Math 1600 Lecture 5, Section 2, 15 Sep 2014

Announcements:

Continue reading Section 1.3 and also the Exploration on cross products for next class. Work through recommended homework questions.

Quiz 1 this week in tutorials. Quiz 1 will cover Sections 1.1, 1.2 and the code vectors part of 1.4. It does not cover the Exploration after Section 1.2. See last lecture for how they run.

Office hours: Monday, 3:00-3:30 and Wednesday, 11:30-noon, MC103B.

Help Centers: Monday-Friday 2:30-6:30 in MC 106 starting Wednesday, September 17. Extra Help Centers: Today and tomorrow (Sept 15 and 16), 4-5pm, MC106.

Review: Section 1.4: Applications: Code Vectors

Example 1.37: A code for rover commands:

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

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**.

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: if we want to send $[0, 1, 1]$ but the first two bits are exchanged, the rover receives $[1, 0, 1]$, which is also a valid command. We'll see codes that can detect transpositions.

New material

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. For example, $3 \cdot 5 + 1 \cdot 0 = 15$ and $3 \cdot 0 + 1 \cdot 5 = 5$, and these are the same modulo 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. **See text** for a worked example.

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.

Question: The Dan code uses vectors in $\mathbb{Z}_4^3$ with check vector $\vec{c} = [3, 2, 1]$. Find the check digit d in the code word $\vec{v} = [2, 2, d]$.

Solution: We compute

$$\begin{aligned} \vec{c} \cdot \vec{v} = [3, 2, 1] \cdot [2, 2, d] &= 3 \cdot 2 + 2 \cdot 2 + 1 \cdot d \\ &= 10 + d = 2 + d \pmod 4 \end{aligned}$$

To make $\vec{c} \cdot \vec{v} = 0 \pmod 4$, we choose $d = 2$.

This is the end of the material for quiz 1. (We aren't covering force vectors.)

Section 1.3: Lines and planes in $\mathbb{R}^2$ and $\mathbb{R}^3$

[These notes are a summary of the material, which will be supplemented by some diagrams on the board.]

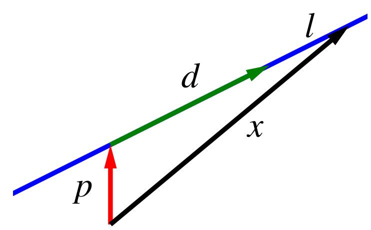
We study lines and planes because they come up directly in applications, but also because the solutions to many other types of problems can be expressed using the language of lines and planes.

Lines in $\mathbb{R}^2$ and $\mathbb{R}^3$

Given a line ℓ, we want to find equations that tell us whether a point (x, y) or (x, y, z) is on the line. We'll write $\vec{x} = [x, y]$ or $\vec{x} = [x, y, z]$ for the position vector of the point, so we can use vector notation.

The **vector form** of the equation for ℓ is:

$$\vec{x} = \vec{p} + t\vec{d}$$



where $\vec{p}$ is the position vector of a point on the line, $\vec{d}$ is a vector parallel to the line, and $t \in \mathbb{R}$. This is concise and works in $\mathbb{R}^2$ and $\mathbb{R}^3$.

If we expand the vector form into components, we get the **parametric form** of the equations for ℓ:

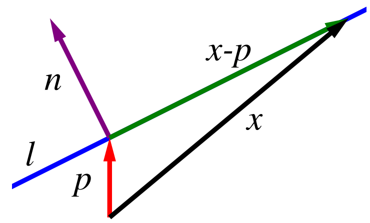
$$\begin{aligned} x &= p_1 + td_1 \\ y &= p_2 + td_2 \\ (z &= p_3 + td_3 \quad \text{if we are in } \mathbb{R}^3) \end{aligned}$$

Lines in $\mathbb{R}^2$

There are additional ways to describe a line in $\mathbb{R}^2$.

The **normal form** of the equation for ℓ is:

$$\vec{n} \cdot (\vec{x} - \vec{p}) = 0 \quad \text{or} \quad \vec{n} \cdot \vec{x} = \vec{n} \cdot \vec{p},$$



where $\vec{n}$ is a vector that is *normal* = *perpendicular* to ℓ.

If we write this out in components, with $\vec{n} = [a, b]$, we get the **general form** of the equation for ℓ:

$$ax + by = c,$$

where $c = \vec{n} \cdot \vec{p}$. When $b \neq 0$, this can be rewritten as $y = mx + k$, where $m = -a/b$ and $k = c/b$.

Note: All of these simplify when the line goes through the origin, as then you can take $\vec{p} = \vec{0}$.

Example: Find all four forms of the equations for the line in $\mathbb{R}^2$ going through $A = [1, 1]$ and $B = [3, 2]$.

Note: None of these equations is *unique*, as $\vec{p}$, $\vec{d}$ and $\vec{n}$ can all change. The general form is closest to being unique: it is unique up to an overall scale factor.

Lines in $\mathbb{R}^3$

Most of the time, one uses the vector and parametric forms above. But there is also a version of the normal and general forms. To specify the direction of a line in $\mathbb{R}^3$, it is necessary to specify **two** non-parallel normal vectors $\vec{n}_1$ and $\vec{n}_2$. Then the **normal form** is

$$\begin{aligned} \vec{n}_1 \cdot \vec{x} &= \vec{n}_1 \cdot \vec{p} \qquad && \text{typo in book in Table 1.3:} \\ \vec{n}_2 \cdot \vec{x} &= \vec{n}_2 \cdot \vec{p} \qquad && \text{there should be no subscripts on } \vec{p} \end{aligned}$$

When expanded into components, this gives the **general form**:

$$\begin{aligned} a_1 x + b_1 y + c_1 z &= d_1, \\ a_2 x + b_2 y + c_2 z &= d_2. \end{aligned}$$

Since *both* equations must be satisfied, this can be interpreted as the intersection of two planes. (We'll discuss planes in a second.)

Question: What are the pros and cons of the different ways of describing a line?

Planes in $\mathbb{R}^3$

Normal form:

$$\vec{n} \cdot (\vec{x} - \vec{p}) = 0 \quad \text{or} \quad \vec{n} \cdot \vec{x} = \vec{n} \cdot \vec{p}.$$

This is *exactly* like the normal form for the equation for a line in $\mathbb{R}^2$.
When expanded into components, it gives the **general form**:

$$ax + by + cz = d,$$

where $\vec{n} = [a, b, c]$ and $d = \vec{n} \cdot \vec{p}$.