Linear Data
Lab 10

Content Creator: Emily J. King

Goals: Properly encode systems of linear equations into a computer and solve it using least squares. Calculate fundamental spaces of matrices, recognizing when returned value is understood as an affine space and when it is an actual vector. Recognize the fundamental geometric linear transformations and compute their determinants. Eigenvalue teaser.

Supplement to MATLAB/Python in-lab portion

Section 1: Traffic problem

(Original problem and image from Steven J. Leon, *Linear algebra with applications*, 7th edition, 2006, pp. 19–20) In the downtown section of a certain city, two sets of one-way streets intersect as shown in Figure 1. The average hourly volume of traffic entering and leaving this section of streets during rush hour is given in Figure 1. We want to determine the amount of traffic between each pair of the four intersections. The units of the variables x_1, x_2, x_3, and x_4 are average hourly volume of traffic.
How do we set up a system of equations to solve this problem? We note that the number of automobiles entering any intersection must be equal to the number leaving. For example, if we consider intersection A, then the number of entering is $x_1 + 450$ and the number leaving is $x_2 + 610$. Thus,

$$x_1 + 450 = x_2 + 610 \quad \text{(intersection } A\text{)}.$$

Similarly,

$$\begin{aligned} x_2 + 520 &= x_3 + 480 \quad \text{(intersection } B\text{)} \\ x_3 + 390 &= x_4 + 600 \quad \text{(intersection } C\text{)} \\ x_4 + 640 &= x_1 + 310 \quad \text{(intersection } D\text{)}. \end{aligned}$$

We can write this system as

$$\begin{cases} x_1 + 450 &= x_2 + 610 \\ x_2 + 520 &= x_3 + 480 \\ x_3 + 390 &= x_4 + 600 \\ x_4 + 640 &= x_1 + 310 \end{cases}$$

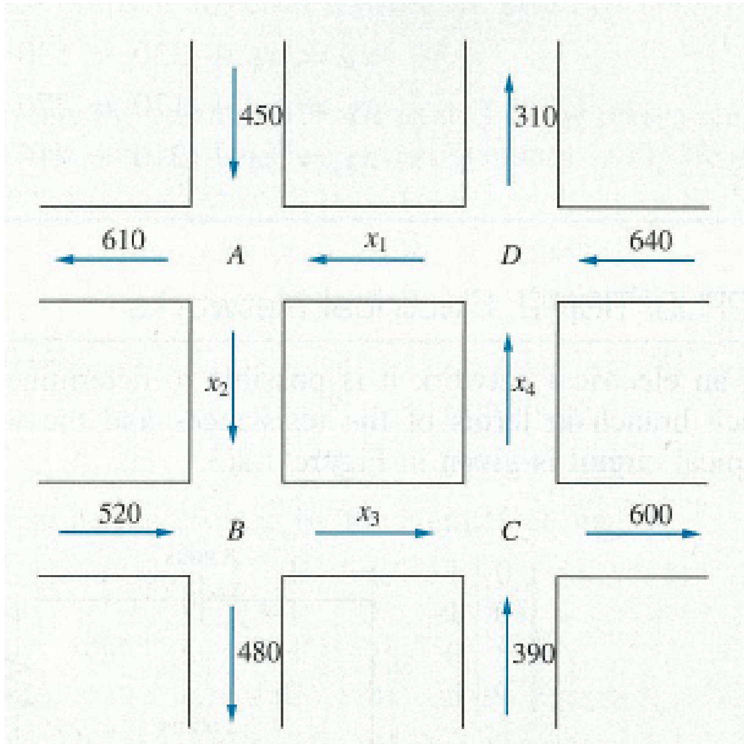

Figure 1: Diagram of the streets (Leon 2006).

which can be rearranged to

$$\begin{cases} 1x_1 & -1x_2 & +0x_3 & +0x_4 & = 160 \\ 0x_1 & +1x_2 & -1x_3 & +0x_4 & = -40 \\ 0x_1 & +0x_2 & +1x_3 & -1x_4 & = 210 \\ -1x_1 & +0x_2 & +0x_3 & +1x_4 & = -330 \end{cases} .$$

The mathematical technique to solve this problem with the restriction that all of the $x_j \geq 0$ is called *linear programming*, which involves linear algebra but is beyond the scope of this class. Instead, we will play around with allowing Matlab/Python to give us various solutions.

Section 3: Eigenvalue teaser

Matrices:

$$\mathbf{D1} = \begin{pmatrix} 0.5 & 0 \\ 0 & 3 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \cos(30) & -\sin(30) \\ \sin(30) & \cos(30) \end{pmatrix}$$