EECS 16A Designing Information Devices and Systems I
Spring 2018 Lecture Notes Note 1

1.1 Introduction to Linear Algebra — the EECS Way

In this note, we will teach the basics of linear algebra and relate it to the work we will see in labs and EECS in general. We will introduce the concept of vectors and matrices, show how they relate to systems of linear equations, and discuss how these systems of equations can be solved using a technique known as Gaussian elimination.

1.1.1 What Is Linear Algebra and Why Is It Important?

- Linear algebra is the study of vectors and their transformations.
- 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.

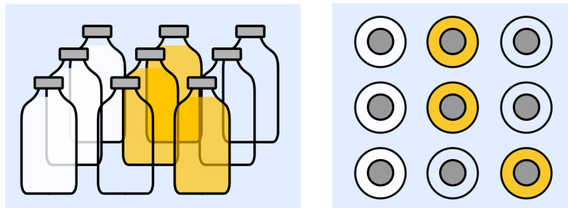
As you will see in the homeworks and labs, these concepts can be used to do many interesting things in real-world-relevant application scenarios. In the previous note, we introduced the idea that all information devices and systems (1) take some piece of information from the real world, (2) convert it 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.

1.1.2 Tomography: Linear Algebra in Imaging

Let's start with an example of linear algebra that relates to this module and uses key concepts from this note: tomography. Tomography allows us to "see inside" of a solid object, such as the human body or even the earth, by taking images section by section 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.

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:



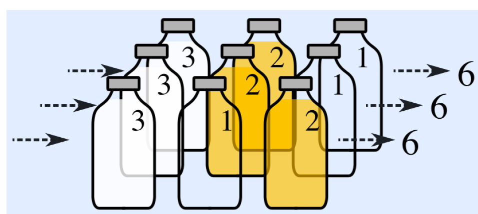
Side view Top view

If we choose symbols such that M=Milk, J=Juice, and O=Empty, we can represent the stack of bottles shown above as follows:

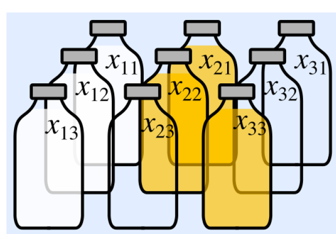
$$\begin{matrix} M & J & O \\ M & J & O \\ M & O & J \end{matrix} \tag{1}$$

Our grocer cannot see directly into the box, but he needs to sort them somehow (without opening every single one). However, suppose he has a device that can tell him how much light an object absorbs. This lets him shine a light at different angles to figure out how much total light the bottles in those columns absorb.

Suppose milk absorbs 3 units of light, juice absorbs 2 units of light and an empty bottle absorbs 1 unit of light. If we shine light in a straight line, we can determine the amount of light absorbed as the sum of the light absorbed by each bottle. In our specific example, shining a light from left to right would look like this, where each row absorbs 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:



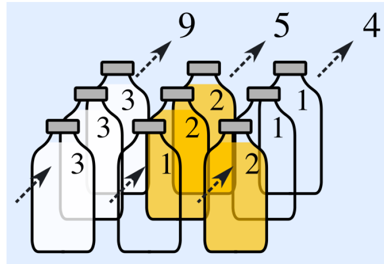
This means that x_{11} would be the amount of light the top left bottle absorbs, x_{21} would be the amount of light the middle left bottle absorbs, and so forth. Shining the light from left to right for our specific example gives the following equations:

$$x_{11} + x_{21} + x_{31} = 6 \tag{2}$$

$$x_{12} + x_{22} + x_{32} = 6 \tag{3}$$

$$x_{13} + x_{23} + x_{33} = 6 \tag{4}$$

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



Which would give the following equations:

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

Now, let's consider how we would proceed if we have the measurement results but cannot open the box. If, for instance, the rows absorb 5, 6, and 4 units of light, we can write down the following equations:

$$x_{11} + x_{21} + x_{31} = 5 \tag{8}$$
$$x_{12} + x_{22} + x_{32} = 6 \tag{9}$$
$$x_{13} + x_{23} + x_{33} = 4 \tag{10}$$

Likewise, if the three columns absorb 6, 3, and 6 units respectively, we have the additional equations:

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

One such configuration that satisfies those equations is:

$$\begin{matrix} 3 & 1 & 1 \\ 2 & 1 & 3 \\ 1 & 1 & 2 \end{matrix} \tag{14}$$

If this is true, we can deduce that the original stack looked like:

$$\begin{matrix} M & O & O \\ J & O & M \\ O & O & J \end{matrix} \tag{15}$$

However, the following configuration also satisfies the above equations:

$$\begin{matrix} 1 & 1 & 3 \\ 3 & 1 & 2 \\ 2 & 1 & 1 \end{matrix} \tag{16}$$

In order to distinguish between these two possible configurations, we need more measurements. To get these measurements, we could shine light at different angles, perhaps diagonally. Are there other possible configurations? How many different directions do we need to shine light through before we are certain of the configuration? By the end of this module, you will have the tools to answer these questions.

In lecture itself, we will work out specific examples to make this more clear. The important thing is that it is not just about the number of measurements we take (or, equivalently, the number of experiments you do), but rather *which* measurements we take. These measurements must be chosen in a way that allows you to extract the information that you are interested in. This doesn't always happen. Sometimes, the measurements that you might think of taking end up being redundant in subtle ways.

1.2 Representing Systems of Linear Equations

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 equations**, which summarizes the known relationships between the variables we want to solve for (x_{11}, x_{12}, x_{13}, etc.) and our measurements. More specifically, the measurements in our tomography example can be characterized as a system of **linear** equations, because each variable relates to the measurement result by a fixed scaling factor.[1] Because writing out all of these equations can be cumbersome, we will typically try to express systems of equations in terms of **vectors** and **matrices** instead. Here, we will briefly introduce the concepts of vectors and matrices in order to illustrate how they can be used to represent systems of equations. In the next few lectures, we will give you a bit more intuition and examples to demonstrate their role in science and engineering applications.

1.2.1 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:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{17}$$

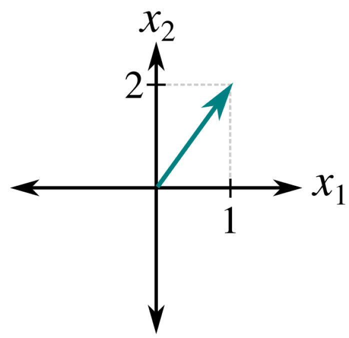
We call $\vec{x}$ a **vector**. Because $\vec{x}$ contains n real numbers, we can use the $\in$ ("in" — i.e., is a member of) symbol to write $\vec{x} \in \mathbb{R}^n$. If the elements of $\vec{x}$ were complex numbers, we would write $\vec{x} \in \mathbb{C}^n$. Each x_i (for i between 1 and n) is called a **component**, or **element**, of the vector. The **size** of a vector is the number of components it contains (n in the example vector, 2 in the example below).

Example 1.1 (Vector of size 2):

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

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:

[1] In the tomography example specifically, the scaling factor was either 1 or 0, but you could imagine different cases where this scaling factor is any constant number. If you imagine plotting the measurement result as a function of each variable, you would get a line — hence the term "linear."



1.2.2 Matrices

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

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} \tag{18}$$

Each A_{ij} (where i is the row index and j is the column index) is a **component**, or **element** of the matrix A. In our simple example of tomography with the grocer, we created a 3×3 matrix to represent the amount of light that each bottle absorbed.

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).

1.2.3 System of Linear Equations

We can represent a system of linear equations in matrix or vector form. Remember that in a linear equation, the variables are only scaled by a constant value and added together. Suppose we have a system of linear equations below

$$\begin{aligned} 3.1x_1 \quad + \quad 2x_2 \quad &= \quad 1 \\ 2x_1 \quad + \quad -4.7x_2 \quad &= \quad 3 \end{aligned} \tag{19}$$

Both equations use the same variables (x_1 and x_2), but have different coefficients to scale these variables (3 and 2 for the first equation, 2 and 4 in the second equation), and sum to different amounts (1 for the first equation, 3 in the second equation). If we create one 2×2 matrix for the scaling factors, one vector for the variables, and another vector for the results of the equation, it turns out that we can represent the system of equations above as

$$\begin{bmatrix} 3.1 & 2 \\ 2 & -4.7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

In later notes, we will give more specifics about the details of matrix and vector multiplication, but in general, we can represent any system of m linear equations with n variables in the form

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

where A is an $m \times n$ matrix, $\vec{x}$ is a vector containing the n variables, and $\vec{b}$ is a vector of size m containing the constants. Why do we care about writing a system of linear equations in the above representation? We will see later in the course that decoupling the coefficients and the variables makes it easier to perform analysis on the system, especially if the system is large.

Example 1.3 (System of linear equations in matrix form): Recall the toy tomography example where we shine light in a straight line. If the rows absorb 5, 6, and 4 units of light, we have the following system of equations — one equation for each measurement:

$$x_{11} + x_{12} + x_{13} = 5 \tag{20}$$
$$x_{21} + x_{22} + x_{23} = 6 \tag{21}$$
$$x_{31} + x_{32} + x_{33} = 4. \tag{22}$$

In this example, there are nine variables (x_{11} through x_{33}) corresponding to the nine bottles in the box, and in each equation the variable has a coefficient of either 1, if it is included in the measurement, or 0, if it is not included. The full system of equations could be written as:

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

But it is much easier to capture this information in matrix-vector form:

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

1.3 Gaussian Elimination

Our goal isn't just to write out systems of equations, but to solve them — namely, to find what value(s) each variable must take in order for all equations to be valid. Gaussian elimination is an **algorithm** (a sequence of programmatic steps) for doing this that can be used to solve any arbitrarily large system of linear equations, or decide that no solution exists. Why does this matter? When we model a measurement, such as total light absorption in a row or column for our tomography example, as the result of some (possibly weighted) combination of variables we would like to solve for, each measurement results in a linear equation that provides more information about these variables. Gaussian elimination is a means of using the system of equations built from these measurements to find the information we care about — the values of the unknown variables.

1.3.1 Solving Systems of Linear Equations

As a specific example, 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. Our approach in Gaussian elimination will be to successively rewrite the system using operations that do not change its solution, such as adding or subtracting multiples of one equation from another.[2] Let's start by seeing how this can be done for the 2 equation case.

Example 1.4 (System of 2 equations): To solve the system of equations above, we would like to find explicit equations 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:

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

[2] Why is this a valid operation? Multiplying two sides of an equation by the same scaling factor or adding/subtracting the same quantity from both sides of an equation will not change the equality. Therefore, these operations will not affect the solution to the system of equations.

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

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

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:

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

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:

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

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

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

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

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

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

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

System of 3 Equations

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

[3] 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.

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:

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

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

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

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. Are there any operations we might need to perform in addition to scaling and adding/subtracting equations?

Example 1.6 (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:

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

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

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

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. $3'$):

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

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:

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

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

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

1.3.2 Gaussian Elimination Algorithm

How about solving systems of more equations? To do so, we need to look at the operations that we have performed in the examples to develop an *algorithm* for solving a system of any number of linear equations. Remember that we can perform an operation to an equation as long as it doesn't change the solution.

1.3.2.1 Algorithm Operations

First, we would like to abstract out the variables by representing the equation as an *augmented matrix*. For instance, the following system of two equations can be represented as a 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 will perform to the rows of a matrix:

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. Switching 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 i^{th} 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. However, will applying these steps *always* result in 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.

1.3.2.2 Gaussian Elimination Examples

Example 1.7 (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. 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.

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

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 1.8 (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.

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

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

Example 1.9 (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 measured results, there would have to be a problem with our modeling assumptions or measurement setup (and this is usually the case when we have noise in our measurements, which we will discuss further in later notes).

Example 1.10 (Canceling intermediate variables):

$$\begin{bmatrix} x & + & y & + & 3z & = & 2 \\ 2x & + & 2y & + & 7z & = & 6 \\ -x & - & y & - & 2z & = & 0 \end{bmatrix} \qquad \left[\begin{array}{ccc|c} 1 & 1 & 3 & 2 \\ 2 & 2 & 7 & 6 \\ -1 & -1 & 2 & 0 \end{array}\right]$$

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

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

Canceling x from rows 2 and 3 has also canceled y, so we eliminate the next variable, z. To do this, we can subtract row 2 from row 3, but this gives a zero row because the rows are identical:

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

Because we now have fewer non-zero rows than variables, this system of equations has an infinite number of solutions, but we can still subtract 3 times row 2 from row 1 to eliminate z from the first equation.

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

We cannot proceed further from here. While we can solve explicitly for z, there are an infinite number of possible values for x and y: for any choice of x, setting y to be $-(4+x)$ will provide a valid solution.

1.3.2.3 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. First, we eliminate the first unknown from all but one equation, then among the remaining equations eliminate the next possible unknown from all but one equation. Repeat this until you reach one of three situations:

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, corresponding to the equation $0 = 0$. From our example with 3 equations and 3 unknowns, this could be an augmented matrix such as:

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

   This means that the system of equations has a **unique solution**. 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 $0 = 0$. From our previous 3-equation, 3-unknown examples, this would be an augmented matrix such as:

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

   If this is the case, 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 $a \neq 0$. From our the 3-equation, 3-unkown example, this could be an augmented matrix such as:

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

   This means that the system of linear equations is inconsistent and there are **no solutions**.

The form of the augmented matrix obtained after running Gaussian elimination is known as its **reduced form**. In discussion section, you will see this explained further. We will also release supplemental material that illustrates how Gaussian elimination can proceed.

1.3.2.4 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 $A \in \mathbb{R}^{m\times(n+1)}$, 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.

1.3.3 Tomography Revisited

How does what we have learned so far relate back to our tomography example? 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.