{R}^3$?

   (a) $\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5$
   (b) $\vec{x}_1, \vec{x}_3, \vec{x}_5$
   (c) $\vec{x}_1, \vec{x}_2, \vec{x}_4$
   (d) $\vec{x}_1, \vec{x}_3, \vec{x}_4, \vec{x}_5$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 8

8.1 Subspace

In previous lecture notes, we introduced the concept of a vector space and the notion of basis and dimension. In this note, we introduce the idea of subspaces, as it is often useful to look at part of the entire set of vectors in a vector space.

Definition 8.1 (Subspace): A subspace $\mathbb{U}$ consists of a subset of the vector space $\mathbb{V}$ that satisfies the following three properties:

- Contains the zero vector: $\vec{0} \in \mathbb{U}$.
- Closed under vector addition: For any two vectors $\vec{v}_1, \vec{v}_2 \in \mathbb{U}$, their sum $\vec{v}_1 + \vec{v}_2$ must also be in $\mathbb{U}$.
- Closed under scalar multiplication: For any vector $\vec{v} \in \mathbb{U}$ and scalar $\alpha \in \mathbb{R}$, the product $\alpha\vec{v}$ must also be in $\mathbb{U}$.

Equivalently, a subspace is a subset of the vectors in a vector space where any linear combination of the vectors in the subset lies within the subset. Just as basis and dimension are defined for vector spaces, they have equivalent definitions for subspaces. A basis of a subspace is a set of linearly independent vectors that span the subspace, and the dimension of a subspace is the number of vectors in its bases.

In the following sections, we will explore a few key subspaces.

Additional Resources For more on subspaces, read *Strang* pages 125 - 127 and try Problem Set 3.1. In *Schaum's*, read pages 117-119 and try Problems 4.8 to 4.12, and 4.77 to 4.82.

8.2 Column Space

We can think of a matrix as a linear function that acts on vectors. Consider the matrix A in $\mathbb{R}^{n\times m}$ – it takes the vectors that live in $\mathbb{R}^m$ (an m-dimensional space) and outputs vectors that live in $\mathbb{R}^n$ (an n-dimensional space). We say that the **range** of an operator is the space of all outputs that the operator can map to.

What is the range of our matrix operator? To answer this we write our matrix in terms of its columns,

$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \dots & \vec{a}_m \\ | & | & & | \end{bmatrix}, \tag{1}$$

where the vectors $\vec{a}$ live in $\mathbb{R}^n$. The matrix A operates on any vector $\vec{x}$ that lives in $\mathbb{R}^m$, where the operation on $\vec{x}$ is $A\vec{x}$. Members of $\mathbb{R}^m$, $\vec{x}$ can be written as

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}. \tag{2}$$

From this we have

$$A\vec{x} = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \dots & \vec{a}_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}. = \sum_{k=1}^{m} x_k \vec{a}_k, \tag{3}$$

so we can conclude that the range of the operator A is the space of all possible linear combinations of its columns, or the *span* of (the columns of) A, which we can write as

$$span(A) = \{\vec{v} \mid \vec{v} = \sum_{i=1}^{m} x_i \vec{a}_i, \text{where } x_i\text{'s are scalars}\}, \tag{4}$$

also called the **column space** of A. Note that the terms **span** and **range** are also often used interchangeably with the term "column space" - for our purposes, all three terms mean exactly the same thing when used to describe a matrix. We know that $span(A)$ is a subset of $\mathbb{R}^n$. However, is $span(A)$ a subspace? Let's see if it satisfies each condition described in Section 8.1.

- We know that the zero vector is in $span(A)$ because A operating on the zero vector gives the zero vector: $A\vec{0} = \vec{0}$.
- If $\vec{v}_1, \vec{v}_2$ are in $span(A)$, then there exist $\vec{u}_1, \vec{u}_2 \in \mathbb{R}^m$ such that $A\vec{u}_1 = \vec{v}_1$ and $A\vec{u}_2 = \vec{v}_2$. Adding the two equations together, we have (due to the distributivity of matrix-vector multiplication):

  $$\vec{v}_1 + \vec{v}_2 = A\vec{u}_1 + A\vec{u}_2 = A\left(\vec{u}_1 + \vec{u}_2\right). \tag{5}$$

  This tells us the $\vec{v}_1 + \vec{v}_2$ is in $span(A)$ as well.
- If $\vec{v}$ is in $span(A)$, then there exists $\vec{u} \in \mathbb{R}^m$ such that $A\vec{u} = \vec{v}$. For any scalar α, we can write

  $$\alpha\vec{v} = \alpha A\vec{u} = A\left(\alpha\vec{u}\right). \tag{6}$$

  This says that $\alpha\vec{v}$ is also in $span(A)$.

As a result, we can see that $span(A)$ is a subspace.

Additional Resources For more on column space, read *Strang* pages 127 - 129. For additional practice with these ideas, try Problem Set 3.1.

8.2.1 Rank: Dimension of the Range

What is the dimension of $span(A)$? A reasonable guess would be n, since the vectors in our span live in $\mathbb{R}^n$, making $span(A)$ a subset of $\mathbb{R}^n$ (meaning it is contained in $\mathbb{R}^n$). In general, however, the matrix operator A will not be able to produce every vector in $\mathbb{R}^n$, so our range will not equal $\mathbb{R}^n$. The dimension of $span(A)$ cannot be greater than n, since $span(A)$ is a subset of $\mathbb{R}^n$, but it can certainly be less. For example, say that A is a zero matrix. In that case, its output would be zero-dimensional, as it can only output $\vec{0}$. Recall that the dimension of a space is the minimum number of parameters needed to describe a vector in that space. Therefore, if the space only contains one vector (such as $\vec{0}$), no parameters are needed to distinguish that vector from any other vector in that space — there are no other vectors. Hence, the dimension of $span(A)$ where A is a zero matrix is just zero.

Considering the definition in Equation (Eq. 4), we see that only m parameters are chosen: $x_1, x_2, \cdots, x_m$. As a result, the dimension of the span cannot be greater than m, even if m is less than n. How can this be, when the vectors in $span(A)$ each have n components? In defining our span, we have constrained the kinds of vectors that can live in our space. Therefore, we may be able to use fewer parameters than components in each vector to distinguish the vectors in this space.

Given our discussion thus far, we might be tempted to say that the span is $\min(m,n)$ — the minimum of m and n — but this is not completely true. The columns of A may be linearly dependent, meaning that some vectors are actually redundant. Any vector in $span(A)$ can always be represented as a linear combination of the linearly *independent* columns of A. For example, take

$$A = \begin{bmatrix} 2 & 0 & 2 \\ 3 & 2 & 5 \\ 5 & 1 & 6 \\ 2 & 2 & 4 \end{bmatrix} \tag{7}$$

The last column is not linearly independent, as it can be obtained by adding the first two columns. Let us take the following linear combination of the columns of A:

$$2\begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \end{bmatrix} + 3\begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} + 4\begin{bmatrix} 2 \\ 5 \\ 6 \\ 4 \end{bmatrix} \tag{8}$$

You can verify that we can obtain the same result by taking a linear combination of only the first two columns:

$$2\begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \end{bmatrix} + 3\begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} + 4\begin{bmatrix} 2 \\ 5 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 12 \\ 32 \\ 37 \\ 26 \end{bmatrix} = 6\begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \end{bmatrix} + 7\begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix}. \tag{9}$$

More generally, you can verify

$$x_1\begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \end{bmatrix} + x_2\begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} + x_3\begin{bmatrix} 2 \\ 5 \\ 6 \\ 4 \end{bmatrix} = \tilde{x}_1\begin{bmatrix} 2 \\ 3 \\ 5 \\ 2 \end{bmatrix} + \tilde{x}_2\begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix}, \quad \tilde{x}_1 = x_1 + x_3, \text{and } \tilde{x}_2 = x_2 + x_3, \tag{10}$$

Since the last column is the sum of the first two columns, we can add x_3 to x_1 and x_2 to obtain any possible linear combination of all columns in A. Since the dimension is given by the smallest number of parameters needed to identify any element in the space, it turns out that the dimension of $span(A)$ is equal to the number of linearly independent columns of A, which will be less than or equal to $\min(m,n)$.

$$\dim(span(A)) \leq \min(m,n). \tag{11}$$

Definition 8.2 (Rank): The rank of a matrix is the dimension of the span of its columns.

$$rank(A) = \dim(span(A))$$

For example, the rank of the matrix

$$A = \begin{bmatrix} 2 & 0 & 2 \\ 3 & 2 & 5 \\ 5 & 1 & 6 \\ 2 & 2 & 4 \end{bmatrix} \tag{12}$$

defined previously is 2, since it has two linearly independent columns.

8.3 Loss of Dimensionality and Nullspace

In the previous example, we saw that the dimension of the output space can be smaller than the dimension of the input space. In this section, we'll explore where is the "remaining dimensionality" is going – somewhere call the **nullspace**.

Definition 8.3 (Nullspace): The nullspace of A consists of all vectors $\vec{x}$ in $\mathbb{R}^m$ such that $A\vec{x} = \vec{0}$:

$$N(A) = \{\vec{x} \mid A\vec{x} = \vec{0},\ \vec{x} \in \mathbb{R}^m\}. \tag{13}$$

The nullspace of A is the set of vectors that get mapped to zero by A.

What is the dimension of the nullspace? We know that it can be at most m, since all of the input vectors have m components. However, unless A is the zero matrix, not every input gets mapped to zero, so in general the dimension should be less than m. The question we need to ask is how many independent ways can we create the zero vector by taking linear combinations of the columns of A. Recall that

$$A\vec{x} = \sum_{k=1}^{m} x_i \vec{a}_k, \tag{14}$$

where again x_i are the free parameters. So our task is to find vectors $\vec{x}$ such that

$$\sum_{i=1}^{m} x_i \vec{a}_i = \vec{0} \tag{15}$$

First note that the only way $\sum_{i=1}^{m} x_i \vec{a}_i = \vec{0}$ (non-trivially) is if the columns of A are not all linearly independent. This holds by definition of linear independence. We can represent A in terms of its linearly independent

columns and dependent columns, $\vec{a}^i$ and $\vec{a}^d$ respectively. Specifically, we can start with an empty set, and then we add columns of A into the set as long as the set stays linearly independent. Once we have no more columns that we can add, the columns in this set are $\vec{a}^i$ and the rest of the columns of A are $\vec{a}^d$. Of course, this set will be different depending on the order you consider the columns of A. However, the size of the set will always be the same, or equal to the dimension of $span(A)$ (Hint: show that this set is a basis for $span(A)$).

Assuming there are $j < m$ linearly independent columns[1],

$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \dots & \vec{a}_m \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & & | & | & & | \\ \vec{a}^i_1 & \dots & \vec{a}^i_j & \vec{a}^d_1 & \dots & \vec{a}^d_{m-j} \\ | & & | & | & & | \end{bmatrix}. \tag{16}$$

We can then break up the summation in equation (Eq. 15) into two summations one for the linearly independent columns and another for the linearly dependent columns,

$$\sum_{k=1}^{j} x^i_k \vec{a}^i_k + \sum_{k=1}^{m-j} x^d_k \vec{a}^d_k = \vec{0}, \tag{17}$$

where x^i and x^d are the parameters of $\vec{x}$ that multiply the linearly independent and dependent columns of A respectively. Rearranging a bit we get

$$\sum_{k=1}^{j} x^i_k \vec{a}^i_k = -\sum_{k=1}^{m-j} x^d_k \vec{a}^d_k \tag{18}$$

Remember we get to choose the parameters in our vector $\vec{x}$ that will satisfy (Eq. 15). We know that in the total summation at least one linearly dependent vector must be multiplied by a nonzero parameter, since the linearly independent vectors alone cannot be linearly combined (non-trivially) to get $\vec{0}$. With this constraint let us then simplify our problem. We will impose that x^d_1 be nonzero, and set the other parameters multiplying linearly dependent vectors equal to zero. That is $x^d_2 = \ldots = x^d_{m-j} = 0$. Since $\vec{a}^d_1$ is linearly dependent on the $\vec{a}^i_k$'s we know that there exist a unique set of numbers $\beta^1_1, \beta^1_2, \ldots, \beta^1_j$ such that

$$\sum_{k=1}^{j} \beta^1_k \vec{a}_k = \vec{a}^d_1. \tag{19}$$

In other words there is a unique linear combination of our linearly independent vectors that equals $\vec{a}^d_1$. **As an aside, if a vector can be represented as a linear combination of linearly independent vectors then this representation is unique. You can try to prove this. Hint: Assume that two representations exists, set the two representations equal to one another, and see if the linear independence still holds.** Rearranging we get

[1] Please note that the linearly independent columns do not all have to be next to another, we just write it this way to ease the presentation. The results we show will still hold even if the linearly independent columns are not side by side.

$$\sum_{k=1}^{j} -\beta_k^1 \vec{a}_k^i + \vec{a}_1^d = \vec{0}, \tag{20}$$

which is also equal to

$$\sum_{k=1}^{j} -\beta_k^1 \vec{a}_k^i + \vec{a}_1^d + \sum_{k=2}^{m-j} 0\vec{a}_k^d = \vec{0}. \tag{21}$$

Notice that the last summation on the left hand side is equal to zero, and we only include it to more clearly show one of the vectors in the nullspace, namely

$$\vec{x} = \begin{bmatrix} -\beta_1^1 \\ -\beta_2^1 \\ \vdots \\ -\beta_j^1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{22}$$

Can we find others? Well the first thing we can do is multiply equation (Eq. 19) by our free parameter x_1^d,

$$x_1^d (\sum_{k=1}^{j} \beta_k^1 \vec{a}_k) = x_1^d \vec{a}_1^d. \tag{23}$$

Similarly we can conclude

$$\sum_{k=1}^{j} -x_1^d \beta_k^1 \vec{a}_k^i + x_1^d \vec{a}_1^d + \sum_{k=2}^{m-j} 0\vec{a}_k^d = \vec{0}. \tag{24}$$

Since x_1^d is a free parameter any vector of the form

$$\vec{x} = \begin{bmatrix} -\beta_1^1 x_1^d \\ -\beta_2^1 x_1^d \\ \vdots \\ -\beta_j^1 x_1^d \\ x_1^d \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} -\beta_1^1 \\ -\beta_2^1 \\ \vdots \\ -\beta_j^1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_1^d \tag{25}$$

will also be in the nullspace! Are there others? Yes. There is nothing special about choosing the parameter of the first linearly dependent vector to be the nonzero parameter. We can repeat the same procedure for each of the linearly dependent columns, to obtain new vectors in the nullspace. For example say that we set $x_1^d = x_3^d = \ldots = x_{m-j}^d = 0$, and leave x_2^d as our nonzero parameter we will find that

$$\vec{x} = \begin{bmatrix} -\beta_1^2 \\ -\beta_2^2 \\ \vdots \\ -\beta_j^2 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_2^d, \tag{26}$$

is also in the nullspace, where $\sum_{k=1}^{j} \beta_k^2 \vec{a}_k = \vec{a}_2^d$. This procedure can be done for each linearly dependent vector, for example if x_3^d is the nonzero parameter we will get

$$\vec{x} = \begin{bmatrix} -\beta_1^3 \\ -\beta_2^3 \\ \vdots \\ -\beta_j^3 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} x_3^d, \tag{27}$$

Furthermore, we can add vectors together from our nullspace together to get other vectors in the nullspace. **Aside: Try to prove this. Hint: if $\vec{x}_1$ and $\vec{x}_2$ are in the nullspace of A, what can be said about $A(\vec{x}_1 + \vec{x}_2)$?** This means that

$$\vec{x} = \begin{bmatrix} -\beta_1^1 \\ -\beta_2^1 \\ \vdots \\ -\beta_j^1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_1^d + \begin{bmatrix} -\beta_1^2 \\ -\beta_2^2 \\ \vdots \\ -\beta_j^2 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_2^d + \begin{bmatrix} -\beta_1^3 \\ -\beta_2^3 \\ \vdots \\ -\beta_j^3 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} x_3^d + \ldots + \begin{bmatrix} -\beta_1^{m-j} \\ -\beta_2^{m-j} \\ \vdots \\ -\beta_j^{m-j} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} x_{m-j}^d \tag{28}$$

is also in the nullspace. Notice that once we choose the x^d parameters then the x^i parameters are fixed. This

is because the x^d parameters control how much of the linearly dependent columns of A are being put into the summation, and the x^i parameters must ensure that the exact amount of linearly independent columns are included to cancel out the linearly dependent columns so that the output be zero. So we finally conclude that the dimension of the nullspace is equal to the number of our x^d parameters, which is equal to the number of linearly dependent columns of A. We will work out some examples in the next section.

The last point we would like to highlight here is that the dimension of the range of A is equal to the number of linearly independent columns, and the dimension of the nullspace of A is equal to the number of linearly dependent columns. Thus

$$m - \dim(span(A)) = \dim(N(A)), \tag{29}$$

so the loss of dimensionality from the input space to the output space shows up in the nullspace! This result is called the **rank-nullity theorem**.

Additional Resources For more on nullspace and rank, read *Strang* pages 135 - 141 and try Problem Set 3.2.

8.4 Computing the Nullspace

So now we will show you how to compute the nullspace of a matrix systematically. Hopefully our analysis in the previous section will prove to be fruitful. For a vector to be in the nullspace it must weight the linearly independent columns appropriately to cancel out the weighted linearly dependent columns, and what we will show next will find all such vectors that do so.

In solving for the nullspace, we are fundamentally trying to solve the system of equations $A\vec{x} = \vec{0}$. We know that row reducing does not affect the solution of the system of equations, so we will assume that we've already performed Gaussian Elimination until we have an upper triangular matrix. If the matrix is not upper triangular then it can first be row reduced and the techniques will apply. Working with upper triangular matricies will make our lives much easier for two reasons: first, it is really easy to find the linearly independent columns (the columns with pivots), and it is easy to figure out how the linearly independent columns should be combined to cancel out the linearly dependent columns. We will work with the matrix

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

First we identify the linearly dependent columns, which in this case could be columns 2 and 4. To be clear,

the set of linearly dependent columns we chose is

$$\{ \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \} \tag{31}$$

and the set of linearly independent columns includes columns 1, 3, and 5,

$$\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \}. \tag{32}$$

Note that the choice of linearly dependent columns need not be unique. All that is needed is that any vector in the linearly dependent columns we choose can be written as a linear combination of the vectors in the set of linearly independent columns and, of course, the columns in the linearly independent set should be linearly independent.

We would want to find all possible scalars x_1, x_2, x_3, x_4, x_5 such that

$$x_2 \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_4 \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_1 \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_3 \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_5 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \vec{0}. \tag{33}$$

Rather than considering all the linearly dependent vectors at once, we can consider them individually and sum up the contribution from each linearly dependent columns at the end. Let us first impose that the first linearly dependent column must have a weight of one in the summation and the other linearly dependent columns have weights of zero. After this, we will repeat this and only allow the second linearly dependent column to show up in the summation. Let's start with the first linearly dependent column, we want to find the unique weighting of the linearly independent columns so that the resulting sum cancels out the first linearly dependent column. It is easy to see that

$$1 \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2 \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{34}$$

Now, as before, the equality would hold if we multiply both sides of the equation by any scalar $\alpha \in \mathbb{R}$

$$\alpha \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2\alpha \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{35}$$

Note that essentially we can treat α as a free variable that can vary its value however we want while satisfying the above equation. Moving everything to the left hand side, we have

$$\alpha \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + (-2\alpha) \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = 0. \tag{36}$$

Now we see that any vector of the form $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -2\alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \alpha$ is in the nullspace of A. Now let's set the weight of the second linearly dependent column to one and set the weight of the first linearly dependent column to zero. Similarly, we can find the unique weightings of the linearly independent columns that sum to the second linearly dependent column.

$$0 \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 1 \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 3 \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 2 \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{37}$$

The equality would still hold if we multiply both sides of the equation by any scalar $\beta \in \mathbb{R}$,

$$0 \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \beta \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 3\beta \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 2\beta \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{38}$$

Again, moving everything to the left hand side of the equation, we have

$$0 \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \beta \times \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + (-3\beta) \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + (-2\beta) \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \vec{0}. \tag{39}$$

We have that any vector of the form $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -3\beta \\ 0 \\ -2\beta \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ -2 \\ 1 \\ 0 \end{bmatrix} \beta$ is in the nullspace of A. We know that if any two vectors are in the nullspace, then their sum is also in the nullspace. Thus we can conclude that the nullspace of A is

$$N(A) = \{ \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} -3 \\ 0 \\ -2 \\ 1 \\ 0 \end{bmatrix} \beta \mid \alpha, \beta \in \mathbb{R} \}. \tag{40}$$

Notice that the dimension of the nullspace is 2 and the dimension of the range is 3.

8.5 Practice Problems

These practice problems are also available in an interactive form on the course website (http://inst.eecs.berkeley.edu/ ee16a/sp19/hw-practice).

1. Let $\vec{v}_1$ and $\vec{v}_2$ be two vectors in a set W. Suppose we know that $\vec{v}_1 + \vec{v}_2$ is not in W. Is W a subspace?

2. Performing Gaussian elimination on $\begin{bmatrix} 1 & -2 & 4 & 3 \\ -1 & 2 & 1 & 2 \end{bmatrix}$ gives $\begin{bmatrix} 1 & -2 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. Find a basis for $\text{Col}(\mathbf{A})$.

   (a) $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix} \right\}$

   (b) $\left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right\}$

   (c) $\left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 4 \\ 1 \end{bmatrix} \right\}$

   (d) $\left\{ \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right\}$

3. True or False: If an $m \times n$ matrix has pivots in every row, then $\text{Col}(\mathbf{A}) = \mathbb{R}^m$.

4. Suppose $\mathbf{A}$ is an $m \times n$ matrix. What is the largest possible dimension of its null space, and what is the largest possible dimension of its column space?

   (a) Null space: m, Column space: m

   (b) Null space: n, Column space: n

   (c) Null space: n, Column space: $\min(m,n)$

   (d) Null space: m, Column space: $\max(m,n)$

5. Given the matrix $\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 & 3 \\ -1 & 2 & 0 & 1 \\ 1 & -2 & 0 & -1 \\ 3 & 5 & 0 & 8 \end{bmatrix}$, find the dimension of the column space.

6. True or false: A square matrix in $\mathbb{R}^{n \times n}$ is invertible if and only its rank is equal to n.

7. True or False: If $\mathbf{A}$ is a square matrix, then $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^2)$.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 9

9.1 Page Rank: Eigenvalues and Eigenvectors in Action

Google's Page Rank algorithm uses a web crawler to estimate the popularity of webpages. We assume that the more popular a website, the more incoming links it has. We use the following setup to model the problem: suppose we have a large population of web surfers scattered at random around the internet. At each time step, all surfers randomly choose an outgoing link on their current page to arrive at a new webpage. After a large number of time steps, we expect the number of viewers on each page to reflect that page's popularity.

It's important to realize that this is a simplified model that leaves out certain characteristics of the real world: for example, people do not click links at random and people often seek out entirely new pages rather than clicking on outgoing links. Page Rank as it's used in practice, while still a simplified model, has many more intricacies than we cover here. However, even this first order approximation is a very powerful tool.

Let's look at an internet consisting of four webpages:

How can we calculate the number of viewers on each page after infinite time steps? In this note, we will introduce three important new concepts to help us do this calculation: **Determinants**, **Eigenvalues**, and **Eigenvectors**.

Let's say we started off with 6 people on each webpage. Then one time step ticks and everyone randomly selects an outgoing link to visit. How many people do we expect to be on each webpage now?

Page 1: Page 3 only links to page 1, so all 6 people currently on 3 will move to 1. Page 4 links to pages 1 and 3, so on average about half the people on page 4 will go to page 1. There are no other incoming links, so we expect page 1 to have about $6 \times 1 + 6 \times \frac{1}{2} = 9$ people in the next time step.

Page 2: Page 1 is the only page that links to page 2. We expect a third of people from page 1 to move to page 2, so we expect page 2 to have $6 \times \frac{1}{3} = 2$ people in the next time step.

Page 3: We can arrive at page 3 from all the other pages. At page 1, the fraction of the people who move to page 3 is $\frac{1}{3}$, at page 2 the fraction of the people who move to page 3 is $\frac{1}{2}$, and at page 4 the fraction of the people who move to page 3 is $\frac{1}{2}$. Hence, we expect $6 \times \frac{1}{3} + 6 \times \frac{1}{2} + 6 \times \frac{1}{2} = 8$ people to be on page 3 in the next time step.

Page 4: We can arrive at page 4 from either page 1 or page 2. At page 1, the fraction of people who move to page 4 is $\frac{1}{3}$; at page 2, the fraction of people who move to page 4 is $\frac{1}{2}$. We expect $6 \times \frac{1}{3} + 6 \times \frac{1}{2} = 5$ people to be on page 4 in the next time step.

Notice that to find the number of people we expect to be on page i, we added up the number of people we expected would move from every other page to page i. In general, if we have n pages, $x_i(k)$ is the number of people on page i at time step k, and $p(x,y)$ is the fraction of people that will jump from page x to page y, then

$$x_i(k+1) \;=\; \sum_{j=1}^{n} x_j(k)\, p(j,i)$$

If we want to compute the expected number of viewers at time $k+1$ for all pages simultaneously, we can use matrix notation as we did with the pump and reservoir system we looked at earlier. Define $\vec{x}(k) = \begin{bmatrix} x_1(k) & x_2(k) & \dots & x_n(k) \end{bmatrix}^T$ as the vector encoding the number of viewers on each page at time k. Then

$$\vec{x}(k+1) \;=\; \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} \;=\; \begin{bmatrix} p(1,1) & p(2,1) & \dots & p(n,1) \\ p(1,2) & p(2,2) & \dots & p(n,2) \\ \vdots & \vdots & \vdots & \vdots \\ p(1,n) & p(2,n) & \dots & p(n,n) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

We call the matrix of fractions P. Check that the entry $x_i(k+1)$ does in fact match the summation above. For our specific example, the matrix equation looks like this:

$$\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix} \;=\; \begin{bmatrix} 9 \\ 2 \\ 8 \\ 5 \end{bmatrix}$$

If we start with initial counts $\vec{x}(0)$ and want to find the expected number of viewers on each page at time k, we compute

$$\vec{x}(k) \;=\; P^k \vec{x}(0) \;=\; \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}^k \vec{x}(0)$$

Since we only care about the relative popularity of each webpage, we can think of $\vec{x}(k)$ as fractions of viewers on each page (instead of the total number of viewers). We assume the viewers are initially distributed equally on all webpages, and we initialize $\vec{x}(0)$ as $\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$. After running the simulation for many time steps,

we hope $\vec{x}(k)$ will reflect the popularity of the webpages. This system is only useful to us if the values of $\vec{x}(k)$ *converge* to some stable fractions, i.e. the fraction of people on each webpage does not change at every timestep.

Once our fractions exactly match these "stable fractions", running the simulation for another time step should not change the values. This means that if the fractions converge, at some point we'll reach fractions $\vec{x}^*$ such that

$$\vec{x}^* = P\vec{x}^*$$

9.2 Eigenvectors and Eigenvalues

In our Page Rank example, $\vec{x}^*$ is an example of an *eigenvector* of P. But eigenvectors have a more general definition:

Definition 9.1 (Eigenvectors and Eigenvalues): Consider a square matrix $A \in \mathbb{R}^{n\times n}$. An **eigenvector** of A is a *nonzero* vector $\vec{x} \in \mathbb{R}^n$ such that

$$A\vec{x} = \lambda\vec{x}$$

where λ is a scalar value, called the **eigenvalue** of $\vec{x}$.

How do we solve this equation for both $\vec{x}$ and λ, knowing only A? It seems we don't have enough information. We start by bringing everything over to one side:

$$A\vec{x} - \lambda\vec{x} = \vec{0}$$

We would like to factor out $\vec{x}$ on the left side, but currently the dimensions don't agree. $\vec{x}$ is an $n \times 1$ vector, while A is an $n \times n$ matrix. To fix this, we replace $\lambda\vec{x}$ with $\lambda I_n\vec{x}$, where I_n is the $n \times n$ identity matrix:

$$(A - \lambda I_n)\vec{x} = \vec{0} \tag{1}$$

Remember, in the definition of an eigenvector, we explicitly excluded the $\vec{0}$ vector. We know at least one element of $\vec{x}$ is nonzero, yet $(A - \lambda I_n)\vec{x} = \vec{0}$. To see what this means, let's define the columns of $A - \lambda I_n$ to be the vectors $v_1 \dots v_n$ and rewrite the product in terms of the columns:

$$\begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} | \\ \vec{v}_1 \\ | \end{bmatrix} x_1 + \begin{bmatrix} | \\ \vec{v}_2 \\ | \end{bmatrix} x_2 + \cdots + \begin{bmatrix} | \\ \vec{v}_n \\ | \end{bmatrix} x_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

From this formulation, we can see that some nonzero linear combination of the columns of $A - \lambda I_n$ results in $\vec{0}$. This means that the columns of this matrix must be linearly dependent. Now we can rephrase our problem: For what values of λwill $A - \lambda I_n$ have linearly dependent columns? To help us answer this question, we introduce the determinant.

Additional Resources For more on eigenvalues and eigenvectors, read *Strang* pages 288 - 291 or read *Schuam's* pages 296-299.

9.3 Determinants

Every square matrix has a quantity associated with it known as the *determinant*. This quantity encodes many important properties of the matrix, and it is also intimately connected to its eigenvalues.

For this class, we only need to know how to compute the determinant of a 2×2 matrix by hand. This is given by

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

.

There is a beautiful recursive definition for determinants of general $n \times n$ matrices which is outside the scope of this class, but you can read the Wikipedia article on determinants if you want to learn more about it.

Suppose we have a $n \times n$ square matrix $A - \lambda I_n$ and a non-zero vector $x \in \mathbb{R}^n$. Recall from the previous section that the columns of $A - \lambda I_n$ are linearly dependent. We will show that if a matrix's columns are linearly dependent, then its determinant is zero.

Let's consider the geometric connection between the column vectors of a matrix and its determinant. For simplicity, first consider a 3×3 matrix. Let its column vectors be a_1, a_2 and a_3. Then consider the parallelepiped formed by these vectors: [1]

We define the determinant of this matrix as the volume of this parallelepiped formed by its column vectors. For a 2×2 matrix, its determinant is the area of the parallelogram generated by its 2 column vectors.

[1] Modified from image by Claudio Rocchini, `https://en.wikipedia.org/wiki/Determinant`

Similarily the determinant of a 4×4 matrix is the 4-dimensional volume formed by its 4 column vectors. You can check the formula for the determinant of a 2×2 matrix by computing the area of the parallelogram below using geometry and then comparing that to the value given by the determinant formula.

$$\det\left(\begin{bmatrix} a_1 & a_2 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 4 & 5 \\ 3 & 12 \end{bmatrix}\right) = 48 - 3 \times 5 = 33$$

So now back to our proof, so what happens to the determinant when the column vectors of the matrix $det(A - \lambda I_n) = 0$ are linearly dependent? Remember from the previous notes, this means that either one vector lies in the plane formed by the other two, or all three lie on the same line! In either case, the parallelepiped will be "compressed" into a plane or line, with zero volume. Thus the determinant (volume of the parallelepiped) has to be zero.

Additional Resources For more on determinants, read *Strang* pages 247 - 253 and pages 254 to 257, and try Problem Set 5.1.
In *Schaum's*, read pages 264-265 and pages 265-266, and try Problems 8.1, 8.38, 8.39, 8.2, and 8.40 to 8.43.

9.4 Computing Eigenvalues with Determinants

Now we can go back to finding eigenvalues and eigenvectors. Recall that we are trying to solve the equation

$$(A - \lambda I_n)\vec{x} = \vec{0}$$

Since we are only interested in solutions where $\vec{x} \neq \vec{0}$, we want to find values of λ such that $(A - \lambda I_n)$ has linearly dependent columns. We just learned that a matrix with linearly dependent columns has determinant equal to zero:

$$(A - \lambda I_n)\vec{x} = \vec{0} \implies \det(A - \lambda I_n) = 0 \tag{2}$$

Now, let's use this to find the eigenvalues of a matrix. Consider the example 2×2 matrix

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$$

We now expand the expression above:

$$A - \lambda I_n = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 2 \\ 4 & 3-\lambda \end{bmatrix}$$

We now take the determinant of this matrix and set it equal to 0:

$$(1-\lambda)(3-\lambda) - 4 \times 2 = 0$$

Expanding this expression, we get

$$\lambda^2 - 4\lambda - 5 = (\lambda+1)(\lambda-5) = 0$$

We find that there are two solutions to this equation, and therefore two eigenvalues: $\lambda = -1, \lambda = 5$. Each eigenvalue will have its own corresponding eigenvector. To find these, we plug each value of λ into the Equation 2 and solve for $\vec{x}$. Starting with $\lambda = 5$:

$$(A - 5I_2)\vec{x} = \begin{bmatrix} -4 & 2 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \vec{0}$$

We see that both of the rows provide redundant information: $4x_1 - 2x_2 = 0$. The eigenvectors associated with $\lambda = 5$ are all of the form

$$\alpha \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha \in \mathbb{R}$$

Next, we will find the eigenvector associated with $\lambda = -1$. We plug this value of λ into the Equation 2 and solve:

$$(A + I_2)\vec{x} = \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \vec{0}$$

Both rows provide the information $x_1 = -x_2$. The eigenvectors associated with $\lambda = -1$ are of the form

$$\alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \alpha \in \mathbb{R}$$

In summary, the matrix $A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$ has eigenvalues $\lambda = 5, -1$ with associated eigenvectors $\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. We

can check by multiplying the eignenvectors with A and seeing that they get scaled by their corresponding eigenvalues.

$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix} = 5 \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = - \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Additional Resources For more on computing eigenvalues, read *Strang* pages 292 - 296. For additional practice, try Problem Set 6.1.

9.5 Repeated and complex eigenvalues

Every 2×2 matrix has two eigenvalues, but they do not have to be real or unique!

Repeated Eigenvalues:

For a 2×2 matrix, it's possible that the two eigenvalues that you end up with have the same value, leading to a phenomenon called a *Repeated Eigenvalues*. This repeated eigenvalue can one or two associated eigenvectors (unlike a single, unrepeated eigenvalue, which will only have one associated eigenvector). If there are two eigenvectors, they form an *eigenspace*, which is the space of all vectors $\vec{v}$ for which $A\vec{v} = \lambda\vec{v}$.

For example, the following matrix has a repeated eigenvalue of λ.

$$A = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

The *eigenspace* of this matrix is all of $\mathbb{R}^2$ since for any vector $\vec{v} \in \mathbb{R}^2$, $A\vec{v} = \lambda\vec{v}$.

Complex Eigenvalues:

Sometimes when we solve $\det(A - \lambda I_n) = 0$, there will be no real solutions to λ. Consider the transformation that rotates any vector by θ, about the origin:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

There are certain specific cases when R has real eigenvalues: When $\theta = 0$, the rotation matrix reduces to the identity matrix, which does not change change vectors, and therefore has eigenvalue $\lambda = 1$. When $\theta = \pi$, the rotation matrix reduces to the negative identity matrix, which changes the sign of vectors and therefore has eigenvalue $\lambda = -1$. In both these cases, all vectors in $\mathbb{R}^2$ are eigenvectors, so $\mathbb{R}^2$ is the eigenspace.

However, when θ does not correspond to 0° or 180° rotation, there are no vectors that are scaled versions of themselves after the transformation. This will results in complex eigenvalues. For example, let's look at

45° rotation:

$$R = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}$$

$$\det(R-\lambda I) = \frac{1}{2}(1-\lambda)(1-\lambda)+\frac{1}{2}$$

Setting this determinant equal to zero and solving yields the complex eigenvalues, $\lambda = \frac{1}{\sqrt{2}}(1+i)$ and $\lambda = \frac{1}{\sqrt{2}}(1-i)$, which makes sense because there are no physical (real) eigenvectors for this transformation.

9.6 Eigenvectors corresponding to distinct eigenvalues are linearly independent.

Let's explore how the eigenvalues of a 2×2 matrix behave in the most general case. For an arbitrary 2×2 matrix A, we know that its eigenvalues λ will satisfy the equation $\det(A-\lambda I)=0$. Letting

$$A = \begin{bmatrix}a & b\\ c & d\end{bmatrix},$$

where a, b, c, and d are real scalars, we find that

$$\det(A-\lambda I) = \det\left(\begin{bmatrix}a-\lambda & b\\ c & d-\lambda\end{bmatrix}\right) = (a-\lambda)(d-\lambda)-bc = 0.$$

Rearranging, we obtain the quadratic equation

$$\lambda^2-(a+d)\lambda+(ad-bc)=0.$$

The polynomial on the left hand side of the above equation is known as the **characteristic polynomial** for the matrix A.

Since the above quadratic equation has real coefficients, we know that there are three possible cases to consider:

1. There are two, real, distinct eigenvalues λ_1 and λ_2 that satisfy the equation.
2. There is a single (repeated) eigenvalue of λ that satisfies the equation.
3. There are two, complex, distinct eigenvalues λ_1 and λ_2 that satisfy the equation.

We won't spend too much time in this course studying the case of complex eigenvalues, so we won't discuss the third case further.

9.6.1 Case 1: Two distinct real eigenvalues

Let's consider the case when we have two distinct, real eigenvalues λ_1 and λ_2 of our matrix A, such that for any $i\in\{1,2\}$, $\det(A-\lambda_i I)=0$. Since $\det(A-\lambda_1 I)=0$, we know that $A-\lambda_1 I$ has a nontrivial nullspace, so

there exists a vector $\vec{v}_1 \neq \vec{0}$ such that

$$(A - \lambda I)\vec{v}_1 = \vec{0} \implies A\vec{v}_1 = \lambda_1 \vec{v}_1.$$

As discussed earlier, $\vec{v}_1$ is an eigenvector of A, corresponding to the eigenvalue λ_1. In a similar manner, there will exist an eigenvector $\vec{v}_2$ corresponding to the eigenvalue λ_2, so $A\vec{v}_2 = \lambda_2 \vec{v}_2$.

We will now pose the following question: are $\vec{v}_1$ and $\vec{v}_2$ related in any way? As it turns out, we can show that $\vec{v}_1$ and $\vec{v}_2$ are *linearly independent*, so we cannot write either vector as a linear combination of the other.

Theorem 9.1: Given two eigenvectors $\vec{v}_1$ and $\vec{v}_2$ corresponding to two different eigenvalues λ_1 and λ_2 of a matrix A, it is always the case that $\vec{v}_1$ and $\vec{v}_2$ are linearly independent.

Proof. We will prove this claim by contradiction. Imagine that $\vec{v}_1$ and $\vec{v}_2$ were linearly dependent, so we could express one of them (say $\vec{v}_1$, without loss of generality) as a scalar multiple of the other. Expressed algebraically, we could write $\vec{v}_1 = \alpha \vec{v}_2$ for a suitable scalar α.

Now, since $\vec{v}_1$ and $\vec{v}_2$ are eigenvectors with known eigenvalues, we know by definition that

$$\begin{aligned} A\vec{v}_1 &= \lambda_1 \vec{v}_1 \\ A\vec{v}_2 &= \lambda_2 \vec{v}_2, \end{aligned}$$

where $\lambda_1 \neq \lambda_2$. Substituting in our expression for $\vec{v}_1$ into the first of the above equations, we find that

$$\begin{aligned} & A(\alpha \vec{v}_2) = \lambda_1 (\alpha \vec{v}_2) \\ \implies \quad & \alpha(A\vec{v}_2 - \lambda_1 \vec{v}_2) = \vec{0}. \end{aligned}$$

Since eigenvectors are, by definition, nonzero, we know that $\vec{v}_1 = \alpha \vec{v}_2 \neq 0$, so $\alpha \neq 0$. Thus, we can divide the above equation by α and rearrange to obtain

$$A\vec{v}_2 = \lambda_1 \vec{v}_2.$$

But since $\vec{v}_2$ is an eigenvector with eigenvalue λ_2, we know that

$$A\vec{v}_2 = \lambda_2 \vec{v}_2.$$

Equating the two right hand sides and rearranging, we obtain

$$\lambda_1 \vec{v}_2 = \lambda_2 \vec{v}_2 \implies (\lambda_1 - \lambda_2)\vec{v}_2 = \vec{0}.$$

Finally, since we said that the eigenvalues were distinct, so $\lambda_1 \neq \lambda_2$, it is clear that $\lambda_1 - \lambda_2 \neq 0$. Dividing by this nonzero scalar, we find that

$$\vec{v}_2 = \vec{0}.$$

But $\vec{v}_2$, being an eigenvector, must be nonzero, so this is a contradiction! Our initial hypothesis, that $\vec{v}_1$ and $\vec{v}_2$ were linearly dependent, must therefore be wrong, so the two eigenvectors are in fact linearly independent! This completes the proof. $\square$

9.6.2 Linear independence of eigenvectors with distinct eigenvalues

More generally, it turns out that all eigenvectors with different eigenvalues are linearly independent! More formally, we claim:

Theorem 9.2: Let $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m$ be eigenvectors of an $n \times n$ matrix with distinct eigenvalues. It is the case that all the $\vec{v}_i$ are linearly independent from one another. The proof of this theorem is out of scope, but is presented below anyway just for reference for those who are interested.

Proof. This proof will be done by induction over m. To establish some notation, let our $n \times n$ matrix be A, and let our distinct eigenvalues be λ_i, so $A\vec{v}_i = \lambda_i \vec{v}_i$ for all integer $1 \leq i \leq m$.

Now, assume for the sake of induction that the first $k < m$ of our eigenvectors (i.e. $\vec{v}_1, \ldots, \vec{v}_k$) are linearly independent. We will aim to show that the first $k+1$ eigenvectors are also linearly independent. To do so, we will do a proof by contradiction. Imagine, for the sake of contradiction, that the first $k+1$ eigenvectors are in fact linearly dependent, so there exist coefficients $\alpha_1, \ldots, \alpha_{k+1}$ (not all of which are zero) such that

$$\alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \ldots + \alpha_k \vec{v}_k + \alpha_{k+1} \vec{v}_{k+1} = \vec{0}.$$

If $\alpha_{k+1} = 0$, then the first k eigenvectors would be linearly dependent, which we know from our inductive hypothesis is not the case. Thus, $\alpha_{k+1} \neq 0$, so we can rearrange the above equation to obtain

$$\vec{v}_{k+1} = -\frac{\alpha_1}{\alpha_{k+1}} \vec{v}_1 - -\frac{\alpha_2}{\alpha_{k+1}} \vec{v}_2 - \ldots - \frac{\alpha_k}{\alpha_{k+1}} \vec{v}_k.$$

For convenience, let $\beta_i = -\alpha_i / \alpha_{k+1}$, so

$$\vec{v}_{k+1} = \beta_1 \vec{v}_1 + \beta_2 \vec{v}_2 + \ldots + \beta_k \vec{v}_k.$$

Notice that, since at least one of the α_i is nonzero, at least one of the β_i is also nonzero.

Now, since $\vec{v}_{k+1}$ is an eigenvector, we have that

$$\begin{aligned} & A\vec{v}_{k+1} = \lambda_{k+1} \vec{v}_{k+1} \\ \implies \quad & A(\beta_1 \vec{v}_1 + \beta_2 \vec{v}_2 + \ldots + \beta_k \vec{v}_k) = \lambda_{k+1}(\beta_1 \vec{v}_1 + \beta_2 \vec{v}_2 + \ldots + \beta_k \vec{v}_k) \\ \implies \quad & \beta_1 (A\vec{v}_1) + \beta_2 (A\vec{v}_2) + \ldots + \beta_k (A\vec{v}_k) = \beta_1 \lambda_{k+1} \vec{v}_1 + \beta_2 \lambda_{k+1} \vec{v}_2 + \ldots + \beta_k \lambda_{k+1} \vec{v}_k. \end{aligned}$$

But since $\vec{v}_1$ through $\vec{v}_k$ are also eigenvectors where $A\vec{v}_i = \lambda_i \vec{v}_i$, we can substitute to obtain

$$\beta_1 \lambda_1 \vec{v}_1 + \beta_2 \lambda_2 \vec{v}_2 + \ldots + \beta_k \lambda_k \vec{v}_k = \beta_1 \lambda_{k+1} \vec{v}_1 + \beta_2 \lambda_{k+1} \vec{v}_2 + \ldots + \beta_k \lambda_{k+1} \vec{v}_k.$$

Rearranging to pull all the terms on one side, we find that

$$\beta_1 (\lambda_1 - \lambda_{k+1}) \vec{v}_1 + \beta_2 (\lambda_2 - \lambda_{k+1}) \vec{v}_2 + \ldots + \beta_k (\lambda_k - \lambda_{k+1}) \vec{v}_k = \vec{0}.$$

For convenience, let

$$\gamma_i = \beta_i (\lambda_i - \lambda_{k+1}),$$

so

$$\gamma_1 \vec{v}_1 + \gamma_2 \vec{v}_2 + \ldots + \gamma_k \vec{v}_k = \vec{0}.$$

Since all the eigenvalues λ_i are distinct, all the $\lambda_i - \lambda_{k+1} \neq 0$ terms in the above equation are nonzero. Since we showed earlier that at least one of the β_i is nonzero, it is clear that at least one of the γ_i is also nonzero. Thus, we have produced a nontrivial linear combination of the first k eigenvectors that equals zero, so the first k eigenvectors are linearly dependent. But this contradicts our inductive hypothesis, so our initial assumption (that the first $k+1$ eigenvectors were linearly dependent) must be false.

Thus, we have shown that, if at least the first $k < m$ eigenvectors are linearly independent, that the first $k+1$ eigenvectors will also be linearly independent. This will be our induction step.

Clearly, the set containing just the first eigenvector will be linearly independent, since all eigenvectors (including the first) are nonzero. Thus, by induction using the above induction step, we can show that the first m eigenvectors will all be linearly independent, which completes our proof! $\square$

9.6.3 Case 2: A repeated real eigenvalue

In the first case, when we have a single eigenvalue, we know that the corresponding eigenspace may be either one- or two-dimensional. For instance, for

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

we can see that the single eigenvalue is $\lambda = 1$, and the eigenspace is the two dimensional space $\mathbb{R}^2 = \text{span}\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$.

However, for matrices such as

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

we see that the characteristic polynomial

$$\det(A - \lambda I) = \det\left(\begin{bmatrix} -\lambda & 1 \\ 0 & -\lambda \end{bmatrix} \right) = \lambda^2$$

yields the single, repeated eigenvalue $\lambda = 0$. However, computing the associated eigenspace $\text{Null}(A - 0I) = \text{Null}(A)$ yields the one-dimensional eigenspace

$$\text{span}\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$$

even though the eigenvalue is repeated. In such a case, when the dimension of an eigenspace is less than the number of times the eigenvalue is repeated (called the *multiplicity* of the eigenvalue), we say that the matrix is *defective*. The concept of defective matrices is out of scope for the course, but will be discussed more in EECS 16B. That being said, it is often useful to be aware of their existence, since they can often serve as easy counter-examples if you're not sure if a theorem is true.

9.7 Examples

In Section 9.4, we saw an example of how to compute the eigenvalues and eigenvectors of a matrix. Here, we'll look at another example where we do the opposite – finding a matrix given it's eigenvalues and eigenvectors. Finally, we'll go back to our page rank example and rank the popularity of webpages by finding the appropriate eigenvector.

Example 9.1 (Finding a matrix from its eigenvalues and eigenvectors): Now we turn to the case where we try to derive the matrix from the provided vectors. This is essentially reverse engineering the previous problem and reworking the derivation of the eigenvalues.

We would like to solve the problem: what is the matrix A that reflects any $2D$ vectors over the vector $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$?

Below is a graphical representation of that vector:

Remember that an eigenvector of a matrix is any vector that results in a scaled version of itself after being transformed by the matrix. Can you think of any eigenvectors of this reflection transformation? First we observe that anything lying on the line of reflection will be unchanged, in other words, scaled by one. Therefore, we can write the eigenvector, eigenvalue pair:

$$\lambda_1 = 1$$

$$\vec{v}_1 = \alpha \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

In addition, any vector lying perpendicular to the line of reflection will be transformed to point in the *opposite* direction, in other words, scaled by -1. This results in another eigenvector, eigenvalue pair:

$$\lambda_2 = -1$$

$$\vec{v}_2 = \alpha \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Now we can use our definition, $A\vec{v} = \lambda\vec{v}$, to set up a system of equations so we can solve for matrix A:

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

We rearrange these linear equations as follows. (*To see these are equivalent, try writing out all four equations in terms of* $a_1 \ldots a_4$.)

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ -2 \end{bmatrix}$$

Finally, we solve the system of equations with Gaussian Elimination to get our transformation matrix, A.

$$A = \begin{bmatrix} .6 & .8 \\ .8 & -.6 \end{bmatrix}$$

Example 9.2 (Page Rank): Let's go back to our page rank example from the beginning of this note. Recall that we have four webpages, with links between them represented by the diagram below:

We can write the tranformation matrix P that tells us how the number of people on each webpage changes over time:

$$\vec{x}(t+1) \;=\; P\vec{x}(t) \;=\; \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \vec{x}(t)$$

where $\vec{x}(t)$ is a vector describing the fraction of people on each page at time t.

We are interested in the *steady state* vector, $\vec{x^*}$ such that

$$\vec{x^*} = P\vec{x^*}.$$

Looking back at our definition of eigenvalues and eigenvectors, we can see that $\vec{x^*}$ is the eigenvector associated with eigenvalue $\lambda = 1$. In this specific case of finding a steady state vector, we already know the eigenvalue, so proceed to finding the associated eigenvector. (More generally, we would use the determinant to find the eigenvalues first).

From Equation 1, we can write that our steady state vector satisfies

$$(P - \lambda I)\vec{x^*} = \vec{0}$$

where $\lambda = 1$.

$$\left(\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right) \vec{x^*} = \vec{0}$$

$$\begin{bmatrix} -1 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & -1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & -1 \end{bmatrix} \vec{x^*} = \vec{0}$$

We solve for $\vec{x^*}$ by finding the null space of the above matrix. After performing Gaussian elimination on the matrix above, we get the following row equivalent system of equations:

$$\begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -\frac{2}{3} \\ 0 & 0 & 1 & -\frac{3}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x^*} = \vec{0}$$

Remember that each row represents an equation:

$$x_1 - 2x_4 = 0$$
$$x_2 - \frac{2}{3}x_4 = 0$$
$$x_3 - \frac{3}{2}x_4 = 0$$

We can represent this set of solutions as a vector where x_4 is a free variable, $x_4 \in \mathbb{R}$.

$$\vec{x}^* = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2x_4 \\ \frac{2}{3}x_4 \\ \frac{3}{2}x_4 \\ x_4 \end{bmatrix}$$

Since we want, the steady state vector to represent the fraction of people on each page, we can choose x_4 so that the total number of people sums to 1.

$$\vec{x}^* = \frac{1}{31}\begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$$

From our steady state solution, we conclude that webpage 1 is most popular, then webpages 3, 4, and 2.

9.8 Steady States

We know that a steady state $\vec{x}^*$ of a transformation matrix A is defined to be such that

$$A\vec{x}^* = \vec{x}^*.$$

In other words, it is an element of the eigenspace of A corresponding to the eigenvalue $\lambda = 1$. The above equation tells us that if we start at a steady state, then we will remain unaffected by the transformation matrix over time.

However, what happens if we *don't* start at a steady state? Can we say anything about what happens to our state as we keep applying the transformation matrix?

9.8.1 Starting at eigenvectors

We will first look at a simpler problem - what happens if our initial state is an eigenvector $\vec{v}$ of A, but with a real eigenvalue λ that may not equal 1? In other words, $A\vec{v} = \lambda\vec{v}$, but it is possible that $\lambda\vec{v} \neq \vec{v}$. Let's try to see what happens over time, as we keep applying A to our initial state $\vec{x}[0] = \vec{v}$. For the first few timesteps, we see that

$$\begin{aligned} & \vec{x}[0] = \vec{v} \\ \implies \quad & \vec{x}[1] = A\vec{x}[0] = A\vec{v} = \lambda\vec{v} \\ \implies \quad & \vec{x}[2] = A\vec{x}[1] = A(\lambda\vec{v}) = \lambda^2\vec{v} \\ \implies \quad & \vec{x}[3] = A\vec{x}[2] = A(\lambda^2\vec{v}) = \lambda^3\vec{v} \\ & \vdots \end{aligned}$$

We can see, therefore, that after t timesteps, our state will become

$$\vec{x}[t] = \lambda^t \vec{v}.$$

But how does this expression behave as time continues, and $n \to \infty$? We can see that there are a number of cases to consider:

- If $\lambda > 1$, then it can be seen that $\lambda, \lambda^2, \lambda^3, \ldots$ will keep increasing to $+\infty$, so our state $\vec{x}[t]$ will keep growing along the direction of $\vec{v}$ towards infinity.
- If $\lambda = 1$, we're just at the steady state, so $\vec{x}[t] = \vec{v}$ as $t \to \infty$.
- If $0 < \lambda < 1$, then $\lambda, \lambda^2, \lambda^3, \ldots$ will decrease towards zero from above, so our state $\vec{x}[t]$ will keep scaling down and approach $\vec{0}$ in the limit.
- If $\lambda = 0$, then $\vec{x}[1] = 0\vec{v} = \vec{0}$, so our state will immediately drop to $\vec{0}$ from the first timestep onwards.
- If $\lambda < 0$, then behavior similar to the above will occur based on the value of $|\lambda|$, except that the sign of the state will keep flipping at each step. For instance, if $\lambda < 0$ but $|\lambda| > 1$ (in other words, $\lambda < -1$), then the magnitude of the state will tend towards infinity but will keep flipping its sign each time step.

9.8.2 General initial states

Still, we're not done! All we've considered so far are steady states of the form $\vec{x}[0] = \vec{v}$ where $\vec{v}$ is an eigenvector of the transformation matrix A. What if we have an arbitrary initial state, which in general may not be an eigenvector of A?

To deal with this issue, we will make our lives a little easier by considering only matrices whose eigenvectors form a basis (i.e.span all of the state space) — in other words, if A is $n \times n$, then *any* initial state $\vec{x}[0]$ can be written as a linear combination

$$\vec{x}[0] = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \ldots + \alpha_n \vec{v}_n,$$

where $\vec{v}_1$ through $\vec{v}_n$ are linearly independent eigenvectors with eigenvalues λ_1 through λ_n, and the α_i are real scalar coefficients. The general case, when this property is not guaranteed to hold, will be discussed in EECS 16B.

Now, we have previously seen that

$$\vec{x}[t] = A^t \vec{x}[0].$$

We will substitute our linear combination for $\vec{x}[0]$ in terms of the eigenvectors into the above equation, to obtain

$$\begin{aligned}
\vec{x}[t] &= A^t(\alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \ldots + \alpha_n \vec{v}_n) \\
&= \alpha_1(A^t \vec{v}_1) + \alpha_2(A^t \vec{v}_2) + \ldots + \alpha_n(A^t \vec{v}_n) \\
&= \alpha_1(\lambda_1^t \vec{v}_1) + \alpha_2(\lambda_2^t \vec{v}_2) + \ldots + \alpha_n(\lambda_n^t \vec{v}_n).
\end{aligned}$$

Thus, to determine the behavior of an (almost) arbitrary linear system with an arbitrary initial state, all we have to do is write the initial state as a linear combination of the eigenvectors, see how each term of the linear combination behaves as $t \to \infty$, and plug back into the linear combination to determine how the initial state evolves as $t \to \infty$.

We have expressed $\vec{x}[t]$ as a linear combination of the $\lambda_i^t \vec{v}_i$, which each represent what happens to each eigenvector after repeated applications of A. And, from the previous section, we know how this behavior depends on the value of λ_i!

9.8.3 Computing linear combinations

In the above description we wrote out $\vec{x}[0]$ as a linear combination of the eigenvectors. In general, how do we write an arbitrary initial state as a linear combination of n linearly independent eigenvectors $\{\vec{v}_i\}$? We'd like to solve for the coefficients $\{\alpha_i\}$ in the equation

$$\vec{x}[0] = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \ldots + \alpha_n \vec{v}_n.$$

Rewriting it as a matrix multiplication using the linear combination interpretation and stacking the unknown α_i into a vector, we obtain the linear system

$$\begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \vec{x}[0],$$

which we can solve using Gaussian elimination. Alternatively, since the matrix on the left is $n \times n$ with n linearly independent columns, we can pre-multiply by its inverse to express our solution vector as

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{bmatrix}^{-1} \vec{x}[0].$$

We will revisit this next lecture when we talk about the diagonalization.

9.8.4 Criteria for convergence

Now, we have all the tools needed to determine what happens to an initial state after repeated application of a transformation matrix.

Consider an arbitrary initial state $\vec{x}[0]$. We know that

$$\vec{x}[t] = \alpha_1(\lambda_1^t \vec{v}_1) + \alpha_2(\lambda_2^t \vec{v}_2) + \ldots + \alpha_n(\lambda_n^t \vec{v}_n).$$

Essentially, we want to know if $\vec{x}[t]$ will ultimately converge to some fixed value, *no matter* what the $\{\alpha_i\}$ are.

Observe that if a single λ_i is such that $|\lambda_i| > 1$, that $\lambda_i^t \vec{v}_i \to \infty$ in the limit. Thus, our state $\vec{x}[t]$ could potentially go to infinity in the limit. Furthermore, observe that if a single $\lambda_i = -1$, that $\lambda_i^t \vec{v}_i$ will oscillate back and forth in the limit, neither decreasing towards zero or growing to infinity. Thus, our state $\vec{x}[t]$ could potentially oscillate indefinitely, and so fail to converge to a fixed value.

However, if all the λ_i are such that $-1 < \lambda_i \leq 1$, then each term in the linear combination will either go to

zero (if the corresponding $\lambda_i \neq 1$) or stay the same (if $\lambda_i = 1$), meaning that the overall state $\vec{x}[t]$ will always converge to a fixed value, no matter what the initial coefficients are.

Consequently, we have come up with a condition that is sufficient to ensure that any initial state of a transformation matrix will reach a steady state in the limit. But to compute what exactly this steady state is, or to determine if a steady state is reached for a *particular* initial state even if it will not always be reached in the general case, we need to go back to our technique of writing our initial state as a linear combination of eigenvectors and computing the limit of each term in the linear combination.

9.9 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. What are the eigenvalues of the matrix $\begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$?
2. What are the eigenvectors of the matrix $\begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$?
3. True or False: If an $n \times n$ matrix $\mathbf{A}$ is not invertible, then it has an eigenvalue $\lambda = 0$.
4. True or False: If an invertible matrix $\mathbf{A}$ has an eigenvalue λ, then $\mathbf{A}^{-1}$ has the eigenvalue $\frac{1}{\lambda}$.
   (a) Always true.
   (b) False.
   (c) True only if $\lambda \neq 0$.
5. Two students find two different eigenvectors associated with the same eigenvalue of a matrix, $\begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -\frac{1}{2} \\ 2 \end{bmatrix}$. Student One insists that the other must have made an error because every eigenvalue is associated with a unique eigenvector. Student Two insists that they are both valid answers. Who is right, Student One or Student Two?
6. True or False: If for some matrix $\mathbf{A}$, $\mathbf{A}\vec{x}_1 = \lambda_1 \vec{x}_1$ and $\mathbf{A}\vec{x}_1 = \lambda_2 \vec{x}_1$, then $\lambda_1 = \lambda_2$.
7. True or False: A matrix with only real entries can have complex eigenvalues.
8. True or False: If $\mathbf{A}^T$ has an eigenvalue λ, then $\mathbf{A}$ also has the eigenvalue λ.
9. True or False: A diagonal $n \times n$ matrix has n distinct eigenvalues.
10. For a square non-invertible $n \times n$ matrix $\mathbf{A}$, what is the maximum and minimum number of distinct eigenvalues it can have?
    (a) Max: 1, Min: 0
    (b) Max: n, Min: 1

(c) Max: n, Min: $n-1$

(d) Max: $n-1$, Min: 0

(e) Max: $n-1$, Min: 1

11. Find the steady state, if it exists, of the following system assuming that the state vector is $\vec{x}[t] = \begin{bmatrix} x_{\text{Google}}[t] \\ x_{\text{Yahoo}}[t] \\ x_{\text{Bing}}[t] \end{bmatrix}$ and that we start off with $\vec{x}[0] = \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}$.

12. What is the matrix $\mathbf{A}$ that reflects any vector in $\mathbb{R}^2$ about the line through the origin in the direction of $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$?

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 10

10.1 Change of Basis for Vectors

Previously, we have seen that matrices can be interpreted as *linear transformations* between vector spaces. In particular, an $m \times n$ matrix A can be viewed as a function $A : U \to V$ mapping a vector $\vec{u}$ from vector space $U \in \mathbb{R}^n$ to a vector $A\vec{u}$ in vector space $V \in \mathbb{R}^m$. In this note, we explore a different interpretation of square, invertible matrices: as a *change of basis*.

Let's first start with an example. Consider the vector $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$. When we write a vector in this form, implicitly we are representing it in the *standard basis* for $\mathbb{R}^2$, $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. This means that we can write $\vec{u} = 4\vec{e}_1 + 3\vec{e}_2$. Geometrically, $\vec{e}_1$ and $\vec{e}_2$ defines a grid in $\mathbb{R}^2$, and $\vec{u}$ is represented by the coordinates in the grid, as shown in the figure below:

What if we want to represent $\vec{u}$ as a linear combination of another set of basis vectors, say $\vec{a}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\vec{a}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$? This means that we need to find scalars β_1 and β_2 such that $\vec{u} = \beta_1\vec{a}_1 + \beta_2\vec{a}_2$. We can write this equation in matrix form:

$$\begin{bmatrix} | & | \\ \vec{a}_1 & \vec{a}_2 \\ | & | \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

Thus we can find β_1 and β_2 by solving a system of linear equations. Since the inverse of $\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}$ is $\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}$, we get $\beta_1 = 4$ and $\beta_2 = 1$. Then we can write $\vec{u}$ as $4\vec{a}_1 + \vec{a}_2$. Geometrically, $\vec{a}_1$ and $\vec{a}_2$ defines a skewed grid from which the new coordinates are computed.

From this example, we can see that the same vector $\vec{u}$ can be represented in multiple ways! In the standard basis $\vec{e}_1, \vec{e}_2$, the coordinates for $\vec{u}$ are $4, 3$. In the skewed basis $\vec{a}_1, \vec{a}_2$, the coordinates for $\vec{u}$ are $4, 1$. It's the same vector geometrically, but with different coordinates.

In general, suppose we are given a vector $\vec{u} \in \mathbb{R}^n$ in the standard basis and want to change to a different basis with linearly independent basis vectors $\vec{a}_1, \cdots, \vec{a}_n$. If we denote the vector in the new basis as $\vec{u}_a = \begin{bmatrix} u_{a_1} \\ \vdots \\ u_{a_n} \end{bmatrix}$, we solve the following equation $A\vec{u}_a = \vec{u}$, where A is the matrix $\begin{bmatrix} \vec{a}_1 & \cdots & \vec{a}_n \end{bmatrix}$. Therefore the change of basis is given by:

$$\vec{u}_a = A^{-1}\vec{u}$$

If we already have a vector $\vec{u}_a$ in the basis $\vec{a}_1, \cdots, \vec{a}_n$, how do we change it back to a vector $\vec{u}$ in the standard basis? We can reverse the change of basis transformation, thus $\vec{u} = A\vec{u}_a$.

Pictorially, the relationship between any two bases and the standard basis is given by:

$$\text{Basis } \vec{a}_1, \cdots, \vec{a}_n \underset{A^{-1}}{\overset{A}{\rightleftarrows}} \text{Standard Basis} \underset{B}{\overset{B^{-1}}{\rightleftarrows}} \text{Basis } \vec{b}_1, \cdots, \vec{b}_n$$

For example, consider a vector in the basis of $\vec{a}_1, \cdots, \vec{a}_n$:

$$\vec{u} = u_{a_1}\vec{a}_1 + \cdots + u_{a_n}\vec{a}_n = A\vec{u}_a$$

Suppose we would like to represent it as a linear combination of different basis vectors $\vec{b}_1, \cdots, \vec{b}_n$. In other words, we'd like to write it as:

$$\vec{u} = u_{b_1}\vec{b}_1 + \cdots + u_{b_n}\vec{b}_n = B\vec{u}_b$$

Setting these equal gives the following relationship:

$$\begin{aligned} B\vec{u}_b = u_{b_1}\vec{b}_1 + \cdots + u_{b_n}\vec{b}_n = \vec{u} = u_{a_1}\vec{a}_1 + \cdots + u_{a_n}\vec{a}_n = A\vec{u}_a \\ B\vec{u}_b = A\vec{u}_a \\ \vec{u}_b = B^{-1}A\vec{u}_a \end{aligned}$$

Thus the change of basis transformation from basis $\vec{a}_1, \cdots, \vec{a}_n$ to basis $\vec{b}_1, \cdots, \vec{b}_n$ is given by $B^{-1}A$.

10.2 Change of Basis for Linear Transformations

Now that we know how to change the basis of vectors, let's shift our attention to linear transformations. We will answer these questions in this section: how do we change the basis of linear transformations and what does this mean? First let's review linear transformations. Suppose we have a linear transformation T represented by a $n \times n$ matrix that transforms $\vec{u} \in \mathbb{R}^n$ to $v \in \mathbb{R}^n$:

$$\vec{v} = T\vec{u}$$

Although T is represented by a matrix, we can think of it as a geometric transformation on vectors. Implicit in this representation is the choice of a coordinate system. Unless stated otherwise, we always assume that vectors lie in the coordinate system defined by the standard basis vectors $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n$.

But what if our vectors represented in a different basis? Suppose we have basis vectors $\vec{a}_1, \cdots, \vec{a}_n \in \mathbb{R}^n$, and the vectors $\vec{u}$, $\vec{v}$ above are represented in this basis:

$$\begin{aligned} \vec{u} =& u_{a_1}\vec{a}_1 + \cdots + u_{a_n}\vec{a}_n \\ \vec{v} =& v_{a_1}\vec{a}_1 + \cdots + v_{a_n}\vec{a}_n \end{aligned}$$

Can we also represent the transformation T above, such that $\vec{v}_a = T_a\vec{u}_a$? Let's start by defining the matrix A:

$$A = \begin{bmatrix} | & & | \\ \vec{a}_1 & \cdots & \vec{a}_n \\ | & & | \end{bmatrix}$$

Since our vectors $\vec{u}$ and $\vec{v}$ are linear combinations of the basis vectors we can write them as $\vec{u} = A\vec{u}_a$ and $\vec{v} = A\vec{v}_a$. This is exactly the formula defined in the previous section to change a vector back to the standard basis! Now, starting with our transformation T in the standard basis, we can plug in these relationships:

$$\begin{aligned} T\vec{u} &= \vec{v} \\ TA\vec{u}_a &= A\vec{v}_a \\ A^{-1}TA\vec{u}_a &= \vec{v}_a \end{aligned}$$

By pattern matching, we see that if we set $T_a = A^{-1}TA$, we get our desired relationship, $T_a\vec{u}_a = \vec{v}_a$.

The correspondences stated above are all represented in the following diagram:

$$\begin{tikzcd}
\vec{u} \arrow[r, mapsto, "T"] \arrow[d, shift right, "A^{-1}"'] & \vec{v} \arrow[d, shift right, "A^{-1}"] \\
\vec{u}_a \arrow[r, mapsto, "T_a"'] \arrow[u, shift right, "A"'] & \vec{v}_a \arrow[u, shift right, "A"']
\end{tikzcd}$$

For example, there are two ways to get from $\vec{u}_a$ to $\vec{v}_a$. First is the transformation T_a. Second, we can trace out the longer path, applying transformations A, T and A^{-1} in order. This is represented in matrix form as $T_a = A^{-1}TA$ (Recall that matrices act from right to left: the closest matrix to the vector will be applied first.)

By the same logic, we can go in the reverse direction: $T = AT_aA^{-1}$. Let's look at whats happing at each step. First, A^{-1} transforms a vector in the standard basis into the A basis. Then, T_a acts on it, returning a new vector which is also in the A basis. Finally, multiplying by A returns a weighted sum of the columns of A, in other words, transforming the vector back into the original standard basis.

10.3 A Diagonalizing Basis

We just saw that transformations are different in different bases. Is there a special basis under which a transformation attains a particularly nice form? Making our transformation into a diagonal matrix is very useful since diagonal matrices are easily invertible, they can be raised to high powers without a lot of computation, and multiplication between diagonal matrices is commutative.

Suppose that we choose our basis vectors $\vec{a}_1, \cdots, \vec{a}_n$ to be the eigenvectors of the transformation matrix T, which have associated eigenvalues $\lambda_1, \cdots, \lambda_n$. What does $T\vec{u}$ look like now? Recall that $\vec{u}$ can be written in the new basis: $u_{a_1}\vec{a}_1 + \cdots + u_{a_n}\vec{a}_n$.

$$\begin{aligned}
T\vec{u} &= T(u_{a_1}\vec{a}_1 + \cdots + u_{a_n}\vec{a}_n) \\
&= u_{a_1}T\vec{a}_1 + \cdots + u_{a_n}T\vec{a}_n
\end{aligned}$$

By the definition of eigenvectors and eigenvalues:

$$\begin{aligned}
&= u_{a_1}\lambda_1\vec{a}_1 + \cdots + u_{a_n}\lambda_n\vec{a}_n \\
&= \begin{bmatrix} | & \cdots & | \\ \vec{a}_1 & \cdots & \vec{a}_n \\ | & \cdots & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} u_{a_1} \\ \vdots \\ u_{a_n} \end{bmatrix} \\
&= AD\vec{u}_a \\
&= ADA^{-1}\vec{u},
\end{aligned}$$

where D is the diagonal matrix of eigenvalues and A is a matrix with the corresponding eigenvectors as its columns. Thus we have shown that in an eigenvector basis, $T = ADA^{-1}$. In particular, T_a, the counterpart of T in the eigenvector basis, is a diagonal matrix.

How is transforming T into a diagonal matrix useful? Say we want apply our transformation T many times. We might be interested in calculating T^k where k could be quite large. Although you could do this calculation with repeated matrix multiplication, it would quite tedious and even with a computer is slow for large matrices.

However, we just showed that we can write $T = ADA^{-1}$ where D is diagonal. Consider raising this to the power of k:

$$T^k = (ADA^{-1})^k$$

We can expand this out k times and then regroup the terms:

$$\begin{aligned} T^k =& (ADA^{-1})(ADA^{-1})\dots(ADA^{-1})(ADA^{-1}) \\ =& AD(A^{-1}A)D(A^{-1}\dots A)D(A^{-1}A)DA^{-1} \\ =& AD(I)D(I\dots)D(I)DA^{-1} \\ =& ADD\dots DDA^{-1}) \\ =& AD^kA^{-1} \end{aligned}$$

To raise a diagonal matrix to a power, one can raise each element to that power (try proving this to yourself by writing out the multiplication element by element). This is a much simpler calculation that repeatedly doing the full matrix multiplication! In addition, this formulation better allows us to write equations describing how the transformation changes with k – having equations will help us better understand and design systems.

10.4 Diagonalization

Under what circumstances is a matrix diagonalizable? **A $n \times n$ matrix T is diagonalizable if it has n linearly independent eigenvectors**. If it has n linearly independent eigenvectors $\vec{a}_1, \cdots, \vec{a}_n$ with eigenvalues $\lambda_1, \cdots \lambda_n$, then we can write:

$$T = ADA^{-1}$$

Where $A = \begin{bmatrix} \vec{a}_1 & \cdots & \vec{a}_n \end{bmatrix}$ and D is a diagonal matrix of eigenvalues. Note that the eigenvalues must be arranged to match the arrangement of eignevectors in A. For example, the eigenvalue associated with $\vec{a}_1$ should be in the first column on D, the eigenvalue associated with $\vec{a}_2$ should be in the second column on D, and so on.

Example 10.1 (A matrix that is not diagonalizable): Let $T = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$. To find the eigenvalues, we solve the equation:

$$\begin{aligned} \det(T - \lambda I) &= 0 \\ (1-\lambda)^2 &= 0 \\ \lambda &= 1 \end{aligned}$$

The eigenvector corresponding to $\lambda = 1$ is $\vec{a} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Since this matrix only has 1 eigenvector but exists in $\mathbb{R}^{2\times 2}$, it is not diagonalizable.

Example 10.2 (Diagonalization of a 3×3 matrix):

$$T = \begin{bmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{bmatrix}$$

To find the eigenvalues, we solve the equation:

$$\begin{aligned} \det(T - \lambda I) &= 0 \\ \lambda^3 - 6\lambda^2 - 15\lambda - 8 &= 0 \\ (\lambda - 8)(\lambda + 1)^2 &= 0 \\ \lambda &= -1, 8 \end{aligned}$$

If $\lambda = -1$, we need to find $\vec{a}$ such that:

$$(T - (-1)I)\vec{a} = 0 \quad \Longrightarrow \quad \begin{bmatrix} 4 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 4 \end{bmatrix} \vec{a} = 0 \quad \Longrightarrow \quad \begin{bmatrix} 2 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{a} = 0$$

Thus the dimension of the nullspace of $T - (-1)I$ is 2, and we can find two linearly independent vectors in this basis:

$$\vec{a}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad \vec{a}_2 = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

If $\lambda = 8$, we need to find $\vec{a}$ such that:

$$(T - (8)I)\vec{a} = 0 \quad \Longrightarrow \quad \begin{bmatrix} -5 & 2 & 4 \\ 2 & -8 & 2 \\ 4 & 2 & -5 \end{bmatrix} \vec{a} = 0 \quad \Longrightarrow \quad \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \vec{a} = 0$$

Thus the dimension of the nullspace of $T - (8)I$ is 1, and we find a vector in the nullspace:

$$\vec{a}_3 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

Now we define:

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 0 & -2 & 1 \\ -1 & 0 & 2 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 8 \end{bmatrix} \quad A^{-1} = \frac{1}{9} \begin{bmatrix} 4 & 2 & -5 \\ 1 & -4 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

Then T can be diagonalized as $T = ADA^{-1}$.

Additional Resources For more on diagonalization, read *Strang* pages 304 - 309. In *Schuam's*, read pages 299 - 301 and try Problems 9.9 to 9.21, and 9.45 to 9.51.

10.5 A Proof with Diagonalization

Not only can diagonalization help us raise matrices to powers, it can also help us understand some underlying properties of matrices. In this example, we use diagonalization to understand when matrices are commutative.

Show that if two $n \times n$ diagonalizable matrices A and B have the same eigenvectors, then their matrix multiplication is commutative (i.e. $AB = BA$)

Let $\vec{v}_1, \ldots, \vec{v}_n$ be the eigenvectors of A and B, and $P = [\vec{v}_1, \ldots, \vec{v}_n]$ be a matrix constructed from the eigenvectors of A and B. Also let $\lambda_{a1}, \ldots, \lambda_{an}$ be the eigenvalues of A, and $\lambda_{b1}, \ldots, \lambda_{bn}$ be the eigenvalues of B.

Using diagonalization, we can decompose A and B such that $A = PD_AP^{-1}$ and $B = PD_BP^{-1}$, where D_A is the diagonal matrix of A, and D_B is the diagonal matrix of B:

$$D_A = \begin{bmatrix} \lambda_{a1} & & \\ & \ddots & \\ & & \lambda_{an} \end{bmatrix} \qquad D_B = \begin{bmatrix} \lambda_{b1} & & \\ & \ddots & \\ & & \lambda_{bn} \end{bmatrix}$$

Now we can write

$$\begin{aligned} AB &= (PD_AP^{-1})(PD_BP^{-1}) \\ &= PD_A(P^{-1}P)D_BP^{-1} \\ &= PD_A(I)D_BP^{-1} \\ &= PD_AD_BP^{-1} \end{aligned}$$

Because D_A and D_B are both diagonal matrices, $D_AD_B = D_BD_A$. To see why this is the case, let's look at it element by element:

$$D_AD_B = \begin{bmatrix} \lambda_{a1} & & \\ & \ddots & \\ & & \lambda_{an} \end{bmatrix} \begin{bmatrix} \lambda_{b1} & & \\ & \ddots & \\ & & \lambda_{bn} \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} \lambda_{a1}\lambda_{b1} & & \\ & \ddots & \\ & & \lambda_{an}\lambda_{bn} \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \lambda_{b1}\lambda_{a1} & & \\ & \ddots & \\ & & \lambda_{bn}\lambda_{an} \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} \lambda_{b1} & & \\ & \ddots & \\ & & \lambda_{bn} \end{bmatrix} \begin{bmatrix} \lambda_{a1} & & \\ & \ddots & \\ & & \lambda_{an} \end{bmatrix} = D_BD_A \tag{4}$$

Thus:

$$\begin{aligned} PD_AD_BP^{-1} &= PD_BD_AP^{-1} \\ &= PD_BID_AP^{-1} \\ &= (PD_BP^{-1})(PD_AP^{-1}) \\ &= BA \end{aligned}$$

We've now shown that A and B are commutative if they have the same eigenvectors. We can use this to design matrices that commute!

10.6 Practice Problems

These practice problems are also available in an interactive form on the course website (http://inst.eecs.berkeley.edu/~ee16a/sp19/hw-practice/).

1. If $\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ is in the standard basis, what is $\vec{v}$ in the basis of $\vec{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \vec{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \vec{a}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$?

2. True or False: A 3×3 matrix with the following eigenvectors is diagonalizable: $\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

3. Which of the following matrices is equivalent to $\begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \end{bmatrix}^k$?

   (a) $\begin{bmatrix} 1 & 0 \\ 0 & 0.5^k \end{bmatrix}$

   (b) $\begin{bmatrix} 1 & 0 \\ 1-0.5^k & 0.5^k \end{bmatrix}$

   (c) $\begin{bmatrix} 0.5^k & 0 \\ 0.5^k - 1 & 1 \end{bmatrix}$

   (d) $\begin{bmatrix} 1 & 0 \\ 1+0.5^k & 0.5^k \end{bmatrix}$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 11

11.1 Introduction to Electrical Circuit Analysis

Our ultimate goal is to design systems that solve people's problems. To do so, it's critical to understand how we go from real-world events all the way to useful information that the system might then act upon. The most common way an engineered system interfaces with the real world is by using sensors and/or actuators that are often composed of electronic circuits; these communicate via electrical signals with processing units, which are *also* composed entirely of electronic circuits. In order to fully understand and *design* a useful system, we will need to first understand **Electrical Circuit Analysis**.

In this note, we will start by intentionally ignoring the underlying physics of electrical circuits, and will instead focus on a standard procedure and set of rules that will allow us to systemically "solve" such circuits, (i.e., given a circuit diagram, solve for all of the relevant electrical quantities in that circuit). By abstracting the physics away during the analysis step, we hope to emphasize that the analysis in and of itself is simply a matter of taking a visual diagram (representing an electrical circuit) and applying a set of rules to it that will convert the diagram into a set of (linear) equations that can then be solved using the techniques we have developed in the first module.

11.2 Basic Circuit Quantities

Let's start with some definitions of basic quantities present in an electrical circuit. **Current** is the flow of charges (e.g. electrons) in the circuit, and **voltage** is the potential energy (per charge) between two points in the circuit. This potential energy is what causes charge to flow (ie. causes current). **Resistance** is the material's tendency to resist the flow of current.

Quantity	Symbol	Units
Voltage	V	Volts (V)
Current	I	Amperes (A)
Resistance	R	Ohms (Ω)

We use these quantities in a **Circuit Diagram**, a visual representation of how a collection of circuit elements are connected. Each circuit element has some voltage *across* it and some current *through* it. We often refer to the voltage difference across some element as the *voltage drop* across the element.

Why is voltage "across" a circuit element? Voltage, or electric potential, is only defined *relative* to another point. A simple analogy is elevation: A mountain's summit could be 9,000 ft above sea level, but 21,000 ft above the bottom of the ocean. In both cases, the elevation is only meaningful relative to another point. For convenience, we frequently define sea level as a reference point with "0 ft of elevation" – then we can state elevation as a single number which is implicitly referenced to sea level (ex. the mountain is 9,000 ft tall). Similarly, in circuits, we will frequently define a reference point, called **ground**, against which other voltages can be measured.

11.3 Basic Circuit Elements

How do our basic circuit quantities interact? It depends on the circuit element! For each element, we will define a relationship between the voltage across the element and the current through it. We will call this the "IV Relationship", because current is represented by I and voltage by V. By convention, for these elements, we will define some positive "+" terminal and some negative "-" terminal, and we define the voltage across the element to be the difference between the positive terminal and the negative terminal. Let's look at some of the most common circuit elements and their IV relationships.

Wire: The most common element is the wire. A wire connects two circuit elements, and we assume current flows freely through it, without any voltage drop. In a circuit diagram, any solid line (that is not a circuit element) is assumed to be a wire. The IV relationship for a wire is:

$V_W = 0$ A wire is an ideal connection with zero voltage across it.
$I_W = ?$ The current through a wire can take any value, and is determined by the rest of the circuit.

Symbol

IV Relationship

Resistor: A resistor is an element that restricts or "resists" current, which is the flow of charge. This resistance causes a voltage difference across the resistor. The IV relationship of a resistor is called "Ohm's Law."

$V_R = I_R R$ The voltage across a resistor is determined by Ohm's Law.
$I_R = \frac{V_R}{R}$ The current through a resistor is determined by Ohm's Law.

Symbol

IV Relationship

Open Circuit: Whereas a wire freely allows current flow with no voltage drop, an open circuit does not allow any current flow through it. As a result, it can have any voltage across it. Physically, a open circuit exists between any two points that are not connected, such as a disconnected wire.

$V_{OC} =$	?	The voltage across an open circuit can take any value, and is determined by the rest of the circuit.
$I_{OC} =$	0	No current is allowed to flow through an open circuit.

Symbol

IV Relationship

Voltage Source: A voltage source is a component that forces a specific voltage V_s across its terminals. The most common voltage sources we encounter are batteries, but these are not very precise in general. In the lab, we may use specific tools to give us very precise voltages. The $+$ and $-$ sign indicates which direction the voltage is pointing. The voltage difference between the "$+$" terminal and the "$-$" terminal is always equal to V_s, no matter what else is happening in the circuit. A voltage source is expected to allow any current through it.

$V_{VS} =$	V_s	The voltage across the voltage source is always equal to the source value.
$I_{VS} =$	?	The current through a voltage source is determined by the rest of the circuit.

Symbol

IV Relationship

Current Source: A current source forces a specific current I_s in the direction specified by the arrow indicated on the schematic symbol. The current flowing through a current source is always equal to I_s, no matter what else is happening in the circuit. A current source is expected to allow any voltage across it. Note the duality between this element and the voltage source.

$V_{IS} =$? The voltage across a current source is determined by the rest of the circuit.
$I_{IS} = I_s$ The current through a current source is always equal to the source value.

Symbol

IV Relationship

Node: A node is a place in a circuit where two or more circuit elements meet. Everything at a given node shares the same voltage potential, which we may also call the *node potential*. Nodes are not necessarily just defined as a single point, but by everything between the circuit elements. Wires are considered parts of nodes because they have 0 voltage drop and so have the same potential as everything it connects. We will take a deeper dive at identifying nodes later in this lecture note.

Here are 3 examples of nodes. Nodes are marked by black dots and labeled, in this case a and b. Note that while the black dot denotes where the node is, the actual node extends all the way to the resistor terminals.

Branches: A branch in a circuit is any connection between two nodes. Generally, this is a single 2-terminal element, such as a resistor or a voltage source. We won't use this term frequently in this class, but you may see it in other literature. A term frequently associated with branches is *branch current*, which is the current through a given branch. This is simply equal to the current through the branch's associated circuit element.

11.4 Rules for Circuit Analysis

In addition to the IV relationships for a single elements, there are also rules govern the current and voltage relationships between multiple elements.

11.4.1 Kirchhoff's Current Law (KCL)

Kirchhoff's Current Law (KCL) states that the net current flowing out of (or equivalently, into) any node of a circuit is zero. To put this more simply, the current flowing into a node must equal the current flowing out of that node.

For example, consider the circuit shown above. We define current flowing out of the node to be positive, and therefore, current flowing into the node is negative. For example, from the left branch, there is i_1 current flowing into the node. We can equivalently say $-i_1$ current flowing out of the node. Similarly, there is i_2 current flowing into the node from the bottom branch and i_3 current flowing out of the node through the right branch. In total, we have $i_1 + i_2$ flowing into the node, and i_3 flowing out of the node. Therefore, the currents must satisfy

$$(-i_1) + (-i_2) + i_3 = 0 \text{ or } i_1 + i_2 = i_3 \tag{1}$$

11.4.2 Kirchhoff's Voltage Law (KVL)

Kirchhoff's Voltage Law (KVL) states that the sum of voltages across the elements connected in a loop must be equal to zero. The illustration below shows an example of a loop in a circuit. In our elevation analogy for voltage, this is equivalent to saying"what goes up must come down". We will not actively use KVL in the analysis procedure we outline next, but it is still an important tool and can be used as an additional rule to double check your final answers.

Mathematically, KVL states that:

$$\sum_{\text{Loop}} V_k = 0. \tag{2}$$

When adding the voltage drops around the loop, we must follow a convention. If the arrow corresponding to the loop goes into the "-" terminal of an element and out the "+" terminal, we add the voltage across that element. (In our elevation analogy, we went "uphill" from a lower voltage to a higher voltage so we gained "elevation.") Conversely, if the arrow goes into the "+" terminal of an element and out of the "-" terminal, we subtract the voltage across that element (this is like going "downhill" in our elevation analogy). Following this convention for the example in above illustration we find:

$$V_A - V_B - V_C = 0. \tag{3}$$

Note that if we had defined the loop in the opposite direction, we would find this instead:

$$-V_A + V_C + V_B = 0. \tag{4}$$

Thinking about an elevation analogy to voltage can help give some intuition to KVL: If you walk in a circle (a loop) so that you end up back where you started, than your total change in elevation must be zero, no matter how much you go up or down. If you walk in a line, ending up somewhere different, than your total change in elevation is equal to the sum of all of the elevation changes along the way.

11.4.3 Ohm's Law and Resistors

As already described when we introduced resistors as an element, for these elements, the voltage across them is directly proportional to the current that flows through them, where the proportionality constant is the "resistance" (R) of the device. This relationship is known as **Ohm's Law**.

$$V_{\text{R}} = I_{\text{R}} R. \tag{5}$$

The unit of R is Volts/Amperes, or more commonly "Ohms" (Ω).

11.5 Guide to Finding Nodes

Solving a circuit means finding all the node voltages and branch currents. How do we know which nodes we care about? Here, we will go over a method you can use to correctly identify all of the nodes in a circuit. We'll go through this method while applying it to an example circuit, shown below.

We'll find nodes one at a time with this method (the order we find the nodes is arbitrary). We start by choosing a color to represent the first node (red, in this example). Then we choose a starting point on the circuit,such as the upper left corner. From this point, we trace (in red) along all of the connected wires until we hit a non-wire component. Everything traced in red is part of a single node.

Then we choose a new color and a new point on the circuit *that is not already colored*. We repeat the process: tracing over all wires that are connected to this new point and stop when a non-wire element is reached. Repeat this process for new colors until all of the wires have been colored.

We've now identified all of the nodes! There are three in this example circuit. Note that one of these would be labeled "ground" in the seven step circuit analysis procedure, so we would only have to solve for two unknown node potentials.

11.5.1 Redrawing Circuit Diagrams

Once we've identified the nodes, it may be useful to redraw the circuit. Redrawing a circuit means drawing a new diagram with the same circuit elements, potentially rearranged, but connected such that the currents and voltages in each element is the same as the original circuit. This new diagram will represent the same circuit behavior as long as the elements are connected the same as before.

The most basic reason to do this is to make a diagram more organized and easier to read. Large circuits can quickly be come messy and complicated, so redrawing can help us quickly reorganize. Keeping your circuit elements aligned and orderly, your wire connections short and straight, etc. will help you communicate your circuits clearly.

Often though, we will redraw circuit diagrams to look like one of many common classes of circuits, such as a resistive voltage divider. Redrawing complicated circuits to look like these common circuits is very useful because it helps us to see patterns in a circuit diagram, making them easier to analyze and help us build intuition. We will begin to explore these circuit classes in the coming weeks.

It is hugely important when we redraw a circuit that we do not change the circuit behavior. If we don't want the circuit to change when we redraw it, **each terminal of each element must stay connected to its original node.** This is because the voltage drop over each element is dependent only on the nodes it is connected to, and the current through each element is determined by the voltage drop and the IV relationship of that element.

Let's try redrawing our example circuit. We can first move R_3 to a different location. Since R_2 and R_3 share a node (u_3), it is useful to draw them in a way that makes it easy to see that. As long as one end of R_3 is still connected to the red node (u_1) and the other end of R_3 is still connected to the green node (u_3), the circuits are equivalent.

For similar reasons, we may want to do the same to R_4. Note that R_5 is connected to the green node (u_3) on both sides. Since it does not interact with the red node (u_1), it is useful to move it "out of the way" so we can focus on the R_3 and R_4:

Note that all of the elements have the same node connections as they did in the original circuit. Therefore this circuit will have the exact same behavior as the original.

11.6 Solving Circuits

In this course, we will learn how to take a real world system and build a circuit diagram that models the behavior of that system, and we will design our own circuits for specific real world tasks. Given a circuit diagram, we want to be able to solve for all the node voltages and element currents. This analysis gives us a complete characterization and understanding of the circuit and can help us solve all sorts of problems.

For example, consider this simple light bulb circuit. This circuit consists of a battery, a resistor, and a lightbulb (represented by a circle with an X). In this circuit, there is some current flowing through the battery, resistor, and light bulb; and there is also some voltage drop across the resistor and light bulb. Solving the circuit will give us these quantities. The current and voltage will give us information about how bright the bulb is, how much power is being dissapated, etc. We can use this information to further design the system for safe amount of current or for a certain brightness.

While this example may be seem simple, the same principles can be used to design touchscreens, biosensors, power grids, etc. All of these boil down to deriving an appropriate circuit model and solving it!

11.7 Node Voltage Analysis (NVA)

Node Voltage Analysis, also known as Nodal Analysis, is a powerful systematic tool for finding all the voltages and currents in an electronic circuit. It can be used to solve any circuit composed of linear components, e.g. voltage sources, current sources, and resistors[1]. Because the equations we use are all linear, we can formulate the problem using matrices and solve them with the same matrix tools we have been using up till now. In fact, one of the earliest circuit solvers SPICE (Simulation Program with Integrated Circuit Emphasis), which was first published here at Berkeley in 1973, used NVA as its backbone, using efficient matrix algorithms to solve general circuit problems.

11.7.1 Objective

Find all voltages (and currents) in an electronic circuit.

11.7.2 Procedure

The method proceeds in several steps, illustrated below for the following example circuit:

[1] We will see in EECS16B that in certain contexts, we can also use NVA to solve circuits with inductors and capacitors.

- **Step 1:** Reference Node

  Select a reference (ground) node. Any node can be chosen for this purpose. In this example, we choose the node at the bottom of the circuit diagram and draw the "ground" symbol on it.

- **Step 2:** Label Nodes with Voltage set by Voltage Sources

  Voltage sources like batteries set the voltage of the node they are connected to. In the example, there is only one source, V_s, and we label the corresponding node u_1 (names are arbitrary, but must be unique).

- **Step 3:** Label Remaining Nodes

  Now we label all remaining nodes in the circuit except the reference. In this example there are two, u_2 and u_3.

- **Step 4:** Label Element Voltages and Currents

  Next we mark the voltage across each elements and the currents through each elements with unique variables.

  Start with labelling the current through each element. The direction is arbitrary (top to bottom, bottom to top, it won't matter, but stick with your choice in subsequent steps).

  Next, mark the element voltages following the passive sign convention. The **passive sign convention** dictates that positive current should *enter* the positive voltage terminal and *exit* the negative voltage terminal of a passive element. This gives us a consistent way to associated the direction of the current and the voltage drop. Below is an example for a resistor:

  When we discuss *power* later in the module, you will see why we call this convention "passive."

  The element voltage for I_s is not marked in the example since it will not be needed in the calculations below. Same for the voltage source. There is no harm in marking those, too.

- **Step 5:** Setup equations with Kirchoff's Current Law (KCL)

  Write KCL equations for all nodes with unknown voltage, u_2 and u_3 in the example.

  At u_2 we get the sum of all currents entering the node equals the sum of all currents exiting:

$$I_{R_1} = I_{R_2} + I_{R_4}$$

Similarly for u_3:

$$I_{R_4} + I_{I_s} = I_{R_3}$$

- **Step 6:** Element Currents

  Find expressions for all element currents in terms of the element voltages and IV charactersitics (e.g. Ohm's law). In this step, we will skip voltage sources, since their currents are not defined by IV characteristics but instead by other parts of the circuit.

  In this example, there are five elements that are not voltage sources: R_1, R_2, R_3, R_4, and I_s. We can define the element current for each of these using their IV characteristic (e.g. Ohm's Law):

$$\begin{aligned} I_{R_1} &= \frac{V_{R_1}}{R_1} \\ I_{R_2} &= \frac{V_{R_2}}{R_2} \\ I_{R_3} &= \frac{V_{R_3}}{R_3} \\ I_{R_4} &= \frac{V_{R_4}}{R_4} \\ I_{I_s} &= I_s \end{aligned}$$

- **Step 7:** Element Voltages

  Substitute all element voltages in your step 6 equations with differences between node voltages. For example, $V_{R_1} = u_1 - u_2$ and $V_{R_2} = u_2 - 0 = u_2$. At this step, we will also process the voltage sources and look at the nodes they define. For example, the source V_s sits between ground and u_1. Therefore, the node voltage at u_1 is $u_1 = 0 + V_s = V_s$. Substituting all our element voltages into the current equations, we get:

$$\begin{aligned} I_{R_1} &= \frac{V_s - u_2}{R_1} \\ I_{R_2} &= \frac{u_2}{R_2} \\ I_{R_3} &= \frac{u_3}{R_3} \\ I_{R_4} &= \frac{u_2 - u_3}{R_4} \\ I_{I_s} &= I_s \end{aligned}$$

- **Step 8:** Substitute Element Currents in KCL Equations

Now we substitute the expressions derived in Step 7 into the KCL equations from Step 5.

$$\frac{V_s - u_2}{R_1} = \frac{u_2}{R_2} + \frac{u_2 - u_3}{R_4}$$
$$I_s + \frac{u_2 - u_3}{R_4} = \frac{u_3}{R_3}$$
.

Let's reorganize this by grouping the terms with unknowns (u_2 and u_3) on the left side and the terms with only known values on the right:

$$u_2(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4}) + u_3(-\frac{1}{R_4}) = \frac{V_s}{R_1}$$
$$u_2(-\frac{1}{R_4}) + u_3(\frac{1}{R_3} + \frac{1}{R_4}) = I_s$$

We have arrived at a system of linear equations, with u_2 and u_3 as our variables.

- **Step 9:** Solve

  Now we can solve for the unknown node voltages, u_2 and u_3 in the example. First, rearrange terms to cast the equations into our familiar linear algebra form $A\vec{x} = b$:

$$\begin{bmatrix} \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} & -\frac{1}{R_4} \\ -\frac{1}{R_4} & \frac{1}{R_3} + \frac{1}{R_4} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{V_s}{R_1} \\ I_s \end{bmatrix}$$

  Then compute the solution using Gaussian Elimination (or let the computer do the work, here using sympy):

```
from sympy import *
init_printing(use_unicode=True)

R1, R2, R3, R4 = symbols('R1 R2 R3 R4')
Y = Matrix([[ 1/R1+1/R2+1/R4, -1/R4], [-1/R4, 1/R3+1/R4]])

V1, I1 = symbols('V1 I1')
b = Matrix([ V1/R1, I1 ])

Vn1, Vn2 = linsolve((Y, b)).args[0]
```

Algebraic result:

```
>>> Vn1
    R₂⋅(I₁⋅R₁⋅R₃ + R₃⋅V₁ + R₄⋅V₁)
─────────────────────────────────────
R₁⋅R₂ + R₁⋅R₃ + R₁⋅R₄ + R₂⋅R₃ + R₂⋅R₄
>>> Vn2
  R₃⋅R₄⋅(I₁⋅(R₁⋅R₂ + R₁⋅R₄ + R₂⋅R₄) + R₂⋅V₁)
─────────────────────────────────────────────
-R₁⋅R₂⋅R₃ + (R₃ + R₄)⋅(R₁⋅R₂ + R₁⋅R₄ + R₂⋅R₄)
```

Numerical result:

```
>>> values = {R1:1, R2:2, R3:3, R4:4, I1:0.5, V1:1}
>>>
>>> f"Vn1 = {Vn1.evalf(3, subs=values)} V"
Vn1 = 0.739 V
>>> f"Vn2 = {Vn2.evalf(3, subs=values)} V"
Vn2 = 1.17 V
```

We have now solved for the node voltages u_2 and u_3, corresponding to Vn1 and Vn2 in the code respectively.

11.7.3 Branch Currents

Now that we have solved for the node voltages, we can easily solve for any element current or branch current. To solve for a branch current, we can plug in the node voltages we found into the element voltage equations from Step 7. For example, the current I_{R_4} through resistor R_4 is

$$I_{R_4} = \frac{V_{R_4}}{R_4} = \frac{u_2 - u_3}{R_4}$$

Using the node voltages we found for u_2 and u_3, we get numerically: I_{R_4} = -0.109 A.

11.8 References

Schaum's Outline of Electric Circuits, Seventh Edition, Section 4.4.

Electronics Reader, Ali Niknejad, 2015, Sections 2.1 and 5.1. (available on course website.)

11.9 Practice Problems

These practice problems are also available in an interactive form on the course website (https://eecs16a.org/#practiceSets)

1. True or False: A voltage source can have any current through it.
2. True or False: A current source can have any voltage across it.
3. True or False: The voltage across R_1 and across R_2 is the same.

4. True or False: The current through the resistors is the same.

5. If you have n nodes in a circuit with k non-wire elements connecting the nodes, how many equations do you need to solve for all node potentials and element currents? Remember that one node needs to be grounded.
6. How many nodes would you need to label to perform nodal analysis? Include nodes for ground and for V_s.

7. How many nodes are in the following circuit?

8. Assume that you have picked the ground node and labeled the node potentials and branch currents as follows.

What are the $+/-$ labels for R_1 and R_2 according to passive sign convention?

9. For the same circuit as above, formulate a system of equations to solve for all node potentials and branch currents.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 12

12.1 Voltage Divider

In order to review the analysis procedure described in the previous note, let's look at a particular circuit called a "voltage divider". As we will see by the end of this note, the voltage divider circuit actually lies at the heart of the resistive touchscreen that we will be studying shortly.
The voltage divider circuit consists of a voltage source (V_s) and two resistors (R_1 and R_2).

We use the circuit analysis algorithm developed in the previous note to analyze this circuit:

- **Step 1:** Pick a node and label it as ground.

- **Step 2:** Label the node connected to the voltage supply as $u_1 (= V_S)$, since the voltage supply goes between this node and ground.

- **Step 3:** Label the remaining node as u_{mid}.

- **Step 4:** Label the voltages and currents through every element in the circuit with V_i and I_i respectively:

- **Step 5:** Write KCL equations for all nodes with unknown voltage - in this case, this is just u_{mid}, since $u_1 = V_S$. The current entering that node is I_{R_1} and the current leaving it is I_{R_2}. Since these currents must be equal,

$$I_{R_1} = I_{R_2}.$$

- **Step 6:** Find expressions for element currents for all elements (except the voltage source) using their

characteristics. Applying Ohm's law to the two resistors, we find that

$$I_{R_1} = \frac{V_{R_1}}{R_1}$$
$$I_{R_2} = \frac{V_{R_2}}{R_2}.$$

- **Step 7:** Substitute in node voltages. From our voltage source, we have

$$u_1 = V_S.$$

From our node voltages, we can express our element voltages as

$$V_{R_1} = u_1 - u_{mid} = V_S - u_{mid}$$
$$V_{R_2} = u_{mid} - 0 = u_{mid}.$$

Substituting, we obtain our element currents in terms of node voltages:

$$I_{R_1} = \frac{V_S - u_{mid}}{R_1}$$
$$I_{R_2} = \frac{u_{mid}}{R_2}.$$

- **Step 8:** Substitute the element currents into our KCL equation. We have

$$I_{R_1} = I_{R_2}$$
$$\implies \quad \frac{V_S - u_{mid}}{R_1} = \frac{u_{mid}}{R_2}.$$

- **Step 9:** Solve the above equation for the unknown node voltage u_{mid}. Rearranging, we find that

$$V_S R_2 - u_{mid} R_2 = u_{mid} R_1$$
$$\implies \quad u_{mid}(R_1 + R_2) = V_S R_2$$
$$\implies \quad u_{mid} = \frac{R_2}{R_1 + R_2} V_S = \frac{1}{1 + \frac{R_1}{R_2}} V_S.$$

The reason this circuit is called a "voltage divider" is that we can control the output voltage of $u_{mid} = \alpha V_S$, $\alpha \in [0,1]$ (assuming that all of the resistance values are non-negative) by varying the **ratio** of the resistor values R_1/R_2. Since only the ratio matters in setting the value of the output voltage, the absolute values we choose for R_1 and R_2 are up to us, as the designer and engineer, and we may make the decision based on a target current or the availablity of parts.

Varying this ratio to changing the output voltage is exactly the mechanism we will use to convert the relative position of a user's touch to a voltage, as we will see in the next section.

12.2 Resistive Touchscreen

Touchscreens have revolutionized how we interact with devices. The majority of real world touchscreens are two dimensional – they find both the horizontal and vertical position of a finger press. In this note we will simplify the touchscreen to a 1D structure where we can only detect the horizontal position of the touch (or equivalently, the vertical position). In later notes we'll extend this to 2D.

Let's first examine the physical structure of a such a 1D touchscreen. The cross-section of the touchscreen is shown below. It consists of two layers of length L: the top layer (red) is flexible, and the bottom layer (black) is non-flexible. Both layers are conductive so current can flow through them. When we touch the screen, the top part contacts the bottom part at the touchpoint (in this example, we assume that the touch causes the entire width of the structure to deform uniformly and make contact to the bottom structure).

Our goal is to find the position of the touchpoint. We define L_{touch} to be the distance from the left side of the structure to the touchpoint, and L_{rest} to be the remaining distance from the touchpoint to the right side of the structure. Determining L_{touch} is equivalent to finding the position of the touchpoint.

How can we measure this physical quantity using an electrical circuit? Next we will introduce some of the physics of circuit components so that we can convert this physical structure into an electrical model. Once we have that model, we can connect additional components around it to build a complete circuit which we can then analyze (using the procedure developed in the previous note). It is worth re-emphasizing here that each of these steps is looking at different levels of abstraction – the modeling step converts a physical system into an electrical model (as we will see in this particular case, a model that is comprised of wires and resistors), and after additional components are connected to that model to create a complete circuit, we can analyze that circuit without necessarily knowing more details than are captured in the model.

12.3 Physics of Circuits

Charge is the basic underlying quantity associated with all electrical systems. Charge can be either positive or negative, although in most materials/systems charge is carried by electrons (which are negatively charged). We measure charges with the unit **coulomb (C)**, and typically use the symbol **Q**.

Current is a measure of the movement of charge, specifically, the net amount of charge crossing through a surface in a unit time. We usually use the symbol **I** to denote current, and is defined by:

$$I = \frac{dQ}{dt}.$$

The unit for current is an **ampere (A)**, which is equivalent to 1 coulomb per second ($1C = 1A * 1sec$). We typically specify the direction of the net positive charge flow using an arrow. For example, the following arrow

$$\xrightarrow{\quad 5A \quad}$$

represents 5 amperes of current traveling to the right. This means that net $+5$ coulombs of charge are crossing the surface per second to the right. We could also say that net -5 coulombs of charge are crossing the surface per second to the left.

Voltage is defined as the amount of energy needed to move a unit charge between two points. We usually denote voltage with the symbol **V**. The unit associated with a voltage is the **Volt (V)**, where 1 Volt is defined such that it will require 1 Joule of energy to move 1 Coulomb (i.e. the unit of charge) between the two points (i.e. $1V = 1J/C$ or $1V * 1C = 1J$). It is important to emphasize that the definition of voltage makes it a relative quantity that can only be measured between two points; an "absolute" voltage is meaningless. As we saw in the analysis procedure, as a shorthand we will sometimes refer to a voltage at a particular point, but if we do so, we are implicitly using some other known point (often, the **ground** that we defined) as a reference against which the voltage at that single point is being measured.

A simple analogy to voltage and the fact that it is a relative quantity can be described using elevation. A mountain's summit could be 9000ft above sea level (where sea level would be the reference of an elevation of 0 ft). Alternatively, we can say that there is 9000ft between the summit and sea level. Instead, if we choose the bottom of the ocean as the reference point (elevation of 0 ft), then there will still be 9000ft between sea level and the summit of the mountain, but the elevation at the mountain referenced to the bottom of the ocean would not be 9000ft (it would be much greater). So, just as sea level is an arbitrary (although convenient) reference level for elevation, the ground node we choose as "0V" to measure the rest of the voltage at the rest of the nodes in our circuit against is also an arbitrary reference point.

Resistance: Conductors are materials that allow current to flow through them (for example, metals such as copper). However, some amount of energy is still required to get charge (current) to move through conductors. This is because electrons (charge) flowing through the conductor occasionally collide with the atoms in the conductor. These collisions cause the atoms to vibrate, generating heat. Since heat is another form of energy, this process implies that energy must be spent to move the electrons through the conductor. The amount of energy that is spent (per unit charge) is described by the *resistance* of the structure.

Since voltage describes the amount of energy needed to move a charge between two points, we can capture the behavior described above by stating that when current flows through a resistor, there is also a voltage drop across that resistor. This statement is known as Ohm's law, which is:

$$V = IR$$

where the resistance R has units of Ohms Ω. Note that this means that 1 Ω = $\frac{1V}{1A}$.

The actual value of resistance for some physical structure is set by two things: (1) material properties, and (2) the structure's dimensions. First let's consider the material properties: every conducting material

has a property known as resistivity ρ which describes how easily electrons flow through the material. For example, gold has a lower resistivity than steel. Hence, for the same physical dimensions, a gold structure will have lower resistance than a steel structure.

Next, let's consider how different dimensions impact the resistance of a structure. If we increase the length L of the structure (i.e., if we increase the dimension of the structure in the same direction as the current flows through it), this will increase the number of atoms that the electrons collide with, and hence we can expect the resistance to increase with L. If we increase the cross-sectional area A of the structure (i.e., increase the dimension in directions perpendicular to the current flow), we are allowing more electrons to flow in parallel with each other, and hence we can expect the resistance to decrease with A.

Therefore, for a physical structure made out of a material with resistivity ρ and having length L and cross-sectional area A, its resistance is given by:

$$R = \rho \times \frac{L}{A}.$$

12.4 Resistive Touchscreen Revisited

Now that you've been introduced to some circuits physics, let's review the actual physical structure of the top (red) layer in the 1-D touchscreen. As said, the touchscreen material is conductive; it will have some resistivity ρ that depends on the material. Though we are only looking at the image in a 1D cross-section, we can assume it has some cross-sectional area A. It also has some length to it. We can put these parameters together and model the screen layer as a resistor!

Since the touchpoint is where we are most interested, we can divide the top (red) layer into two parts and see what happens at the touchpoint. Given that the top (red) layer has a resistivity ρ and a cross-sectional area A, the resistance of the top layer from the touchpoint to the right end is given by $R_1 = \rho \frac{L_{rest}}{A}$, and the resistance of the top layer from the left end to the touchpoint is given by $R_2 = \rho \frac{L_{touch}}{A}$. Thus we can model this system as two resistors connected one after another, or "in series".

We then connect a voltage source V_s to the two ends of the circuit model we just built.

This is the same circuit model that we already analyzed at the beginning of this note — it's a voltage divider circuit! From our previous analysis, we know that

$$u_{mid} = \frac{R_2}{R_1 + R_2} \times V_s. \tag{1}$$

Now we plug in our relationships between the resistances and our physical model, $R_2 = \rho \frac{L_{touch}}{A}$ and $R_1 = \rho \frac{L_{rest}}{A}$:

$$u_{mid} = \frac{\rho \frac{L_{touch}}{A}}{\rho \frac{L_{rest}}{A} + \rho \frac{L_{touch}}{A}} \times V_s. \tag{2}$$

ρ and A are shared in the denominator and the numerator so they cancel. Then u_{mid} becomes:

$$u_{mid} = \frac{L_{touch}}{L_{touch} + L_{rest}} \times V_s. \tag{3}$$

What is $L_{touch} + L_{rest}$? It is L, the physical length of the touchscreen. Hence, we can simplify our equation:

$$u_{mid} = \frac{L_{touch}}{L} \times V_s. \tag{4}$$

By connecting a voltage source to the two ends of the touchscreen and measuring u_{mid}, we can figure out the relative position of L_{touch}.

The relationship we have found between u_{mid} and V_s is very convenient because u_{mid} is not dependent on any material property such as ρ and A. This means that the top layer can be built with any material and the relationship between u_{mid} and V_s is still valid. We therefore have the freedom to choose a material for the top layer that is good for display purposes (rather than needing a specific material for the touchscreen to work).

One thing has not yet been explained – what is the function of the bottom (black) layer in the touchscreen? In the next note, we will explain how to use the bottom (black) layer to measure u_{mid}.

12.5 Practice Questions

These practice problems are also available in an interactive form on the course website (https://eecs16a.org/#practiceSets)

1. How many nodes would you need to label to perform nodal analysis? Include nodes for ground and for V_s.

2. By what factor does the resistance of a rectangular block of metal change if you double each dimension of the block?

   (a) 1
   (b) 2
   (c) $\frac{1}{2}$
   (d) $\frac{1}{4}$

3. Assume that you have a cylindrical piece of metal. The cylinder has a length of 10 cm and a radius of 1 mm. You measure the resistance between the two circular surfaces and a get a reading of 0.54 mΩ. What is the resistivity of the metal?

(a) $1.7 \times 10^{-5}\,\Omega\,\text{m}$
(b) $1.7 \times 10^{-6}\,\Omega\,\text{m}$
(c) $1.7 \times 10^{-8}\,\Omega\,\text{m}$
(d) $5.4 \times 10^{-9}\,\Omega\,\text{m}$

4. Find the voltage across the resistor R_1.

(a) V_s
(b) $\frac{R_2}{R_1+R_2}V_s$
(c) 0
(d) Undefined

5. Given the circuit below, which resistors have a voltage drop equal to V_s?

6. In the same circuit, we add a new resistor R_6, which resistors still have a voltage drop equal to V_s?

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 13

13.1 Resistive Touchscreen - expanding the model

Recall the physical structure of the simple resistive touchscreen given in the prior lecture:

Previously, we ignored the black "bottom plate"; But why do we want it at all? Let's return to the model of our touchscreen and build from that. The simple version is shown below, where u_{mid} represents the point where the finger is touching. Note that a voltage source has been connected around it, with the positive terminal connected to one edge of the top plate and the negative terminal connected to the opposite edge.

To understand what the bottom plate is doing, we will first assume it is a perfect conductor ($\rho = 0$), so the circuit will look as follows. Notice again that a voltage source is connected to the terminating edges of the plates:

Does the voltage at u_{mid} change in this case? Thinking about it in terms of nodal analysis, we have not actually added any new nodes to our circuit - you can that ideal wires (with zero voltage drop) are all that

we have actually added. Therefore, we see that with a perfect conductor (analogous to our ideal wire), $V_{out} = u_{mid}$.

So why have the bottom plate at all? It lets us take the measurement for V_{out} using connection points on the edge of the plate instead of having to put a probe or wire at the actual touch point every time! As we have shown, in the case of an ideal bottom plate this does not affect the measurement result itself.

What if the bottom plate is not a perfect conductor? This means that the bottom plate would also have an associated resistance, calculated just like the resistance of the top electrode. Our new circuit schematic would look like this:

There are now more nodes in the circuit. We now have two options; we can either go back and perform nodal analysis with this new expanded circuit, or we can try and build off previous understanding to see if we can discern how the addition of resistors R_3 and R_4 affect circuit behavior.

Consider this abbreviated schematic:

Note that there is a current shown through the two new resistors, which may lead one to think that we need to conduct nodal analysis again. However, both of these resistors are followed by an open circuit. From the definition of an open circuit, we know that zero current will flow through it. If $i_{mid} = i_{R4} + i_{R3}$ from KCL, and $i_{R4} = i_{R3} = 0$ from the definition of an open circuit, from Ohm's Law, the voltage across these new resistors will be 0. This means that, even with an imperfectly conductive bottom plate, the voltage V_{out} will still be equal to u_{mid}, even with the addition of these new resistors.

To measure an output voltage, we need to put some device at the open circuit labeled V_{out}. Therefore, from the analysis above, we would like our device to function like an "open circuit". In the prior case, it allowed us to neglect the non-idealities of our bottom plate. This extends to an important question: **How can I guarantee that whatever I connect to my circuit to measure V_{out} or i_{out} does not influence/change the circuit itself?** To seek a general answer, we will introduce a new quantity: power.

13.2 Power

Note the two fundamental quantities that we may want to measure, voltage and current. To measure these quantities we use tools called a *voltmeter* and an *ammeter*, respectively. We can draw an abstracted diagram that looks like these for each case:

Note that the voltmeter measures voltage *across* the circuit, while the ammeter needs to be put in-line with the circuit so that the current flows *through* it. It turns out that the most complete and concise way of guaranteeing these measurement tools do not influence the circuit is to state that **they do not allow any power dissipated through the measurement device**.

So what exactly *is* power? Power P is the rate of change of energy, i.e. $P = \frac{dE}{dt}$. Recall that voltage V is the amount of energy needed to move a unit charge between two points (see Note 12). Thus, $dE = V\,dQ$ where dE is a differential unit of energy and dQ is a differential unit of charge. Taking the time derivative of both sides yields:

$$\frac{dE}{dt} = V\frac{dQ}{dt} \tag{1}$$
$$P = V\frac{dQ}{dt} \tag{2}$$

Recall that current I is defined as the rate of change of charge, $\frac{dQ}{dt}$. Therefore,

$$P = V\frac{dQ}{dt} \tag{3}$$
$$P = VI \tag{4}$$

The power dissipated (positive) or supplied (negative) by a component with voltage V and current I is thus equal to their product IV. Note that according to passive sign convention, positive current goes into the positive terminal of the component.

When $P = IV$ is positive, power is being dissipated; when $P = IV$ is negative, power is being generated/delivered. Being able to tell when power is dissipated versus delivered using the sign is the reason why passive sign convention was introduced in the first place! Note that, once we have completed nodal analysis, we should therefore be able to solve for power delivered/dissipated in every circuit element using this relationship.

$P = IV$ is a fundamental relationship that will be used repeatedly and is worth memorizing. It can also take other useful forms through combination with Ohm's Law, given here for reference:

$$P = IV \tag{5}$$
$$P = \frac{V^2}{R} \tag{6}$$
$$P = I^2R \tag{7}$$

Returning to the question of how to build an ideal measurement device, remember that we want the device to have 0 effect on the power dissipation/generation of the circuit.

Because our voltmeter is made to measure voltage, we can naturally assume that the voltage will or can be non-zero; this means that a voltmeter *must* have 0 current going into it to ensure $P = IV = 0$. In the ideal

case, the voltmeter will resemble an open-circuit, which has exactly zero current through. If we instead assume that the voltmeter does have some associated resistance, what do we want it to be? Remember Ohm's Law, $V = IR$. For a given voltage, the higher the associated resistance, the lower the current and therefore the lower the dissipated power.

On the other hand, the ammeter has the circuit's current flowing *through* it. Therefore, in order to ensure $P = IV = 0$, the ammeter needs V across it to equal zero. In the ideal case, the ammeter will therefore resemble a short-circuit (ideal wire), which has exactly zero voltage drop across it. If we instead assume that the ammeter does have some associated resistance, what do we want it to be? Remember Ohm's Law, $V = IR$. For a given current, the lower the associated resistance, the lower the associated voltage drop and therefore the lower the dissipated power.

13.3 Practice Questions

These practice problems are also available in an interactive form on the course website (http://ee16a.com/hw-practice#/).

1. What does it mean for the power of a particular component to be negative, for the following components?
   (a) Resistor
   (b) Voltage source
   (c) Current source
2. How much power is dissipated in R_2 in the following circuit?

3. In the same circuit as above, how much power is dissipated in the voltage source?

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 14

14.1 Voltage Divider Circuit Review

Before modeling the 2D touchscreen, let's review the most important concept that has enabled 1D touchscreen modeling: the voltage divider circuit.

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_s. \tag{1}$$

By using a voltage divider circuit, we can map u_{mid} to L_{touch}. A relationship between u_{mid} and L_{touch} exists such that:

$$u_{mid} = \frac{R_2}{R_1 + R_2} \times V_s = \frac{L_{touch}}{L} \times V_s. \tag{2}$$

14.2 EE16A Physics Revisited

Before we dive into the modeling of 2D resistive touchscreen, let's review the I-V characteristics for some basic circuit elements.

From the I-V plots, although a resistor, a wire and an open circuit can behave quite differently, their behaviors are exactly the same at (0, 0). This means that at (0, 0), these three circuit elements can be replaced by one another and the same behavior ($I = 0, V = 0$) is still expected.

14.3 An Interesting Circuit

Let's look at an example of different circuit elements behaving in the same way. The circuit we will analyze next (and the corresponding thought process) will be important when we analyze our 2D resistive touchscreen at the end of this note. Consider the following circuit:

If we want to solve for u_1, u_2, u_3, we could use our general analysis procedure. However, we can simplify our analysis by noticing that this circuit is very similar to the voltage divider we already analyzed. In fact, it is two voltage dividers — one consisting of resistors R_1, kR_1 and another consisting of R_2, kR_2.

Therefore, we can apply our voltage divider equation twice to find u_2 and u_3. Note that the total voltage drop over both voltage dividers is V_s.

$$u_2 = \frac{kR_1}{R_1 + kR_1} V_s = \frac{k}{1+k} V_s.$$
$$u_3 = \frac{kR_2}{R_2 + kR_2} V_s = \frac{k}{1+k} V_s.$$

We see that regardless of the resistances R_1 and R_2, the potentials u_2 and u_3 are the same! This holds as long as k is constant.

$$u_2 = u_3 = \frac{k}{1+k} V_s$$

Now, let's add another resistor R_3 to the circuit.

Once again, we could analyze this circuit from scratch using our circuit analysis procedure, but maybe we can simplify the analysis. Let's make a bold assumption that adding R_3 will not affect circuit operation and therefore $u_2 = u_3$ from our analysis above. We can determine if this assumption is true by analyzing the circuit and seeing if there are any contradictions that arise. If there are no contradictions, then we know that this bold assumption is true.

First, analyze the current flow through R_3.

$$R_3 i_3 = u_2 - u_3 \tag{3}$$

Under the assumption that R_3 does not affect circuit operation, then $u_2 = u_3$. Plugging this in tells us that the current flowing through R_3 is zero. In addition, we can calculate that the voltage drop across R_3 is $(u_2 - u_3)$ which is also zero. This means that R_3 is at the special (0, 0) point on the I-V plot, where it behaves the same way as a wire or open circuit. This means that there is no contradiction from our bold assumption since a resistor and open circuit have the same current and voltage at this point! Our bold assumption that R_3 will not affect circuit operation is correct!

To complete our analysis, we can replace R_3 with an open circuit:

Now, we can write u_2, u_3 directly, using the voltage divider equation as we did above:

$$u_2 = u_3 = \frac{k}{1+k} V_s. \tag{4}$$

When is it ok to replace R_3 with an open circuit? If R_1, kR_1, R_2, kR_2 were four arbitrary resistors, could we still replace R_3 with an open circuit? No, this simplification is only possible because we have set our resistor values so that $u_2 = u_3$. This means that the voltage drop over R_3 and current flowing in R_3 are both zero. In this case, the resistor R_3 is operating at (0, 0) on the I-V plot, so we can replace it without affecting circuit operation.

Next, we will introduce the 2D resistive touchscreen — and we'll see a very similar circuit appear in our model!

14.4 2D Resistive Touchscreen

Now, let's introduce the physical structure of a 2D touchscreen: it consists of a top red plate and a bottom black plate. When a finger touches the screen, the top red plate is pushed into contact with the bottom black plate at the touch point.

The top and bottom ends of the top red plate as well as the left and right ends of the bottom black plate are made of materials that have very low resistivities ρ, we can treat them as ideal wires ($\rho = 0$). The materials of the transparent screen that we touch in the middle have much higher resistivity.

In a 2D touchscreen, we want to figure out the vertical position and the horizontal position of the touch point: $L_{\text{touch, vertical}}$, $L_{\text{touch, horizontal}}$.

Let's first analyze the physical structure of the top red plate. We can treat the red plate as a bunch of vertical resistor strips, where each vertical strip is connected to the strips next to it by horizontal resistors as well. When we touch the plate, we split it into a top and bottom half, or R_{rest} and R_{touch}.

To simplify analysis, rather than considering many vertical strips, we will divide the red plate into just three vertical segments (of equal width) represented by resistors, which are connected by horizontal resistors R_{h1} and R_{h2}. By the end of this analysis you will see that increasing the number of vertical strips does not change the result.

As we did with the 1D touchscreen, we connect a voltage supply V_s to the top and bottom ends of the top red plate:

Let's analyze the circuit we just built with the top red plate and a voltage supply V_s:

Does this circuit remind you of the "interesting circuit" we analyzed in the previous section? Since R_{rest} and R_{touch} are the same for each segment, we know that $u_2 = u_3 = u_4$. As with the "interesting circuit" can replace horizontal resistors R_{h1}, R_{h2} with open circuits.

After replacing horizontal resistors with open circuit, we can use a voltmeter and measure u_3. Once again, using the voltage divider equation, we get:

$$u_3 = \frac{R_{touch}}{R_{rest} + R_{touch}} \times V_s. \tag{5}$$

Given $R_{touch} = \rho \frac{L_{touch}}{A}$, $R_{rest} = \rho \frac{L_{rest}}{A}$, u_3 can be further simplified:

$$u_3 = \frac{L_{touch}}{L} \times V_s \quad \text{where} \quad L_{touch} = L_{\text{touch, vertical}} \tag{6}$$

This means that u_3 is mapped to the vertical position touched in the same way as the 1D touchscreen. When measuring the vertical position touched ($L_{\text{touch, vertical}}$), the bottom black plate connects to a voltmeter and measures u_3, the same way it did in the 1D touchscreen. Note that, although we have represented the top red plate by three segments of equal width in the circuit model we built, the value of u_3 will remain the same if we choose to represent the top red plate by an infinite number of segments.

Now that we found $L_{\text{touch, vertical}}$, how can we find $L_{\text{touch, horizontal}}$? We know from linear algebra that if we want to find two values (i.e. vertical and horizontal position), we will need two measurements. What's another useful measurement that we can take? Well, the black bottom plate is rotated 90° compared to the red plate, so we can repeat this procedure on the black plate to get the horizontal touch position.

To do this, we connect the supply voltage source V_s to the bottom black plate, and connect the top red plate to a voltmeter. As before, we choose to represent the bottom black plate by three segments of equal width which are connected in between by vertical resistors R_{v1}, R_{v2}.

Let's analyze the circuit model for the bottom black plate:

Once again, we see that this is very similar to the "interesting circuit" and we can replace R_{v1} and R_{v2} with open circuits. Then we perform the same analysis as for top red plate, and we can derive u_3, which is:

$$u_3 = \frac{R_{touch}}{R_{touch} + R_{rest}} \times V_s \tag{7}$$

Here, $R_{touch} = \rho \frac{L_{\text{touch, horizontal}}}{A}$ in which $L_{\text{touch, horizontal}}$ is the horizontal position touched.

The measurement of vertical and horizontal positions ($L_{\text{touch, vertical}}$, $L_{\text{touch, horizontal}}$) for a 2D touchscreen can be summarized as follows:

- **Vertical Position Measurement**
  We connect a voltage source V_s to the top red plate and connect a voltmeter to the bottom black plate. We can map the voltage measured to the vertical position touched:

$$V_{out} = \frac{L_{\text{touch, vertical}}}{L} \times V_s. \tag{8}$$

- **Horizontal Position Measurement**
  We connect a voltage source V_s to the bottom black plate and connect a voltmeter to the top red plate. We can map the voltage measured to the horizontal position touched:

$$V_{out} = \frac{L_{\text{touch, horizontal}}}{L} \times V_s. \tag{9}$$

The important simplification used is replacing R_{h1}, R_{h2} with open circuits for the $L_{\text{touch, horizontal}}$ measurement and R_{v1}, R_{v2} for the $L_{\text{touch, vertical}}$ measurement. However, this kind of simplification is valid only if the resistor is at (0, 0) on the I-V plot, which means the resistor has zero current flow and therefore zero voltage drop ($IR = V$).

14.5 Faster Circuit Analysis

In the previous section, we used the fact that all of the segments had equal R_{rest} and R_{touch}. In this section, we'll consider a similar circuit where the resistances are all different! We'll solve the circuit using analysis that's theoretically equivalent to the 7 step procedure, except this time we will do the steps in a smart way that makes everything faster and easier.

We'll consider the following circuit:

Our goal is to solve for the voltage at each node.

- **Step 1: Label the circuit.** We will label **all unknown node voltages** (u_1, u_2, and u_3).

- **Step 2: Write equations for the nodes with voltage sources between them.** Here, the ground node and u_1 have V_s between them, which tells us that

$$u_1 = V_s.$$

- **Step 3: Write KCL for any unknown nodes, using the $V = IR$ relationship, and taking into account any current sources connected to the node.** In this case, our unknown nodes are u_2 and u_3, and we have no current sources (an example with a current source is below). We will treat the current as always flowing **out** of the node.

  Recall that $V = IR$. If we consider node u_2, the sum of the currents flowing out of u_2 through R_1, R_5, and R_3 is 0. Following passive sign convention, since the currents are treated as flowing out of the

node, so the voltage drop is $u_2 - u_i$ for some adjacent node u_i. So as an example, the current flowing from u_2 to u_1 through R_1 is $\frac{u_2-u_1}{R_1}$. Following this pattern for all the currents, and replacing $u_1 = V_s$, we get

$$\frac{u_2 - V_s}{R_1} + \frac{u_2}{R_3} + \frac{u_2 - u_3}{R_5} = 0, \qquad \frac{u_3 - V_s}{R_2} + \frac{u_3}{R_4} + \frac{u_3 - u_2}{R_5} = 0.$$

You can check that the signs of your voltage drops are correct by checking that the node in question is always positive in the numerators. So in the first equation, u_2 is positive, while in the second equation, u_3 is positive.

Now, we have just two equations and two unknowns (u_2 and u_3)! So all the remains to be done is to solve the equations by hand.

What if a node has a current source entering? How would we write the KCL equation in that case?

Remember that we treat all currents as flowing out of the node. We should also take the direction of the current source into account. In this case, the left branch of u_1 has a current source I_S flowing in, so $-I_S$ current flows out. So our equation is

$$-I_S + \frac{u_1 - u_2}{R_1} + \frac{u_1 - u_3}{R_2} = 0.$$

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 15

15.1 Introduction: Superposition and Equivalence

Circuit analysis can be cumbersome to do by hand, and it can be difficult to understand the high-level behavior of complicated circuits given a schematic. We need tools we can use to both lessen the burden of analysis, and help us think about circuits and understand how they behave. Ultimately, our goal is to **design** interesting circuits, and in this note we will build up additional tools to help us.

15.2 Dependent Sources

At this point, we will introduce a new circuit element: dependent sources. Dependent sources act like the independent sources we've studied so far, but instead of a producing a single voltage/current, the voltage/current is *controlled* by something else in the circuit. There are four different types of dependent sources, shown below:

Voltage-controlled voltage source

Voltage-controlled current source

Current-controlled voltage source

Current-controlled current source

Here, the diamond symbol represents the source, which produces voltage or current proportional to a different voltage or current elsewhere in the circuit. Let's make this more concrete by looking at an example:

In this case, the current in the dependent source is equal a constant (g_m) times the voltage v_c, which is defined elsewhere in the circuit. Just like an independent current source, this dependent source will produce the same current, regardless of what is connected to it (unless v_c changes – then the current will change appropriately). As we'll see in future notes, dependent sources are useful tools for modeling many advanced circuit elements.

15.3 Superposition

In this section, we are going to look at circuits with multiple voltage or current sources. In particular, we would like to introduce a very useful idea in working with circuits of this type – superposition.

Let's think back to our seven-step circuit analysis procedure. To solve for the currents and node potentials in a circuit, we set up a matrix problem of the form $A\vec{x} = \vec{b}$ where $\vec{x}$ contained the unknown currents and node potentials, $\vec{b}$ contained the independent current and voltage sources, and A described the relationship between them. Since this matrix equation describes a real system, we know that there is a unique solution. Therefore, A is invertible:

$$\vec{x} = A^{-1}\vec{b}$$

This means that we can describe any current or node potential (ie. any element of $\vec{x}$) as a linear combination of the independent current and voltage sources (the elements of $\vec{b}$). For example, consider a circuit with n independent sources voltage sources $V_{s1} \ldots V_{sn}$, and m independent current sources $I_{s1} \ldots I_{sm}$. An arbitrary node potential u_i (or equivalently, an arbitrary current i_i) can be written as

$$u_i = \alpha_1 V_{s1} + \ldots + \alpha_n V_{sn} + \beta_1 I_{s1} + \ldots \beta_m I_{sm}$$

where the α's and β's are coefficients from inverting A. Since this equation is linear, we can calculate each term of this equation separately and then add them together at the end. For example, if we want to calculate the first term, $\alpha_1 V_{s1}$ we can set all of the other voltage and current sources to zero, then solve for u_i. Repeating this for every source then adding the results is equivalent to calculating u_i with all of the sources present. However, splitting up the calculations can help us see simplifications and patterns that might be less obvious with all of the sources present.

This procedure is known as **superposition** and can be summarized as follows:

For each independent source k (either voltage source or current source)
- **Set all other *independent* sources to** 0
  - **Voltage source: replace with a wire**
  - **Current source: replace with an open circuit**
- **Compute the circuit voltages and currents due to this source k**

Compute V_{out} by summing the $V_{out,k}$s for all k.

Now we ask the question: why does it make sense to replace voltage sources with wires? If we look at the I-V plot of a voltage source V_S, where I is the current going through the voltage source, then the plot would be a vertical line:

Now if we want to zero out this voltage source, we are setting $V_S = 0$. Then the I-V plot is exactly the y-axis.

What does this mean? This means that it allows any current to go through, however the voltage drop always remains zero. This is exactly what a wire element (sometimes called a short circuit) does.
Now let's look at why we replace current sources with open circuits. If we plot the I-V graph of a current source I_S, we get the following:

What if we turn off the current source? Then the I-V graph becomes the x-axis, i.e., the line $I = 0$.

What does this mean? This means no matter what voltage you apply, there will be no current. This is equivalent to an open circuit.

Now let's illustrate this idea on the circuit below, where we would like to figure out V_{out}.

We first compute the output voltage due to V_1 and hence source V_2 will be replaced with a wire:

We can recognize this as a voltage divider circuit, and therefore we know that $V_{out,1} = \frac{R_2}{R_1+R_2}V_1$. Next we compute the output voltage due to V_2 and hence source V_1 will be replaced with a wire.

We again recognize that this is just a voltage divider circuit and therefore we can see that $V_{out,2} = \frac{R_1}{R_1+R_2}V_2$. Finally, to get the output voltage V_{out} of the original circuit, we add the contributions from each voltage source $V_{out} = V_{out,1} + V_{out,2} = \frac{R_2}{R_1+R_2}V_1 + \frac{R_1}{R_1+R_2}V_2$.

As a side note, we can apply the idea of replacing elements with equivalent elements (e.g. replacing a $V = 0$ voltage source with a wire) to resistors as well. When do resistors have an equivalent representation? We will try and demonstrate this graphically. Recall that by Ohm's law, the I-V graph across a resistor looks like

We know that the slope of the line is equal to $\frac{1}{R}$. What happens in the limit where R trends towards infinity? Then the line becomes the x-axis, which corresponds to an open circuit as we've seen earlier. Now what happens in the limit where R trends towards zero? The line becomes the y-axis, which corresponds to a wire.

To summarize, **zero voltage source and zero resistance are equivalent to wires (i.e. short circuits); zero current source and infinite resistance are equivalent to open circuits.**

15.4 Equivalence

One aspect of circuit design that is distinctly different than most software engineering is that when we assemble a large circuit out of a component blocks, each of the blocks can potentially influence the behavior of the others. Does this mean that every addition or change to a circuit means that we need to completely re-analyze the entire system? No, because luckily, the ways they interact are limited in a very specific way that we will discuss. It turns out they actually interact through only 2 parameters, current I and voltage V. This leads to a new tool we will develop to help us when describing more complicated/complete circuit models; the concept of equivalence.

Equivalent circuits are used to simplify interactions between circuits. Let's take the simplest case where interactions are only through one pair of nodes. In that case, we just have two possible quantities: the voltage across the nodes and the current flowing through the connections. The relationship between this current and this voltage would then fully define the interactions between the circuits. This is where the idea of equivalence comes in. If we have a circuit that exhibits the same $I-V$ relationship from the standpoint of a pair of nodes, the other circuit (the one you are interacting with) can't tell the difference. The idea of equivalence is to be able to replace one (or both) of the interacting circuits with a simpler circuit that will give us the same overall behavior.

Before we move on, let's clarify what we mean by "equivalent": **Two circuits are equivalent if they have the same $I-V$ relationship.** (An example of an $I-V$ is that of a resistor, i.e., $V = IR$ or $I = \frac{V}{R}$). This is *exactly* what we mean by equivalence; be careful not to overextend this definition or apply others. For example, equivalence tells us nothing about the *power* in a circuit and one should be careful not to assume it does.

Now why is this possibly intuitively? Since voltage and current are governed by a linear relationship for all of the circuit elements we've learned about, and a line can be uniquely determined by exactly two points, we can capture the original circuit with a simplified circuit that has exactly two components: a voltage (or current) source and a resistor.

Definition 15.1 (equivalent circuit): **If we pick two terminals within a circuit, we say that another circuit is equivalent to the original circuit if it exhibits the same $I-V$ relationship at those two terminals.**

Note: From the standpoint of any other nodes in the circuit (i.e. any pairs of nodes), the circuit may or may not be equivalent. Furthermore, looking at the same circuit but examining a different pair of terminals may not produce equivalent $I-V$ relationship.

At a high level, what does it take (at a minimum) to construct a line? We can either use two points along the line, or one point and the slope of the line. Remember, the equivalent circuit of a circuit will have an

identical IV curve, which is a line. In this class, we will construct these equivalent curves using a point and the slope. The two easiest points to collect along a line are the x-intercept (point with 0 current) and the y-intercept (point with 0 voltage).

There are two types of equivalent circuits we will construct: the *Thevenin* and the *Norton*. For the Thevenin equivelant we look at the intersection with the x-axis (zero current); for the Norton, we look at the intersection with the y-axis (zero voltage).

Next we figure out the slope of the line; remember, for an $I \times V$ curve, the slope is equal to the resistance.

We call the first circuit below, containing a voltage source and a resistor the **Thevenin equivalent circuit**; we call the second circuit, containing a current source and a resistor, the **Norton equivalent circuit.** Once we simplify the original circuit to one of the above, we can easily figure out V_{out} no matter what resistor it is connected to on the right. In fact, we can convert **any** circuit into any one of these equivalent forms.

or

15.5 Thevenin Equivalent Circuit

Now how would you figure out V_{Th} and R_{Th} for the Thevenin equivalent circuit?

Concretely, the procedure to solve for the Thevenin equivalent is as follows:

Step 1, find V_{Th}: Connect an open circuit across the two output terminals and measure the voltage across them. This measured V_{OC} equals V_{Th}.

Step 2, find R_{Th}: Zero out any independent sources. Remember, this means voltage sources turn into a wire and current sources turn into an open circuit. Then apply either a test current into the terminal and measure the resultant voltage, or apply a test voltage and measure the resultant current. $R_{Th} = \frac{V_{test}}{I_{test}}$

15.6 Norton Equivalent Circuit

What about solving for the Norton equivalent circuit? First, note that R_{No} is equal to R_{Th}, since the slope of the IV curve is the same. Now, instead of looking at the V axis intercept, we find the intersection with the I-axis: At the intersection with the I-axis, the voltage drop between A and B is zero, which is equivalent to placing a wire between A and B (i.e. shorting A and B). We denote the current through the wire be I_{SC}.

To put it in terms of our standard procedure:

Step 1, find I_{No}: Connect a short circuit across the two output terminals and measure the current through it. This measured I_{SC} equals I_{No}.

Step 2, find R_{No}: Zero out any independent sources. Remember, this means voltage sources turn into a short circuit and current sources turn into an open circuit. Then apply either a test current into the terminal and measure the resultant voltage, or apply a test voltage and measure the resultant current. $R_{Th} = \frac{V_{test}}{I_{test}}$

Note that the second step doesn't change because R_{No} is equal to R_{Th}!

15.7 Equivalence Examples

Here will we find the Thevenin equivalents for a set of simple circuits.

15.7.1 Series Resistors

Consider the schematic:

Let's follow the procedure given above.

Step 1: Note that there is already an open circuit is already connected between terminals a and b. In this case there is no voltage or current source in the circuit. Therefore, the voltage at every node is the same, and therefore, $V_{ab,OC} = 0$. Remember, $V_{ab,OC} = V_{Th}$, so $V_{Th} = 0$.

Step 2: There is no source to zero out in this case. Since it will turn out to be the easier choice, we will apply a test current and measure the resulting voltage, as shown:

There is only one loop, and therefore all the currents in this circuit are the same.

$$V_{R1} = I_{\text{Test}} R1 \tag{1}$$
$$V_{R2} = I_{\text{Test}} R2 \tag{2}$$
$$V_{\text{Test}} = V_{R1} + V_{R2} = I_{\text{Test}} R_1 + I_{\text{Test}} R_2 \tag{3}$$
$$V_{\text{Test}} = (R_1 + R_2) I_{\text{Test}} \tag{4}$$
$$R_{Th} = \frac{V_{\text{Test}}}{I_{\text{Test}}} = R_1 + R_2 \tag{5}$$

We see that equivalent resistance of these two resistors is simply their sum. We call these resistors in **series**. Note that in order to be in series, the resistors have to have the exact same current through them.

15.7.2 Parallel Resistors

Another way to arrange a circuit with two resistors and no voltage source is as follows:

Let's again follow the procedure given above to find our equivalent circuit.

Step 1: Note that there is already an open circuit is already connected between terminals a and b. For the same reason as the prior example, $V_{ab,OC} = 0$ in this case. Therefore, $V_{Th} = 0$.

Step 2: There is no source to zero out in this case. Since it will turn out to be the easier choice, apply a test voltage and measure the resulting current, as shown:

To analyze this circuit, first we notice that the voltage drop over each resistor is equal to V_{Test}. This is because the voltage drop between node a and b is V_{Test}, and each resistor is connected to node a on one side and node b on the other.

First we use the I-V relationship of R_1.

$$V_{\text{Test}} = i_1 R_1 \tag{6}$$

$$i_1 = \frac{V_{\text{Test}}}{R_1} \tag{7}$$

Then we use the I-V relationship of R_2.

$$V_{\text{Test}} = i_2 R_2 \tag{8}$$

$$i_2 = \frac{V_{\text{Test}}}{R_2} \tag{9}$$

Finally, they are combined to calculate the equivalent resistance.

$$I_{\text{Test}} = i_1 + i_2 = \frac{V_{\text{Test}}}{R_1} + \frac{V_{\text{Test}}}{R_2} \tag{10}$$

$$\frac{I_{\text{Test}}}{V_{\text{Test}}} = \frac{1}{R_{Th}} = \frac{1}{R_1} + \frac{1}{R_2} \tag{11}$$

Rearranging this expression gives our final resistance:

$$R_{Th} = \frac{R_1 R_2}{R_1 + R_2} \tag{12}$$

We call these resistors in **parallel**. Note that in order to be in parallel, the voltage across them has to be the same.

This mathematical relationship comes up often enough that it actually has a name: the "parallel operator", denoted $\|$. When we say $x \parallel y$, it means $\frac{xy}{x+y}$. Note that this is a mathematical operator and does not say anything about the actual configuration. In the case of resistors the parallel operator is used for parallel resistors, but for other components (like capacitors) this is *not* the case.

From these analyses, we now have a simple rule to tell if elements are in series or parallel. Series elements will have the exact same current through them due to KCL. Parallel elements will have the exact same voltage across them due to KVL.

15.7.3 Voltage Divider

Now let's apply our analysis above to a voltage divider circuit shown below (which is very similar to the touchscreen). To figure out V_{th}, we solve for V_{oc} in the following circuit

Note that the same current flows through the two resistors. In addition, the voltage drop over the two resistors

sums to V_s, so we can write $V_{Rab} = V_s - V_{AB,oc}$. Therefore, using Ohm's Law:

$$\begin{aligned} I_{Rab} &= I_{Rbc} \\ \frac{V_{Rab}}{R_{ab}} &= \frac{V_{AB,oc}}{R_{bc}} \\ \frac{V_s - V_{AB,oc}}{R_{ab}} &= \frac{V_{AB,oc}}{R_{bc}} \\ \frac{V_s}{R_{ab}} - \frac{V_{AB,oc}}{R_{ab}} &= \frac{V_{AB,oc}}{R_{bc}} \\ V_{AB,oc} &= \frac{R_{bc}}{R_{ab} + R_{bc}} V_s \end{aligned}$$

To figure out R_{th}, we zero out the independent source and apply a test voltage, measuring the resultant current.

We can see that this is the same as the parallel resistor case we examined above: therefore, $R_{Th} = \frac{V_{\text{Test}}}{I_{\text{Test}}} = R_{ab} \parallel R_{bc}$.

This gives us a resulting Thevenin equivalent circuit of:

What if we instead chose the upper two nodes (instead of the lower two nodes) as the two terminals (nodes A and B)? We can follow the same procedure to find an equivalent Thevenin circuit from the standpoint of these new nodes,

After following the same procedure we get the following equivalent circuit:

This is not the same result! In this case, the Thevenin voltages in the two circuits are different. This example shows how different pairs of nodes in the same circuit result in different equivalent circuits. In general, there is no guarantee that circuits will behave in the same way from the standpoint of different pairs of nodes.

15.8 Summary for Finding Equivalent Resistance

In general, there are three ways of finding the Thevenin/Norton equivalent resistance of a circuit. However, some of them only work in certain situations, so need to be used with caution.

1. Zero out all independent sources and apply a V_{test} or I_{test} to calculate the resulting I_{test} or V_{test} respectively. $R_{eq} = \frac{V_{test}}{I_{test}}$.

   This is the method that we described in detail in the examples above, because **this method works for any circuit**. When in doubt, this method is the most reliable.

2. Zero out all independent sources and reduce the entire remaining circuit into a single resistor using the series and parallel resistor formulas that were derived in Sections 15.7.1 and 15.7.2.

This method does not work if there are dependent sources. Remember that only independent sources are zeroed out, and there are no resistor formulas for dependent sources. In addition, some resistor configurations cannot be decomposed into combinations of parallel and series resistances.

3. Calculate V_{th} and I_{no}, $R_{eq} = \frac{V_{th}}{I_{no}}$. This is an efficient method of finding R_{eq} if both the Thevenin and Norton equivalent circuits are being derived. Why does this work? Since the IV relationship is linear, we can calculate the slope (which is the reciprocal of resistance) from any two points. V_{th} and I_{no} are the points where the IV curve crosses the V and I axes, respectively (see the left-hand figure below).

   However, this method does not work if V_{th} and I_{no} do not provide two unique points on the IV curve (see the right-hand figure below). Specifically, **this method only works if there is at least one independent source in the circuit.** When there are no independent sources, $V_{th} = I_{no} = 0$ which does not provide enough information to calculate R_{eq}.

$R_{eq} = \frac{V_{th}}{I_{no}}$

R_{eq} cannot be calculated from I_{no} and V_{th} alone

15.9 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. True or False: For the following circuit, $V_{\text{out}} < V_{\text{in}}$ for any 2 positive resistors R_1 and R_2.

2. Find the equivalent resistance of the following network between a and b.

(a) $(R_1 + R_4) \parallel R_2 \parallel (R_3 + R_5)$
(b) $(R_1 \parallel R_2 + R_4) \parallel (R_3 + R_5)$
(c) $R_1 \parallel R_2 \parallel (R_3 + R_5)$
(d) $R_1 \parallel R_2 \parallel R_3 + R_4 \parallel R_5$

3. True or False: The power generated by the Thévenin equivalent circuit equals the total power generated in the original circuit.

4. Consider the following circuit:

Find R_{th} and V_{th} between a and b.

5. Consider the following circuit:

Find R_{th} and V_{th} between a and b.

6. True or False: Resistors in parallel have an equivalent resistance that is smaller than any of the individual resistances (positive resistance only).

7. What is the voltage between the nodes a and b?

8. For the same circuit above, what is the equivalent Thevenin resistance between the nodes a and b?

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 16

16.1 Introduction to Capacitive Touchscreen

We've seen how a resistive touchscreen works by using the concept of voltage dividers. Essentially, for a resistive touchscreen, we map the value of analog voltage to the position touched. Another way to design a touchscreen is to break the screen down into an array of pixels. At each pixel, we detect whether the finger is touching it or not. This allows the touchscreen to sense multiple touch points at a time. In order to do this efficiently, we need a new element: **capacitors**. One of the nice features of the capacitive touchscreen is that you don't have to bend anything like we did for the resistive touchscreen — the presence or absence of the finger can modify the capacitance directly!

16.2 Capacitor

A **capacitor** is a circuit element that stores charge. Capacitors have an associated quantity called **capacitance**, which represents how much charge can be stored for a given amount of voltage. We usually denote capacitance with C. Let's represent voltage across the capacitor as V_C and the charge stored on the capacitor as Q. This relation is often written as:

$$Q = CV_C. \tag{1}$$

When we draw circuits, the symbol for a capacitor is:

The unit of capacitance is **Farad (F)**. A capacitor with capacitance of 1 F stores 1 C of charge across it when it has 1 V voltage across its electrodes.

Let's examine the defining relationship of a capacitor:

$$Q = CV_C. \tag{2}$$

Differentiating both sides with respect to t and assuming a constant capacitance, we have

$$\frac{dQ}{dt} = \frac{d(CV_C)}{dt} = C\frac{dV_C}{dt}. \tag{3}$$

However, we know that current $I = \frac{dQ}{dt}$. Hence,

$$I = C\frac{dV_C}{dt}. \tag{4}$$

Thus, the current through a capacitor is the product of the capacitance and the rate of change of the voltage across the capacitor. This yields the following graph, with slope equal to C:

An important implication of this is that **current is only flowing through the capacitor if the voltage across the capacitor is changing with time.** If the voltage is no longer changing, then the current through the capacitor will equal 0. Hence, we will now need to solve for the time-varying (dynamic) behavior of circuits with capacitors.

How can we solve for the voltage across the capacitor? In other words, how do we find $V_C(t)$? There are many ways to solve Equation 4, but we will use separation of variables and then integrate both sides, shown below. Note that it's okay if you haven't seen this technique before, it is not required for this class.

$$I = C\frac{dV_C}{dt}, \tag{5}$$

$$Idt = CdV_C, \tag{6}$$

$$\int_0^t Idt = \int_0^{V_C} CdV_C. \tag{7}$$

To integrate the left-hand side, we assume that the current I is constant over time, so we have $\int_0^t Idt = It$. When we integrate the right-hand side, we must account for the initial voltage across the capacitor, $V_C(0)$.

$$It = C(V_C(t) - V_C(0)), \tag{8}$$

$$V_C(t) = \frac{I}{C}t + V_C(0). \tag{9}$$

This yields the following graph, with slope equal to $\frac{I}{C}$. Although Eq. 4 is fully general, Eq. 9 is only valid when the current is constant over time.

Let's see what happens when we connect a capacitor to a voltage source.

We know that initially, charges are going to build up on the two plates — positive charges on the top plate and negative charges on the bottom plate. Once enough charge has been stored, the voltage across the capacitor becomes $V_C = V_S$; because the voltage can no longer increase past this point, the current I flowing through the circuit becomes zero.

16.3 Capacitor Equivalence

Note that when we initially talked about equivalence it was in terms of $I - V$. But what about for capacitors? Now we will specify equivalence in terms of $I - \frac{dV}{dt}$, since the current through a capacitor depends on the rate of change of the voltage across the capacitor. That means we need to apply I_{test} and measure $\frac{dV_{test}}{dt}$ or apply $\frac{dV_{test}}{dt}$ and measure I_{test}. Then we can calculate the equivalent capacitance:

$$C_{eq} = \frac{I_{\text{test}}}{\frac{dV_{test}}{dt}}$$

Just like resistors, we can connect capacitors in series and in parallel. We'll first focus on capacitor-only circuits, which means that we don't need to find Thevenin/Norton equivalent circuits; it will just be a capacitor.

16.3.1 Capacitors in parallel

Given the following circuit:

What is the equivalent capacitance? As with our resistor case, we will try and apply a test voltage to this circuit (the $\frac{dV_{\text{test}}}{dt}$ represents a voltage source whose voltage has a constant rate of change):

$$i_1 = C_1 \frac{dV_{test}}{dt}$$

$$i_2 = C_2 \frac{dV_{test}}{dt}$$

$$i_{test} = i_1 + i_2 = (C_1 + C_2)\frac{dV_{test}}{dt}$$

$$C_1 + C_2 = \frac{i_{test}}{\frac{dV_{test}}{dt}}$$

$$\Rightarrow C_{eq} = C_1 + C_2$$

We can thus reduce our original circuit to:

We find that in general, the equivalent capacitance of capacitors in parallel is the sum of their capacitance. Intuitively, this makes sense because two capacitors with the same voltage across them should be able to store proportionately more charge.

16.3.2 Capacitors in series

Given the circuit:

What is the equivalent capacitance here? As with the resistor case, let's apply a test current and measure the resulting voltage:

From KCL, we know that all of the currents are equal, since there is just one loop for current to flow in:

$$i_{C_1} = i_{C_2} = I_{test}$$

For each capacitor, we plug in our $I - \frac{dV}{dt}$ relationship:

$$i_{C_1} = I_{test} = C_1 \frac{du_1}{dt}$$

$$i_{C_2} = I_{test} = C_2 \frac{d(u_2 - u_1)}{dt} = C_2 \left(\frac{du_2}{dt} - \frac{du_1}{dt} \right)$$

Next, we eliminate u_1 from the equations above and rearrange.

$$\frac{du_1}{dt} = \frac{I_{test}}{C_1} \Rightarrow I_{test} = C_2 \frac{du_2}{dt} - \frac{C_2}{C_1} I_{test}$$

$$I_{test} = \frac{C_2}{1 + \frac{C_2}{C_1}} \frac{du_2}{dt}$$

Finally, we plug in that $u_2 = V_{test}$

$$I_{test} = \frac{C_2}{1 + \frac{C_2}{C_1}} \frac{dV_{test}}{dt},$$

and solve for the equivalent capacitance with $C_{eq} = I_{test} / \frac{dV_{test}}{dt}$

$$\Rightarrow C_{eq} = \frac{C_2}{1 + \frac{C_2}{C_1}} = \frac{C_1 C_2}{C_1 + C_2}$$

Note that this equation for capacitors in *series* is similar to the resistors in *parallel* equation.

We can thus reduce our original circuit to:

16.4 Physics of the Parallel Plate Capacitors

Now that we know the properties of capacitors, let's take a look at the underlying physics. This will be useful when we want to build models for physical structures, such as our capacitive touchscreen. Remember that the physics we teach in this course is not designed to be exact or fully detailed, but is close enough to get us the understanding we need to create circuit models from physical structures.

A simple way to build a capacitor is having any two pieces of conductive material, typically metal, separated by some other material that is not in general conductive (aka an insulator). Capacitors with this structure are called parallel plate capacitors.

If there is voltage across the two pieces of conductor, current can flow into the plates and charges will build up on the surface of the capacitor. As depicted below, when we apply voltage V across the two plates, positive charges build up on the (bottom) surface of the plate connected to the positive terminal and negative charges build up on the (top) surface of the plate connected to the negative terminal. While the charge is flowing onto the plates, current flows in the circuit. Charge never flows across the insulator material between the plates.

The capacitance is set by the dimensions of the conductors and some properties of the material separating them. In particular, the capacitance of a parallel plate capacitor is:

$$C = \varepsilon \frac{A}{d} \tag{10}$$

where A is the area of the surface of the plates facing each other, d is the separation between the two plates (illustrated below), and ε is the "permittivity" of the material between the plates. (Permittivity describes how a material reacts under an external electrical field. A material with higher permittivity has weaker internal electrical field than a material with lower permittivity when the same external electrical field is applied across them.) We can deduce that ε has unit F/m. For example, if the material between the capacitors is air (vacuum), then $\varepsilon = 8.85 \times 10^{-12} F/m$.

When a capacitor is charged, there is voltage across it. What does this mean? This means that there is energy stored in the capacitor. Why is there energy stored? Recall that like charges repel each other. So for example if we take a positive charge and move it closer to another positive charge, we need to exert force and thus supply energy to do so. Let's now ask the question: when the capacitor is fully charged, how much energy is stored in it? We know from previous lectures that the energy required to store an additional dq amount of charge when the voltage across the capacitor is V_C is

$$dE = V_C dq. \tag{11}$$

We also know that $dq = CdV_C$. With this in mind, we have

$$dE = V_C (CdV_C) = CV_C dV_C. \tag{12}$$

Integrating both sides, we have

$$\int_0^E dE = \int_0^V CV_C dV_C = C \int_0^V V_C dV_C \tag{13}$$

$$E = \frac{1}{2} CV_C^2. \tag{14}$$

Hence, the energy stored in the capacitor after it's fully charged is $\frac{1}{2}CV^2$ where V is the voltage applied across the capacitor. The charges across the capacitor cannot just disappear for no reason, and therefore the energy continues to be stored - that means we can use this element as a tool to design circuits for measuring stored charge/energy.

Let's return to our capacitor equivalence models and try to gain some more physical intuition.

Physical Intuition of Parallel Capacitors:
Suppose we combine two capacitors in parallel as follows (the left parallel plate capacitor C_1 has a surface area A_1 and a separation d, and the right parallel plate capacitor C_2 has a surface area A_2 and a separation d)

If we look at the combined capacitor above, its capacitance is equal to

$$C_{eq} = \varepsilon \frac{A_1 + A_2}{d}, \tag{15}$$

which is equal to the sum of the capacitance of the two capacitors

$$C_{eq} = \varepsilon \frac{A_1 + A_2}{d} = \varepsilon \frac{A_1}{d} + \varepsilon \frac{A_2}{d} = C_1 + C_2. \tag{16}$$

This matches the equivalence equation for parallel capacitors! In other words, connecting capacitors in parallel is analogous to summing the surface area of the two plates.

Physical Intuition of Series Capacitors:
If we take the following capacitor and put a (very thin) piece of conductor plate in between, then it acts the same as two capacitors connected in series

Assume that the surface area is A. Intuitively, we know that the equivalent capacitance should be smaller than both C_1 and C_2 since the plates are farther from each other. We know that $C_1 = \varepsilon \frac{A}{d_1}$ and $C_2 = \varepsilon \frac{A}{d_2}$. Let the capacitance of the combined capacitor be C_{eq}. It satisfies

$$\frac{1}{C_{eq}} = \frac{1}{\varepsilon} \frac{d_1 + d_2}{A} = \frac{1}{\varepsilon} \frac{d_1}{A} + \frac{1}{\varepsilon} \frac{d_2}{A} = \frac{1}{C_1} + \frac{1}{C_2}. \tag{17}$$

Hence, the equivalent capacitance is equal to

$$C_{eq} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} = \frac{C_1 C_2}{C_1 + C_2}. \tag{18}$$

Notice that the examples above keep one physical dimension constant between the capacitors. This is not really necessary to arrive at these expressions for series and parallel capacitors. The expressions for equivalent capacitance that we formally derived in Section 16.3 work for capacitors of arbitrary shape/values.

16.5 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Given a parallel plate capacitor, you can increase the effective capacitance in the following way(s):

   (a) Increasing the overlapping area of the two plates
   (b) Decreasing the distance between the two plates
   (c) Using an insulator with a higher permittivity
   (d) All of the above

2. True or False: Capacitors in series have an equivalent capacitance that is smaller than any of the individual capacitances (assuming that all capacitances are positive capacitance).

3. True or False: Capacitors in parallel have an equivalent capacitance that is smaller than any of the individual capacitances (assuming that all capacitances are positive).

4. True or False: Given the current through a capacitor, is it possible to determine the voltage across it at time t?

5. Consider the circuit below. Find an expression for $v_{\text{out}}(t)$ in terms of I_s, C_1, C_2, C_3, C_4, and t. Assume that all capacitors are initially uncharged.

6. Find the equivalent capacitance between the terminals a and b in terms of C.

7. What is the total energy stored in capacitors in the circuit below?

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 17

17.1 Capacitive Touchscreen

Viewing the physical structure corresponding to one pixel on the capacitive screen, we want to be able to tell if there is a finger touch on top of the pixel.

Our finger touches the dielectric at the top, C_2 is the capacitance between our finger (F) and electrode E_2, and C_1 is the capacitance between our finger (F) and electrode E_1. Now, we can draw the equivalent circuit model corresponding to the pixel we're looking at:

A smart phone can only sense something that is between electrodes E_1 and E_2 as it is impractical to physically connect a measuring device to our finger. So, we redraw our circuit as follows:

Since we care about the circuit behavior at nodes E_1 and E_2, we can further simplify the circuit by replacing the series combination of C_1 and C_2 with some capacitance $C_\Delta = \frac{C_1 C_2}{C_1 + C_2}$.

If we consider the parallel combination of C_0 and C_Δ as some equivalent capacitance C_{eq}, we know that C_{eq} will have some value greater than C_0 because $C_{\text{eq}} = C_0 + C_\Delta$ and $C_\Delta > 0$.

When a finger presses on top of a pixel, the equivalent capacitance between E_1 and E_2 will become greater than the default capacitance without a finger pressing on top. This characteristic allows us to tell if there is a finger pressing on top of each pixel on our 2D capacitive touchscreen.

Given 4 pixels and their corresponding locations, we can measure the capacitance at (x_1, y_1) to know if there is a finger pressing on top of pixel 1. Similarly, to know if there is a finger pressing on top of pixels 2, 3, 4, we can measure the capacitances at (x_1, y_2), (x_2, y_1), (x_2, y_2). If we sense an increase in the capacitance at the pixel location, we know there is a finger pressing on top of this pixel.

17.2 Capacitance Measurement

To detect a finger touch, we need to be able to measure capacitance. There are various ways you can use to measure capacitance. In this lecture, we will cover one specific method to measure capacitance. First, let's review some basic physics for a capacitor. As you know:

$$I = C \times \frac{dV}{dt}, \tag{1}$$

$$Q = C \times V. \tag{2}$$

We can build a circuit with a current source I_s and a capacitor C. If we charge the capacitor C using the current source I_s for a time interval t,

we can then find V to be:

$$V = \frac{I_s \times t}{C}. \tag{3}$$

V measured depends on the amount of charge $Q = I_s \times t$ and the capacitance value C. So, if we charge a capacitor using some current source for a finite time period, we can map the voltage across the capacitor to capacitance. However, building a current source is not that easy, so we have to come up with some workarounds to measure capacitance.

- **Attempt #1** Our first attempt is connecting the capacitor to a voltage source and measuring the voltage across the capacitor directly:

  We can immediately see that no matter how C_{eq} changes, the voltage measured V_{out} is always going to be equal to V_s.

- **Attempt #2** The issue in our first attempt is that after we charge C_{eq} by the voltage source, the voltage across C_{eq} will stay at V_s. In order to use the voltage source as the power source in our circuit, we must disconnect V_s from C_{eq} after charging it, and then somehow discharge C_{eq} so the voltage across it, V_{out}, is different from V_s and the difference must depend on the value of C_{eq}.
  How to disconnect V_s from C_{eq}? We can add switches to the circuit. A switch has two states, at its on state, the switch is closed and becomes a perfectly conducting wire with zero resistance, and when the switch is turned off, it becomes an open circuit.

To discharge C_{eq}, we will connect it to a reference capacitor C_{ref} in parallel so some of the charge on C_{eq} is transferred to C_{ref}, and we will measure the voltage V_{out} across them.

There are two cases (phases) of this circuit.

- **Phase 1:** s_1 is closed and s_2 is open.

  In phase 1, C_{eq} is connected to voltage source V_s and C_{ref} is not connected to anything.

- **Phase 2:** s_1 is open and s_2 is closed.

  In phase 2, C_{eq} is disconnected from V_s and C_{ref} is connected to C_{eq}. However, there is one serious problem with this circuit, since we are not sure what is the initial charge of C_{ref}, we cannot perform any circuit analysis. Our second attempt fails because the initial condition of C_{ref} is not clear.

- **Attempt #3** We learned from our second attempt that we must somehow know the initial condition of C_{ref}. How do we know the initial condition of C_{ref}? It turns out that we can acquire this missing piece of information by adding another switch s_3!

  - **Phase 1:** s_1 is closed, s_2 is open, s_3 is closed.

    In this phase, the voltage across C_{eq} is V_s, and the voltage across C_{ref} is zero.

  - **Phase 2:** s_1 is open, s_2 is closed, s_3 is open.

    In this phase, the voltage across C_{eq} is the same as the voltage across C_{ref}.

17.3 Circuit Analysis with Charge Sharing

How do we analyze this circuit? We know that the total charge of the two capacitors must be preserved between phase 1 and phase 2, this indicates that:

$$Q_{\text{phase 1}} = Q_{\text{phase 2}}. \tag{4}$$

The total charge in each phase can be calculated by summing up Q_{eq} and Q_{ref} for each phase.

$$Q_{\text{phase 1}} = Q_{\text{eq}} + Q_{\text{ref}} = C_{\text{eq}} \times V_s + C_{\text{ref}} \times 0\text{V}. \tag{5}$$

$$Q_{\text{phase 2}} = Q_{\text{eq}} + Q_{\text{ref}} = C_{\text{eq}} \times V_{out} + C_{\text{ref}} \times V_{out}. \tag{6}$$

From equations (5) and (6) , we have:

$$C_{\text{eq}} \times V_s + C_{\text{ref}} \times 0\text{V} = C_{\text{eq}} \times V_{out} + C_{\text{ref}} \times V_{out}. \tag{7}$$

Therefore, we can derive the following equation for V_{out}:

$$V_{out} = \frac{C_{\text{eq}}}{C_{\text{eq}} + C_{\text{ref}}} \times V_s. \tag{8}$$

This particular circuit analysis approach we have used is called "charge sharing". Charge sharing refers to the sharing of charge among capacitors. We can use the charge sharing approach to solve many capacitor problems.

Now we can solve for C_{eq} given V_{out}. Recall that we only want to figure out whether or not our finger is touching this pixel, i.e. is C_{eq} greater than its default value when there is no finger touch?

17.4 Comparator & Op-amp Basics

We want to figure out if there is a finger pressing on top of a pixel from V_{out}. In order to do so, we can use a comparator. Before explaining how a comparator works, we will first introduce the concept of op-amp.

Op-amp (operational amplifier), by definition, is an amplifier that can amplify something small into something much bigger. For example, a speaker is an audio amplifier, if you connect your smart phone to a speaker, it can generate sounds much louder than your phone can. The circuit symbol for an op-amp is:

We have two input terminals named V_+ and V_-, two power supply terminals named V_{DD} and V_{SS}, and one output terminal named V_{out}. The op-amp circuit can transform a voltage signal $(V_+ - V_-)$ into V_{out}. For the op-amp to function correctly, we need to connect the two external voltage sources V_{DD}, V_{SS} to the op-amp (typically $V_{DD} > 0$ and $V_{SS} \leq 0$). The symbol actually represents the following equivalent circuit:

In the circuit we see a new symbol — a diamond with +/- signs inside. This represents a voltage-controlled voltage source where the voltage across it depends on the voltage(s) in other parts of the circuit. (An introduction to dependent sources can be found in Note 15 Section 15.2.) In this case, the voltage across the top voltage-controlled voltage source is $A\,(V_+ - V_-)$ where A is a constant. For good op-amps, the constant A term is very large – approaching infinity, but the output voltage cannot go to infinity and is limited by the two external voltage supplies ($V_{SS} \leq V_{out} \leq V_{DD}$). The bottom voltage-controlled voltage source is dependent on the power supplies of the op-amp. If V_+ and V_- are exactly equal, V_{out} is the average of the supply voltages V_{DD} and V_{SS}. Note that the dependent voltage sources are referenced to V_{SS}, not ground.
Equivalently, we have the following plot that describes the relationship between V_{out} and $(V_+ - V_-)$.

For any ideal op-amp, when $V_+ < V_-$ ($V_+ - V_- < 0$), we have V_{out} equals V_{SS}; when $V_+ - V_- > 0$, we have V_{out} equals V_{DD}. If we zoom in around $V_+ - V_- = 0$ by a million times, we will be able to see there is actually some finite slope associated with the transition region.
Now, we add an op-amp to the circuit model we built to extract C_{eq}.

Since the two input voltages to the op-amp are the output of the circuit V_{ana}, and our threshold voltage V_{ref}, if the voltage V_{ana} is greater than V_{ref}, the output voltage V_{out} will be 3.3 V. If $V_{\text{ana}} < V_{\text{ref}}$, V_{out} will be 0V. Recall for V_{ana}, we have

$$V_{\text{ana}} = \frac{C_{\text{eq}}}{C_{\text{eq}} + C_{\text{ref}}} \times V_s. \tag{9}$$

When there is touching, $C_{\text{eq}} = C_0 + C_\Delta$ in which $C_\Delta > 0$, we have:

$$V_{\text{ana, touch}} = \frac{C_0 + C_\Delta}{C_0 + C_\Delta + C_{\text{ref}}} \times V_s \tag{10}$$

When there is no touching:

$$V_{\text{ana, no-touch}} = \frac{C_0}{C_0 + C_{\text{ref}}} \times V_s \tag{11}$$

For any finite capacitance $C_\Delta > 0$, $V_{\text{ana, touch}}$ is always greater than $V_{\text{ana, no-touch}}$.
V_{ana} is connected to the positive voltage input (V_+) of the op-amp, V_{ref} is connected to the negative voltage input (V_-) of the op-amp. If we set up V_{ref} such that:
(1) $V_{\text{ana, touch}} > V_{ref}$, i.e. $V_+ - V_- > 0$, V_{out} will be equal to 3.3 V when there is touching.
(2) $V_{\text{ana, no-touch}} < V_{ref}$, i.e. $V_+ - V_- < 0$, V_{out} will be equal to 0 V when there is no touching.
Now we have built a **comparator** circuit that can compare V_{ana} against V_{ref} using an op-amp!
How should we set up the threshold voltage V_{ref}? Intuitively, we want to set it to be halfway between $V_{\text{ana, no-touch}}$ and $V_{\text{ana, touch}}$. This actually allows our comparator circuit to be robust against non-idealities, we will have the same amount of margin for any V_{ana} that deviates from its ideal values, either from $V_{\text{ana, touch}}$ or from $V_{\text{ana, no-touch}}$.
A second important question is **how should we set up C_{ref}?** Recall that $V_{ana} = \frac{C_{\text{eq}}}{C_{\text{eq}}+C_{\text{ref}}} \times V_s$. We realize that V_{ana} is associated with C_{ref}. As long as we do not set C_{ref} to some value much greater (for example, 1000 times greater) than C_{eq} so that V_{ana} becomes too small (for our op-amp to pick up the signal), we can get some reasonable V_{ana}, which then allows us to set up V_{ref} according to $V_{\text{ana, touch}}$ and $V_{\text{ana, no-touch}}$.
Practically, we will set up one of C_{ref}, V_{ref} first, and then set up the other value according to $V_{\text{ref}} = \frac{1}{2}(V_{\text{ana, touch}} + V_{\text{ana, no-touch}})$.

17.5 Comparators in Practice

In the previous section, we used an ideal op-amp to function as a comparator for our capacitive touchscreen design. Even though for ideal op-amps which have infinite open loop gain, A, this is okay to do, in practice

circuit designers rarely use op-amps as comparators. Since op-amps are optimized for linear signal amplification with output voltages always staying between the supply rails (V_{DD}, V_{SS}), and for stability, they tend to be very slow when having to resolve a small input difference into a large digital output (V_{DD} or V_{SS}).

Dedicated comparators, by contrast, are special, distinct components designed to output only the supply voltages, (V_{DD} or V_{SS}), and are preferred to op-amps for their faster operation. The plot of input and output voltages for a comparator is identical to that of an ideal op-amp in the comparator configuration, shown in the previous section.

For any ideal comparator, (just like for any ideal op-amp), when $V_+ - V_- < 0$, we have V_{out} equals V_{SS}; when $V_+ - V_- > 0$, we have V_{out} equals V_{DD}.

The circuit symbol for a comparator is shown below:

In summary, in the context of this class we will say that *ideal* op-amps can operate as *ideal* comparators. Keep in mind though, that the actual circuit design and implementation of these two circuit blocks (more of which you can see in classes like EE 105, EE140 and EECS151) are fundamentally different.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 17B

This handout outlines a detailed algorithmic procedure that solves two-phase switch capacitor circuit problems.
Goal: Find the voltage of all floating nodes in a 2-phase switched capacitor circuit at the end of phase 2.
Main Principle: Charge Conservation
Useful Definition: A **floating node** is a node out of or into which no charge can flow. You can identify those nodes as the nodes connected only to capacitor plates, or, as you'll see later in the course, op-amp inputs or comparator inputs.
Notation: In each phase, the circuit has a different configuration by opening/closing different switches. In the circuit diagram, a switch labeled ϕ_1 will be closed in phase 1 and open in phase 2, and a switch labeled ϕ_2 will be open in phase 1 and closed in phase 2. Generally, a switch labeled $\phi_i (i = 1, 2, 3, \ldots)$ will be closed in phase i and open in all other phases.

We present this algorithm by applying it to two examples. Note, that this method can be extended to circuits with more than 2 phases.

Example 1: For the switch capacitor circuit below, calculate the value of all node voltages at the end phase 2, as a function of the voltage source V_s and the capacitors C_1, C_2.

Step 1: Label the voltages across all the capacitors. Choose whichever direction (polarity) you want for each capacitor - this means you can mark any one of the plates with the "+" sign, and then you can mark the other plate with the "−" sign. Just make sure you stay consistent with this polarity across phases.

Step 2: Draw the equivalent circuit during both phases. For this problem, phase 1: ϕ_1 closed, ϕ_2 open, and phase 2: ϕ_1 open, ϕ_2 closed. Also, label all node voltages on the circuit for both phases. No need to try and maintain the same names, since certain nodes of the phase 1 circuit might be merged or split in phase 2.

Phase 1:

Phase 2:

Step 3: Identify all "floating" nodes in your circuit during phase 2. A floating node is a node out of or into which no charge can flow. You can identify those nodes as the nodes connected only to capacitor plates, op-amp inputs or comparator inputs. These will be the nodes where we apply charge sharing.

In this case the only node that is floating during phase 2 is node u_3. (Node u_2 is connected to the voltage source, i.e. $u_2 = V_s$, and the 3rd node is the ground node).

Step 4: For steps 4-6 we will **examine each phase 2 floating node individually**. Pick a floating node from the ones you found in step 3 and identify all capacitor plates connected to that node during phase 2. Then, calculate the charge on each of these plates in the steady state of phase 1.
To do so, identify all nodes in your circuit during phase 1. Label all node voltages, and write the voltages across each capacitor as functions of node voltages (step 2 should help you with that). Do this according to the polarities you have selected. Then the charge is found as $Q = CV_C$ (where V_C is the voltage *across* a capacitor).
Careful: The plate marked with the "$-$" sign will have $Q = -CV_C$ and the plate marked with the "$+$" sign will have $Q = CV_C$ stored onto them.
Careful: We assume here that you know all node voltages during phase 1. If you don't, before starting this procedure calculate the node voltages you need using one of the previously introduced circuit analysis techniques (most likely KVL will do the job).

Looking at our single floating node we can see that the "+" plates of C_1 and C_2 are connected to it during phase 2. Let's calculate the charge on these plates connected to the floating node u_3 during **phase** ϕ_1, denoted

by $Q_{u_3}^{\phi_1}$.

$$\begin{aligned} Q_{u_3}^{\phi_1} &= V_1^{\phi_1} C_1 + V_2^{\phi_1} C_2 \\ &= (V_s - 0)C_1 + 0 \\ &= V_s C_1 \end{aligned}$$

Step 5: Find the total charge on each of the floating nodes in the steady state of phase 2 as a function of node voltages. Use the same process as in Step 4, but this time using the node voltages during phase 2 to write the voltages across each capacitor. Make sure you kept the polarity same and pay attention to the sign of each plate.

$$\begin{aligned} Q_{u_3}^{\phi_2} &= V_1^{\phi_2} C_1 + V_2^{\phi_2} C_2 \\ &= (u_3 - u_2)C_1 + (u_3 - 0)C_2 \\ &= (u_3 - V_s)C_1 + u_3 C_2 \end{aligned}$$

Step 6: The charge in the steady state of phase 1 we calculated in Step 4, $Q_{u_3}^{\phi_1}$, is the initial charge of phase 2. Charge must be conserved at a floating node, so the steady state charge of phase 2, $Q_{u_3}^{\phi_2}$, should equal the initial charge $Q_{u_3}^{\phi_1}$. We can then write in terms of the unknown(s).

$$\begin{aligned} Q_{u_3}^{\phi_1} &= Q_{u_3}^{\phi_2} \\ V_s C_1 &= (u_3 - V_s)C_1 + (u_3 - 0)C_2 \\ u_3 &= \frac{2C_1}{C_1 + C_2} V_s \end{aligned}$$

Step 7: Repeat steps 4-6 for every floating node. This will give you one equation per floating node (i.e. if you have m floating nodes you will have m equations). You can then solve the system of equations to find the node voltages during phase 2 (unknowns). It should have a unique solution!

In this problem we did not go through step 7 since we only had one floating node during phase 2. This means we have only one unknown node voltage (u_3) for which we solved using our single equation from Step 6. We will be using step 7 in our second example!

Example 2: Find the voltage V_{out} across capacitor C_x at the end of phase 2, as a function of C, C_x, and V_s. Assume that $C_1 = C_2 = C$.

Step 1: Capacitor voltage labelling.

Step 2: Redrawing the circuit during both phases.

Phase 1:

Phase 2:

Step 3: Identify all floating nodes during phase 2.

In this problem the floating nodes are u_2 and u_3 since no current is flowing into or out of them (they are only connected to capacitor plates).

Step 4: Calculate the total charge accumulated on phase 2 floating nodes during phase 1.

Let's look at u_2 first: "$-$" plate of C_1 and "$+$" plate of C_2 are connected to that node during phase 2. Calculating the charge stored in them during phase 1 we get:

$$\begin{aligned}
Q_{u_2}^{\phi_1} &= -V_1^{\phi_1} C_1 + V_2^{\phi_1} C_2 \\
&= -(u_1 - 0)C_1 + (u_1 - 0)C_2 \\
&= -u_1 C_1 + u_1 C_2 \\
&= -V_s C_1 + V_s C_2 \\
&= 0
\end{aligned}$$

Where in the last step we used that $C_1 = C_2 = C$.

Step 5: Calculate the total charge accumulated on phase 2 floating nodes during phase 2 as a function of node voltages.

$$\begin{aligned}
Q_{u_2}^{\phi_2} &= -V_1^{\phi_2} C_1 + V_2^{\phi_2} C_2 \\
&= -(0 - u_2)C_1 + (u_2 - u_3)C_2 \\
&= (2u_2 - u_3)C
\end{aligned}$$

Step 6: Equate charge calculated during phases 1 and 2.

$$\begin{aligned}
Q_{u_2}^{\phi_1} &= Q_{u_2}^{\phi_2} \\
0 &= (2u_2 - u_3)C_2
\end{aligned}$$

$$u_2 = \frac{u_3}{2} \tag{1}$$

Step 7: Repeat steps 4-6 for every floating node.

Let's repeat Steps 4-6 for node u_3:

Step 4:
u_3 is connected to the "$-$" plate of C_2 and "$+$" plate of C_x. Charge during phase 1:

$$\begin{aligned}
Q_{u_3}^{\phi_1} &= -V_2^{\phi_1} C_2 + V_x^{\phi_1} C_x \\
&= -(u_1 - 0)C_2 + (0 - 0)C_x) \\
&= -u_1 C_2 \\
&= -V_s C
\end{aligned}$$

Step 5: Charge during phase 2:

$$\begin{aligned} Q_{u_3}^{\phi_2} &= -V_2^{\phi_2} C_2 + V_x^{\phi_2} C_x \\ &= -(u_2 - u_3)C_2 + (u_3 - 0)C_x) \\ &= -u_2 C_2 + u_3(C_x - C_2) \end{aligned}$$

Step 6: Equate the charge from steps 4 and 5:

$$Q_{u_3}^{\phi_1} = Q_{u_3}^{\phi_2}$$

$$-V_s C = -u_2 C + u_3(C + C_x) \tag{2}$$

Now we have two unknowns, u_2 and u_3, and two equations, Eq. (1) and (2). Finally, solve the system of equations (1), (2): Plugging Eq. (1) into Eq. (2) we get:

$$\begin{aligned} -V_s C &= -\frac{u_3}{2} C + u_3(C + C_x) \\ u_3 &= -\frac{C}{\frac{C}{2} + C_x} V_s = -\frac{2V_s}{1 + \frac{2C_x}{C}} \end{aligned}$$

Plugging this solution for u_3 into Eq. (1), we get:

$$u_2 = \frac{u_3}{2} = -\frac{V_s}{1 + \frac{2C_x}{C}}$$

From our circuit diagram, $V_{out} = u_3$.

Important Notes:

1. A node that is floating during phase 2 is not necessarily floating during phase 1. In fact, it could have been two separate nodes during phase 1 depending on how your switches are configured. We only care about the floating nodes during phase 2 since those are the nodes with unknown voltages.
   What about floating nodes during phase 1? You should be able to calculate the voltage of these nodes based on the initial condition of the circuit and the circuit techniques that you have learned so far (most likely KVL).

2. When handling charge sharing problems you should avoid using any parallel or series capacitance formulae that you have learned. Any simplification of the circuit might lead to mistakes since the circuit in different phases is configured differently and some capacitors placed in series during one phase may be in parallel or even not connected at all in another phase!

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 18

18.1 Introduction: Op-amps in Negative Feedback

In the last note, we studied how to use an op-amp as a comparator. We did this by assuming the internal gain of an op-amp, A, is very large (approaching infinity) and modelling the output as a piecewise function between the two supply rails V_{DD} and V_{SS}. In reality though, this gain, A, is finite, so one may consider using this as an *amplifier* to amplify the input difference $(V_+ - V_-)$. However, since the gain, A, is very large (between 1000-100000 depending various factors), in practice, any input V_+ and V_- with a slight difference between them will result in an output $V_{out} = A(V_+ - V_-)$ that is saturated or *railed* to either V_{DD} or V_{SS}.

In this note, we will look at how to use op-amps to scale the input difference $(V_+ - V_-)$ without causing the op-amp to rail. We will first motivate this with a design example. Then, we will show techniques using **negative feedback** to build blocks that depend only on ratio of physical quantities as long as the op-amp has high enough (but not precise) internal gain.

18.2 Design example – Digital to Analog Converter (DAC)

A digital to analog converter (DAC) is a component that translates digital input signals into output analog voltages. Like an op-amp, DACs have supply rails V_{DD} and V_{SS}, and the output voltage will be restricted between those values, i.e the maximum output voltage is V_{DD} and the minimum is V_{SS}. Commonly, we model DACs as simply a voltage source with a series resistor. The symbol is shown below (left), along with an example model circuit (right).

Consider a song stored on your computer, encoded in some digital values (some sequence of 1s and 0s). We would like to convert it to some analog (continuous value) signal to play on a speaker. Below is a simple depiction of our speaker, which takes an input signal between 0 V and 10 V (as indicated by the arrow) and outputs some corresponding sound. We can model our speaker simply as a small resistor, 8 Ω for example.

To design an audio system, the simplest thing we can do is directly connect the DAC to the speaker. The DAC takes in digital bits and converts them into an analog signal, then this signal is fed into a speaker.

Using the models from above, we can draw this complete circuit model:

We want the input voltage to the speaker to be between 0V and 10V (in order to make the speaker loud enough). To reach that maximum and minimum voltage, we may try to make $V_{DD} = 10V$ and $V_{SS} = 0V$ for the DAC. Unfortunately, these are not always available to us. Depending on the system, we may be restricted to using only $5V$, $3.3V$, or even less.

For this example, we will use $V_{DD} = 3.3\,\text{V}$ and $V_{SS} = 0\,\text{V}$, realistic values for many computer systems. This means the maximum voltage the DAC can produce is 3.3V, and the minimum voltage the DAC can produce is 0V. From the circuit model above, we see that:

$$V_{speaker} = \frac{8}{8+1000} \times V_{TH} \tag{1}$$

which is much smaller than V_{TH}. However, to match the speaker's input voltage range, we actually want to be able to map voltages from 0 to 3.3V at the DAC output to voltages from 0 to 10V at the speaker input, so $V_{speaker}$ should be three times $(10/3.3)$ larger than V_{TH} to scale the voltage correctly. This circuit isn't going to do what we would like!

Instead of directly connecting the DAC and speaker, we need some intermediate circuit. This circuit needs to accomplish three things: it must provide a gain of roughly 3, it must not affect the output voltage of the

DAC (so that it does not change the signal), and it must be able to supply any current required by the speaker (so that the speaker does not affect the expected output voltage). So we want something like the following

This looks just like the internal of an op-amp! Now notice that based on what we know so far about op-amps as comparators, we can't just scale the voltage linearly as we we would have wished: if we connect an ideal op-amp (infinite internal gain) with $V_{DD} = 10V$ and $V_{SS} = 0V$, the output voltage would either be $10V$ or $0V$, but not something in between. To achieve intermediate values and use the op-amp as an amplifier, we need another tool, which we will introduce in the next section.

18.3 Negative feedback

Negative feedback is used just about everywhere, including electronics, biology, mechanics, robotics, and more. Many systems and processes will drift into unknown or undesirable states if left unattended or uncontrolled. Negative feedback occurs when some function of the output of a system is fed back into the input, in order to correct the system and keep the output at some controlled value. Let's turn this high level description into a more mathematical one.

Concretely, we want to get a certain known gain out of our op-amp. Currently we have an op-amp with some very large uncertain internal gain. We can describe this problem using a block diagram; a collection of drawings (mathematical in nature) that operate on quantities of interest using simplified representations.

Let's take a look at a generic block diagram for negative feedback systems.

In this diagram, the arrows represent the direction of signals. If an arrow is going into a block, it is an input to that block. If the arrow points away from a block, it is an output of that block. The blocks labelled A and f are multiplier elements, i.e. they take some input x and output that input multiplied by some factor Ax. The circle with a $+$ symbol is an addition element, i.e. it takes some inputs a and b and outputs the sum

$a+b$. With addition elements, we may add signs to the inputs to signal if we want to add or subtract them. For example, in this diagram, the $-$ symbol next to the feedback signal signifies we will be subtracting it, i.e. error = input $-$ feedback.

In words, this diagram takes the difference between the input and the *feedback*, which is a scaled version of the output (i.e. f times the output). This difference is called the *error*, and we apply gain A on it to again produce the output.

The "negative" aspect of this feedback comes from taking the difference between the input and the feedback signal. Observe that if the feedback signal for some system is increased, the error (input - feedback) signal will decrease (move down), which then causes the output to go down as well, i.e., the loop has the ability to suppress the original change in the feedback signal!

Now we can get an intuitive idea of how negative feedback can be useful. When we want to get a system to have a desired output, negative feedback loops can help re-adjust to the value of the desired output when the output is too high or too low relative to the target value.

Now let's see how negative feedback loops can be realized in op-amps. Consider the following circuit:

We want to map the components here to the feedback block diagram above. We have an input signal V_{in} and an output V_{out}. The voltage divider formed by R_1 and R_2 multiplies our output V_{out} to give us the feedback signal V_{fb}, which is connected to the $-$ terminal of the op-amp. This is highlighted in red. Recall that our op-amp takes the difference between the inputs and amplifies that difference at the output. Thus the op-amp performs both the difference operation between the input and feedback signals internally, and then amplifies the error (difference) signal to create the output.

To help analyze circuits of this kind, we will introduce two "golden rules" that could make our lives simpler.

18.4 Golden Rules

Much like we can use our KVL and KCL rules and Ohm's law to solve circuits with sources and resistors, these op-amp golden rules help us solve circuits that include op-amps.

Recall the op-amp symbol and equivalent circuit, assuming $V_{SS} = -V_{DD}$. For simplicity, we will usually not draw the supply terminals in negative feedback circuits and assume that the output voltage is between whatever V_{DD} and V_{SS} we choose.

For an ideal op amp, i.e. an op-amp with internal gain approaching infinity, the **"golden rules"** are

- (1) **The current into both the input nodes is zero, i.e.** $I_+ = I_- = 0$**.** Consider the equivalent circuit for the op-amp above. Notice that there is no closed circuit connected to the positive or negative input terminal of the op-amp. Thus, no current can flow into the positive or negative input terminal. **This rule holds regardless of whether there is negative feedback or not.**
- (2) **When there is negative feedback, in steady state, the input and feedback voltages will be equal, i.e.** $V_+ = V_-$. When a circuit is first connected in negative feedback, there may be initially be some error signal. The feedback loop will try to correct this by changing the output voltage. This leads to a change in the feedback voltage, which leads to a change in the error signal, leading to another change in the output voltage, etc. Eventually, the error difference will go to zero, and the output will reach a steady state. **This rule only holds when there is negative feedback and holds only for large intrinsic op-amp gain,** $A \to \infty$**.** We will explore this in more depth in the next section. When we analyze circuits using this rule, we will assume that the op-amp gain is large and that the circuit has already reached steady state.

Now let's use the golden rules to analyze the circuit we saw earlier:

First we apply KCL at the junction between R_1 and R_2 to get the following relationship

$$I_1 = I_2 + I_-. \tag{2}$$

By the first golden rule, we know that $I_+ = I_- = 0$. Hence,

$$I_1 = I_2. \tag{3}$$

Now let's apply the second golden rule, $V_+ = V_-$. Using this, we have

$$V_{in} = V_{fb}. \tag{4}$$

Now we can solve for I_2 using Ohm's law, $V_{fb} = I_2 R_2$, hence

$$I_2 = \frac{V_{fb}}{R_2}. \tag{5}$$

Using Ohm's law on R_1, we also have $V_{out} - V_{fb} = I_1 R_1$. Hence,

$$I_1 = \frac{V_{out} - V_{fb}}{R_1}. \tag{6}$$

However, we know that $I_1 = I_2$, which gives us the following relationship

$$I_1 = \frac{V_{out} - V_{fb}}{R_1} = \frac{V_{fb}}{R_2} = I_2, \tag{7}$$

which is equivalent to

$$I_1 = \frac{V_{out} - V_{in}}{R_1} = \frac{V_{in}}{R_2} = I_2. \tag{8}$$

Moving terms around, this gives us

$$V_{out} = V_{in}\left(1 + \frac{R_1}{R_2}\right). \tag{9}$$

Notice that here the ratio $\frac{V_{out}}{V_{in}}$ only depends on the ratio $\frac{R_1}{R_2}$. This is a great property since it is rather difficult to produce resistors with a particular absolute resistance. Since R_1 and R_2 are always positive, the coefficient $1 + \frac{R_1}{R_2}$ is always positive. We call this the **non-inverting amplifier**.

18.5 Second golden rule revisited

Recall that in the last section, we stated the second golden rule $V_+ = V_-$, i.e., the voltage potential at the positive input terminal (relative to ground) and the voltage at the negative input terminal (relative to the same ground) are the same when there is negative feedback. Now we would like to justify why this is the case. We return to the block diagram we drew earlier for a general negative feedback loop, but now focus on its application in op-amp circuits.

Let's build some inituition of what happens if something changes and how we may arrive back in steady state.

Suppose we are initially in steady state, and V_{in} suddenly increases. Since $V_{error} = V_{in} - V_{fb}$, V_{error} will also increase. Then since A is a positive number, V_{out} also goes up, which causes V_{fb} to go up. This in turn causes the magnitude of V_{error} to go down, meaning that the system is going to stablize itself. [1]

Now what if we change the minus sign to a plus sign in the diagram, i.e., add the feedback signal instead of subtract it? This changes the system into a positive feedback system. With a similar logic, you could verify that if V_{in} goes up, V_{fb} goes up, but V_{error} goes up, which further causes V_{out} to go up. We see that it is not possible to stabilize the system.

Let's look at the negative feedback op-amp circuit we've seen earlier,

We know that when V_{in} increases, V_{out} also increases since $V_{out} = A\left(V_{in} - V_{fb}\right)$. When V_{out} increases, $V_{fb} = \frac{R_2}{R_1+R_2}V_{out}$ also increases, which then causes V_{error}, and hence, V_{out} to go down. Relating this circuit back to the block diagram above, we can say that the feedback $f = \frac{R_2}{R_1+R_2}$. Now let's derive why $V_+ = V_-$ in this case. We know that in the above circuit, $V_+ = V_{in}$ and $V_- = V_{fb}$. Let's redraw the block diagram.

Assume the system has reached steady state, so we can use the golden rules. We have

$$V_{error} = V_+ - V_- \tag{10}$$

$$V_{out} = AV_{error} = A\left(V_+ - V_-\right) \tag{11}$$

$$V_- = fV_{out} \tag{12}$$

[1] This description implies some "delay" between each of the voltages changing. The impact of this delay, well as the values of A and f, are explored more in EECS16B and EE128. Most feedback systems are designed with appropriate values of A, f, and delay so that the steady state occurs and we can use our golden rules.

Combining the last two equations, we have

$$V_{out} = A\left(V_+ - fV_{out}\right), \tag{13}$$

which gives us

$$V_{out}\left(1 + Af\right) = AV_+. \tag{14}$$

Finally, we have

$$V_{out} = \frac{A}{1+Af}V_+. \tag{15}$$

Hence,

$$V_- = fV_{out} = \frac{fA}{1+Af}V_+. \tag{16}$$

Now we know that the gain A is very large, hence fA is very large. Hence, the ratio

$$\frac{fA}{1+Af} \approx 1. \tag{17}$$

Thus, when $A \to \infty$, which is what we assume for an ideal op-amp, $V_+ = V_-$.

18.6 Inverting Op-amp Amplifier

Let's apply what we've learned so far about Golden rules and negative feedback to the following op-amp circuit: [2]

Given an ideal op-amp (with power rails of sufficiently large magnitude), what is V_{out} if we input an arbitrary voltage of V_{in}?

The first golden rule says that $I_- = I_+ = 0$. Hence using KCL at the node labelled with voltage V_-, we have

$$I_{in} = I_- + I_f = 0 + I_f = I_f. \tag{18}$$

[2] We will develop tools to determine if a circuit is in negative feedback in the next note. For now, let's accept it as a given.

We have

$$I_{in} = I_f. \tag{19}$$

Now, let's apply the second golden rule, $V_+ = V_-$. Since the positive input terminal is connected to ground, $V_+ = 0$. Hence, we have

$$V_+ = V_- = 0. \tag{20}$$

By Ohm's law,

$$I_{in} = \frac{V_{in} - V_-}{R_{in}} = \frac{V_{in}}{R_{in}} \tag{21}$$

$$I_f = \frac{V_- - V_{out}}{R_f} = -\frac{V_{out}}{R_f}. \tag{22}$$

Since $I_{in} = I_f$, we have

$$\frac{V_{in}}{R_{in}} = -\frac{V_{out}}{R_f}. \tag{23}$$

Moving terms around, we have

$$V_{out} = -\frac{R_f}{R_{in}} V_{in}. \tag{24}$$

Observe that the output voltage is a multiple of the input voltage with a scaling factor of $-\frac{R_f}{R_{in}}$. Depending on chosen resistors, we can make this ratio anything between 0 and infinity. In addition, notice that V_{out} and V_{in} are of opposite signs. Since the output is amplified and inverted, this type of circuit is called an **inverting amplifier**.

18.7 Inverting Summing Amplifier

Now let's take a look at a slightly more complicated op-amp circuit example with two voltage sources:

First, let's apply the first golden rule, $I_- = I_+ = 0$. Applying KCL at the node labelled V_-, we have

$$I_1 + I_2 = I_- + I_3 = 0 + I_3. \tag{25}$$

Hence, we have

$$I_1 + I_2 = I_3. \tag{26}$$

Now by the second golden rule and the fact that the positive input terminal is connected to ground, we have

$$V_+ = V_- = 0. \tag{27}$$

Applying Ohm's law at each of the three resistors, we have

$$I_1 = \frac{V_1 - V_-}{R_1} = \frac{V_1}{R_1} \tag{28}$$

$$I_2 = \frac{V_2 - V_-}{R_2} = \frac{V_2}{R_2} \tag{29}$$

$$I_3 = \frac{V_- - V_{out}}{R_3} = -\frac{V_{out}}{R_3} \tag{30}$$

$$\tag{31}$$

Plugging in the above result to equation 26, we have

$$\frac{V_1}{R_1} + \frac{V_2}{R_2} = -\frac{V_{out}}{R_3}. \tag{32}$$

Multiplying both sides by R_3, we have

$$V_{out} = -\frac{R_3}{R_1}V_1 - \frac{R_3}{R_2}V_2, \tag{33}$$

The output of this circuit V_{out} is a sum of of the two inputs V_1 and V_2, multiplied by the ratios $\frac{-R_3}{R_1}$ and $\frac{-R_3}{R_2}$. We call this circuit the **inverting summing amplifier**.

Note that we can also derive this final result from superposition. If we short V_2 and keep V_1 on, we get $V_{out} = -\frac{R_3}{R_1}V_1$. Similarly, if we short V_1 and keep V_2 on, we get $V_{out} = -\frac{R_3}{R_2}V_2$. Adding these together, we get $V_{out} = -\frac{R_3}{R_1}V_1 - \frac{R_3}{R_2}V_2$.

18.8 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. True or False: We can use both Golden Rules to analyze any op-amp circuit.
2. True or False: An op-amp can operate without externally supplied power.

3. Find an expression for the output voltage V_{out} in terms of the input voltage V_{in} if the comparator is not railing (the output is between V_{DD} and V_{SS}). Assume that the op-amp has a gain of A.

4. An op-amp does not change the voltage of the circuit it is connected to because:

   (a) it has infinite gain.

   (b) it has infinite input resistance.

   (c) the Thévenin equivalent resistance at the output is 0.

5. If we switched the negative terminal and the positive terminal on an op-amp in negative feedback, would the gain across the amplifier be the same? For example, consider the op amp below.

   (a) Yes, it would be the same.

   (b) No, it would be the inverse of the original gain.

   (c) No, because we can no longer assume that V_+ and V_- are equal.

6. True or False: An ideal op-amp behaves as though it has infinite gain.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 19

19.1 Golden Rules Revisited

Recall for an ideal op-amp, the "golden rules" are

- (1) $I_+ = I_- = 0$. No current can flow into the positive or negative input terminal. **This rule holds regardless of whether there is negative feedback or not.**
- (2) For a circuit with negative feedback, in steady state, $U_+ = U_-$. **This rule only holds when there is negative feedback**.

$U_+ = U_-$ means that the error voltage $v_{err} = U_+ - U_-$ is $0V$ for an op-amp. In order to understand how v_{out} can be non-zero for $v_{err} = 0$, let's review the block diagram of an op amp in negative feedback.

We have negative feedback gain f. For an op-amp in negative feedback, we have

$$v_{in} - f \cdot v_{out} = v_{err}. \tag{1}$$

We know $v_{out} = A \cdot v_{err}$ and for our op-amp in negative feedback, we derived:

$$v_{out} = \frac{A}{1 + A \cdot f} v_{in}. \tag{2}$$

When $v_{in} = 0V$, $v_{err} = 0V$. Note that mathematically, $\lim_{x\to\infty} kx \cdot \frac{1}{x} = k$ even though $\lim_{x\to\infty} \frac{1}{x} = 0$. Similarly, for an ideal op-amp, v_{out} can be non-zero even if $v_{err} = 0$ given $A = \infty$.

19.2 Signals vs. Supply Voltages

As we saw in the last note (and will continue to explore here), op-amps allow us to perform mathematical operations on the input voltages. In real systems, these input voltages are typically small signals that we have

measured, for example, changes in light intensity or electrical brain activity read from an EEG. Although these signals are voltages, they are different from the supply voltages of an op-amp because they are typically much smaller and they change over time.

To distinguish, we will use the following symbol to represent small, time-varying voltage signals:

We will continue to use our standard voltage source symbol to represent static voltages that are used as power supplies (such as those connected to V_{SS} and V_{DD} of an op-amp). Voltage *signals* generally should not be used to power an op-amp since they typically too small and this would create time-varying behavior of the op-amp.

19.3 Checking for Negative Feedback

Recall that we can only use golden rule #2 if our circuit is in negative feedback. While it is tempting to say "if the feedback voltage is connected to the negative op-amp terminal, then we have negative feedback," this is not always true. Here is a two-step procedure for determining if a circuit is in negative feedback:

- **Step 1: Zero out all independent sources**, replacing voltage sources with wires and current sources with opens as we did in superposition. You do not need to zero out the voltage sources that serve as the power supplies to the op-amp, since they are not treated as signals and almost considered part of the op-amp.
- **Step 2: Wiggle the output and check the loop**. The goal is to see how the feedback loop responds to a change. Assume that the output increases slightly. Check the direction of change of the feedback signal and the error signal from the circuit. Any change in the error signal will cause a new change in the output. This change is the feedback loop's response to the initial change.
  - If the error signal decreases, then the output must also decrease. This is the *opposite direction* we initially assumed, i.e. the loop is trying to correct for the change. So the circuit is in negative feedback.
  - If the error signal instead increased, then the output would also increase. This is the *same direction* we initially assume, i.e. the initial increase lead to further increase. We call this positive feedback.

Let's use the rules to check some example circuits.

Example circuit 1

Step 1: (zero-out all independent sources) We can zero-out the independent source v_{in} in the circuit, and the circuit diagram becomes:

Step 2: (Wiggle v_{out} and check the effect on v_{err}) If we increase v_{out}, we know u_2 is going to increase because R_1 and R_2 form a voltage divider circuit. u_2 is connected to u_-, so if u_- increases, then $v_{err} = u_+ - u_-$ decreases, and thus v_{out} decreases. So the change in the error signal in the opposite direction, and this circuit is in negative feedback.

Example circuit 2

Step 1: (zero-out all independent sources) We can zero-out the independent source v_{in} in the circuit, and the circuit diagram becomes:

Step 2: (Wiggle v_{out} and check the effect on v_{err}) This circuit is now the same as the previous example, so using the same reasoning, this circuit is in negative feedback.

Example circuit 3

Step 1: (Zero out all independent sources).

Step 2: (Wiggle v_{out} and check the effect on v_{err}) No current flows into u_1, so $u_1 = v_{out}$. If we increase v_{out}, then u_1 will also increase. u_1 is connected to u_+, so u_+ increases, therefore $v_{in} = u_+ - u_-$ increases, and

thus v_{out} increases. So the feedback from the circuit is in the same direction, and this circuit is in positive feedback.

Example circuit 4

Step 1: (Zero out all independent sources). We do not zero out the dependent (voltage-controlled) voltage source $-1 \cdot v_f$ because it is not an independent voltage source.

Step 2: (Wiggle v_{out} and check the effect on v_{err}) If we increase v_{out}, v_f increases which then decreases $v_- = -1 \cdot v_f$, which then increases $v_{in} = u_+ - u_-$, thus v_{out} increases. The feedback circuit response to the increase in v_{out} is in the same direction. So this circuit is in positive feedback.

19.4 Non-inverting & Inverting Amplifiers

There are two important basic categories of op-amp amplifier circuits that you should be familiar with: non-inverting amplifier and inverting amplifier:

- **Non-inverting Amplifier**

For the non-inverting amplifier circuit, we have

$$v_{out} = (1 + \frac{R_1}{R_2}) \cdot v_{in}. \tag{3}$$

- **Inverting Amplifier**

For the inverting amplifier circuit, we have

$$v_{out} = -\frac{R_2}{R_1} \cdot v_{in}. \tag{4}$$

19.5 Artificial Neuron

Neurons in our brain are interconnected. The output of a single neuron is dependent on inputs from several other neurons. We can represent this idea with vector-vector multiplication, in which the output is a linear combination of several inputs. As an example, let's write the linear system for an ouput that is the linear combination of just two inputs, v_1 and v_2:

$$\begin{bmatrix} a_1 & a_2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = a_1 \cdot v_1 + a_2 \cdot v_2 \tag{5}$$

An artificial neuron circuit must accomplish this linear combination, i.e. addition and multiplication. The following model can be used to build a circuit representation of an artificial neuron:

We have analyzed this circuit in a previous lecture note, and we arrived at the final output:

$$v_{out} = -\frac{R_3}{R_1}v_1 - \frac{R_3}{R_2}v_2 \tag{6}$$

This circuit is called the "inverting summing amplifier", because it adds two inputs, scaled by negative values (R_1, R_2, R_3 are resistors and have positive values, so the coefficients a_1 and a_2 are all negative). This circuit acheives both the multiplication and addition goals desired in linear combination.

What if we want to make a_1 and a_2 positive instead? By adding another inverter amplifier circuit at v_{out}, we can get positive coefficients.

At the output of the inverting summing amplifier, v_{ana}, we have:

$$v_{ana} = -\frac{R_3}{R_1}v_1 - \frac{R_3}{R_2}v_2. \tag{7}$$

as input to an the inverting amplifier. So the new v_{out} is:

$$v_{out} = -\frac{R_4}{R_4}v_{ana}. \tag{8}$$

Hence,

$$v_{out} = \frac{R_3}{R_1}v_1 + \frac{R_3}{R_2}v_2. \tag{9}$$

and we have positive coefficients $a_1 = \frac{R_3}{R_1}$, $a_2 = \frac{R_3}{R_2}$.

Note that the voltage v_{ana} is unaffected by the second inverting amplifier, because the output of the op-amp is a (dependent) voltage source.

19.6 Loading and Buffering

Consider the following *unity gain buffer* (we'll see why it's called that shortly):

To analyze this circuit, we first notice that $u_+ = v_{in}$ and $u_- = v_{out}$. We apply the second golden rule, $u_+ = u_-$, which tells us

$$v_{in} = v_{out}.$$

The input and output voltages are the same! (This is where the term *unity gain* comes from). Does that mean that this circuit is useless? No! To see why, let's look back at the DAC-speaker example from the last note. Recall that we want to drive our speaker using the output voltage from our DAC (let's suppose that no amplification is necessary). We model our DAC as a voltage source with a series resistors, and we model the speaker as an $8\,\Omega$ resistor connected to ground. In this example, we'd like $V_{speaker} = V_{DAC}$, so that the DAC output is directly played on the speaker. We try connecting the DAC and speaker together directly:

Using the voltage divider equation, we calculate that

$$V_{speaker} = \frac{8\Omega}{1000\Omega + 8\Omega} V_{DAC} = 0.0079\, V_{DAC}$$

which is much smaller than we'd like.

The speaker has a small resistance, so it draws a lot of current causing a large voltage drop over R_{TH}, much larger than when the speaker was not present. This effect is called **loading**. Loading effects make it much harder to design circuits because they mean that the DAC will behave differently depending on what it's connected to. Specifically, if the circuit afterwards draws current through R_{TH}, then the output voltage seen at $V_{speaker}$ drops. Furthermore, the amount it drops depends on the speaker's resistance — but we don't want to have to redesign our circuit every time we buy a new speaker!

This is where our unity gain buffer from before can help. Rather than connecting the DAC and speaker directly, we include the buffer in between as follows:

From the first golden rule, we know that $i_+ = 0$. This means there is no current through R_{TH}, and therefore no voltage drop across the resistor. In other words, the op-amp does not load the DAC since it has infinite internal resistance! Since there is no voltage drop over R_{TH}, we know that $u_+ = V_{DAC}$. Combining this with our previous analysis of the buffer, we see that $V_{DAC} = V_{speaker}$, as desired. [1]

Buffers are a powerful tool because they allow us to split circuits into blocks that we can analyze separately and then combine later. When circuit blocks behave the same way regardless of what they're are connected to, we don't need to worry about what's inside, making it much easier to design complex circuits.

19.7 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. Is the following amplifier in an inverting or a non-inverting configuration?

[1] At this point, one might ask if the speaker has a loading effect on the buffering op-amp. Note that the output of the op-amp is connected to a (dependent) voltage source, so loading cannot change the output voltage. All the more reason we like buffers and op-amps!

2. In the circuit above, if $R_s = 50\,\Omega$, $R_f = 100\,\Omega$, and $V_s = 2\,\text{V}$, what is the value of V_{out}?

3. Is the following op-amp in negative feedback?

4. For the circuit above, find V_{out} in terms of V_{in}.

5. True or False: An inverting amplifier can have any negative gain (for positive resistor values).

6. True or False: A non-inverting amplifier can have any non-negative gain (for positive resistor values).

7. If we placed a buffer, which is an op-amp in negative feedback that has a gain of 1, between two subcircuits, would it be the same as connecting the subcircuits with a wire?

(a) Yes, because the buffer isn't changing the value between the subcircuits.

(b) No, because the buffer does not change the voltage of the circuit it is connected to.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 20

20.1 Design Procedure

Now that we've analyzed many circuits, we are ready to focus on *designing* interesting circuits to perform specific tasks. Circuit design is generally more challenging than circuit analysis. Usually, we are given a prompt about what type of circuit is desired and some specifications, and then we must decide what components to use and how to connect them. Circuit design problems are very similar to proof problems, which we looked at earlier in this course; Both cases are open ended, and we frequently have to integrate many ideas and explore several different possibilities to reach a solution.

When faced with a design problem, a good place to start is to follow the *design procedure* outlined here:

Step 1 (Specification): Concretely restate the goals for the design.
Frequently, a design prompt will include a lot of text, so we'd like to restate all of the most important features of our design. We'll refer to these specifications later to determine if our design is complete.

Step 2 (Strategy): Describe your strategy (often in the form of a block diagram) to achieve your goal.
To do this, start by thinking about what you can measure vs. what you want to know. For example in our capacitive touchscreen, we want to know if there is a touch and we can measure voltage. Since we know that a touch can change the capacitance, we break this down into the following block diagram:

Step 3 (Implementation): Implement the components described in your strategy.
This is where pattern matching is useful: remind yourself of blocks you know, (ex. voltage divider, inverting amplifier) and check if any of these can be used to implement steps of your strategy. If you don't know of a block that does what you want, think about how to modify or extend the blocks you know.

Step 4 (Verification): Check that your design from Step 3 does what you specified in Step 1.
It's tempting to think that you're done after implementation, but verification is critical! In particular, check block-to-block connections, as these are the most common point for problems. Does one block load another block causing it to behave differently than expected? Are there any contradictions (ex. a voltage source with both ends connected by a wire, or a current source directed into an open circuit)? Repeat previous steps if necessary to make sure that your final circuit meets the specifications.

20.2 Design Example: Countdown Timer

Let's practice the design procedure with an example: Your boss comes to you and asks you to build a countdown timer that will turn on an LED 2 sec after a button is pressed. She tells you that the LED turns on when 2V is applied across it.

Step 1 (Specification): Build a circuit that, after a button is pressed, measures 2 seconds and then applies two volts across an LED.

Step 2 (Strategy): Let's figure out a strategy to tackle this problem: When the button is pressed we'd like to turn on a timer. Then we'll need to know if the time elapsed has reached 2 sec, which we can determine with a comparator. When our comparator tells us that the time has exceeded 2 sec, we want to apply 2 V across the LED. This is summarized in the following block diagram:

Note that "time" will need to be represented as either a current or voltage, since those are the only circuit quantities we can control. Therefore, we will also have to determine what current/voltage is equivalent to 2 sec in our representation.

Step 3 (Implementation): Now we'd like to implement each of the blocks above. To do this, we start by considering which component(s) could be used to build each block.

The first block we need to build is to "start the timer." What component can changes based on a user input? A switch — we will assume we have a switch that either opens or closes when the button is pressed.

The next block is the timer itself. To build a timer, we need a component that changes either voltage or current as a function of time. A resistor won't work because its IV relationship is static in time. But a capacitor is a good candidate since it's IV relationship is time-varying. We know that for a capacitor:

$$I_c = C\frac{dV_c}{dt} \tag{1}$$

This relationship is easiest to work with when the current is constant, so let's apply a constant current source over the capacitor:

By integrating both sides of Eq. (1) (or referring to the lecture notes on capacitors) we can derive that

$$V_c(t) = \frac{I_s}{C_{ref}} t + V_c(0) \tag{2}$$

where the voltage is linearly increasing with time.

Let's combine our first two blocks: the switch and the timer:

When the switch closes at $t = 0$, we start charging the capacitor. The voltage over the capacitor will indicate how much time has elapsed since the switch closed (i.e. since the button was pressed).

Now let's look at the third block of our block diagram, the comparator. We already know a circuit element for this block - an op-amp used as a comparator (meaning, the op-amp is not in negative feedback). Based on our block diagram, we'd like one input to the comparator to be the timer output, and the other input should be a static voltage equal to the timer output voltage after 2 sec.

That means V_{ref} should be set to the voltage on the capacitor after charging for 2s (assuming the capacitor is initially uncharged):

$$V_{ref} = \frac{2I_s}{C_{ref}} \tag{3}$$

Before 2 seconds, the comparator output will be V_{SS}, and after 2 seconds has elapsed, the output will be V_{DD} From our problem specification, we know that we want to apply $2V$ over the LED when it is on, so we set $V_{DD} = 2V$ and $V_{SS} = 0V$.

Our final block from the diagram is to apply $2V$ over the LED. We do this by connecting one terminal of the LED to ground and the other to the op-amp output.

Step 4 (Verfication): Now, that we have a complete design, we must verify that it works as desired. Since we designed each block separately, there may be mistakes when they are connected.

Let's consider what happens before the button is pressed. The left hand side of the circuit looks like this:

Uh oh – this is a contradiction! To see why, we apply KCL at the upper left node:

$$I_s = I_L \tag{4}$$

I_s is the constant current supplied by the current source, I_L is the current flow into the switch. Before the button is pressed, I_L must equal 0 since there is an open circuit. However, the current source guarantees that I_s is nonzero. These two equations cannot simultaneously be true, leading to a mathematical contradiction. In real life, at least one of the elements will not behave as expected, possibly causing elements to break.

Therefore, we'd like to think of a new circuit that achieves our goals without any mathematical contradictions. To do this we can add another switch in the circuit:

Before the button is pressed, the switch is closed and can be replaced by a wire. In this case, the current from the current source can flow through this wire to ground, eliminating the contradiction that happened before. In addition, when the switch is closed, it connects the top of the capacitor to ground, so the voltage drop over the capacitor remains constant at 0 and the capacitor is discharged.

After the button is pressed, the switch is off and can be replaced by an open circuit, and everything functions as it did before.

This change in circuit architecture has another important benefit: it guarantees that the initial voltage on the capacitor is 0V. This is important because we used this assumption when setting V_{ref} in Eq.(3). Without this change, we would not know the initial voltage, and therefore we could not choose V_{ref} to correctly correspond to 2 sec.

We put all of this together into our final design, where the switch is initially closed and it opens when the button is pressed.

Can you build this countdown timer in lab? Almost! The only thing that is missing is the current source. In the next section, we'll explore building an ideal current source out of commonly available components.

20.3 Design Example: "Almost" current source

In this section, we will use resistors, voltage sources, and op-amps to build a current source that can be used in the countdown timer above. Let's follow our design procedure.

Step 1 (Specification): First we restate the problem clearly: Build a current source that outputs constant current, I_s, regardless of the voltage across it.

Step 2 (Strategy): We can use a voltage source, but we need a way to transform the voltage into a constant current. Ohm's law tell us

$$I = \frac{V}{R},$$

so maybe we can use a resistor to convert the voltage source output, V_s, into a constant current, I_s. This is summarized in the following block diagram:

Step 3 (Implementation): Based on our strategy from Step 2, we make our first attempt to build the current source. Let's take a voltage source and connect it to a resistor:

Attempt #1

In the above circuit,

$$u_1 = V_s \tag{5}$$

$$I_s = \frac{V_s}{R} \tag{6}$$

This looks promising, but if we hook up an element between u_2 and ground, there is a loading effect and we find that the current through R_s is no longer the constant $I_s = \frac{V_s}{R_s}$. For example, if the element is a resistor R_l, then the current drops to $I_s = \frac{V_s}{R_s+R_l}$.

Recall that it is important to guarantee a constant current output from the current source regardless of the element added to the circuit. We need to modify our design.

Attempt #2 Although our Attempt 1 was unsuccessful, we have learned an important lesson: if we can somehow set u_2 to $0V$ without physically connecting it to ground, then the current through R_s will always equal $\frac{V_s}{R_s}$ ($I_s = \frac{u_1-u_2}{R_s} = \frac{V_s-0}{R_s}$). Do we know of an element that can set two points to have the same voltage without physically connecting them? Yes! According to golden rule #2, we can set both U_+ and U_- to $0V$ if an op-amp circuit is in negative feedback.

Indeed, we will now use an op-amp to build a current source!

According to golden rules:

$$I_- = 0 \tag{7}$$

$$U_- = U_+ = 0V \tag{8}$$

We also know that the current across R_s will always be:

$$I_s = \frac{u_1 - u_2}{R_s} \tag{9}$$

Solving the above equations, we can get the value of I_s:

$$I_s = \frac{V_s}{R_s} \tag{10}$$

By setting u_2 (U_-) to $0V$ by using a negative feedback circuit, we have successfully built a current source! It is important to keep in mind that setting u_2 to $0V$ by using a negative feedback circuit is very different from physically connecting the node of u_2 to ground.

Step 4 (Verification): Now let's hook up a resistor R_L to the circuit and prove that current flow through R_L is constant:

According to KCL:

$$I_L = I_s = \frac{V_s}{R_s} \tag{11}$$

From Eq. (11), we can see immediately that I_L is not affected by changes in R_L. How does the circuit maintain the constant current flow through R_L? The op-amp outputs a negative voltage to maintain the constant current. In other words, the op-amp sets a voltage drop across R_L such that

$$V_{R_L} = \frac{V_s}{R_S} R_L. \tag{12}$$

We've now verified that our current source works for a resistor load. However, we also want this source to work in our countdown timer, which is more complicated. To verify, we'll connect our current source, then analyze the result. Recall that the first part of the timer looks like this:

When we add our current source, the circuit becomes:

Let's consider the case after the button is pressed and the switch is open. We are supposed to be charging the capacitor to keep track of time. However, u_3 is connected to ground, and $u_2 = 0V$ by the second golden rule. Therefore, the voltage across C_{ref} is

$$V_{time} = u_2 - u_3 = 0V - 0V = 0V \tag{13}$$

which means that the voltage is not increasing with time as desired!

In addition, there is another problem: For the op-amp to be in negative feedback, the controlled voltage source inside the op-amp must make small changes to compensate for changes in the input. However, since the output of the op-amp is always 0V, there is no feedback and we cannot assume the second golden rule. To solve this problem, we get rid of the ground connection at u_3.

Now we analyze the circuit above:

$$I_s = C_{ref}\frac{dV_{time}}{dt} \tag{14}$$

$$I_s = C_{ref}\frac{d(u_2 - u_3)}{dt} \tag{15}$$

According to the second golden rule, $u_2 = 0V$.

$$I_s = C_{ref}\frac{d(0V - u_3)}{dt} = C_{ref}\frac{d(-u_3)}{dt} \tag{16}$$

Solving the above equation:

$$u_3(\mathrm{t}) = -\frac{I_s}{C_{ref}} \times t + u_3(\mathrm{t} = 0\mathrm{s}) \tag{17}$$

$u_3(\mathrm{t})$ is dependent on the initial value $u_3(\mathrm{t} = 0\mathrm{s})$. Before touch, the switch $S1$ is closed, which sets $u_3(\mathrm{t} = 0\mathrm{s})$ to $0V$. Therefore,

$$u_3(\mathrm{t}) = -\frac{I_s}{C_{ref}}t = -\frac{V_s}{R_s C_{ref}}t \tag{18}$$

Note there is a term $R_s C_{ref}$ in the denominator. What are the units of $R_s C_{ref}$?

$$\text{Unit of } RC = \frac{\text{volts}}{\text{ampere}} \times \frac{\text{columbs}}{\text{volt}} = \frac{\text{columbs}}{\text{ampere}} = \frac{\text{columbs}}{\frac{\text{columbs}}{\text{second}}} = \text{seconds} \tag{19}$$

The units of RC are seconds. This comes up frequently in the timing analysis of circuits.

Caveats of the "Almost" Current Source

Although this current source has many things in common with an ideal current source, there 2 important points where it differs:

- Do not connect the output of the current source to ground externally. Doing so may force V_{out} of the op-amp to $0V$ and lead to nonidealities.
- The circuit element we hook up to the current source must still keep the op-amp circuit in its negative feedback state. Being in negative feedback allows us to set the node u_2 to $0V$ without physically connecting it to ground and hence allows a constant current output.

20.4 Capacitive Touchscreen Revisted

In this section, we'll use our current source design to implement an alternative capacitive touchscreen. Recall that the presence or absence of a finger changes the capacitance between two electrodes, $C_{\text{eq, E1-E2}}$. We can design the following circuit to detect these changes:

Here, the current source, I_s, outputs a period square wave that alternates between I_1 and $-I_1$. The voltage over the capacitor V_c increases linearly, with a slope that depends on $C_{\text{eq, E1-E2}}$. The switch is opened at $t = \frac{T}{4}$.

The periodic current source is critical in this design, but not readily available. Can we use our design of the "almost" current source in our capacitive touchscreen? The "almost" current source outputs a constant current, while we need a periodic current (shown in red above). However, we can achieve this by powering the "almost" current source with a periodic voltage instead of a constant voltage:

V_s has the following time-varying response:

Since $I_s = \frac{V_s}{R_s}$, the current supplied to the load element will be either $\frac{V_1}{R_s}$ or $-\frac{V_1}{R_s}$. Now let's attach this current source to our touchscreen circuit.

Let's look at the op-amp on the right side. Note that it's positive terminal is directly connected to u_2, and $u_2 = 0$. This means that u_+ of the right most op-amp is always zero and does not depend on the capacitance $C_{\text{eq,E1-E2}}$. Therefore, V_{out} cannot detect changes in $C_{\text{eq,E1-E2}}$ and the circuit does not work as intended.

How can we fix this? We should not connect u_2 to the op-amp. Instead we can connect the other end of the capacitor as follows:

By the second golden rule (applied to the left op-amp), $u_2 = U_+ = 0V$. We can use KVL to determine that

$$\begin{aligned} u_3 &= u_2 - V_c(t) \\ &= -V_c(t) \end{aligned}$$

This is good because u_3 will depend on the capacitance, unlike the previous configuration. Let's analyze what $u_3 = -V_c$ is as a function of the capacitance:

Since no current flows into either op-amp, $I_s = \frac{V_s}{R_s} = I_c$. Using the equation for a capacitor charging under constant current, we find

$$V_c(t) = \frac{I_c}{C_{\text{eq, E1-E2}}} t = \frac{V_1}{R_s C_{\text{eq, E1-E2}}} t$$

This equation only applies when the current is constant. When V_s becomes negative, I_s will also be negative, and then V_c will decrease at the same rate (see Note 17 for a more in-depth derivation of this effect). Overtime, we get the following waveform for $V_c(t)$ (assuming the switch is opened at $t = \frac{T}{4}$)

Recalling that $C_{\text{eq, E1-E2}}$ increases when there is a touch, we can plot $u_+ = -V_c$ when the finger is present and absent. We'd like to set V_{ref} halfway between the peak of u_+ with a touch and u_+ without a touch.

Using the equations for a comparator, we find

$$V_{out} = \begin{cases} 0V & \text{if } u_+ < V_{ref} \\ 3.3V & \text{if } u_+ > V_{ref} \end{cases}$$

We get the following waveforms when there is a touch:

Touch

When there is no touch, there is an increase in V_{out} every period, as shown below:

V_{out} has the same structure as when we used an ideal current source! However, since one terminal of our "almost" current source is always 0V, the analysis was slightly different and we had to rethink the circuit architecture. That's why Step 4 (Verification) of the design procedure is so important – we can't assume different circuit blocks will act the same when we connect them.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 21

21.1 Module Goals

In this module, we introduce a family of ideas that are connected to **optimization** and **machine learning**, which are some of the modern pillars of EECS. You'll learn more about these concepts in EECS 127 and CS 189.

This module uses the practical application of localization — figuring out where you are — as a source of examples and a connection to the lab. However, just like in previous modules, the ideas we will learn and the skills that we develop are far more generally applicable.

21.2 Introduction: Positioning Systems

GPS (Global Positioning System) is a navigational system that we use all the time to tell us where we are. In this module, we'll explore the *signal processing* underlying GPS, and you'll build your own acoustic positioning system in the lab.

How does GPS work? A receiver, such as your cell phone, receives messages from satellites orbiting the earth. From these messages, the receiver is able to determine how far away it is from each satellite. The receiver also knows the positions of each satellite.

The distance to the first satellite defines a set of possible locations where the receiver could be – a circle (or a sphere in 3D) around the satellite. Each other satellite also defines a circle of possible locations. The intersection of all these circles tells us the position of the reciever!

However, in real life, our measurements will be noisy. Can we use extra measurements from more satellites to reduce the effects of noise? In this module, we'll also look at how to combine many measurements together to get a better estimate of position, when there is noise.

We can break this process into two parts, which we'll cover in detail in the upcoming notes:

1. Find the distances from the receiver to the satellites, based on the messages received.
2. Combine the measurements from each satellite to determine location.

In the upcoming notes, we'll explore this process in more detail. In this note, we'll introduce some fundamentals that will be built on in the rest of the module.

21.3 Inner Products

Definition: The (Euclidean) inner product between two vectors $\vec{x}, \vec{y} \in \mathbb{R}^n$ is defined as:

$$\langle \vec{x}, \vec{y} \rangle \equiv \vec{x} \cdot \vec{y} \equiv \vec{x}^T \vec{y} \tag{1}$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{2}$$

$$= x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \tag{3}$$

$$= \sum_{i=1}^{n} x_i y_i \tag{4}$$

In physics, inner products are often called **dot products** (denoted by $\vec{x} \cdot \vec{y}$) and can be used to calculate values such as work. In this class we will use the notation $\langle \vec{x}, \vec{y} \rangle$ for the inner product.

Example 21.1 (Inner product of two vectors): Compute the inner product of the two vectors $\begin{bmatrix} -1 & 3.5 & 0 \end{bmatrix}^T$ and $\begin{bmatrix} 1 & 0 & 2 \end{bmatrix}^T$.

$$\left\langle \begin{bmatrix} -1 \\ 3.5 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \right\rangle = \begin{bmatrix} -1 & 3.5 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \tag{5}$$

$$= -1 \times 1 + 3.5 \times 0 + 0 \times 2 \tag{6}$$

$$= -1 + 0 + 0 = -1. \tag{7}$$

21.4 Properties of Inner Products

Commutative: The inner product is a *commutative* or *symmetric* operation; that is, it remains the same even if we switch its arguments.[1]

Proof:

$$\begin{aligned} \langle \vec{x}, \vec{y} \rangle &= x_1 y_1 + \cdots + x_n y_n \\ &= y_1 x_1 + \cdots + y_n x_n \\ &= \langle \vec{y}, \vec{x} \rangle \end{aligned} \tag{8}$$

Scalar multiplication: If we scale either vector by a real number c, then the inner product will be scaled by c as well.

Proof:

$$\begin{aligned} \langle c\vec{x}, \vec{y} \rangle &= (cx_1) y_1 + \cdots + (cx_n) y_n \\ &= c(x_1 y_1) + \cdots + c(x_n y_n) \\ &= c \langle \vec{x}, \vec{y} \rangle \end{aligned} \tag{9}$$

By a similar argument (exercise: prove it yourself),

$$\langle \vec{x}, c\vec{y} \rangle = c \langle \vec{x}, \vec{y} \rangle \tag{10}$$

Distributive over vector addition: What happens when we take the inner product between a sum of two vectors and another vector? We can write:

$$\begin{aligned} \langle \vec{x} + \vec{y}, \vec{z} \rangle &= (x_1 + y_1) z_1 + \cdots + (x_n + y_n) z_n \\ &= x_1 z_1 + y_1 z_1 + \cdots + x_n z_n + y_n z_n \\ &= x_1 z_1 + \cdots + x_n z_n + y_1 z_1 + \cdots + y_n z_n \\ &= \langle \vec{x}, \vec{z} \rangle + \langle \vec{y}, \vec{z} \rangle \end{aligned} \tag{11}$$

[1] In later classes, you'll see that this ceases to be true if we are working with complex numbers. But since we only work with real-valued vectors in this class, commutativity is always true for our purposes.

By a similar argument:

$$\langle \vec{x}, \vec{y} + \vec{z} \rangle = \langle \vec{x}, \vec{y} \rangle + \langle \vec{x}, \vec{z} \rangle \tag{12}$$

21.5 Orthogonal Vectors

Two vectors $\vec{x}$, $\vec{y}$ in $\mathbb{R}^n$ are **orthogonal** if their inner product is zero, i.e. $\langle \vec{x}, \vec{y} \rangle = 0$. The words perpendicular and orthogonal mean the same thing, but when working in dimensions greater than 2D or 3D, the term orthogonal tends to be more commonly used.

Here's an example, in $\mathbb{R}^3$. Let's say $\vec{x} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$ and $\vec{y} = \begin{bmatrix} 4 \\ -1 \\ -1 \end{bmatrix}$. Then, $\langle \vec{x}, \vec{y} \rangle = (1)(4) + (1)(-1) + (3)(1) = 0$. Thus $\vec{x}$ and $\vec{y}$ are orthogonal.

Note that the standard unit vectors are always orthogonal to each other.

21.6 Interpretation of the Inner Product

Now that we have defined the inner product, it's time to get an intuition about what an inner product is.

First we take the unit vector $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and a general unit vector in $\mathbb{R}^2$, $\vec{x} = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}$. $\vec{x}$ is a unit vector because $\|\vec{x}\| = \sqrt{\cos^2\alpha + \sin^2\alpha} = 1$.

When we draw the vectors, the angle between them is α. We can calculate

$$\begin{aligned} \langle \vec{e}_1, \vec{x} \rangle &= \left\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} \right\rangle \\ &= (1)(\cos\alpha) + (0)(\sin\alpha) \\ &= \cos\alpha \end{aligned}$$

This shows that the inner product of two unit vectors is the cosine of the angle between them when one of the vectors is pointed along the x-axis. Let's try to prove this for two general unit vectors in $\mathbb{R}^2$, $\vec{x} = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}$ and $\vec{y} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}$. (All unit vectors can be represented in this form).

$$||\vec{x}|| = 1, \quad ||\vec{y}|| = 1$$

Let's plug these vectors into the definition of the inner product. We'll use the trigonometric identity $\cos\alpha\cos\beta + \sin\alpha\sin\beta = \cos(\alpha-\beta)$.

$$\begin{aligned}\langle \vec{x}, \vec{y}\rangle &= \left\langle \begin{bmatrix}\cos\alpha\\ \sin\alpha\end{bmatrix}, \begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix}\right\rangle \\ &= \cos\alpha\cos\beta + \sin\alpha\sin\beta \\ &= \cos(\alpha-\beta) \\ &= \cos\theta\end{aligned} \tag{13}$$

We've shown now that the inner product is the cosine of the angle between any two unit vectors.

What about any two general (nonzero) vectors $\vec{x}$ and $\vec{y}$? (Now we will let $\vec{x}$ and $\vec{y}$ be any length). We can first convert them into the unit vectors by dividing by the norm. This is called *normalization*.

$$\hat{x} = \frac{\vec{x}}{||x||} \qquad \hat{y} = \frac{\vec{y}}{||y||}$$

Since the direction of a vector does not change when it's multiplied by a scalar (and the norm is a scalar), then we know that the normalized vectors will point in the same direction as the original vectors. The angle between $\vec{x}$ and $\vec{y}$ is only dependent on their directions, so θ is unchanged by dividing by the norm. Therefore, we can apply Eq. ((Eq. 13)) on the normalized vectors:

$$\left\langle \frac{\vec{x}}{||x||}, \frac{\vec{y}}{||y||}\right\rangle = \cos\theta \tag{14}$$

We can then use scalar multiplication property of inner products (Eq. (Eq. 9)) since norms are scalars:

$$\begin{aligned}\frac{\langle \vec{x}, \vec{y}\rangle}{||x||\,||y||} &= \cos\theta \\ \langle \vec{x}, \vec{y}\rangle &= ||x||\,||y||\cos\theta\end{aligned} \tag{15}$$

Now we have a geometric interpretation of the inner product: the inner product of vectors $\vec{x}$ and $\vec{y}$ is their lengths multiplied by the angle between them. One remarkable observation is that the inner product *does not depend on the coordinate system the vectors are in*, it only depends on the relative angle between these vectors and their length. This is the reason it is very useful in physics, where the physical laws do not depend on the coordinate system used to measure them. This is also the reason why this property holds in higher dimensions as well. For any two vectors we can look at the plane passing through them and the angle between them is the angle θ measured in the plane.

21.7 Norms

The **Euclidean Norm** of a vector is defined as:

$$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} = \sqrt{\langle \vec{x}, \vec{x} \rangle} \tag{16}$$

Why is the norm important? The norm of a vector is also the magnitude (or length) of the vector. This corresponds to the usual notion of distance in $\mathbb{R}^2$ or $\mathbb{R}^3$.

21.8 Properties of Norms

The following properties are true for all norms (not just the 2D norm).

Non-negativity: For $\vec{x}$ in $\mathbb{R}^n$,

$$\|\vec{x}\| \geq 0$$

Zero Vector: The zero vector is the only vector with a norm of zero. For $\vec{x}$ in $\mathbb{R}^n$,

$$\|\vec{x}\| = 0 \text{ if and only if } \vec{x} = \vec{0}$$

Scalar Multiplication: For vector $\vec{x}$ in $\mathbb{R}^n$ and scalar α,

$$\|\alpha\vec{x}\| = |\alpha|\|\vec{x}\|$$

Triangle Inequality: For vectors $\vec{x}$ and $\vec{y}$ in $\mathbb{R}^n$,

$$\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$$

21.9 The Cauchy-Schwarz Inequality

The Cauchy-Schwarz inequality relates the inner product of two vectors to their length. The Cauchy-Schwarz inequality states:

$$\left| \langle \vec{x}, \vec{y} \rangle \right| \leq \|\vec{x}\| \|\vec{y}\| \tag{17}$$

We can prove this by recognizing that $|\cos\theta| \leq 1$ for all real θ. Thus:

$$\begin{aligned} \left|\langle \vec{x}, \vec{y} \rangle\right| &= \left| \|\vec{x}\| \|\vec{y}\| \cos\theta \right| \\ &= \|\vec{x}\| \|\vec{y}\| \cdot \left|\cos\theta\right| \\ &\leq \|\vec{x}\| \|\vec{y}\| \end{aligned} \tag{18}$$

21.10 p-Norms [optional]

Sometimes, we write the Euclidean norm of a vector as $\|\vec{x}\|_2$, with a subscript 2. The subscript differentiates the Euclidean norm (or 2-norm) from other useful norms. In general, the p-norm is defined as:

$$\|\vec{x}\|_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{1/p} \tag{19}$$

These other norms might feel esoteric but turn out to be useful in many engineering settings. Follow-on courses like EE 127 and EE 123 will show you how these can be useful in applications like machine learning. If no subscript is used, you can assume the 2-norm.

21.11 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. We can verify that 2 vectors are orthogonal to each other by:
   (a) Multiplying their magnitudes together and checking if it is 1.
   (b) Taking the outer product of the two and checking if it is 0.
   (c) Multiplying the magnitudes and the cosine of the angle between them and checking if it is 0.
   (d) Multiplying the magnitudes and dividing by the cosine of the angle between them and checking if it is 1.
2. True or False: If two vectors $\vec{u}$ and $\vec{v}$ are orthogonal, then $\|\vec{u}+\vec{v}\| = \|\vec{u}\| + \|\vec{v}\|$.
3. True or False: If two vectors $\vec{u}$ and $\vec{v}$ are linearly dependent, then $|\langle \vec{u}, \vec{v} \rangle| = \|\vec{u}\| \|\vec{v}\|$.
4. Does the following inner product definition (for $\mathbb{R}^2$) satisfy the all of the inner product properties? $\langle \vec{x}, \vec{y} \rangle = \vec{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \vec{y}$
5. Let $\vec{x}, \vec{y} \in \mathbb{R}^n$ such that $\|\vec{x}\| = 1$. Find $\text{proj}_{\vec{x}}\vec{y}$.
   (a) $\frac{\langle \vec{y}, \vec{x} \rangle}{\langle \vec{y}, \vec{y} \rangle} \vec{x}$
   (b) $\frac{\langle \vec{y}, \vec{x} \rangle}{\langle \vec{x}, \vec{x} \rangle} \vec{y}$
   (c) $\langle \vec{y}, \vec{x} \rangle$

(d) $\langle \vec{y}, \vec{x} \rangle \vec{x}$

6. What is the projection of the vector $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ onto the vector $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$?

7. Assume that you receive a shifted signal $\vec{r}$ and you correlate it with the original periodic signal $\vec{s}$, such that $\mathbf{C}_{\vec{s}}\vec{r} = \text{circcorr}(\vec{r}, \vec{s}) = \begin{bmatrix} -32 \\ -32 \\ 16 \\ 16 \\ 96 \\ 24 \end{bmatrix}$. What would you estimate as the sample delay between you and the transmitter?

8. True or False: $\|\mathbf{C}_{\vec{x}}\vec{y}\| = \left\|\mathbf{C}_{\vec{y}}\vec{x}\right\|$.

9. True or False: The projection of $\vec{y}$ onto $\vec{v}$ is the same as the projection of $\vec{y}$ onto $c\vec{v}$ whenever $c \neq 0$.

EECS 16A Designing Information Devices and Systems I
Fall 2019 Lecture Notes Note 22

22.1 Positioning Sytems: Trilateration and Correlation

In this note, we'll introduce two concepts that are critical in our positioning system, *trilateration* and *correlation*.

To begin, let's review the underlying concepts of GPS, introduced in the previous note. Recall that a GPS system consists of two parts: (1) a receiver, with unknown location, and (2) a set of "beacons", that send messages and have known location. In GPS, these are satellites, but they could be anything that can broadcast a message, such as a speaker sending audio signals or a radio transmitter on the ground. The goal is to determine the location of the receiver from the beacon's messages.

We split the process of localizing the receiver into two steps:

1. Find the distances from the receiver to each beacon, based on the messages received.
2. Combine the distance measurements to determine the receiver's location.

We'll used *correlation* to determine the distances, and we'll use *trilateration* to find the position based on the distance measurements and known beacon locations. Let's start with trilateration.

22.2 Trilateration

In this section, we'll assume that we've already determined the distance from the receiver to each beacon (we'll explain how to measure this distance in the next section).

Let's imagine we have a situation like the one below. We know the locations of the beacons $\vec{a}_1, \vec{a}_2, \vec{a}_3$, but don't know the location of the point at $\vec{x}$ (we'll be trying to find out what $\vec{x}$ is). We do know the distances d_1, d_2, d_3. We're trying to find the coordinates of $\vec{x}$ in this diagram:

We know that:

1. $\|\vec{x}-\vec{a}_1\|^2 = d_1^2$
2. $\|\vec{x}-\vec{a}_2\|^2 = d_2^2$
3. $\|\vec{x}-\vec{a}_3\|^2 = d_3^2$

Rewriting these using transpose notation we get:

$$\vec{x}^T\vec{x} - 2\vec{a}_1^T\vec{x} + \|\vec{a}_1\|^2 = d_1^2 \tag{1}$$
$$\vec{x}^T\vec{x} - 2\vec{a}_2^T\vec{x} + \|\vec{a}_2\|^2 = d_2^2 \tag{2}$$
$$\vec{x}^T\vec{x} - 2\vec{a}_3^T\vec{x} + \|\vec{a}_3\|^2 = d_3^2 \tag{3}$$

But we have squared terms here involving unknown variable $\vec{x}$. That's not really conducive to our style of computation - we prefer only linear terms in unknowns, so we can use linear algebra.

For this, we use a trick. Let's subtract equation 1 from equation 2, and separately again from equation 3. Then we get:

$$2(\vec{a}_1-\vec{a}_2)^T\vec{x} = \|\vec{a}_1\|^2 - \|\vec{a}_2\|^2 - d_1^2 + d_2^2$$

and

$$2(\vec{a}_1-\vec{a}_3)^T\vec{x} = \|\vec{a}_1\|^2 - \|\vec{a}_3\|^2 - d_1^2 + d_3^2$$

These two equations are linear in $\vec{x}$, so we can then stick them into a matrix:

$$\begin{bmatrix} 2(\vec{a}_1-\vec{a}_2)^T \\ 2(\vec{a}_1-\vec{a}_3)^T \end{bmatrix} \vec{x} = \begin{bmatrix} \|\vec{a}_1\|^2 - \|\vec{a}_2\|^2 - d_1^2 + d_2^2 \\ \|\vec{a}_1\|^2 - \|\vec{a}_3\|^2 - d_1^2 + d_3^2 \end{bmatrix}.$$

This system can be solved for a location. Three circles uniquely define a point in 2D. The argument extends in 3D to four spheres. And in 4D to five hyper-spheres. (In the real-world, time is the fourth dimension we need to solve for.)

22.3 Finding distances with correlation

Now, we'll explore how to find the distances between the receiver and the beacons. To do this, we'll need to be a little more explicit about what messages the beacons are sending. We will generally say that the beacons are sending "signals."

22.3.1 Signals

A **signal** is a message that contains information as a function of time[1]. We call a signal a *discrete-time signal* if it is only defined at specific points in time (for example, every minute). In contrast, a *continuous-time signal* is defined over all time. In this class we'll focus on discrete-time signals.

We can represent a discrete-time signal as a list of numbers. It's convenient to write the signal as a vector where each element is the value at a single time point – then we can use all of the concepts of matrix algebra on signals.

Consider the following example: a beacon in our GPS system is sending a message using radio waves. The amplitude of the radio wave as a function of time is a signal, and we only consider discrete times (for example, every second). We can represent the signal graphically or as a vector:

$$\vec{s} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

Every element of the vector represents the signal value at one timestep. We'll use the notation $s[k]$ to represent the k-th element of the vector where initial element is at $k = 0$. For example, in the signal $\vec{s}$ above, $s[0] = 0$, $s[1] = 1$, etc.

22.4 Cross-Correlation

The **cross-correlation** is a measure of the similarity between two signals $\vec{x}$ and $\vec{y}$ based on the inner product. We compute the inner product $\langle \vec{x}, \vec{y} \rangle$, and then the same with $\vec{y}$ shifted by 1, then shifted by 2, and so on. First we'll define cross-correlation more formally, then we'll explore how we can use this concept to find the distance between a beacon and receiver.

22.4.1 Cross-Correlation

Cross-correlation is defined as follows:

$$\text{corr}_{\vec{x}}(\vec{y})[k] = \sum_{i=-\infty}^{\infty} x[i]\, y[i-k] \tag{4}$$

where $\vec{x}$ and $\vec{y}$ are input signals and $\text{corr}_{\vec{x}}(\vec{y})[k]$ is the k-th element of their cross-correlation.

[1] Signals can also be functions of other variables, such as space (like in an image). However, for simplicity we'll define them as a function of time, while knowing that they can be used more generally.

Let's break this down: If we would like to find the value of the cross correlation at 0, we plug $k = 0$ into the equation:

$$\text{corr}_{\vec{x}}(\vec{y})[0] = \sum_{n=-\infty}^{\infty} x[n]y[n]$$

Recall that the inner product of two vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$ is

$$\langle \vec{u}, \vec{v} \rangle = \sum_{i=1}^{n} u_i v_i$$

which is very similar to the cross correlation at 0. However, there is one difference: the summation in the cross correlation equation goes from $-\infty$ to ∞. This means that we will need to know $x[n]$ and $y[n]$ for all n, even if $\vec{x}$, $\vec{y}$ were only defined over a finite range. **We make the assumption that $x[n]$ and $y[n]$ are 0 for any n that they are not defined for.** (This assumption is what differentiates a linear cross-correlation from a circular cross-correlation, which we will introduce in the next section.)

Consider two vectors, $\vec{s}_1 \in \mathbb{R}^3$ and $\vec{s}_2 \in \mathbb{R}^2$:

$$\vec{s}_1 = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \qquad \vec{s}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

We cannot calculate their inner product because they are not the same dimension, but we can calculate $\text{corr}_{\vec{s}_1}(\vec{s}_2)[0]$ because we assume that the smaller vector has another zero at the end, making them the same size.

$$\text{corr}_{\vec{s}_1}(\vec{s}_2)[0] = (1)(2) + (3)(1) + (2)(0) = 5$$

Now that we can find the cross correlation at 0, how do we find it at other k values? Looking back at the definition in Eq. (**??**), we see that $y[i-k]$ has all of the same values as $y[i]$, but shifted to the right (or down) by k. Therefore, to find $\text{corr}_{\vec{x}}(\vec{y})[k]$, we shift the second signal, $\vec{y}$, to the right by k units, then take the inner product between $\vec{x}$ and the shifted version of $\vec{y}$. As before, we assume that any values not defined in either vector are 0.

The figure below summarizes the linear cross correlation for the example vectors $\vec{s}_1$ and $\vec{s}_2$ defined above. Note that both signals have been padded with zeros on either side. Each value of the cross-correlation is found by shifting $\vec{s}_2$ (right for positive values, left for negative values) and then taking the inner product.

Eventually, the two signals no longer overlap (for example, when $\vec{s}_2$ is shifted by -2 units). When the signals don't overlap, the inner product will be 0, since every non-zero element will be multiplied by zero. At this point we don't need to calculate more of the cross correlation. If we write all of the non-zero values in a vector, we get the following in this example:

$$\text{corr}_{\vec{s}_1}(\vec{s}_2) = \begin{bmatrix} 1 \\ 5 \\ 8 \\ 4 \end{bmatrix}.$$

This is the same as what you get in python with `numpy.correlate([1, 3, 2], [2, 1], 'full')`. The 'full' tag tells python to return all of the non-zero values. However, python doesn't tell you which output corresponds to 0 shift of the signal – you must calculate that yourself based on the length of the input signals.

22.4.2 Correlation Interpretation

Now that we've introduced cross-correlation, what does it mean? We can think of the cross-correlation as a sliding inner product. Recall that the inner product can be written as

$$\langle \vec{x}, \vec{y} \rangle = \|\vec{x}\| \, \|\vec{y}\| \cos\theta$$

where θ is the angle between the vectors. If θ is low, it indicates that the vectors are pointed in almost the same direction, meaning the vectors are very similar. As θ approaches $90°$, the vector point increasingly different directions, indicating that the vector are not similar.

Therefore, when the inner product is large, $\vec{x}$ and $\vec{y}$ are more similar, and when it's small, they are more different. The cross-correlation checks the inner product at all relative shifts between the signals, so it tells us how similar two signals are at every shift.

22.4.3 Correlation Properties

It is important to note that correlation is not commutative:

$$\text{corr}_{\vec{x}}(\vec{y}) \neq \text{corr}_{\vec{y}}(\vec{x})$$

We can think of circular correlation as a special case of linear correlation, so it is not commutative either:

$$\text{circcorr}(\vec{x},\vec{y}) \neq \text{circcorr}(\vec{y},\vec{x})$$

In future classes, you'll learn about the *convolution* which is an operation similar to correlation but commutative. (A mathematical structure in machine learning called a *convolutional neural network* takes its name from this operation.)

We can take a correlation between a signal and itself. This is call an **autocorrelation**. The autocorrelation tells us how similar a signal is to all shifts of itself.

22.5 Modeling the positioning problem

Now that we've defined cross-correlation, we are ready to use it to solve our positioning problem. Recall that we have several beacons that are sending signals that our receiver picks up. We want to determine the distance from the receiver to each beacon. Let's start with the case where there is a single beacon a distance d away from the receiver.

The beacon sends a periodic signal using radio waves (or something else that travels through air, such as light or sound). Since radio waves travel at a finite velocity, there is a delay between the when the beacon sends the signal and when the receiver sees it. For example, in the diagram below, the value marked in red is sent at $t = 0$ and received at $t = 2$, indicating that there is a 2 second delay (assuming there is a sample ever second).

Our goal is to determine this time delay, τ. Then we can use the velocity of radio waves (or light, or sound), v, to determine the distance: $d = v\tau$.

We'll assume that the beacon is sending a known periodic signal, $\vec{s}$, with period N. The receiver will measure the first N samples. We can model the receiver signal, $\vec{r}$ as

$$\begin{aligned} \vec{r} &= \alpha \vec{s}^{(\tau)} \\ r[k] &= \alpha\, s[k-\tau] \end{aligned} \tag{5}$$

where α is a scalar that accounts for beacon's signal becoming weaker as it travels. τ is the time delay that we would like to measure.

We have N equations (from each of our N measurements) and two unknowns α and τ. However, τ is related to $\vec{r}$ and $\vec{s}$ in a nonlinear way, so we cannot use our standard linear algebra to solve this problem. What can we do?

Recall that the cross-correlation tells us how much of a signal is present at every shift. Therefore, we can estimate the true shift by finding the index of the cross-correlation with the largest value!

Let's walk through this with the example above. We know that the beacon is sending the following signal, $\vec{s}$ of length $N = 4$ that repeats periodically. We measure the first N samples at the receiver to get $\vec{r}$.

$$\vec{s} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 4 \end{bmatrix} \qquad \vec{r} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

We'll try taking the inner product of our received signal with every possible circular shift of our known beacon signal.

$$\begin{aligned} \text{circcorr}(\vec{r}, \vec{s}) &= \begin{bmatrix} \left\langle \vec{r}, \vec{s}^{(0)} \right\rangle \\ \left\langle \vec{r}, \vec{s}^{(1)} \right\rangle \\ \left\langle \vec{r}, \vec{s}^{(2)} \right\rangle \\ \left\langle \vec{r}, \vec{s}^{(3)} \right\rangle \end{bmatrix} \\ &= \begin{bmatrix} - & \vec{s}^{(0)^T} & - \\ - & \vec{s}^{(1)^T} & - \\ - & \vec{s}^{(2)^T} & - \\ - & \vec{s}^{(3)^T} & - \end{bmatrix} \begin{bmatrix} | \\ \vec{r} \\ | \end{bmatrix} \\ &= \begin{bmatrix} 2 & 4 & 6 & 4 \\ 4 & 2 & 4 & 6 \\ 6 & 4 & 2 & 4 \\ 4 & 6 & 4 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} 28 \\ 32 \\ 36 \\ 32 \end{bmatrix} \end{aligned}$$

The maximum is 36 at $k = 2$ which indicates that there is a two sample delay between $\vec{s}$ and $\vec{r}$ (remember, we the first value is at index 0 because there is no shift). Looking back at the figure above, we can see that this is true shift.

We can write this process for estimating τ using optimization notation:

$$\hat{\tau} = \arg\max_k (\text{circcorr}(\vec{r}, \vec{s})[k])$$

This means that our estimate of τ (denoted $\hat{\tau}$) is the value of k that makes $\text{circcorr}(\vec{r}, \vec{s})[k]$ largest. This style of notation is very common in optimization and machine learning.

22.6 Other definitions of Cross-Correlation [optional]

The following definitions are very similar to cross-correlation in spirit, but have some small differences in their actual computation. They are all out of scope for this course, but are presented here just if you're interested, or if you see them appear in materials from older versions of this course.

22.6.1 Circular Cross-Correlation

A circular cross-correlation with period N is defined as follows:

$$\text{circcorr}(\vec{x}, \vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[(i-k)_N] \tag{6}$$

There's some new notation here: $(i)_N$ is another way of writing "i mod N." $(i)_N$ is the remainder when i is divided by N. For example, if $N = 3$,

$$(0)_N = 0 \text{ mod } 3 = 0 \qquad (1)_N = 1 \text{ mod } 3 = 1 \qquad (2)_N = 2 \text{ mod } 3 = 2$$
$$(3)_N = 3 \text{ mod } 3 = 0 \qquad (4)_N = 4 \text{ mod } 3 = 1 \qquad (5)_N = 5 \text{ mod } 3 = 2$$

This notation allows us to count in a circular way: we count from 0 to $N-1$ and then repeat, starting at 0.

Now let's go back to our definition of circular cross-correlation. Once again, we'll start by considering when $k = 0$:

$$\text{circcorr}(\vec{x}, \vec{y})[0] = \sum_{i=0}^{N-1} x[i]\, y[(i)_N]$$

We are able to remove the N subscript in this case because i is always less than N due to the bounds on the summation:

$$\text{circcorr}(\vec{x}, \vec{y})[0] = \sum_{i=0}^{N-1} x[i]\, y[i]$$

Now this is exactly the inner product of the $\vec{x}$ and $\vec{y}$ if they are both length N. If either vector is longer than N, then it is the inner product of the first N elements. However, this case is less useful since some of the elements are never used in the computation. Therefore, for circular correlations we restrict ourselves to the case where $\vec{x}$ and $\vec{y}$ are both length N.

How about when k is not 0? Now, we take the inner product of $\vec{x}$ and a version of $\vec{y}$ that is circularly shifted by k. We'll denote this circularly shifted vector as $\vec{y}^{(k)}$. For example:

$$\vec{y} = \vec{y}^{(0)} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \qquad \vec{y}^{(1)} = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix} \qquad \vec{y}^{(2)} = \begin{bmatrix} 3 \\ 4 \\ 1 \\ 2 \end{bmatrix}$$

Using this notation

$$\text{circcorr}(\vec{x},\vec{y})[k] = \left\langle \vec{x}, \vec{y}^{(k)} \right\rangle = \vec{y}^{(k)^T}\vec{x}$$

We can write all of the values of the correlation in vector format:

$$\text{circcorr}(\vec{x},\vec{y}) = \begin{bmatrix} \left\langle \vec{x}, \vec{y}^{(0)} \right\rangle \\ \left\langle \vec{x}, \vec{y}^{(1)} \right\rangle \\ \vdots \\ \left\langle \vec{x}, \vec{y}^{(N-1)} \right\rangle \end{bmatrix}$$

Then we expand this into matrix-vector multiplication

$$\begin{aligned}
\text{circcorr}(\vec{x},\vec{y}) &= \begin{bmatrix} - & \vec{y}^{(0)^T} & - \\ - & \vec{y}^{(2)^T} & - \\ & \vdots & \\ - & \vec{y}^{(N-1)^T} & - \end{bmatrix} \begin{bmatrix} | \\ \vec{x} \\ | \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} y[0] & y[1] & \dots & y[N-2] & y[N-1] \\ y[N-1] & y[0] & \dots & y[N-3] & y[N-2] \\ \vdots & & & & \vdots \\ y[2] & y[3] & \dots & y[0] & y[1] \\ y[1] & y[2] & \dots & y[N-1] & y[0] \end{bmatrix}}_{\mathbf{C}_{\vec{y}}} \begin{bmatrix} | \\ \vec{x} \\ | \end{bmatrix} \\
&= \mathbf{C}_{\vec{y}}\vec{x}
\end{aligned}$$

Here, $C_{\vec{y}}$ is a matrix containing all of the shifted versions of $\vec{y}$ in its rows. This is called a **circulant matrix**, and is another way to write a circular correlation.

$$\mathbf{C}_{\vec{y}} = \begin{bmatrix} y[0] & y[1] & \dots & y[N-2] & y[N-1] \\ y[N-1] & y[0] & \dots & y[N-3] & y[N-2] \\ \vdots & & & & \vdots \\ y[2] & y[3] & \dots & y[0] & y[1] \\ y[1] & y[2] & \dots & y[N-1] & y[0] \end{bmatrix}$$

22.6.2 Linear, Circular, and Periodic Correlations

In order to understand the relationship between linear and circular correlations, we will introduce one more correlation: the **linear correlation for periodic signals**, sometimes called the **periodic correlation.** Before defining it, we'll motivate it by thinking about when linear and circular correlations (defined above) are equivalent.

Recall that a linear correlation is

$$\text{corr}_{\vec{x}}(\vec{y})[k] = \sum_{i=-\infty}^{\infty} x[i]\,y[i-k]$$

and a circular correlation is

$$\text{circcorr}(\vec{x},\vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[(i-k)_N]$$

In what case are they the same? If we assume that the second input, $\vec{y}$, is repeated indefinitely, then a circular shift of $\vec{y}$ is the same as a linear shift. (If we circularly shift $\vec{y}$, the end of the vector appears at the beginning. This is exactly what happens if there is another copy of $\vec{y}$ before the initial copy.) However, if we plug infinite repeating signals into the linear correction equation, we'll get infinity because the bounds on the sum are from from $-\infty$ to ∞.

Therefore, we can think of a circular correlation as a subset of a linear correlation where the signals are periodic with period N (i.e. repeating ever N samples) *and* we sum only over one period. We call this the **linear correlation for periodic signals** or **periodic correlation** which is defined as

$$\text{corr}_N(\vec{x},\vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[i-k]$$

We distinguish this from the linear correlation with the N subscript. (Linear correlations are denoted "corr" and periodic correlations are denoted "corr_N"). When we calculate $\text{corr}_N(\vec{x},\vec{y})$ we assume that $\vec{x}$ and $\vec{y}$ are periodic with period N.

Now let's examine the relationship between the three correlations we've introduced:

22.6.3 Circular Correlation vs. Periodic Correlation

Circular Corrlation:

$$\text{circcorr}(\vec{x},\vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[(i-k)_N]$$

Periodic Correlation:

$$\text{corr}_N(\vec{x},\vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[i-k]$$

In the circular correlation, when we shift $\vec{y}$, we shift it in a circular fashion – i.e. elements that shift off the end of the signal reappear at the beginning. This is mathematically described by the term $y[(i-k)_N]$ in the equation above (recall that the subscript means mod N). Note that we are not making any assumptions about the signal when we do this. Rather, we define the shift to be circular.

In contrast, in the periodic correlation we assume that the signal is periodic with period N and we shift $\vec{y}$ in a linear fashion. However, shifting a periodic signal linearly is the same as shifting a non-periodic signal in circular way! As we shift periodic $\vec{y}$ to the right, a copy of $\vec{y}$ appears on the left since the signal is repeated. This is exactly the same as if we circularly shifted $\vec{y}$.

In conclusion, these two correlations yield the exact same result! We can conceptually think of the circular correlation as shifting the signal in a circle, and we can think of the period correlation as shifting a repeating signal. However, since they give the same result, we can also interpret the the circular correlation as shifting a repeated signal (and vice versa). For positioning we will use periodic signals so periodic correlations or equivalently, circular correlations, are the natural operations to use.

22.6.4 Linear Correlation vs. Periodic Correlation

Linear Correlation:

$$\text{corr}_{\vec{x}}(\vec{y})[k] = \sum_{i=-\infty}^{\infty} x[i]\, y[i-k]$$

Periodic Correlation:

$$\text{corr}_N(\vec{x},\vec{y})[k] = \sum_{i=0}^{N-1} x[i]\, y[i-k]$$

In both linear and periodic correlations, we shift $\vec{y}$ in a linear fashion. However, in a linear correlation, we assume that all of the values outside of the defined range are zero. In a periodic correlation, we assume that the signal is repeating with period N.

In general, our signals are not infinite length. Therefore, when we assume zeros outside defined range (as in linear correlation), the sum is generally finite. However, if we assume a repeating signal (as in periodic correlation) the sum could be infinite, which is a problem. Therefore, we only define the periodic correlation over one period from 0 to $N-1$.

In conclusion, a periodic correlation (or equivalently, a circular correlation – see above section) is the same as a linear correlation where the signal is periodic and we sum over only one period.

22.7 Practice Problems

These practice problems are also available in an interactive form on the course website (`http://ee16a.com/hw-practice#/`). Note that some of them may reference definitions from the optional section above, and so are out of scope of the course.

1. True or False: We can always uniquely determine our position in an n-dimensional world using $n+1$ beacons.

2. Assume that you receive a shifted signal $\vec{r}$ and you correlate it with the original periodic signal $\vec{s}$, such that $\text{corr}_{\vec{s}}(\vec{r}) = \begin{bmatrix} -32 \\ -32 \\ 16 \\ 16 \\ 96 \\ 24 \end{bmatrix}$. What would you estimate as the sample delay between you and the transmitter?

3. True or False: $\|\mathbf{C}_{\vec{x}}\vec{y}\| = \left\|\mathbf{C}_{\vec{y}}\vec{x}\right\|$.

4. True or False: The projection of $\vec{y}$ onto $\vec{v}$ is the same as the projection of $\vec{y}$ onto $c\vec{v}$ whenever $c \neq 0$.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 23

23.1 Introduction: Least Squares

In our previous exploration of linear algebra, we developed techniques such as Gaussian Elimination and matrix inversion to solve systems of linear equations exactly. Gaussian Elimination and inversion work best when we have perfect measurements of our system, i.e. when there are no errors or noise in the system. However, given a system which might be prone to noisy measurements, Gaussian Elimination might not be able to solve it; since the equations might end up being inconsistent and have no solution.

For example, in the GPS problem, we use cross-correlation to measure the distance between the satellites (beacons) and the receiver to generate the distances to feed into the trilateration algorithm. However, interference in the wireless signals or errors in correlation can easily lead to noisy measurements of the distance between the receiver and the transmitter.

We'd like to develop a method to solve a system of linear equations even when the system might be inconsistent due to noise. In particular, our goal is to understand how collecting more equations than unknowns can help us reduce the impact of noise and find a solution to the system of equations that might not be perfect, but is as close as possible.

This note develops the *Least Squares* technique for approximately solving systems of linear equations in the presence of noise. Our focus will be on *overdetermined* systems of equations (more equations than unknowns), so we can use the extra equations to reduce the impact of noise. Least squares is the fundamental idea behind *data fitting* and *machine learning*: In data fitting, we find lines or curves that best match the data. In machine learning, we use a best-fit curve to make predictions about new, unseen data.

23.2 Approximate Solutions to Linear Systems

So far, we have studied many problems that ultimately reduce to linear equations of the form

$$A\vec{x} = \vec{b}.$$

When given a system of linear equations, we have sought to assign values to variables that satisfy *all* of the equations *exactly*.

However, when there is noise, this may result in an *inconsistent* system, where no $\vec{x}$ exists that satisfies all the equations exactly.

Fundamentally, least squares deals with choosing a solution $\vec{x}$ that may not satisfy all equations, but minimizes the magnitude of the *error* in our solution. The error of the (approximate) solution $\vec{x}$ is defined as:

$$\vec{e} = \vec{b} - A\vec{x}$$

Let us write out the columns of an $n \times m$ matrix A explicitly and rearrange the above expression, to obtain

$$\begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \vec{e} = \vec{b}.$$

Expanding out the matrix multiplication, we obtain

$$(x_1\vec{a}_1 + x_2\vec{a}_2 + \ldots + x_m\vec{a}_m) + \vec{e} = \vec{b},$$

In other words, we are trying to find the "closest" vector to $\vec{b}$ in the span of the columns $\{\vec{a}_i\}$ of A.

23.2.1 Least Squares: The 2D Case

How can we find such a vector? To gain some intuition, let's simplify the problem slightly, working in $n = 2$ dimensional space, where A has $m = 1$ columns i.e, A is a 2×1 column vector $\vec{a}$. Then we would like to find a solution x that minimizes the error $\|\vec{e}\| = \left\|\vec{b} - x\vec{a}\right\|$ (note that x is a scalar solution, and $\vec{b}$ and $\vec{e}$ are therefore 2×1 column vectors:

Since x is a scalar, the vector $x\vec{a}$ will have the same direction as $\vec{a}$. To minimize the magnitude of the error vector, we therefore want to find the closest point on the line defined by the vector a, to the vector b. The diagram sugggests that we should drop a perpendicular from $\vec{b}$ onto $\vec{a}$ to find the closest point (known as the *projection* of $\vec{b}$ onto $\vec{a}_1$), making the error $\vec{e}$ orthogonal to $\vec{a}$. We can make this intuition rigorous:

Consider a different solution $\alpha \in \mathbb{R}$ such that $\alpha\vec{a}$ lies in the span of $\vec{a}$ (since α is a scalar, this will always be true). Since we assume that $\alpha \neq x$, we can assume without loss of generality that $\alpha < x$. Consider the error between $\alpha\vec{a}$ and $\vec{b}$: $\vec{e}' = \vec{b} - \alpha\vec{a}$. Plotting this on a diagram, we see that $\vec{e}'$ forms the hypotenuse of a right triangle, where one leg is the vector $\vec{e}$:

Thus, by the Pythagorean Theorem, $\left\|\vec{e'}\right\|^2 = \|\vec{e}\|^2 + \|x\vec{a} - \alpha\vec{a}\|$. Since $\alpha \neq x$, we know that the last term is strictly positive. Hence, $\left\|\vec{e'}\right\| > \|\vec{e}\|$. Therefore the error with $\alpha\vec{a}$ is greater than the error with $x\vec{a}$, and hence α is not an optimal solution. Since α was an arbitrary solution, this shows that x is indeed optimal.

We have shown that the projection of $\vec{b}$ onto $\vec{a}$ is the solution that minimizes the error $\vec{e}$. But what is this projection? Well, we know that $\vec{e} \perp x\vec{a}$. Using the properties of inner products, we get:

$$\begin{aligned}
& & \langle \vec{e}, x\vec{a} \rangle &= 0 \\
&\implies & x\langle \vec{e}, \vec{a} \rangle &= 0 \\
&\implies & \left\langle \vec{b} - x\vec{a}, \vec{a} \right\rangle &= 0 \\
&\implies & \left\langle \vec{b}, \vec{a} \right\rangle - x\langle \vec{a}, \vec{a} \rangle &= 0 \\
&\implies & x &= \frac{\left\langle \vec{b}, \vec{a} \right\rangle}{\langle \vec{a}, \vec{a} \rangle}.
\end{aligned}$$

This is known as the *projection* of $\vec{b}$ onto the span of $\vec{a}_1$. It is also commonly called an *orthogonal projection*. Since the orthogonal projection gives the smallest error solution to $\mathbf{A}\vec{x} = \vec{b}$, we have essentially derived the Least Squares algorithm in two dimensions.

23.2.2 Least Squares: The General Case

Now, how do we generalize beyond two dimensions? Well, we argued in 2D that, in order to minimize the magnitude of the error, we should choose x_1 such that $\vec{e} \perp x_1$ - i.e $\vec{e}$ is orthogonal to the subspace in which $x_1\vec{a}_1$ lies. The analogous claim in higher dimensions, given that we wish to choose $\vec{x}$ to minimize the magnitude of the error

$$\vec{e} = \vec{b} - A\vec{x},$$

would be to choose $\vec{x}$ that make $\vec{e}$ orthogonal to the span of all possible $A\vec{x}$ - i.e $\vec{e}$ that is orthogonal to the column space of A. Can we justify this claim?

We can generalize what we did earlier in the $2D$ case to higher dimensions to derive the general Least Squares algorithm. What are all the possible values that $A\vec{x}$ can take? This is the set of all linear combinations of the columns of A, i.e the columnspace of A. This is a hyperplane in n-dimensional space that passes through the origin. To help us visualize this, we will illustrate the span of the columns of A as a 2D plane embedded in 3D space:

In the above diagram, the vector $\vec{b}$ does not lie in the column space of A, which is why doing Gaussian Elimination on $A\vec{x} = \vec{b}$ would lead to an inconsistent set of equations that cannot be solved.

Our goal is to find that vector $A\vec{x}$ in span(A) that is closest to $\vec{b}$ (i.e. we want to minimize the magnitude of the error $\|\vec{e}\| = \left\|\vec{b} - A\vec{x}\right\|$). This will give us a valid solution $\vec{x}$ (such that $A\vec{x}$ lies in the column space of A) and is as close as possible to one that generates the measurements $\vec{b}$.

We drop a perpendicular from the tip of the vector $\vec{b}$ (denoted by P) to the plane span(A), and label the point of intersecton with the columnspace of A point Q. Consider the vector $A\vec{x}$ that connects the origin O to point Q.

As before, our goal is to show that this choice of $\vec{x}$ minimizes the magnitude of the error $\|\vec{e}\|$. To do so, by analogy with our 2D geometric proof, we will consider any other $\vec{x'}$, where $A\vec{x'}$ is the point R on the plane span(A). Adding it to our figure, where $\vec{e'} = \vec{b} - A\vec{x'}$, we obtain

We'd like to show that $\left\|\vec{e'}\right\| \geq \|\vec{e}\|$. In the 2D case, we were able to do so by drawing a right triangle where $\vec{e}$ was a leg and $\vec{e'}$ the hypotenuse. Can we do something similar here?

From the figure, we see that the triangle formed by P, Q, and R is a triangle with the desired side lengths, as illustrated below:

But is it a *right* triangle? Observe that the bottom leg of the triangle is the vector $\vec{e}-\vec{e}' = A(\vec{x}-\vec{x'})$, which lies in the column space of A. Thus, by construction, it is orthogonal to $\vec{e}$, so this is indeed a right triangle!

Therefore, we can simply apply the Pythagorean theorem to conclude that $\left\|\vec{e'}\right\| \geq \|\vec{e}\|$, as desired. So as we have shown that no $\vec{x'}$ can produce a smaller error than $\vec{x}$, we have proven that $\vec{x}$ is the optimal solution!

23.2.2.1 Algebraic Proof

We can also show this using an algebraic proof. Assume that an $\vec{x}$ exists such that $\vec{e} = \vec{b} - A\vec{x}$ is orthogonal to all elements in the column space of A. Now, imagine for the sake of *contradiction* that there existed an alternative $\vec{x'}$ with corresponding error $\vec{e'} = \vec{b} - A\vec{x}$, such that

$$\left\|\vec{e'}\right\| < \|\vec{e}\|.$$

In other words, we are imagining that there exists a strictly *better* solution to $A\vec{x} \approx \vec{b}$ than $\vec{x}$. From the equation relating the norms of the errors, we have that

$$\begin{aligned} & \left\|\vec{e'}\right\| < \|\vec{e}\| \\ \implies & \left\|\vec{e'}\right\|^2 < \|\vec{e}\|^2 \\ \implies & \left\langle \vec{e'}, \vec{e'} \right\rangle < \langle \vec{e}, \vec{e} \rangle \end{aligned}$$

So far, all we've done is make simple manipulations to our inequality, squaring to write the norm of a vector in terms of inner products. Now, we will use a small trick. We need to use the fact that $\vec{e}$ is orthogonal to any element in the span of A *somewhere* in our proof. In other words, we want an expression of the form $\langle \vec{e}, A\vec{v} \rangle$ (for some vector $\vec{v}$) to appear somewhere, so that we can set it to 0 and apply this fact.

Where could such an expression $A\vec{v}$ come from? One way is to recognize that

$$\begin{aligned}\vec{e'} &= \vec{b} - A\vec{x'} \\ &= \vec{b} - A\vec{x} + A\vec{x} - A\vec{x'} \\ &= \vec{e} + A(\vec{x} - \vec{x'}),\end{aligned}$$

adding and subtracting the same term $A\vec{x}$ in order to bring $\vec{e}$ back into the equation. Making this substitution, we find that

$$\left\langle \vec{e} + A(\vec{x} - \vec{x'}), \vec{e} + A(\vec{x} - \vec{x'}) \right\rangle < \langle \vec{e}, \vec{e} \rangle .$$

Now, we will apply the distributive property of inner products on the left-hand-side of the inequality, to obtain

$$\langle \vec{e}, \vec{e} \rangle + \left\langle \vec{e}, A(\vec{x} - \vec{x'}) \right\rangle + \left\langle A(\vec{x} - \vec{x'}), \vec{e} \right\rangle + \left\langle A(\vec{x} - \vec{x'}), A(\vec{x} - \vec{x'}) \right\rangle < \langle \vec{e}, \vec{e} \rangle .$$

Recall that inner products of real vectors are commutative, so

$$\left\langle \vec{e}, A(\vec{x} - \vec{x'}) \right\rangle = \left\langle A(\vec{x} - \vec{x'}), \vec{e} \right\rangle .$$

Using the above identity to combine terms, and cancelling equal terms on both sides of the inequality, we find that

$$2\left\langle A(\vec{x} - \vec{x'}), \vec{e} \right\rangle + \left\langle A(\vec{x} - \vec{x'}), A(\vec{x} - \vec{x'}) \right\rangle < 0.$$

And now we can use our orthogonality assumption! Since $A(\vec{x} - \vec{x'})$ lies in the span of A, we know that it is orthogonal to $\vec{e}$, so

$$\left\langle A(\vec{x} - \vec{x'}), \vec{e} \right\rangle = 0.$$

Substituting, we find that

$$\left\langle A(\vec{x} - \vec{x'}), A(\vec{x} - \vec{x'}) \right\rangle < 0 \Rightarrow \left\| A(\vec{x} - \vec{x'}) \right\| < 0.$$

However, we know that norms are always nonnegative, and so the left-hand-side of the above expression must be greater than or equal to 0. We have therefore reached a contradiction.

Thus, our original assumption that an $\vec{x'}$ existed that was a better least squares solution than $\vec{x}$ must have been false. So we have shown *if* an $\vec{x}$ exists such that $\vec{b} - A\vec{x}$ is orthogonal to the span of A, that it must be the optimal solution to our least squares problem!

23.2.3 Least Squares

We know now that, *if* there exists an $\vec{x}$ such that $\vec{e} = \vec{b} - A\vec{x}$ is orthogonal to *every vector* in the column space of A, that such a vector would be the optimum solution to our least squares problem and minimizes $\|\vec{e}\| = \left\|\vec{b} - A\vec{x}\right\|$.

Theorem 23.1: A vector $\vec{e}$ is orthogonal to every vector in the column space of A if and only if it is orthogonal to each of the columns $\vec{a}_i$ that form the basis of its column space.

Proof. Observe that if $\vec{e}$ is orthogonal to every vector in the column space of A, then it is orthogonal to each of the $\vec{a}_i$, as each of the $\vec{a}_i$ are in the column space of A.

Now, we will try to prove the converse. Consider an arbitrary vector $\vec{v} \in \text{span}(A)$. By definition, we know that there exist coefficients α_i such that we can express

$$\vec{v} = \alpha_1 \vec{a}_1 + \alpha_2 \vec{a}_2 + \ldots + \alpha_m \vec{a}_m.$$

Now, consider our vector $\vec{e}$, that is orthogonal to each of the $\vec{a}_i$. In other words, we know that for any valid i,

$$\langle \vec{e}, \vec{a}_i \rangle = 0.$$

We wish to show that $\vec{e}$ is orthogonal to $\vec{v}$ as well. To show this, it is natural to try evaluating $\langle \vec{e}, \vec{v} \rangle$, in the hope that this turns out to be 0. Doing so, substituting in our various definitions, and applying the distributive property of inner products, we see that

$$\begin{aligned}
\langle \vec{e}, \vec{v} \rangle &= \langle \vec{e}, \alpha_1 \vec{a}_1 + \alpha_2 \vec{a}_2 + \ldots + \alpha_m \vec{a}_m \rangle \\
&= \langle \vec{e}, \alpha_1 \vec{a}_1 \rangle + \langle \vec{e}, \alpha_2 \vec{a}_2 \rangle + \ldots + \langle \vec{e}, \alpha_m \vec{a}_m \rangle \\
&= \alpha_1 \langle \vec{e}, \vec{a}_1 \rangle + \alpha_2 \langle \vec{e}, \vec{a}_2 \rangle + \ldots + \alpha_m \langle \vec{e}, \vec{a}_m \rangle \\
&= \alpha_1 \cdot 0 + \alpha_2 \cdot 0 + \ldots + \alpha_m \cdot 0 \\
&= 0,
\end{aligned}$$

as desired! In other words, if our $\vec{e}$ is orthogonal to all the basis vectors of a subspace, it is orthogonal to *every vector* in that subspace as well! □

We know now that, for our error vector $\vec{e}$ to be orthogonal to every vector in the column space of A, it suffices to say that it is orthogonal to each of the columns $\vec{a}_i$. Now, recall that the inner product of two vectors can be represented as a matrix multiplication, so

$$\langle \vec{e}, \vec{a}_i \rangle = 0 \iff \vec{a}_i^T \vec{e} = 0.$$

Since we have $\vec{a}_i^T \vec{e} = 0$ for all i, we can stack these scalar equations to form the single matrix equation

$$\begin{bmatrix} - & \vec{a}_1^T & - \\ - & \vec{a}_2^T & - \\ & \vdots & \\ - & \vec{a}_1^T & - \end{bmatrix} \vec{e} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

But we have,

$$\begin{bmatrix} - & \vec{a}_1^T & - \\ - & \vec{a}_2^T & - \\ & \vdots & \\ - & \vec{a}_1^T & - \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_m \\ | & | & & | \end{bmatrix}^T = A^T,$$

so we can simplify our equation to just

$$A^T \vec{e} = \vec{0}.$$

What do we do with this equation? We're interested in solving for $\vec{x}$, so it makes sense to substitute for

$\vec{e} = \vec{b} - A\vec{x}$, to obtain

$$\begin{aligned} & A^T(\vec{b} - A\vec{x}) = \vec{0} \\ \implies \quad & A^T A\vec{x} = A^T\vec{b}. \end{aligned}$$

Now, observe that (unlike A) $A^T A$ is a square $m \times m$ matrix! Is it invertible? We will show below that it is, in the case where the columns of A are linearly independent. In this case $A^T A$ is a square matrix with only a trivial nullspace, we know that it has a unique inverse $(A^T A)^{-1}$. Pre-multiplying our earlier equation by this inverse (which we know exists), we obtain

$$\begin{aligned} & A^T A\vec{x} = A^T\vec{b} \\ \implies \quad & (A^T A)^{-1} A^T A\vec{x} = (A^T A)^{-1} A^T\vec{b} \\ \implies \quad & \vec{x} = (A^T A)^{-1} A^T\vec{b}. \end{aligned}$$

This is the general least squares solution for $A\vec{x} \approx \vec{b}$ when A has independent columns, and is our final result!

23.3 Is $A^T A$ invertible?

Let us figure out when exactly we can invert $A^T A$. First, we will prove a helper theorem.

Theorem 23.2: $\text{Null}(A^T A) = \text{Null(A)}$, even when A has a nontrivial nullspace.

Proof. To see this, consider an arbitrary $\vec{v} \in \text{Null}(A^T A)$. By definition, we have that

$$A^T A\vec{v} = \vec{0}.$$

Now, we will make the "magic" step[1] of pre-multiplying by $\vec{v}^T$, to obtain

$$\vec{v}^T A^T A\vec{v} = \vec{v}^T\vec{0} = 0.$$

Rearranging the left-hand-side of the above identity, we find that

$$(A\vec{v})^T(A\vec{v}) = 0 \implies \langle A\vec{v}, A\vec{v}\rangle = \|A\vec{v}\|^2 = 0.$$

Thus, it is clear that $A\vec{v} = 0$, so $\vec{v} \in \text{Null}(A)$. Thus, we have shown that $\text{Null}(A^T A) \subseteq \text{Null}(A)$. Now, consider an arbitrary vector $\vec{v'} \in \text{Null}(A)$. Pre-multiplying by A^T, we have that

$$\begin{aligned} & A\vec{v'} = \vec{0} \\ \implies \quad & A^T A\vec{v'} = \vec{0}, \end{aligned}$$

so $\vec{v'} \in \text{Null}(A^T A)$.

[1] As an aside, make sure to understand the intuition behind our "magic" step - we wanted to write the left-hand-side as an inner product, to obtain an equation involving the norm of $A\vec{v}$. This kind of pre-multiplication in order to get an inner product is fairly common in these sorts of proofs, and is a good thing to have in your "toolbox" of algebraic tricks.

Consequently, $\text{Null}(A) \subseteq \text{Null}(A^T A)$. Combining this with the earlier result, we have that

$$\text{Null}(A) = \text{Null}(A^T A),$$

as desired! $\square$

We can now return to our least squares argument in the special case where A has linearly independent columns. Since all the $\vec{a}_i$ are independent, it is a consequence that A has only a trivial null space. From our helper theorem, we can now see that $A^T A$ also only has a trivial nullspace!

23.4 Application of Least Squares

Gauss used this technique to predict where certain planets would be in their orbit. A scientist named Piazzi made 19 observations over the period of a month in regards to the orbit of Ceres. Gauss used some of these observations. He also knew the general shape of the orbit of planets due to Kepler's laws of planetary motion. Gauss set up equations like:

$$\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \varepsilon y = 1$$

We can set up a matrix like:

$$\begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_n^2 & \dots & \dots & \dots & y_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

Here, the x's and y's are known: they are the coordinates of the measured positions of Ceres. The unknowns are $\alpha, \dots, \varepsilon$. We write the above equation with matrix/vector notation as follows

$$A\vec{v} = \vec{b}$$

where define

$$A = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_n^2 & \dots & \dots & \dots & y_n \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

Now we can use the least squares formula to estimate the unknown coefficients in $\vec{v}$:

$$\vec{v} = (A^T A)^{-1} A^T \vec{b}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{bmatrix} = (A^T A)^{-1} A^T \vec{b}$$

Once solved, we can now translate the coefficients, $\alpha, \ldots, \varepsilon$, back into an ellipse using the original equation $\alpha x^2 + \beta xy + \gamma y^2 + \delta x + \varepsilon y = 1$. A possible result might look like this:

As we can see, the least squares method is useful for fitting noisy or approximate measurements to a curve, provided that we know the general shape of the curve. In addition, this example shows least squares is not limited to lines.

23.5 Practice Problems

These practice problems are also available in an interactive form on the course website.

1. True or False: Least squares is a method for solving an underdetermined system of linear equations.
2. Find the least squares solution to $\begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 0 \\ -8 \end{bmatrix}$.
3. True or False: Least squares always has a unique solution given by $\vec{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b}$.
4. True or False: We can use the least squares method to perform regression with a sine function, solving for $y = a \cdot \sin(bx)$. Make sure to also understand the reasoning why.
5. Let's say that we have a scenario where we have a set of data which includes information about a given set of patients. We have their height, weight, age, and white blood cell count. We are trying to create a predictor function for white blood cell count by solving an equation of the form $\mathbf{A}\vec{x} = \vec{b}$. What information should be in the $\mathbf{A}$ matrix?
   (a) The white blood cell counts.
   (b) The height, weight, age, and white blood cell count of each patient.
   (c) The unknown parameters $\alpha_1, \alpha_2, \alpha_3$.
   (d) The height, weight, and age for each patient.

6. True or False: Let $\vec{x} = \text{proj}_{\text{Col}(\mathbf{A})}\vec{b}$ be the projection of $\vec{b}$ onto the column space of a matrix $\mathbf{A}$. Then, $\mathbf{A}^T(\vec{b} - \vec{x}) = \vec{0}$.

7. True or False: The projection of a vector $\vec{b}$ onto a set of vectors $\{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_k\}$ is equal to $\frac{\langle \vec{b}, \vec{a}_1 \rangle}{\langle \vec{a}_1, \vec{a}_1 \rangle}\vec{a}_1 + \frac{\langle \vec{b}, \vec{a}_2 \rangle}{\langle \vec{a}_2, \vec{a}_2 \rangle}\vec{a}_2 + \cdots + \frac{\langle \vec{b}, \vec{a}_k \rangle}{\langle \vec{a}_k, \vec{a}_k \rangle}\vec{a}_k$.

8. True or False: Given an arbitrary cost function, the error vector corresponding to the best approximation of a vector $\vec{b}$ to the column space of $\mathbf{A}$ is always orthogonal to $\text{Col}(\mathbf{A})$.

9. Find the best approximation $\hat{x}$ to the system of equations $\begin{cases} a_1 x = b_1 \\ a_2 x = b_2 \end{cases}$ given the cost function $\text{cost}(x) = 2(b_1 - a_1\hat{x})^2 + (b_2 - a_2\hat{x})^2$.

   (a) $\frac{a_1 b_1 + a_2 b_2}{a_1^2 + a_2^2}$

   (b) $\frac{2a_1 b_1 + a_2 b_2}{a_1^2 + a_2^2}$

   (c) $\frac{2a_1 b_1 + a_2 b_2}{2a_1^2 + a_2^2}$

   (d) $\frac{a_1 b_1 + 2a_2 b_2}{a_1^2 + 2a_2^2}$

EECS 16A Designing Information Devices and Systems I
Fall 2020 Lecture Notes Note 24

24.1 Introduction: Trilateration with multiple beacons

In the previous notes we learned how to find the distance from a beacon to a receiver using cross-correlation. Then we used trilateration to combine several of these measurements and find the the location of the receiver. However, we only looked at cross-correlation for finding the distance to *one* beacon, but we know that we need at least 3 beacons to find our location in 2D. How can we detect our distances from many beacons simultaneously?

In this note, we'll introduce an algorithm called **orthogonal matching pursuit (OMP)** that will allow us to unmix the signals from multiple beacons.

This algorithms isn't only useful for estimating distances – in addition, the beacons can encode messages in the signals they send, and using OMP we can recover these messages even when multiple beacons are transmitting at the same time.

24.2 Orthogonal Matching Pursuit (OMP)

We'll walk through the OMP algorithm using an example with three beacons. At the end of the note we'll provide a general formulation.

Consider three beacons (in GPS, these would be satellites) which are each transmitting their own unique periodic code. We'll denote these codes $\vec{s}_1$, $\vec{s}_2$, and $\vec{s}_3$. Each of these is length $N = 10$. For concreteness, let's say that the codes are as follows:

$$\vec{s}_1 = \begin{bmatrix} -4 \\ -1 \\ 1 \\ -1 \\ 1 \\ 0 \\ -3 \\ 4 \\ -4 \\ -3 \end{bmatrix} \qquad \vec{s}_2 = \begin{bmatrix} -4 \\ -5 \\ -4 \\ -4 \\ -2 \\ -2 \\ 4 \\ 2 \\ 1 \\ 3 \end{bmatrix} \qquad \vec{s}_3 = \begin{bmatrix} -2 \\ -1 \\ -1 \\ -4 \\ -4 \\ 5 \\ 2 \\ 4 \\ -4 \\ 1 \end{bmatrix}$$

Beacon 1 transmits $\vec{s}_1$ repeatedly, beacon 2 transmits $\vec{s}_2$ repeatedly, and so on. In addition, the beacons can each multiply their code by a scalar to encode additional information. For example, each beacon could tell the receiver the temperature at its location by multiplying its code by the temperature. (This could be helpful in determining which parts of the walls or doors need more insulation).

Our receiver is a different distance away from each beacon, so it receives a circularly shifted version of each code. It can't distinguish between the signals sent by the different beacons, so it measures the sum of the three incoming signals. We measure the first $N = 10$ samples at the receiver. Putting all of this together, we can model our received signal, $\vec{y}$, as

$$\vec{y} = a_1 \vec{s}_1^{(\tau_1)} + a_2 \vec{s}_2^{(\tau_2)} + a_3 \vec{s}_3^{(\tau_3)} \tag{1}$$

where $\vec{s}_1^{(\tau_1)}$ is $\vec{s}_1$ circularly shifted by τ_1 samples. The a values are the scalars that the beacons can use to send messages.

From our known codes $\vec{s}_1, \ldots, \vec{s}_3$ and our measured receiver signal $\vec{y}$, we'd like to find the coefficients $a_1, \ldots, a_3$ and the shifts $\tau_1, \ldots, \tau_3$.

Suppose our received signal is:

$$\vec{y} = \begin{bmatrix} -11 & 2 & -3 & 12.5 & 4 & 0.5 & 6.5 & -11 & -12 & -2.5 \end{bmatrix}^T$$

We will now decompose it into the scaled and shifted versions of the beacon codes with OMP. OMP is an *iterative* method, which means we will repeat the same set of steps several times before reaching an answer.

Iteration 1

We know that our receiver signal consists of the sum of several shifted and scaled codes. First we will find the code (and it's corresponding shift) that is most similar to the received signal. To do this, we'll **take the circular cross correlation of $\vec{y}$ with each of the different possible codes** ($\vec{s}_1, \vec{s}_2, \vec{s}_3$). These values are shown below graphically.

The shifted code that is most similar to the received signal is the one with largest cross-correlation with the received signal. Therefore, we **find the code and shift with the maximum (in absolute value) cross-correlation.** Why absolute value? The transmitted code could have been scaled by a negative number, which will result in a negative value of the cross-correlation when these signals are similar. In this example, the maximum cross-correlation is given by $\vec{s}_2^{(7)}$, i.e. code $\vec{s}_2$ at shift 7.

Now we will guess that this shifted code is the only one in the received signal. In other words, we'll create an estimate of $\vec{y}$, denoted $\hat{\vec{y}}$, which takes the form

$$\hat{\vec{y}} = x_1 \vec{s}_2^{(7)}.$$

We want to understand how well the signal $\vec{s}_2$ at shift 7 can explain the received signal $\vec{y}$. For this, we **set up and solve a least squares problem**. We define

$$A = \vec{s}_2^{(7)} \qquad \vec{x} = x_1 \qquad \hat{\vec{y}} = A\vec{x}.$$

To get the best estimate of $\vec{x}$, we try to find $\vec{x}$ that minimizes the difference between $\vec{y}$ and $\hat{\vec{y}}$. We do this with least squares:

$$\vec{x} = (A^T A)^{-1} A^T \vec{y}$$

Plugging in the values in our example gives

$$\vec{x} = 2.14.$$

so our estimate of the received signal is

$$\hat{\vec{y}} = A\vec{x} = 2.14\,\vec{s}_2^{(7)}$$

However, our estimate of the received signal $\hat{\vec{y}}$, does not match our actual received signal. Next we will **calculate the difference between the actual received signal and estimated received signal, called the residual or error.** We'll denote the error signal $\vec{e}$.

$$\vec{e} = \vec{y} - \hat{\vec{y}}$$
$$\vec{e} = \vec{y} - A\vec{x}$$

The values of $\vec{y}$, $\hat{\vec{y}}$, and $\vec{e}$ for our example are shown below:

The residual is not zero, so we know that there are more codes from different beacons in the received signal. To find them, we will repeat the same process, but instead of starting with our received signal, we'll start with the residual. The logic is that the first shifted code we find is the strongest and will possibly block us from seeing other weaker codes. We first found this strongest code, and then removed it from the receiver signal to give us a better chance of finding the other weaker codes.

We're now ready for the second iteration of OMP.

Iteration 2

In the first iteration, the first thing we did was cross-correlate the receiver signal $\vec{y}$ with each code. We'll do the same thing this time but use the residual $\vec{e}$ instead of the receiver signal. Below are the circular cross correlations between the residual and each code:

Once again, we will find the shifted code that is most similar to the residual by taking the maximum (in absolute value) of the cross correlation. In this example, we find that the maximum is code $\vec{s}_1$ at shift 3 (it's a negative value, but biggest in magnitude). Note that the cross-correlation with $\vec{s}_2$ at shift 7 (the maximum from the first iteration) is 0. This is because we completely removed this component from the residual, forcing $\vec{e}$ to be orthogonal to $\vec{s}_2^{(7)}$. This is where the "orthogonal" in "orthogonal matching pursuit" comes from.

Now we update our estimate of the received signal to include the two shifted codes that we've found. (Recall that in the first iteration we found $\vec{s}_2$ at shift 7, and in this iteration we found $\vec{s}_1$ at shift 3).

$$\hat{\vec{y}} = x_1 \vec{s}_2^{(7)} + x_2 \vec{s}_1^{(3)}$$

We don't assume that x_1 is the same value as before, since we want to account for the additional codes that we've found. Instead we solve for both coefficients, x_1 and x_2, using least squares. We setup our least squares problem by defining

$$A = \begin{bmatrix} | & | \\ \vec{s}_2^{(7)} & \vec{s}_1^{(3)} \\ | & | \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \hat{\vec{y}} = A\vec{x}$$

so that

$$\vec{x} = (A^T A)^{-1} A^T \vec{y}$$

If we plug in the values in our example, we get

$$\vec{x} = \begin{bmatrix} 2.0 \\ -1.12 \end{bmatrix}$$

and our estimate of our received signal is now

$$\hat{\vec{y}} = 2.0\, \vec{s}_2^{(7)} - 1.1\, \vec{s}_1^{(3)}$$

.

Finally, we calculate our residual using our updated estimate $\hat{\vec{y}}$.

$$\vec{e} = \vec{y} - \hat{\vec{y}}$$
$$\vec{e} = \vec{y} - A\vec{x}$$

The residual is still not zero, and we know there were three beacons, so we will do a third iteration to find another beacon code.

Iteration 3

First, we take the cross-correlation between the new residual and each of the beacon codes.

Next, we find the code and shift with the maximum (in absolute value) cross-correlation. In this example, the maximum is $\vec{s}_3$ at shift 4. Once again, notice that the codes we found in previous iterations ($\vec{s}_2$ at shift 7 and $\vec{s}_1$ at shift 3) have zeros in the cross-correlation. This is because the least squares step ensures that the residual is orthogonal to all previous codes.

Now, we update our estimate of the received signal to include all three shifted codes that we've found.

$$\hat{\vec{y}} = x_1 \vec{s}_2^{(7)} + x_2 \vec{s}_1^{(3)} + x_3 \vec{s}_3^{(4)}$$

We setup a least squares problem by defining

$$A = \begin{bmatrix} | & | & | \\ \vec{s}_2^{(7)} & \vec{s}_1^{(3)} & \vec{s}_3^{(4)} \\ | & | & | \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \hat{\vec{y}} = A\vec{x}$$

so that

$$\vec{x} = (A^T A)^{-1} A^T \vec{y}$$

If we plug in the values in our example, we get

$$\vec{x} = \begin{bmatrix} 2.0 \\ -1 \\ 0.5 \end{bmatrix}$$

and our estimate of our received signal is now

$$\hat{\vec{y}} = 2.0\,\vec{s}_2^{(7)} - 1\,\vec{s}_1^{(3)} + 0.5\,\vec{s}_3^{(4)}$$

Finally, we calculate our residual using our updated estimate $\hat{\vec{y}}$.

$$\vec{e} = \vec{y} - \hat{\vec{y}}$$
$$\vec{e} = \vec{y} - A\vec{x}$$

Our residual is $\vec{0}$! We've found all of the codes:

$$\vec{y} = \hat{\vec{y}} = -1\,\vec{s}_1^{(3)} + 2.0\,\vec{s}_2^{(7)} + 0.5\,\vec{s}_3^{(4)}$$

24.3 Stopping Condition: How many iterations?

In the example above we did three iterations of OMP and at the end the residual was $\vec{0}$. However, in most examples there will be some noise in the measurement, so the residual will never completely vanish. How do we know how many iterations to run? We have two options:

1. We can run iterations until the residual is small. We set a threshold value th and when the norm of the residual is less than th, we will assume that the signal contains only noise and we'll stop iterating.
2. From the problem statement, we may already know how many beacon codes are contained in the received signal, k. Each iteration provides one new code, so the number of codes present will determine the number of iterations. In practice, we may not know the exact number, but we can guess and then do a few more iterations just in case.

We will use both of these criteria and we'll stop iterating when either is met. In other words, we'll set a number of iterations k, but we'll stop before k iterations if $\|\vec{e}\| < th$.

24.4 OMP with many beacons

In the previous example, we looked at using OMP with three beacons, each with their own code. However, we can do OMP with many more beacons! These could be things like temperature sensors, light-meters, and humidity sensors. As in the previous example, each beacon has its own unique code to transmit, and the beacons send information by multiplying their codes with a scalar, a_i. The value of a_i is the message.

For OMP to work, each code should be "almost" orthogonal to all of the other codes, and to all the shifted versions of itself. (Orthogonal means that the inner product of two codes is exactly zero. "Almost" orthogonal means that the magnitude of the inner product is close to zero.) The "gold codes" that we looked at in the homework are good examples of codes that fulfill these properties and would work well for OMP.

Now imagine we have 2,000 beacons, each with its own code: $\vec{s}_0, \vec{s}_1, \ldots, \vec{s}_{1999}$. Each message is a real number that scales the code: $a_0, a_1, \ldots, a_{1999}$, for each of the 2,000 sensors. Each of the messages from the transmitters is delayed by a different amount, given by: $\tau_0, \tau_1, \ldots, \tau_{1999}$. Each code is periodic with period $N = 400$.

The signal received ($\vec{y} \in \mathbb{R}^{400}$) is a linear combination of shifted codes from all transmitting users given by:

$$\vec{y} = a_0 \vec{s}_0^{(\tau_0)} + a_1 \vec{s}_1^{(\tau_1)} + \ldots + a_{1999} \vec{s}_{1999}^{(\tau_{1999})}$$

Here, the subscript of $\vec{s}$ indicates which beacon is transmitting, and the superscript in parenthesis represents the how many samples the code is shifted by.

We have 400 measurements ($\vec{y} \in \mathbb{R}^{400}$) and 2000 unknown values of a. How can we possibly determine the messages? We need more information! The additional information is that only a small number (say 10) beacons are transmitting at the same time. This means that only 10 of the beacons have non-zero a_i coefficients, but we don't know which beacons (and we don't know their shifts). If we put all of the a_i's in a vector, most of the vector would contain 0's. We call this a **sparse** vector and the **sparsity level** k is the number of non-zero elements.

In the simplest case, there will be only one beacon transmitting data to the receiver, for example. If beacon 130 is sending a message of value 1 with a delay of 10, the received signal will be:

$$\vec{y} = 1 \cdot \vec{s}_{130}^{(10)}$$

Now if we perform cross-correlation of the received signal with the code from beacon 130, $\vec{s}_{130}$, the result will have a peak at shift index 10, with a normalized amplitude of 1, shown below:

Correlation of received signal with circular shifts of $\vec{S_{130}}$,
showing a peak at the true shift index of 10

If we correlate with all codes and plot the maximum from each one, we get the results shown below. There is a single peak at the beacon ID 130, telling us that beacon 130 was "on" and sending a value of 1.

Correlation of received signal with circular shifts of all the codes.
This is only on peak, at a shift of 10 for beacon 130.

What happens when multiple transmitters are "on"?

Imagine beacons 40, 100, and 312 are simultaneously transmitting with delays of 13, 20, and 45 (respectively) and message values of 10, 10, and 8 (respectively). Our received signal is now:

$$\vec{y} = 10\,\vec{s}_{40}^{(13)} + 10\,\vec{s}_{100}^{(20)} + 8\,\vec{s}_{312}^{(45)}$$

If we cross-correlate the received signal with every code, find the maximum peak of each cross-correlation, and plot it for each beacon ID (normalizing by the length of the code), we get the plot below.

Maximum value of correlation of received signal with circular shifts of all the codes.

We can see above, that beacons 40, 100, 312 are all "on".

Let us look at one more example in which 4 beacons (beacon ID 40, 100, 312 and 350) transmit simultaneously and their corresponding messages are 100, 10, 8 and 0.02. The result of correlating the received signal with all of the different codes is shown below:

Maximum value of correlation of received signal with circular shifts of all the codes.

The peaks corresponding to sensor 100, 312 and 350 don't seem to appear at all! Even if we zoom in, we can't distinguish peaks corresponding to these sensors. What could possibly cause this? Perhaps the very high value of user 40's made it so that we cannot see the codes from users 100, 312 and 350.

However, as discussed in the previous example, OMP identifies the dominant code in the received signal, then removes it! We can use OMP in this example to recover the weaker codes as well as the dominant ones.

Here is a description of OMP with an example received signal of : $\vec{y} = a_{40}\vec{s}_{40}^{(13)} + a_{100}\vec{s}_{100}^{(8)}$ where $a_{40} = 100$ and $a_{100} = 10$.

1. Find the code with the highest correlation with $\vec{y}$, and find the shift that maximizes the correlation. We'll denote this shifted code as vector $\vec{s}_*^{(*)}$, where $*$ indicates the code ID and shift that maximize the cross-correlation. What does it mean for a vector $\vec{s}_*^{(*)}$ to have the highest correlation with $\vec{y}$? The highest correlation between two vectors means the error vector between them is the lowest. We find this vector via a cross-correlation with all the circular shifts of all the codes, the same first step we have always done. This gives us $\vec{s}_{40}^{(13)}$.

2. Use least squares to solve for $\vec{x}$ in the equation $A\vec{x} = \vec{b}$ where $\vec{b}$ is the received signal $\vec{y}$ and A is $\begin{bmatrix} | \\ \vec{s}_{40}^{(13)} \\ | \end{bmatrix}$: $\vec{x} = (A^T A)^{-1} A^T \vec{y}$. In this case, $\vec{x} = 100 = a_{40}$. The orthogonal projection of the least squares solution onto the subspace spanned by A (aka our 'ideal message') is given by: $A\vec{x}$.

3. Now find the residual of $\vec{y}$, denoted $\vec{e}$, left over by subtracting our ideal message from the received signal: $\vec{e} = \vec{y} - A\vec{x}$.

4. Repeat the above steps now using the residual $\vec{e}$ instead of the received signal $\vec{y}$. A new correlation between $\vec{e}$ and all of possible codes would find that the next strongest code is: $\vec{s}_{100}^{(8)}$. We then update our matrix A to be $\begin{bmatrix} | & | \\ \vec{s}_{40}^{(13)} & \vec{s}_{100}^{(8)} \\ | & | \end{bmatrix}$ Finally, we solve $\vec{x}$ via least squares. For this step, we use the received signal $\vec{y}$ (not the residual), $\vec{x} = (A^T A)^{-1} A^T \vec{y}$. We find that $\vec{x} = \begin{bmatrix} 100 \\ 10 \end{bmatrix}$. We have now recovered both of our message values!

5. We stop iterating when we have gone $k = 10$ steps, the known the sparsity of the signal, OR until the norm of the residual is below some threshold value, meaning the residual contains only noise.

24.5 OMP Summary

Setup:
Let the number of unique codes be m. Each code is length n. We represent each of the m codes with a vector $\vec{s}_i$ where i is an integer between 0 and $m-1$. Each code can potentially carry a message a_i along with it. Then the measurement at the receiver is

$$\vec{y} = a_0\vec{s}_0^{(\tau_0)} + a_1\vec{s}_1^{(\tau_1)} + \ldots + a_{m-1}\vec{s}_{m-1}^{(\tau_{m-1})}$$

We will assume that the number of codes that are being broadcast at the same time is very small (for example, 10). If a code is not being broadcast, it's coefficient a_i will be zero. Therefore, there are at most k non-zero a_i's.

Inputs:

- A set of m codes, each of length n: $\mathbf{S} = \{\vec{s}_0, \vec{s}_1, ..., \vec{s}_{m-1}\}$
- An n-dimensional received signal vector: $\vec{y}$
- The sparsity level k of the signal. This is the number of codes with non-zero coefficients.
- Some threshold, th. When the norm of the signal is below this value, the signal is assumed to contain only noise.

Outputs:

- A set of codes that were identified, F, which will contain at most k elements.
- A vector, $\vec{x}$ containing the coefficients of the codes (a_1, etc.), which will be of length k or less.
- An n-dimensional residual $\vec{e}$

Procedure:

- Initialize the following values: $\vec{e} = \vec{y}$, $j = 1$, $A = [\ \ ]$, $F = \{\emptyset\}$
- while $((j \leq k)$ & $(\| \vec{e} \| \geq th))$:
  1. Cross correlate $\vec{e}$ with each of the codes. Find the code index, i, and the shifted version of the code, $\vec{s}_i^{(\tau_i)}$, with which the received signal has the highest correlation value.
  2. Add i to the set of code indices, F.
  3. Column concatenate matrix A with the correct shifted version of the code: $A = [A \mid \vec{s}_i^{(\tau_i)}]$
  4. Use least squares to obtain the code coefficients: $\vec{x} = (A^T A)^{-1} A^T \vec{y}$
  5. Update the residual value $\vec{y}$ by subtracting: $\vec{e} = \vec{y} - A\vec{x}$
  6. Update the counter: $j = j + 1$
- Stop iterating when you've done k iterations (i.e. you've reached the pre-determined sparsity level of the signal) or when the norm of the residual drops below the threshold th. We'll stop at which ever one happens first.

24.6 Connections to Machine Learning

In this note, we showed you how by thinking systematically using the design philosophies taught in 16A, you could have systematically have come up with the algorithm that is commonly known as orthogonal matching pursuit or OMP. While this specific algorithm is certainly practically important in its own right as a building block, it is also a prime representative of a family of closely related algorithms that are very important in machine learning — namely what are called "boosting" algorithms (in case you want to read further on your own — there's no rush, these are topics that are usually covered if/when you take 189).

For those who want to know more, specifically "gradient boosting" algorithms can be understood as nothing more than a version of these ideas to be used when the relevant distance is not Euclidean squared distance, but something else. But all the main ideas are already present in this note about OMP. The most famous

example of a gradient boosting algorithm is called AdaBoost, which is sometimes referred to as“the best out-of-the-box classificatio” algorithm for practical machine learning. In fact, modern machine learning is largely based on working through ideas whose seeds are taught to you across 16A and 16B.

To be more specific, you saw how from the hypothetical real world goal of being able to extract signals from many potential beacons, which might be at differing levels of signal strength, you could still find them by iteratively reducing interference one at a time.

The first approach you could think of was to simply ignore the interference and just search for the beacons one at a time. You saw that the problem was that a strong beacon signal could completely mask your ability to see a weaker signal.

To deal with this, you naturally thought of trying to find the signals in decreasing order of signal strength. First, you find the strongest one. Then, you remove its estimated effect from what you observed. And then, you tried to find the signal which was presumably the next strongest. And then remove its effect, and so on. This pattern is what is called “matching pursuit” in the literature. (Because it can be understood as trying to greedily reach a destination by picking the available direction that would bring you as close as possible, actually moving along that direction, and then repeating from your new vantage point.)

You saw that while this worked better, it didn’t revisit its estimate of the first found signal in light of the second signal that was found, and so on. This made errors in the residual signal potentially compound as you found more and more signals, making it harder to find weaker signals as you went on. So, you made one final version that took advantage of what you knew about projection to keep improving estimates as we found more signals — allowing us to better clean up the residual. This is what gave us orthogonal matching pursuit OMP.

While doing all this, you also noticed something else. OMP allows us to potentially break what seemed like a fundamental barrier at the beginning of 16A — namely that we need to have n equations/measurements if we want to solve for n unknowns. OMP tells us that in fact, we often can find n unknowns with fewer than n equations, if we somehow happened to know that most of the n potential unknowns were going to come out to be zero. This is the powerful idea of sparsity. This will be explored further in parts of 16B and beyond, but is actually something that is believed to be at the heart of what is sometimes considered the “unreasonable effectiveness” of machine learning algorithms in the real world.

EECS 16A Designing Information Devices and Systems I
Spring 2021 Lecture Notes Note 25

25.1 Trilateration

Recall that GPS works by receiving signals from many satellites, computing the time delay t of each signal, and using these time delays to determine the receiver's location.

In the previous note, we saw how to use correlation to compute these time delays. Now, we will see how to use these time delays to determine the actual position of our receiver, in a process known as *trilateration*.

25.2 Computing distances from time delays

Imagine that each of the beacons start broadcasting their signal at time $t = 0$. Assume also that our receiver starts recording at this same time $t = 0$. Since our signals travel at a finite velocity v (such as the speed of light for radio waves, or the speed of sound for audio signals), it will take some time for each signal to reach our receiver. Specifically, if beacon i is at some distance d_i from our receiver, then it will take time d_i/v to reach the receiver, so the time delay becomes

$$t_i = \frac{d_i}{v}.$$

Thus, we can rearrange the above to compute all the distances as $d_i = vt_i$.

25.3 Determining our location

Let's imagine we have a situation like the one below. We know the locations of the beacons $\vec{a}_0, \vec{a}_1, \vec{a}_2$, but don't know the location of the point at $\vec{x}$. Our goal will be to solve for $\vec{x}$, given the distances d_0, d_1, d_2.

By the Pythagorean theorem, we have that

$$\begin{aligned}
\|\vec{a}_0 - \vec{x}\|^2 &= d_0^2 \\
\|\vec{a}_1 - \vec{x}\|^2 &= d_1^2 \\
\|\vec{a}_2 - \vec{x}\|^2 &= d_2^2.
\end{aligned}$$

We wish to solve for $\vec{x}$. It is natural to rewrite the squared norms using inner products and expand out the squared binomial terms, in order to separate out the terms involving $\vec{x}$ from the terms that are known.

For example, we expand the first equation $\|\vec{a}_0 - \vec{x}\|^2 = d_0^2$ below:

$$\begin{aligned}
\|\vec{a}_0 - \vec{x}\|^2 &= d_0^2 \\
\langle \vec{a}_0 - \vec{x}\,,\, \vec{a}_0 - \vec{x} \rangle &= d_0^2 \\
\langle \vec{a}_0 - \vec{x}, \vec{a}_0 \rangle - \langle \vec{a}_0 - \vec{x}, \vec{x} \rangle &= d_0^2 \\
\langle \vec{a}_0, \vec{a}_0 \rangle - \langle \vec{x}, \vec{a}_0 \rangle - \langle \vec{a}_0, \vec{x} \rangle + \langle \vec{x}, \vec{x} \rangle &= d_0^2 \\
\vec{a}_0^T \vec{a}_0 - 2\vec{a}_0^T \vec{x} + \vec{x}^T \vec{x} &= d_0^2.
\end{aligned}$$

In total this yields 3 equations:

$$\begin{aligned}
\vec{a}_0^T \vec{a}_0 - 2\vec{a}_0^T \vec{x} + \vec{x}^T \vec{x} &= d_0^2 \\
\vec{a}_1^T \vec{a}_1 - 2\vec{a}_1^T \vec{x} + \vec{x}^T \vec{x} &= d_1^2 \\
\vec{a}_2^T \vec{a}_2 - 2\vec{a}_2^T \vec{x} + \vec{x}^T \vec{x} &= d_2^2.
\end{aligned}$$

We're interested in solving for $\vec{x}$. If the above system were a linear system, then we could do so using Gaussian elimination. Unfortunately, the $\vec{x}^T \vec{x} = (x_1)^2 + (x_2)^2 + \ldots$ quadratic terms prevent us from doing so. To eliminate these terms, we can exploit the fact that each equation contains $\vec{x}^T \vec{x}$ with the same sign and size so we may subtract the first of our equations from each of the other two. In doing so, the $\vec{x}^T \vec{x}$ term will subtract out, leaving us with a linear system!

Doing so, we obtain the two equations:

$$\begin{aligned}
\vec{a}_1^T \vec{a}_1 - \vec{a}_0^T \vec{a}_0 - 2\vec{a}_1^T \vec{x} + 2\vec{a}_0^T \vec{x} &= d_1^2 - d_0^2 \\
\vec{a}_2^T \vec{a}_2 - \vec{a}_0^T \vec{a}_0 - 2\vec{a}_2^T \vec{x} + 2\vec{a}_0^T \vec{x} &= d_2^2 - d_0^2.
\end{aligned}$$

Simplifying and collecting terms within each equation, we can rewrite the above system as

$$\begin{bmatrix} 2(\vec{a}_0^T - \vec{a}_1^T) \\ 2(\vec{a}_0^T - \vec{a}_2^T) \end{bmatrix} \vec{x} = \begin{bmatrix} \vec{a}_1^T \vec{a}_1 - \vec{a}_0^T \vec{a}_0 + d_0^2 - d_1^2 \\ \vec{a}_2^T \vec{a}_2 - \vec{a}_0^T \vec{a}_0 + d_0^2 - d_2^2 \end{bmatrix}.$$

Now, we have two linear equations. So if we are working in two dimensions, we can solve for the two components of $\vec{x}$ and recover our position. If we are working in three dimensions, however, then we do not have enough equations - we would need to add one more beacon $\vec{a}_3$ in order to get our third linear equation. More generally, if we are working in n-dimensional space, we need at least $n+1$ beacons in order to determine our position, since we "lose" one beacon in our subtraction trick in order to linearize the system.

25.4 Unsynchronized Clocks

There is one critical assumption baked into the above approach. To find the time delays t_i, assume that our receiver starts recording at the same time $t = 0$ that the beacons start broadcasting. Then we use cross-correlation to compute each t_i, and then solve for d_i and do trilateration as described above.

But how does our receiver know to start recording at exactly that time $t = 0$? It must have some clock that is synchronized with the beacons, in order to know when to begin recording. In reality, however, such a clock does not exist - while the beacons can all be synchronized to start broadcasting at the same time, the receiver cannot guarantee that it starts recording at exactly that same time. Instead, it begins recording at some time Δt, representing the offset from $t = 0$ when we start recording. Note that Δt may be either positive or negative, but is unknown to the receiver.

Consider the following figure, representing the signals broadcast by the beacons and received by the receiver:

Notice that the received signal is delayed by some amount, due to its travel time. As we discussed earlier, letting d_i be the distance between the receiver and the beacon, this delay will equal d_i/v.

If the receiver were able to synchronize its recording with the time at which the signal was broadcast, then we could use cross-correlation to compute the above delay, recover d_i, and perform trilateration in the manner described above. However, what if we were unable to start our recording at exactly $t = 0$, but instead started at some unknown time $t = \Delta t$?

As shown above, the recording begins only after some time Δt. When we perform cross-correlation, we determine that the signal has been delayed by some time t_i after we began recording. But as shown in the above diagram, the signal has in fact been traveling for a time $t_i + \Delta t$, not simply t_i.

So we may write our observations t_i

$$\begin{aligned} t_0 &= \frac{d_0}{v} - \Delta t \\ t_1 &= \frac{d_1}{v} - \Delta t \\ &\vdots \\ t_{n-1} &= \frac{d_{n-1}}{v} - \Delta t. \end{aligned}$$

So what do we do now? Since Δt is unknown, we can't recover the d_i directly and use our previous approach to solve for our position. Instead, we can rearrange and write the d_i in terms of both t_i (which are known) and Δt (which is not):

$$\begin{aligned} d_0 &= v(\Delta t + t_0) \\ d_1 &= v(\Delta t + t_1) \\ &\vdots \\ d_{n-1} &= v(\Delta t + t_{n-1}). \end{aligned}$$

Observe that this system of equations looks very similar to the system in the previous section, except in the previous system of equations $\Delta t = 0$. This makes sense, since in our previous setup we were told that the receiver began recording at time 0, whereas now all we know is that it starts recording at some unknown time Δt. Thus, we see from these equations that the time taken for the ith signal to travel from the beacon to the receiver, known as the "time of flight," is $\Delta t + t_i$.

Note that since all the *beacons* are still synchronized, they all broadcast together at the same time, so we only have one unknown Δt, not n of them.

Unfortunately, since we don't know Δt, we can't use the same approach that we did earlier. Ideally, we'd like to eliminate Δt from our equations, so that the distances are the only remaining unknowns. One natural way of doing this is to subtract the first equation from all the rest, to obtain:

$$\begin{aligned} d_1 - d_0 &= v(t_1 - t_0) \\ d_2 - d_0 &= v(t_2 - t_0) \\ &\vdots \\ d_{n-1} - d_0 &= v(t_{n-1} - t_0). \end{aligned}$$

For convenience, we will define the "time difference of arrival", or TDOA, of each beacon i to be $\tau_i = t_i - t_0$. Observe that while we cannot directly measure the time of flight' $t_i - \Delta t$ since Δt is unknown, we *can* measure these time differences of arrival, as all the t_i *are* known.

Observe that d_0 occurs many times in the above system of equations. To make future calculations simpler, we will choose our origin to be the position of the 0th beacon and place the receiver at the position $\vec{x}$ in this coordinate system, so

$$d_0^2 = \vec{x}^T \vec{x}.$$

As before, we let the coordinates of the ith beacon be $\vec{a}_i$, with $\vec{a}_0 = \vec{0}$.

Substituting in the distances computed using the Pythagorean theorem, we obtain the equations

$$\begin{aligned}
\sqrt{(\vec{x}-\vec{a}_1)^T(\vec{x}-\vec{a}_1)} - \sqrt{\vec{x}^T\vec{x}} &= v\tau_1 \\
\sqrt{(\vec{x}-\vec{a}_2)^T(\vec{x}-\vec{a}_2)} - \sqrt{\vec{x}^T\vec{x}} &= v\tau_2 \\
&\vdots \\
\sqrt{(\vec{x}-\vec{a}_{n-1})^T(\vec{x}-\vec{a}_{n-1})} - \sqrt{\vec{x}^T\vec{x}} &= v\tau_{n-1}.
\end{aligned}$$

How can we solve these equations? The square roots certainly aren't helpful, so let's see if we can get rid of them. Squaring each equation would remove some square roots, but then we'd get a cross term caused by the product of the two terms on the left hand side, which looks difficult to deal with. Instead, let's pull one square root to the other side, so that when we then square, we won't get as ugly a cross term.

Rearranging in such a manner, we obtain

$$\begin{aligned}
\sqrt{(\vec{x}-\vec{a}_1)^T(\vec{x}-\vec{a}_1)} &= v\tau_1 + \sqrt{\vec{x}^T\vec{x}} \\
\sqrt{(\vec{x}-\vec{a}_2)^T(\vec{x}-\vec{a}_2)} &= v\tau_2 + \sqrt{\vec{x}^T\vec{x}} \\
&\vdots \\
\sqrt{(\vec{x}-\vec{a}_{n-1})^T(\vec{x}-\vec{a}_{n-1})^2} &= v\tau_{n-1} + \sqrt{\vec{x}^T\vec{x}}.
\end{aligned}$$

And now squaring, we obtain

$$\begin{aligned}
(\vec{x}-\vec{a}_1)^T(\vec{x}-\vec{a}_1) &= (v\tau_1)^2 + \vec{x}^T\vec{x} + 2v\tau_1\sqrt{\vec{x}^T\vec{x}} \\
(\vec{x}-\vec{a}_2)^T(\vec{x}-\vec{a}_2) &= (v\tau_2)^2 + \vec{x}^T\vec{x} + 2v\tau_2\sqrt{\vec{x}^T\vec{x}} \\
&\vdots \\
(\vec{x}-\vec{a}_{n-1})^T(\vec{x}-\vec{a}_{n-1}) &= (v\tau_{n-1})^2 + \vec{x}^T\vec{x} + 2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}
\end{aligned}$$

Does this look any better? Ideally, we'd get a linear equation, which we could then solve using Gaussian elimination or least squares, depending on the number of beacons. But there are still terms of the form $\vec{x}^T\vec{x}$ and $2\sqrt{\vec{x}^T\vec{x}}$ that keep this system nonlinear. Can we get rid of these terms?

The first thing to observe is that, by expanding out the left-hand-side, we get a $\vec{x}^T\vec{x}$ term, which cancels with the right hand side. Awesome! Performing this expansion, we obtain

$$\begin{aligned}
\vec{x}^T\vec{x} - 2\vec{a}_1^T\vec{x} + \vec{a}_1^T\vec{a}_1 &= (v\tau_1)^2 + \vec{x}^T\vec{x} + 2v\tau_1\sqrt{\vec{x}^T\vec{x}} \\
\vec{x}^T\vec{x} - 2\vec{a}_2^T\vec{x} + \vec{a}_2^T\vec{a}_2 &= (v\tau_2)^2 + \vec{x}^T\vec{x} + 2v\tau_2\sqrt{\vec{x}^T\vec{x}} \\
&\vdots \\
\vec{x}^T\vec{x} - 2\vec{a}_{n-1}^T\vec{x} + \vec{a}_{n-1}^T\vec{a}_{n-1} &= (v\tau_{n-1})^2 + \vec{x}^T\vec{x} + 2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}.
\end{aligned}$$

And canceling, we then obtain

$$\begin{aligned}
-2\vec{a}_1^T\vec{x}+\vec{a}_1^T\vec{a}_1 &= (v\tau_1)^2+2v\tau_1\sqrt{\vec{x}^T\vec{x}}\\
-2\vec{a}_2^T\vec{x}+\vec{a}_2^T\vec{a}_2 &= (v\tau_2)^2+2v\tau_2\sqrt{\vec{x}^T\vec{x}}\\
&\vdots\\
-2\vec{a}_{n-1}^T\vec{x}+\vec{a}_{n-1}^T\vec{a}_{n-1} &= (v\tau_{n-1})^2+2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}.
\end{aligned}$$

Is this system of equations linear? Recall that the $\vec{a}_i$ are constants (since their values are known), so the presence of $\vec{a}_i^T\vec{a}_i$ terms *does not* make the system nonlinear, since they are also just constant terms.

However, the $2\sqrt{\vec{x}^T\vec{x}}$ term is a nonlinear term, since $\vec{x}$ is an unknown. So our system isn't linear just yet. To get rid of this term, we could try to use the same "trick" we used in the previous section, where we subtracted equation 1 from all the other equations to get rid of the $\vec{x}^T\vec{x}$ term. Unfortunately, here there are multiplicative coefficients in front of the $2\sqrt{\vec{x}^T\vec{x}}$ term, so we can't directly subtract it out. Instead, we need to first scale each equation in order to get rid of the constant, and then do the subtraction trick.

Dividing each equation by $v\tau_i$ to get rid of the constant in front of $2\sqrt{\vec{x}^T\vec{x}}$, we obtain the system

$$\begin{aligned}
\frac{\vec{a}_1^T\vec{a}_1-2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_1+2\sqrt{\vec{x}^T\vec{x}}\\
\frac{\vec{a}_2^T\vec{a}_2-2\vec{a}_2^T\vec{x}}{v\tau_2} &= v\tau_2+2\sqrt{\vec{x}^T\vec{x}}\\
&\vdots\\
\frac{\vec{a}_{n-1}^T\vec{a}_{n-1}-2\vec{a}_{n-1}^T\vec{x}}{v\tau_{n-1}} &= v\tau_{n-1}+2\sqrt{\vec{x}^T\vec{x}}.
\end{aligned}$$

Now, we can subtract the first equation from each of the others individually, to obtain

$$\begin{aligned}
\frac{\vec{a}_2^T\vec{a}_2-2\vec{a}_2^T\vec{x}}{v\tau_2}-\frac{\vec{a}_1^T\vec{a}_1-2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_2-v\tau_1\\
\frac{\vec{a}_3^T\vec{a}_3-2\vec{a}_3^T\vec{x}}{v\tau_3}-\frac{\vec{a}_1^T\vec{a}_1-2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_3-v\tau_1\\
&\vdots\\
\frac{\vec{a}_{n-1}^T\vec{a}_{n-1}-2\vec{a}_{n-1}^T\vec{x}}{v\tau_{n-1}}-\frac{\vec{a}_1^T\vec{a}_1-2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_{n-1}-v\tau_1.
\end{aligned}$$

Whew, that was a bit of messy algebra! But have we ended up with a linear system of equations? Well, notice that the only "quadratic" terms remaining are of the form $\vec{a}_i^T\vec{a}_i$. But since all the $\vec{a}_i$ are known to us, these are just constant terms. The only terms involving the unknown $\vec{x}$ are linear terms, so we have indeed obtained a linear system of equations!

Let's rewrite our equations in a more familiar form, collecting the constant terms on the right hand side, and

stacking everything into a matrix-vector system:

$$\begin{bmatrix} 2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_2^T/(v\tau_2) \\ 2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_3^T/(v\tau_3) \\ \vdots \\ 2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_{n-1}^T/(v\tau_{n-1}) \end{bmatrix} \vec{x} = \begin{bmatrix} v\tau_2 - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_2^T\vec{a}_2}{v\tau_2} \\ v\tau_3 - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_3^T\vec{a}_3}{v\tau_3} \\ \vdots \\ v\tau_{n-1} - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_{n-1}^T\vec{a}_{n-1}}{v\tau_{n-1}} \end{bmatrix}.$$

Now, we have made the linear structure of our system of equations entirely explicit. Observe that we end up with $n-2$ equations and a single unknown vector $\vec{x}$. Let x be in d-dimensional space. Thus, we need at least $n \geq d+2$ beacons to determine our location when our clocks are not synchronized with those of the beacons - one more than we did in the case of synchronized clocks.

Intuitively, this can be justified by recognizing that we really have one more unknown - the offset Δt - so we clearly need at least $d+1$ beacons. The fourth and last equation is needed to "linearize" our system by subtracting out the quadratic terms. Of course, having more equations doesn't hurt. If we have more than four equations, then we end up with an overdetermined system, which can still be solved using least squares.