Math 2500: Linear Algebra

Math 2500: Linear Algebra

Contents

Preface

This is a set of class notes designed to guide an inquiry-based course on linear algebra. This is not a complete resource! Rather, this text is meant to accompany the book Introduction to Linear Algebra by Gilbert Strang. Industrious students might also use it for a self-study course.

An important feature of this text is the integration of the Sage Mathematical Software System. Students in this course will learn to use Sage to perform long tedious computations (which linear algebra has in spades) and create visualizations. I thank the Sage community, in particular William Stein, for creating such a wonderful tool and making it open-source.

I also thank Rob Beezer for creating the MathBook project which makes this particular book project available on the web.

Chapter 1

Vectors, the Dot Product, and Matrices

This first chapter introduces the basic objects of linear algebra. You will meet vectors, the dot product, and matrices.

Vectors are a generalization of the concept of number. Where real numbers can help us model the geometry of points on a line, vectors will allow us to model the geometry of a plane, or (three-dimensional) space, or even "spaces" with higher dimensions. We begin by learning about the algebra of vectors, and making connections to the geometry.

The dot product is a funny kind of multiplication. It plays an important role in mathematics because it captures all of the basics of measurement. We shall learn how to use the dot product to measure lengths and angles. By its definition, the dot product is connected with the concept of a linear equation, so it will make frequent appearances in our work.

Matrices are another way to generalize the concept of number. (In fact, they generalize the concept of vector.) We start here by learning about the algebra of matrices. The whole rest of this course will focus on matrices, their uses, and their properties.

A running theme for this course is the use of the Sage mathematical software system. In order to get started, the fourth section of this course is dedicated to getting started with Sage using the SageMathCloud (SMC). You will make an account and run through an introductory workshop with SMC. Also, throughout this workbook you will find little embedded pieces of Sage code. These are implemented using the Sage SingleCell server. Most of these Sage cells are mutable. You can change the content in them and re-evaluate your new Sage code. I encourage you to play with these—it is a good way to learn the basics of Sage.

The fifth and final section of the chapter is a short assignment designed to consolidate learning. You will get a chance to practice your skills and to think more deeply about the concepts you have learned.

1.1 Vectors

1.1.1 The Assignment

- Read section 1.1 of *Strang* (pages 1-7).
- Read the following and complete the exercises below.

1.1.2 Learning Objectives

Before class, a student should be able to:

- Add vectors.
- Multiply a vector by a scalar.
- Compute linear combinations.
- Draw pictures which correspond to the above operations.

At some point, a student should be able to:

- Solve linear combination equations involving unknown coefficients.
- Solve linear combination equations involving unknown vectors.

1.1.3 Some Discussion

Algebraically, a **vector** is a stack of numbers in a set of parentheses or brackets, like this

$$\begin{pmatrix} 2 \\ 7 \\ 9 \end{pmatrix}, \text{ or } \begin{bmatrix} 2 \\ 7 \\ 9 \end{bmatrix}, \text{ or } \begin{pmatrix} 2 & 7 & 9 \end{pmatrix}.$$

The individual numbers are called the **components** or **entries** or **coordinates** of the vector. For example, 7 is the second component of the vectors above.

The first two vectors above are called **column** vectors because they are stacked vertically. The third is called a **row** vector because it is arranged horizontally. For this class, we will always use column vectors, but to save space, we might sometimes write them as row vectors. It is up to you to make the switch. (We will see later how this matters!)

Vectors can take lots of different sizes. The vectors above are all 3-vectors. Here is a 2-vector:

$$\begin{pmatrix} 71 \\ -12 \end{pmatrix}.$$

Here is a 4-vector:

$$\begin{pmatrix} \pi \\ 0 \\ -\pi \\ 1 \end{pmatrix}.$$

The main value in using vectors lies in their standard interpretations. Let's focus on 3-vectors for now. The vector $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ can represent

- A point in space described in the standard three-dimensional rectangular coordinate system with x coordinate equal to a, y-coordinate equal to b and z coordinate equal to c.
- An arrow in space which points from the origin $(0, 0, 0)$ to the point (a, b, c).
- An arrow in space which points from some point (x, y, z) to the point $(x + a, y + b, z + c)$.

Operations on Vectors

There are two operations on vectors which are of utmost importance for linear algebra. (In fact, if your problem has these operations in it, there is a chance you are doing linear algebra already.)

Scalar Multiplication Given a number $\lambda \in \mathbb{R}$ and a vector $v = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$, we form the new vector

$$\lambda v = \begin{pmatrix} \lambda a \\ \lambda b \\ \lambda c \end{pmatrix}.$$

Addition Given a vector $v = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ and a vector $w = \begin{pmatrix} d \\ e \\ f \end{pmatrix}$ of the same size, we form their **sum**

$$v + w = \begin{pmatrix} a+d \\ b+e \\ c+f \end{pmatrix}$$

These operations have "obvious" generalizations to vectors of different sizes. Because things go entry-by-entry, these are often called **coordinate-wise** operations.

Combining these two operations gives us the notion of a *linear combination*. If λ and μ are numbers and v and w are vectors of a common size, then the vector

$$\lambda v + \mu w$$

is a linear combination of v and w.

1.1.4 Sage Instructions

Basic Constructions A vector in Sage is constructed by applying the `vector` command to a list. Lists are entered in square brackets with entries separated by commas, so the typical way to create a vector looks like this:

```
u = vector([1,1,2])
```

Notice that nothing was displayed. Sage just put the vector u into memory. We can ask for it by calling it.

```
u
```

```
(1, 1, 2)
```

Sage defaults to displaying vectors horizontally, which is different from how we normally write them by hand. This is okay. You will get used to it quickly.

Sage knows how to add, multiply by scalars, and form linear combinations, and the notation for it is just as easy as you would expect.

```
v = vector([-1,-1,2])
u + v
```

```
(0, 0, 4)
```

```
pi * u
```

```
(pi, pi, 2*pi)
```

```
3*u + 4*v
```

```
(-1, -1, 14)
```

If you ask Sage to plot a vector, you get this kind of picture:

```
plot(v)
```

And in two dimensions something similar...

```
a = vector([-1,1])
plot(a)
```

If you find that you want a vector to have its tail someplace that is not the origin, use the `arrow` command.

```
plot(arrow([1,1],[2,3], color='red',
           arrowsize=2, width=2),
     figsize=5, aspect_ratio=1)
```

Note that Sage cut off some of this plot! Also, I used some options just to show them off. The `arrow` command works in three dimensions, too.

Interactive Demonstrations This is a Sage "interact." You can use this to explore the idea of linear combinations of 2-vectors.

```
@interact(layout= {'top':[['a','c','e'],['b','d','f'],['l','m']]})
def two_dim_plot(a=input_box(1,width=10),
     b=input_box(2,width=10),c=input_box(2,width=10),
     d=input_box(1,width=10),
                 l=input_box(1,width=10), m=input_box(1,width=10),
                      e=input_box(2,width=10),f=input_box(2,width=10)):
    two_dim = arrow([0,0], [a,b], color ='red') +
         arrow([0,0],[c,d],color='blue')
    two_dim+= arrow([0,0], [l*a,l*b], color='red') +
         arrow([m*c,m*d],[l*a+m*c,l*b+m*d],color='red')
    two_dim+= arrow([0,0], [m*c,m*d], color='blue') +
         arrow([l*a,l*b],[l*a+m*c,l*b+m*d],color='blue')
    two_dim+= point([e,f],size=20,color='black',zorder=2)+
         arrow([l*a,l*b],[l*a+m*c,l*b+m*d],color='blue')
    two_dim+= text('v = (a,b)', [a-.1,b+.1], color='red') + text('w=(c,d)',
         [c+.1,d-.1],color='purple')
    two_dim+= text('l*v + m*w', [l*a+m*c+.1, l*b+m*d+.1],color='purple') +
         text('P=(e,f)', [e+.1,f+.1],color='black')
    two_dim+= arrow([0,0],[l*a+m*c,l*b+m*d],color='purple', arrowsize=1,
         width=1)
    two_dim.show(axes=True)
```

This is a different Sage interact. You can use this one to explore linear combinations of 2-vectors.

```
@interact(layout= {'top':[['a','c','e'],['b','d','f'],['l','m']]})
def two_dim_plot(a=input_box(1,width=10),
     b=input_box(2,width=10),c=input_box(2,width=10),
     d=input_box(1,width=10),
                 l=input_box(1,width=10), m=input_box(1,width=10),
                      e=input_box(2,width=10),f=input_box(2,width=10)):
    two_dim = arrow([0,0], [a,b], color ='red') +
         arrow([0,0],[c,d],color='blue')
    two_dim+= arrow([0,0], [l*a,l*b], color='red') +
         arrow([m*c,m*d],[l*a+m*c,l*b+m*d],color='red')
    two_dim+= arrow([0,0], [m*c,m*d], color='blue') +
         arrow([l*a,l*b],[l*a+m*c,l*b+m*d],color='blue')
    two_dim+= point([e,f],size=20,color='black',zorder=2)+
         arrow([l*a,l*b],[l*a+m*c,l*b+m*d],color='blue')
    two_dim+= text('v = (a,b)', [a-.1,b+.1], color='red') + text('w=(c,d)',
         [c+.1,d-.1],color='purple')
    two_dim+= text('l*v + m*w', [l*a+m*c+.1, l*b+m*d+.1],color='purple') +
         text('P=(e,f)', [e+.1,f+.1],color='black')
    two_dim+= arrow([0,0],[l*a+m*c,l*b+m*d],color='purple', arrowsize=1,
         width=1)
    two_dim.show(axes=True)
```

Exercise 1.1. Find an example of numbers λ and μ so that

$$\lambda \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \mu \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

or describe why no such example can exist.

Exercise 1.2. Find a vector $b = \left(\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right)$ so that

$$\begin{pmatrix} 2 \\ 7 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 10 \\ -3 \end{pmatrix}$$

or describe why no such example can exist.

Exercise 1.3. Find a vector $b = \left(\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right)$ so that this equation has at least one solution λ

$$\begin{pmatrix} 1 \\ -2 \end{pmatrix} + \lambda \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

or describe why no such example can exist.

Exercise 1.4. Give examples of numbers a and b such that

$$a \begin{pmatrix} 2 \\ 1 \end{pmatrix} + b \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$

or explain why no such numbers exist.

In the situations like the last exercise, the pair of numbers a, b is called a **solution** to the equation.

Exercise 1.5. Give an example of a vector $X = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ so that the equation

$$a \begin{pmatrix} 2 \\ 1 \end{pmatrix} + bX = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$

has no solution (a, b), or explain why no such example exists.

Exercise 1.6. Give an example of a number λ so that

$$\lambda \begin{pmatrix} 7 \\ -1 \\ 2 \end{pmatrix} + 3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 49 \\ -7 \\ 20 \end{pmatrix}$$

or explain why no such number exists.

Exercise 1.7. Give an example of numbers λ and μ which are a solution to the equation

$$\lambda \begin{pmatrix} 7 \\ -1 \\ 2 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 49 \\ -7 \\ 20 \end{pmatrix}$$

or explain why no such solution exists.

Exercise 1.8. Give an example of a vector $w = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ so that the equation

$$a \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

has no solution (a, b), or explain why no such vector exists.

Exercise 1.9. Give an example of a vector $w = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ so that the equation

$$a \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

has exactly one solution (a, b), or explain why no such vector exists.

Exercise 1.10. Give an example of a vector $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ such that the equation

$$a \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + c \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

has no solutions (a, b, c), or explain why no such vector exists.

Exercise 1.11. Give an example of a vector $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ such that the equation

$$a \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + c \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

has exactly one solution, or explain why no such vector exists.

Exercise 1.12. Give an example of a vector $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ such that the equation

$$a \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + b \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} + c \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ 7 \end{pmatrix}$$

has no solutions, or explain why no such vector exists.

1.2 The Dot Product

1.2.1 The Assignment

- Read section 1.2 of *Strang*
- Read the following and complete the exercises below.

1.2.2 Learning Objectives

Before class, a student should be able to

- Compute the dot product of two given vectors.
- Compute the length of a given vector.
- Normalize a given vector.
- Recognize that $u \cdot v = 0$ is the same as "u and v are orthogonal."
- Compute the angle between two given vectors using the cosine formula.

At some point, a student should be able to

- Interpret the statements $u \cdot v < 0$ and $u \cdot v > 0$ geometrically.
- Pass back and forth between linear equations and equations involving dot products.
- Make pictures of level sets of the dot product operation.

1.2.3 Discussion

The dot product is a wonderful tool for encoding the geometry of Euclidean space, but it can be a bit mysterious at first. As Strang shows, it somehow holds all of the information you need to measure lengths and angles.

What does this weird thing have to do with linear algebra? A dot product with a "variable vector" is a way of writing a linear equation. For example,

$$\begin{pmatrix} 7 \\ 3 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = 7x + 3y - 2z.$$

Sometimes this will allow us to connect linear algebra to geometry, and use geometric thinking to answer algebraic questions.

1.2.4 Sage and the dot product

Basic Commands

Sage has a built-in dot product command `u.dot_product(v)`. This will return the dot product of the vectors u and v.

```
u = vector([1,1,1]); v = vector([-1,0,1])
u.dot_product(v)
```

```
0
```

It also has a built-in command for computing lengths. Here sage uses the synonym "norm": `u.norm()`. Of course, you can also call this like a function instead of like a method: `norm(u)`.

```
u.norm(), v.norm()
```

```
(sqrt(3), sqrt(2))
```

There is no built-in command for angles. You just have to compute them using the cosine formula, like below. (I will break up the computation, but it is easy to do it all with one line.)

```
num = u.dot_product(v)
den = u.norm() * v.norm()
angle = arccos(num / den)
angle
```

```
1/2*pi
```

Of course, Sage's `arccos` command returns a result in *radians*. To switch to degrees, you must convert.

```
angle*180/pi
```

```
90
```

Often, it is helpful to normalize a vector. You can do this with the `normalized` method like this:

```
u.normalized()
```

```
(1/3*sqrt(3), 1/3*sqrt(3), 1/3*sqrt(3))
```

Sage Interacts

Interacts would go here...

1.2.5 Exercises about the Dot Product

Exercise 1.13. What shape is the set of solutions $\binom{x}{y}$ to the equation

$$\begin{pmatrix}3\\7\end{pmatrix} \cdot \begin{pmatrix}x\\y\end{pmatrix} = 5?$$

That is, if we look at all possible vectors $\binom{x}{y}$ which make the equation true, what shape does this make in the plane?

What happens if we change the vector $\binom{3}{7}$ to some other vector? What happens if we change the number 5 to some other number?

What happens if instead of 2-vectors, we use 3-vectors?

Exercise 1.14. Find an example of two 2-vectors v and w so that $\binom{1}{2} \cdot v = 0$ and $\binom{1}{2} \cdot w = 0$, or explain why such an example is not possible.

Exercise 1.15. Let $v = \binom{3}{-1}$. Find an example of a pair of vectors u and w such that $v \cdot u < 0$ and $v \cdot w < 0$ and $w \cdot u = 0$, or explain why no such pair of vectors can exist.

Exercise 1.16. Find an example of three 2-vectors u, v, and w so that $u \cdot v < 0$ and $u \cdot w < 0$ and $v \cdot w < 0$, or explain why no such example exists.

Exercise 1.17. Find an example of a number c so that

$$\begin{pmatrix}1\\-1\end{pmatrix} \cdot \begin{pmatrix}x\\y\end{pmatrix} = c$$

has the vector $\binom{4}{7}$ as a solution, or explain why no such number exists.

Exercise 1.18. Let $v = \left(\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\right)$ and $w = \left(\begin{smallmatrix} -3 \\ 4 \end{smallmatrix}\right)$. Find an example of a number c so that

$$v \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = c \quad \text{and} \quad w \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = c,$$

or explain why this is not possible.

Exercise 1.19. Let $P = \left(\begin{smallmatrix} -3 \\ 4 \end{smallmatrix}\right)$. Find an example of numbers c and d so that

$$\begin{pmatrix} 2 \\ -1 \end{pmatrix} \cdot P = c \quad \text{and} \quad \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot P = d,$$

or explain why no such example is possible.

Now we move to three dimensions!

Exercise 1.20. Let $V = \left(\begin{smallmatrix} 1 \\ 1 \\ 1 \end{smallmatrix}\right)$. Find a unit vector of the form $X = \left(\begin{smallmatrix} x \\ y \\ 0 \end{smallmatrix}\right)$ so that $V \cdot X = \sqrt{2}$, or explain why no such vector exists.

Exercise 1.21. Find an example of numbers c, d, and e so that there is no solution vector $X = \left(\begin{smallmatrix} x \\ y \\ z \end{smallmatrix}\right)$ which simultaneously satisfies the three equations

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot X = c, \qquad \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \cdot X = d, \qquad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot X = e,$$

or explain why no such numbers exist.

Exercise 1.22. Find an example of numbers c, d, and e so that there is no solution vector $X = \left(\begin{smallmatrix} x \\ y \\ z \end{smallmatrix}\right)$ which simultaneously satisfies the three equations

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot X = c, \qquad \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \cdot X = d, \qquad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot X = e,$$

or explain why no such numbers exist.

1.3 Matrices

1.3.1 The Assignment

- Read *Strang* section 1.3 (pages 22-27).
- Read the following and complete the exercises below.

1.3.2 Learning Goals

Before class starts, a student should be able to:

- multiply a matrix times a vector
  - as a linear combination of columns
  - as a set of dot products, row times column
- translate back and forth between our three representations
  - a system of linear equations,
  - a linear combination of vectors equation, and
  - a matrix equation $Ax = b$.
- Correctly identify the rows and columns of a matrix
- describe what is meant by a lower triangular matrix

At some point, as student should be comfortable with these concepts, which get a very brief informal introduction in this section:

- linear dependence and linear independence
- the inverse of a matrix

1.3.3 Discussion

A **matrix** is a two-dimensional array of numbers like this:

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.$$

Sometimes it helps to think of a matrix as a collection of its **rows** which are read across:

$$M = \begin{pmatrix} \longrightarrow \\ \longrightarrow \end{pmatrix}$$

and sometimes it helps to think of a matrix as a collection of its **columns** which are read down:

$$M = \begin{pmatrix} \downarrow & \downarrow \end{pmatrix}.$$

It is often more clear to describe a matrix by giving the sizes of its rows and columns. An m by n matrix is one having m rows and n columns. It is really easy to get these reversed, so be careful. For example, this is a 2×3 matrix, because it has two rows and three columns:

$$B = \begin{pmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{pmatrix}$$

A matrix is called a **square** matrix when the number of rows and the number of columns is equal. The matrix A that I wrote down above is square because it is a 2×2 matrix.

Multiplying Matrices and Vectors

It is possible to multiply a matrix by a vector like this:

$$\begin{pmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{pmatrix} \begin{pmatrix} 13 \\ 21 \\ 34 \end{pmatrix} = \begin{pmatrix} 102 \\ 416 \end{pmatrix}$$

For this to work, it is absolutely crucial that the sizes match up properly. If the matrix is m by n, then the vector must have size n. In the above example $m = 2$ and $n = 3$.

Later, we shall see that the word "multiplication" is not really the best choice here. It is better to think of the matrix as "acting on" the vector and turning it into a new vector. For now, the word multiplication will serve.

How exactly does one define this matrix–vector multiplication?

Linear Combination of Columns Approach The first way to perform the matrix-vector multiplication is to think of the vector as holding some coefficients for forming a linear combination of the columns of the matrix. In our example, it looks like this:

$$\begin{pmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{pmatrix} \begin{pmatrix} 13 \\ 21 \\ 34 \end{pmatrix} = 13 \begin{pmatrix} 1 \\ 3 \end{pmatrix} + 21 \begin{pmatrix} 1 \\ 5 \end{pmatrix} + 34 \begin{pmatrix} 2 \\ 8 \end{pmatrix} = \begin{pmatrix} 102 \\ 416 \end{pmatrix}$$

Dot Products with the Rows Approach The second way is to think of the matrix as a bundle of vectors lying along the rows of the matrix, and use the dot product. In our example above, this means that we consider the vectors

$$r_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \quad r_2 = \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix}, \text{ and } v = \begin{pmatrix} 13 \\ 21 \\ 34 \end{pmatrix}$$

(notice I've rewritten the rows as columns) and then perform this kind of operation:

$$\begin{pmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{pmatrix} \begin{pmatrix} 13 \\ 21 \\ 34 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} v = \begin{pmatrix} r_1 \cdot v \\ r_2 \cdot v \end{pmatrix} = \begin{pmatrix} 102 \\ 416 \end{pmatrix}.$$

Two important remarks:

- Note that these operations only make sense if the sizes match up properly.
- Note that the two versions of the operation give you the same results.

Matrix Equations

There are many situations in linear algebra that can be rewritten in the form of an equation that looks like this:

$$Av = b$$

where A is a matrix, and v and b are vectors. The interesting case is when we know A and b, but we want to find the unknown v. We will call this a **matrix-vector equation**.

Let's consider the case where you are given some **square** matrix A. Sometimes one can find another matrix B so that no matter what vector b is chosen in the matrix-vector equation above, the solution vector takes the form $v = Bb$. When

this happens, we say that A is **invertible** and call B the **inverse** of A. It is common to use the notation A^{-1} in place of B. This is a wonderful situation to be in! Eventually, we will want to figure out some test for when a given matrix is invertible, and find some ways to compute the inverse.

A Note about Vectors

This reading also has a brief introduction to the idea of a set of vectors being **linearly depedent** or **linearly independent**. Strang is coy about the precise definition, so here it is:

A set of vectors $v_1, v_2, \ldots, v_n$ is called **linearly depdendent** when there is some choice of numbers $a_1, a_2, \ldots, a_n$ which are not all zero so that the linear combination

$$a_1v_1 + a_2v_2 + \cdots + a_nv_n = 0$$

A set of vectors which is not linearly dependent is called **linearly independent**.

This is a little funny the first time you read it. Note that for any set of vectors, you can make a linear combination of those vectors come out as 0. Simply choose all of the coefficients to be zero. But that is so easy to do we call it **trivial**. What the definition is asking is that we find a *nontrival linear combination of the vectors to make zero.*

1.3.4 Sage and Matrices

Sage has many useful commands for working with linear algebra, and given the central role played by matrices in this subject, there are lots of things Sage can do with matrices. We'll focus here on just basic construction and matrix–vector multiplication.

The matrix construction command

The command to construct a matrix is pretty straightforward. One types `matrix(r, c, [list of entries` where `r` is the number of rows and `c` is the number of columns. The entries should be read across the rows starting with the top row.

```
A = matrix(2,3, [1,2,3,5,8,13]); A
```

```
[ 1  2  3]
[ 5  8 13]
```

If you wish, you can structure that list of entries to be a list of lists, where each sublist is a row in your matrix.

```
B = matrix(2,3, [[1,2,3], [5,8,13]]); B
```

```
[ 1  2   3]
[ 5  8  13]
```

Every once in a while, it might matter to you what kinds of numbers you put into the matrix. Sage will let you specify them by putting in an optional argument like this: `matrix(number type, r, c, [list of entries])`

```
C = matrix(ZZ, 2, 2, [2,1,1,1])
C # the best matrix
```

```
[2 1]
[1 1]
```

The notation `ZZ` means "integers." There are other sets of numbers here:

- `QQ` the rational numbers (with exact arithmetic)
- `RR` the real numbers (with computer precision arithmetic)
- `CC` the complex numbers
- `AA` the set of all algebraic numbers, that is, all of the numbers that are roots of some polynomial with integer coefficients

You can find out what kind of entries a matrix thinks it has by calling the `.parent()` method on it.

```
A.parent()
# this should say something about the integers
```

```
Full MatrixSpace of 2 by 3 dense matrices over Integer Ring
```

```
D = matrix(QQ, 3,3, [[1,0,1],[2/3, 1, 0],[0,0,9/5]])
# this should say something about the rationals
D.parent()
```

```
[  1   0   1]
[2/3   1   0]
[  0   0 9/5]
Full MatrixSpace of 3 by 3 dense matrices over Rational Field
```

Building a matrix from rows or columns

It is possible to build a matrix by bundling together a bunch of vectors, too. Let's start with an example made using rows.

```
v1 = vector([2,1]); v2= vector([3,4])
# construct E with rows v1 and v2, then display
E = matrix([ v1, v2]); E
```

```
[2 1]
[3 4]
```

Sage prefers rows. I wish it were the other way, but I am sure there is a good reason it prefers rows. If you want to make a matrix whose columns are the vectors `v1` and `v2`, you can use the `transpose` method. We'll talk more about the operation of transpose later, but it basically "switches rows for columns and vice versa."

```
F = matrix([v1,v2]).transpose(); F
```

```
[2 3]
[1 4]
```

Matrix action on vectors

Of course, Sage knows how to perform the action of a matrix on a vector.

```
C; v1
```

```
[2 1]
[1 1]
(2, 1)
```

```
C*v1
```

```
(5, 3)
```

And if you get the sizes wrong, it will return an error.

```
A; v1
```

```
[ 1  2  3]
[ 5  8 13]
(2, 1)
```

```
A*v1
```

```
Error in lines 1-1
...
TypeError: unsupported operand parent(s) for '*': 'Full MatrixSpace of
    2 by 3 dense matrices over Integer Ring' and 'Ambient free module
    of rank 2 over the principal ideal domain Integer Ring'
```

If you really need it, Sage can tell you about inverses.

```
A.is_invertible()
```

```
False
```

```
C.is_invertible()
```

```
True
```

```
C.inverse()
```

```
[ 1 -1]
[-1  2]
```

1.3.5 Exercises

Exercise 1.23. Make an example of a matrix $\left(\begin{smallmatrix} 1 & \bullet \\ -1 & \bullet \end{smallmatrix}\right)$ so that the equation

$$\begin{pmatrix} 1 & \bullet \\ -1 & \bullet \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

has exactly one solution, or explain why this is not possible.

Interpret this as a statement about 2-vectors and draw the picture which corresponds.

Exercise 1.24. Make an example of a matrix $\left(\begin{smallmatrix} 4 & 8 & \bullet \\ 3 & 6 & \bullet \\ 1 & 2 & \bullet \end{smallmatrix}\right)$ so that the equation

$$\begin{pmatrix} 4 & 8 & \bullet \\ 3 & 6 & \bullet \\ 1 & 2 & \bullet \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 6 \\ 2 \end{pmatrix}$$

has exactly one solution, or explain why this is not possible.

Interpret this as a statment about 3-vectors and draw the picture which corresponds.

Exercise 1.25. Make an example of a matrix $\left(\begin{smallmatrix} 2 & -1 \\ \bullet & \bullet \end{smallmatrix}\right)$ so that the equation

$$\begin{pmatrix} 2 & -1 \\ \bullet & \bullet \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \end{pmatrix}$$

has exactly one solution, or explain why this is not possible.

Interpret this as a statement about a pair of lines in the plane and draw the picture which corresponds.

Exercise 1.26. Make an example of a matrix $\left(\begin{smallmatrix}1&0&1\\1&1&3\\\bullet&\bullet&\bullet\end{smallmatrix}\right)$ so that the equation

$$\begin{pmatrix}1&0&1\\1&1&3\\\bullet&\bullet&\bullet\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}1\\1\\1\end{pmatrix}$$

has no solutions, or explain why this is not possible.

Interpret this as a statement about a planes in space and draw the picture which corresponds.

Exercise 1.27. Find a triple of numbers x, y, and z so that the linear combination

$$x\begin{pmatrix}1\\2\\3\end{pmatrix}+y\begin{pmatrix}4\\5\\6\end{pmatrix}+z\begin{pmatrix}7\\8\\9\end{pmatrix}$$

yields the zero vector, or explain why this is not possible.

Rewrite the above as an equation which involves a matrix.

Plot the three vectors and describe the geometry of the situation.

Exercise 1.28. The vectors

$$r_1=\begin{pmatrix}1\\4\\7\end{pmatrix},\qquad r_2=\begin{pmatrix}2\\5\\8\end{pmatrix},\qquad\text{and}\qquad r_3=\begin{pmatrix}3\\6\\9\end{pmatrix}$$

are linearly dependent because they lie in a common plane (through the origin). Find a normal vector to this plane.

Since the vectors are linearly dependent, there must be (infinitely) many choices of scalars x, y, and z so that $xr_1+yr_2+zr_3=0$. Find two sets of such numbers.

Exercise 1.29. Consider the equation

$$\begin{pmatrix}2&1\\1&1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}b_1\\b_2\end{pmatrix}.$$

We are interested in being able to solve this for x and y for any given choice of the numbers b_1 and b_2. Figure out a way to do this by writing x and y in terms of b_1 and b_2.

Rewrite your solution in the form

$$\begin{pmatrix}x\\y\end{pmatrix}=b_1\begin{pmatrix}\bullet\\\bullet\end{pmatrix}+b_2\begin{pmatrix}\bullet\\\bullet\end{pmatrix}.$$

How is this related to the inverse of the matrix $A=\left(\begin{smallmatrix}2&1\\1&1\end{smallmatrix}\right)$?

Exercise 1.30. Find an example of a number c and a vector $\left(\begin{smallmatrix}b_1\\b_2\end{smallmatrix}\right)$ so that the equation

$$\begin{pmatrix}3&51\\c&17\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}b_1\\b_2\end{pmatrix}$$

does not have a solution, or explain why no such example exists.

Explain your solution in terms of

- lines in the plane,
- 2-vectors and linear combinations, and
- invertibility of a matrix.

1.4 Getting Started with Sage

1.4.1 The Assignment

It it is time to get comfortable with the basics of using Sage. Begin by setting up an account with SageMathCloud.

- Be sure that you have login access to computers in student labs across campus. Before class, drop by a campus lab and make sure your credentials work properly and you can sign in to a machine. For our tutorial session we will use one of the labs on the first floor of Wright Hall, so one of those would make a good choice.
- Go to the course web site and follow the links to the Sage Intro workshop.
- Complete the first two steps of the workshop set up process:
    1. Start: where you make an account at SageMathCloud.
    2. Get: where you create your first "project" and populate it with files for a tutorial.

1.4.2 Learning Objectives

At the end of this assignment, a student should have an account at SageMathCloud and should be able to log into the service without trouble. The student will also have a first project with some files.

1.5 Going Further with the Basic Objects

1.5.1 The Assignment

Now we take a short break to revisit and consolidate the learning you have done so far. Revisit the reading and the exercises you have done in Chapter One: Basic Objects. The important feature of this work should be learning to think about your own thinking. This sort of **meta-cognition** characterizes expert learners. Eventually, you want to be able to monitor your work at all times and recognize when you understand deeply and when you do not. This will allow you to self-correct.

To help you get started with meta-cognition, I listed learning goals in each section. To go further, you need to explicitly go through the process of reviewing what you can do and what you cannot. Here are some prompts to help you get started with this process.

- Review the learning goals from each section. Can you do the things described? Can you do them sometimes, or have you mastered them so you can do them consistently?
- Go back through the exercises in this workbook. Complete any items you did not complete the first time through. Prepare any that we have not discussed in class so that you will be ready to present them.
- As you look through all of the tasks, go deeper into them. Can you connect each exercise to one of our pictures? Try to build a mental model of how the exercise and its solution work.
- If your first solution to an exercise involve a "guess-and-check" approach, can you now complete the exercise in a *purposeful* and systematic manner?
- Make a list of concepts or exercises that are not clear to you. Phrase each item in your list as a question, and make each question as specific as possible. Talk with fellow students or your instructor until you can answer your own questions.

Chapter 2

Linear Equations

This chapter is dedicated to the first major problem of linear algebra: solving systems of linear equations. Restated as a set of questions, we will consider these.

- What is the set of solutions to a given system of linear equations?
- When does a given system have solution, and when does it not?
- If there is a solution, how many solutions are there?
- What ways do we have of describing the collection of solutions?
- Is there a computationally effective way to find those solutions?

Though we begin with the first question, we answer the last one first. As we explore the process of finding solutions, we will start to build the tools we need to finish answering the other questions in a later chapter. In this chapter, we aim to get as complete understanding as we can for at least a special case: **square systems**.

We will begin with a study of the two ways that vectors let us make pictures of systems of linear equations. Then we take up a basic process for finding solutions, where matrices will appear as a convenient notational device. But as we dig a little further, matrices will become interesting in their own way, so we will study those. But what we study will relate back to the fundamental issue of solving systems of linear equations.

This chapter has two short review sections in it. One just after we learn about elimination, and another at the end of the chapter.

2.1 Three Geometric Models

2.1.1 The Assignment

- Read section 2.1 of *Strang* (pages 31-40).
- Read the following and complete the exercises below

2.1.2 Learning Goals

Before class, a student should be able to:

- Translate back and forth between the three algebraic representations:
  - A system of linear equations.

  - An equation involving a linear combination of vectors.
  - A matrix equation.
- Can write down the $n \times n$ **identity matrix**.

Sometime in the near future, a student should be able to:

- Given a system, interpret and plot the "row picture".
- Given a system, interpret and plot the "column picture".
- Use a matrix as a model of a **transformation**, including stating the **domain** and the **range**.

2.1.3 Discussion

Now we have a little experience with vectors and related things, it is time to be aware of what we have done so we can use it as a foundation for future work. So far, we have talked about two geometric interpretations for a system of linear equations, the **row picture** and the **column picture**.

Does the following picture make sense to you?

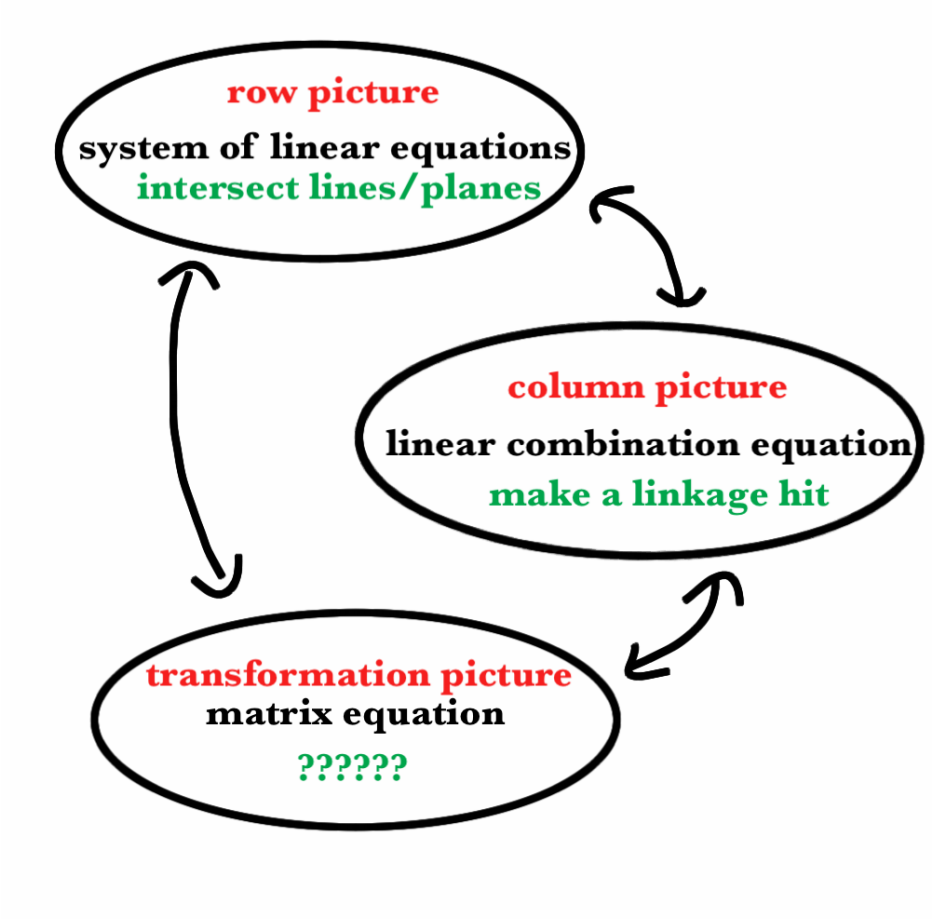

Figure 2.1: The three geometric models of linear algebra

A deep understanding of linear algebra will involve a level of comfort with each of these three views of the subject in the diagram, and also the ability to pass back and forth between them.

The Transformational View

We have seen that matrices can be made to "act upon" vectors by a kind of multiplication. In particular, if A is an $m \times n$ matrix, then A can be multiplied (on the left) with a column vector of size n, and the result is a column vector of size m.

This makes A into a kind of **function**. (We will use the synonyms **mapping** or **transformation**, too.) For every vector v of size n, the matrix A allows us to compute a new vector $T_A(v) = Av$ of size m. This is the basic example of what we will eventually call a **linear transformation**.

$$\mathbb{R}^m \xrightarrow{T_A} \mathbb{R}^n$$

$$v \longmapsto Av.$$

One of our long term goals is to find a way to think about the geometry of linear algebra from this viewpoint, too.

2.1.4 Sage and Plotting for Linear Algebra

There are a few new Sage commands that might be useful here. We have already seen how to take linear equations and turn them into vectors and then turn the vector equation into a matrix equation. But Sage can help us move in the other direction, too.

The keys are commands to pull out the rows and columns from a given matrix. Let's start with a simple situation where the matrix equation is

$$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x & y \end{pmatrix} = \begin{pmatrix} 3 & 4 \end{pmatrix}.$$

```
A = matrix(QQ, 2,2, [2,1,1,1]); A
```

```
[2 1]
[1 1]
```

```
x, y = var('x y')
X = vector([x,y]); X
```

```
(x, y)
```

```
b = vector([3,4]); b
```

```
(3, 4)
```

To get Sage to pull out the columns, we can use the `.columns()` method. If we want just one column, we can use the `.column()` method, but then we have to remember to specify which column we want.

```
A.columns() # this will return a list
```

```
[(2, 1), (1, )]
```

```
A.column(1)
```

```
(1, 1)
```

Big important note: Sage always numbers lists starting with zero. so the first element of every list is the 0 entry, and the second element is the 1 entry.

Now it is possible to make Sage do things like this:

```
column_plot = plot(A.column(0), color='red')
column_plot+= plot(A.column(1), color='blue')
column_plot+= plot(b, color='purple')
show(column_plot, figsize=5, aspect_ratio=1)
```

This is an example of the a column picture.

One can also pull out the rows with corresponding row methods. And if you recall the way that matrix multiplication works if you think of rows, you can make a row picture.

```
A.rows()
```

```
[(2, 1), (1, 1)]
```

```
expr1 = A.row(0).dot_product(X) == b[0]
expr2 = A.row(1).dot_product(X) == b[1]

print expr1
print expr2
```

```
2*x + y == 3
x + y == 4
```

And now the picture:

```
row_plot = implicit_plot(expr1, [x,-5,5], [y,-1,9], color='blue')
row_plot+= implicit_plot(expr2, [x,-5,5], [y,-1,9], color='red')
show(row_plot, axes=True)
```

2.1.5 Exercises

Exercise 2.2. Make an example of a system of linear equations which some students might find challenging to change into an equation involving a linear combination. Explain what the challenge is and how you can think clearly to overcome it.

Exercise 2.3. Make an example of a linear combination equation which some students might find challenging to change into a system of linear equations. Explain what the challenge is and how you can think clearly to overcome it.

Exercise 2.4. Consider the matrix equation

$$\begin{pmatrix} 1 & 2 & 4 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \end{pmatrix}.$$

1. Draw a diagram representing the row picture of this matrix equation.
2. Draw a diagram representing the column picture of this matrix equation.

Exercise 2.5. Make an example of a system of linear equations so that the corresponding column picture is about linear combinations of four 2-vecs becoming the zero vector.

Exercise 2.6. Find a linear combination equation so that the corresponding system of linear equations corresponds to finding the intersection of three lines in the plane.

Exercise 2.7. Find an example of a vector b so that the equation

$$\begin{pmatrix} -1 & 2 \\ 5 & -9 \end{pmatrix} v = b$$

has no solution v, or explain why it is impossible to find such an example.

Exercise 2.8. In each of the below, find an example of a matrix B which has the described effect.

1. $B\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}$
2. Rotates vectors through $45°$ counter-clockwise.
3. Reflects vectors across the $y$axis.
4. $B\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x+y \\ y \end{pmatrix}$.

2.2 Solving Systems

2.2.1 The Assignment

- Read section 2.2 of Strang (pages 45-51).
- Read the following and complete the exercises below.

2.2.2 Learning Goals

Before class, a student should be able to do the following things.

- Clearly state and use the following vocabulary words: pivot, multiplier, triangular matrix, back substitution, singular, non-singular

Sometime after class, a student should be able to the following things.

- Perform elimination to put a system of linear equations into triangular form.
- Solve small systems by hand.
- Explian the two failure modes for elimination, and describe which leads to no solutions, and which leads to infinitely many solutions.
- Solve larger systems with the help of a computer algebra package (Sage).

2.2.3 Discussion: Elimination for Solving Systems of Linear Equations

Now we begin the process of learning how to solve a system of linear equations systematically through a process called **elimination**.

Some terminology

A typical system looks something like this:

$$\begin{cases} 3x_1 + 2x_2 - \pi x_3 = 0 \\ -4x_1 - 33x_2 + x_3 = 12 \end{cases}$$

This situation is *two* equations in *three* unknowns. The unknowns here are the three numbers x_1, x_2 and x_3 for which we search. Usually, we bundle the numbers together as a vector (x_1, x_2, x_3). If we can find a vector which makes all of the equations true simultaneously, we call that vector a **solution**.

Keep in mind that the process involves eliminating instances of the variable below **pivots**. Strang describes the process pretty well, and gives good examples. What Strang describes in this section is sometimes called **the forward pass** elimination.

Watch out for situations which are **singular** in that they have fewer pivots than unknowns. A system is called **non-singular** if it has as many pivots as unknowns.

Keeping track of things

Playing with all of the equations is nice, but all that really matters is the collection of coefficients, and the numbers on the right hand sides of the equal signs. Experienced solvers get tired of copying notation from line to line in a computation, so they only keep track of the matrix of coefficients, **augmented** by the vector on the right-hand side. In the example above, that augmented matrix is

$$\left(\begin{array}{ccc|c} 3 & 2 & -\pi & 0 \\ -4 & -33 & 1 & 12 \end{array}\right)$$

All of the row operations can be performed on just this augmented matrix, without losing any of the essential information.

2.2.4 Sage and Row Operations

The process of elimination for systems of equations involves performing operations on the equations. When translated to matrix form, it involves operations on the rows of the coefficient matrix. The corresponding matrix methods come in two types.

The first type of method modifies the matrix "in place", which means that it *Changes the input matrix.*

- `A.rescale_row(r, num)` multiplies row `r` by the factor of `num`.
- `A.swap_rows(r1, r2)` switches the places of rows `r1` and `r2`.
- `A.add_multiple_of_row(target, useful, num)`. This adds `num` times row `useful` to row `target`.

Throughout, please remember that Sage uses 0-based indexing! So the rows are labeled `0, 1, 2, ...`

```
A = matrix(QQ, 3,3, [0,2,4, 1,1,5, 6,2,5]); A
```

```
[0 2 4]
[1 1 5]
[6 2 5]
```

```
A.swap_rows(0,1); A
```

```
[1 1 5]
[0 2 4]
[6 2 5]
```

```
A.add_multiple_of_row(2,0,-6)
A # this should add -6 times row 0 to row 2
```

```
[  1   1   5]
[  0   2   4]
[  0  -4 -25]
```

```
A.rescale_row(1,1/2); A
```

```
[  1   1   5]
[  0   1   2]
[  0  -4 -25]
```

```
A.add_multiple_of_row(2,1,4)
A # this should add 4 times row 2 to row 2
```

```
[  1   1   5]
[  0   1   2]
[  0   0 -17]
```

```
A.rescale_row(2,-1/17); A
```

```
[1 1 5]
[0 1 2]
[0 0 1]
```

This just did the whole process of **forward pass elimination**. (Well, we did a bit more than Strang would. He wouldn't rescale the rows.)

Sometimes you do not want to change the matrix `A`. If instead, you want to leave `A` alone, you can use these methods, which return a new object and do not change `A`.

- `A.with_rescaled_row(r, num)`
- `A.with_swapped_rows(r1, r2)`
- `A.with_added_multiple_of_row(t, u, num)`

Let's do the same operations as above, but without changing `A`. This will mean making a bunch of new matrices. In fact, let's also change the name of our matrix to `B`

```
B = matrix(QQ, 3,3, [0,2,4, 1,1,5, 6,2,5]); B
```

```
[0 2 4]
[1 1 5]
[6 2 5]
```

```
B1 = B.with_swapped_rows(0,1); B1
```

```
[1 1 5]
[0 2 4]
[6 2 5]
```

```
B2 = B1.with_added_multiple_of_row(2,0,-6)
B2 # this should add -6 times row 0 to row 2
```

```
[  1   1   5]
[  0   2   4]
[  0  -4 -25]
```

```
B3 = B2.with_rescaled_row(1,1/2); B3
```

```
[  1   1   5]
[  0   1   2]
[  0  -4 -25]
```

```
B4 = B3.with_added_multiple_of_row(2,1,4)
B4 # this should add 4 times row 2 to row 2
```

```
[  1   1   5]
[  0   1   2]
[  0   0 -17]
```

```
B5 = B4.with_rescaled_row(2,-1/17); B5
```

```
[1 1 5]
[0 1 2]
[0 0 1]
```

This second option has some advantages. At any point, you can revise your work, because the original matrix is still in memory, and so are all of the intermediate steps. Let's display all six of the matrices at once to see that they all still exist.

```
B, B1, B2, B3, B4, B5
```

2.2.5 Exercises

Exercise 2.9. Use the elimination method to transform this system into an easier one. (Can you make it triangular?) Circle the pivots in the final result.

$$\begin{cases} 2x & + & 3y & + & z & = & 8 \\ 4x & + & 7y & + & 5z & = & 20 \\ & - & 2y & + & 2z & = & 0 \end{cases}$$

What two operations do you use to do this efficiently? Now use back substitution to solve the system.

Exercise 2.10. *(Sage Exercise)*: Because the last system can be transformed in two operations, there are three equivalent systems generated through the process (the original, the intermediate, and the final).

Make row picture plots for each of the three systems. [Hint: Sage] How do the operations transform the pictures?

Exercise 2.11. Suppose that a system of three equations in three unknowns has two solutions (a, b, c) and (A, B, C). Explain why the system must have other solutions than these two. Describe clearly two other solutions in terms of a, b, c, A, B, C.

Exercise 2.12. Find three examples of numbers a so that elimination will fail to give three pivots for this coefficient matrix:

$$A = \begin{pmatrix} a & 2 & 3 \\ a & a & 4 \\ a & a & a \end{pmatrix}$$

Exercise 2.13. How many ways can two lines in the plane meet? Make examples to represent as many *qualitatively different* situations as you can.

Exercise 2.14. Complete the following to make an example of a system of two equations in two unknowns which is singular but still has a solution, or explain why no such example exists.

$$\begin{cases} 2x & + & 3y & = & 1 \\ \bullet x & + & \bullet y & = & \bullet \end{cases}$$

Exercise 2.15. Complete the following to a system of three equations in three unknowns which is singular and does not have a solution, or explain why no such example exists.

$$\begin{cases} & & 3y & - & z & = & 1 \\ 2x & - & y & + & 3z & = & 0 \\ \bullet x & + & \bullet y & + & \bullet z & = & \bullet \end{cases}$$

Exercise 2.16. Complete the following to a system of three equations in three unknowns which is singular but still has a solution, or explain why no such example exists.

$$\begin{cases} x & + & y & + & z & = & 1 \\ 2x & + & y & + & 2z & = & 0 \\ \bullet x & + & \bullet y & + & \bullet z & = & \bullet \end{cases}$$

Exercise 2.17. How many ways can three planes in three dimensional space meet? Make examples to represent as many *qualitatively different* situations as you can.

2.3 Elimination using Matrices

2.3.1 The Assignment

2.3.2 Learning Goals

2.3.3 Discussion

2.3.4 Sage and xxx

2.3.5 Exercises

2.4 Systems: Going Further

2.4.1 The Assignment

2.4.2 Learning Goals

2.4.3 Discussion

2.4.4 Sage and xxx

2.4.5 Exercises

2.5 Matrix Algebra

2.5.1 The Assignment

2.5.2 Learning Goals

2.5.3 Discussion

2.5.4 Sage and xxx

2.5.5 Exercises

2.6 Matrix Inverses

2.6.1 The Assignment

2.6.2 Learning Goals

2.6.3 Discussion

2.6.4 Sage and xxx

2.6.5 Exercises

2.7 The LU Decomposition

2.7.1 The Assignment

2.7.2 Learning Goals

2.7.3 Discussion

2.7.4 Sage and xxx

2.7.5 Exercises

2.8 Permutation Matrices

2.8.1 The Assignment

2.8.2 Learning Goals

2.8.3 Discussion

2.8.4 Sage and xxx

2.8.5 Exercises

2.9 Matrix Algebra: Going Further

2.9.1 The Assignment

2.9.2 Learning Goals

2.9.3 Discussion

2.9.4 Sage and xxx

2.9.5 Exercises

Chapter 3

Vector Spaces and Subspaces

Here is some introductory text.

Chapter 4

Orthogonality

Here is some introductory text.

Chapter 5

Determinants

Here is some introductory text.

Chapter 6

Eigendata and the Singular Value decomposition

Here is some introductory text.