Math 1600A Lecture 7, Section 002

Announcements:

More texts, solutions manuals and packages **have arrived**!

Read Sections 2.0, 2.1 and 2.2 for next class. Work through recommended homework questions. Scans of the text up to Section 2.1 are available from the course home page, but will be removed soon.

Quiz 2 is this week, and will cover the material until the end of Section 1.4.

Office hour: today, 1:30-2:30, MC103B. Also, if you can't make it to my office hours, feel free to attend Hugo Bacard's office hours, listed on the course home page.

Help Centers Monday-Friday 2:30-6:30 in MC 106.

Partial review of previous lectures:

Recall that $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$ with addition and multiplication taken modulo m. That means that the answer is the remainder after division by m.

For example, in $\mathbb{Z}_{10}$, $8 \cdot 8 = 64 = 4 \pmod{10}$.

$\mathbb{Z}_m^n$ is the set of vectors with n components, each of which is in $\mathbb{Z}_m$.

New material

Section 1.4: Applications: Code Vectors (we aren't covering force vectors)

We're going to study a way to encode data that allows us to detect transmission errors. Used on CDs, UPC codes, ISBN numbers, credit card numbers, etc.

Example 1.37: Suppose we want to send the four commands "forward", "back", "left" and "right" as a sequence of 0s and 1s. We could use the following code:

$$\text{forward} = [0,0], \quad \text{back} = [0,1], \quad \text{left} = [1,0], \quad \text{right} = [1,1].$$

But if there is an error in our transmission, the Mars rover will get the wrong message and will drive off of a cliff, wasting billions of dollars of taxpayer money (but making for some good NASA jokes).

Here's a more clever code:

$$\text{forward} = [0,0,0], \quad \text{back} = [0,1,1], \quad \text{left} = [1,0,1], \quad \text{right} = [1,1,0].$$

If any single *bit* (binary digit, a 0 or a 1) is flipped during transmission, the Mars rover will notice the error, since all of the **code vectors** have an **even** number of 1s. It could then ask for retransmission of the command.

This is called an **error-detecting code**. Note that it is formed by adding a bit to the end of each of the original code vectors so that the total number of 1s is even.

In vector notation, we replace a vector $\vec{b} = [v_1, v_2, \ldots, v_n]$ with the vector $\vec{v} = [v_1, v_2, \ldots, v_n, d]$ such that $\vec{1} \cdot \vec{v} = 0 \pmod{2}$, where $\vec{1} = [1, 1, \ldots, 1]$.

Exactly the same idea works for vectors in $\mathbb{Z}_3^n$; see Example 1.39 in the text.

Note: One problem with the above scheme is that **transposition** errors are not detected.

Example 1.40 (UPC Codes): The Univeral Product Code (bar code) on a product is a vector in $\mathbb{Z}_{10}^{12}$, such as

$$\vec{u} = [6,7,1,8,6,0,0,1,3,6,2,4].$$

Instead of using $\vec{1}$ as the **check vector**, UPC uses

$$\vec{c} = [3,1,3,1,3,1,3,1,3,1,3,1].$$

The last digit is chosen so that $\vec{c} \cdot \vec{u} = 0 \pmod{10}$.

For example, if we didn't know the last digit of $\vec{u}$, we could compute

$$\vec{c} \cdot [6,7,1,8,6,0,0,1,3,6,2,d] = \cdots = 6 + d \pmod{10}$$

and so we would find that we need to take $d = 4$, since $6 + 4 = 0 \pmod{10}$.

This detects any single error. The pattern in $\vec{c}$ was chosen so that it detects many transpositions, but it doesn't detect when digits whose difference is 5 are transposed. The problem is that $2 \cdot 5 = 0 \pmod{10}$.

Example 1.41 (ISBN Codes): ISBN codes use vectors in $\mathbb{Z}_{11}^{10}$. The check vector is $\vec{c} = [10,9,8,7,6,5,4,3,2,1]$. Because 11 is a prime number, this code detects all single errors and *all* single transposition errors.

Summary: To create a code, you choose m (which determines the allowed digits), n (the number of digits in a code word), and a check vector $\vec{c} \in \mathbb{Z}_m^n$. Then the valid words $\vec{v}$ are those with $\vec{c} \cdot \vec{v} = 0$. If $\vec{c}$ ends in a 1, then you can always choose the last digit of $\vec{v}$ to make it valid.

Note: This kind of code can only reliably detect one error, but more sophisticated codes can detect multiple errors. There are even **error-correcting codes**, which can *correct* multiple errors in a transmission without needing it to be resent. In fact, you can drill small holes in a CD, and it will still play the entire content perfectly. We'll learn about these codes later.

Section 2.1: Systems of Linear Equations

Definition: A **linear equation** in the variables $x_1, x_2, \ldots, x_n$ is an equation that can be written in the form

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

where the **coefficients** $a_1, \ldots, a_n$ and the **constant term** b are constants.

Linear equations:

$$2x - 5y = 10, \quad r + \frac{1}{2}\, s = 0.5t - 2, \quad x_1 - \sqrt{2} x_2 - (\sin \frac{\pi}{5}) x_3 = 0.$$

Non-linear equations:

$$xy + z = 1, \quad x_1^2 + x_2^2 = 2, \quad \sin(x) = 0, \quad 2^y + z = 16.$$

A **solution** to $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = b$ is a vector $[s_1, \ldots, s_n]$ such that the equation is true when we substitute $x_1 = s_1, \ldots, x_n = s_n$. For example, $[10, 2]$ is a solution to $2x - 5y = 10$.

When a linear equation has two unknowns, its solutions form a line in $\mathbb{R}^2$. To describe the solutions in parametric form, we can solve for one of the variables in terms of the other.

For example, for $2x - 5y = 10$, we can write $y = \frac{2}{5}x - 2$. If we set x to a parameter t, we get parametric solutions $[t, \frac{2}{5}t - 2]$.

The same works when there are n variables: we can solve for one in terms of all of the others, and get a solution with $n - 1$ parameters.

Systems of linear equations

Definition: A **system of linear equations** is a finite set of linear equations, each with the same variables. A **solution** to the system is a vector that satisfies *all* of the equations.

Example:

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

Is $[1, 1]$ a solution? How about $[-1, 3]$? How can we find all solutions? What's happening geometrically?

Example:

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

Is $[1, 1]$ a solution? How about $[-1, 3]$? How can we find all solutions? What's happening geometrically?

Example:

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

Is $[1, 1]$ a solution? How about $[-1, 3]$? How can we find all solutions? What's happening geometrically?

A system is **consistent** if it has one or more solutions, and **inconsistent** if it has no solutions. We'll see later that a consistent system always has either one solution or infinitely many.

Solving a system

We started with the system on the left and produced the system on the right:

$$\begin{aligned} x + y &= 2, \\ -x + y &= 4, \end{aligned} \qquad \begin{aligned} x + y &= 2 \\ 2y &= 6 \end{aligned}$$

The system on the right was **easy** to solve. These two systems are said to be **equivalent** because they have exactly the same solutions. (The geometry is different, though!)

Example: Similarly, a large system such as

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

is easy to solve, because of its **triangular** structure. The method is called **back substitution**:

$$\begin{aligned} &z = 2 \\ &y = 5 - 3z = 5 - 6 = -1 \\ &x = 2 + y + z = 2 - 1 + 2 = 3. \end{aligned}$$

So the unique solution is $[3, -1, 2]$.

Let's see how a general system can be converted into a system with a triangular form.

Example: We'll solve the system on the left

$$\begin{aligned} x - y - z &= 2 \\ 3x - 3y + 2z &= 16 \\ 2x - y + z &= 9 \end{aligned} \qquad \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 3 & -3 & 2 & 16 \\ 2 & -1 & 1 & 9 \end{array}\right]$$

but to save time, we can write it as the **augmented matrix** on the right.

Today, we'll show the equations as well.

To put it into triangular form, the first step is to eliminate the xs in equations 2 and 3.

Replace row 2 with row 2 - 3(row 1):

$$\begin{aligned} x - \ \ y - \ \ z &= 2 \\ 5z &= 10 \\ 2x - \ \ y + \ \ z &= 9 \end{aligned} \qquad \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 2 & -1 & 1 & 9 \end{array}\right]$$

Replace row 3 with row 3 - 2(row 1):

$$\begin{aligned} x - \ \ y - \ \ z &= 2 \\ 5z &= 10 \\ y + 3z &= 5 \end{aligned} \qquad \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 0 & 1 & 3 & 5 \end{array}\right]$$

Now we can exchange rows 2 and 3, to end up in triangular form:

$$\begin{aligned} x - \ \ y - \ \ z &= 2 \\ y + 3z &= 5 \\ 5z &= 10 \end{aligned} \qquad \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 1 & 3 & 5 \\ 0 & 0 & 5 & 10 \end{array}\right]$$

Hey! This is the system we solved earlier, so now we know that the solution is $[3, -1, 2]$.

This system and the original system have *exactly* the same solutions. Explain.
We say they have the same **solution set** and therefore that they are **equivalent** systems.