Quantitative Engineering Analysis I

Fifth Edition

Spring 2020

Contents

Module I

Faces: Linear Algebra Through Facial Recognition

Chapter 1

Day 1: Facial Recognition: The Big Picture

1.1 Schedule

- 0900-0920: Welcome - A letter to my future self
- 0920-0950: A simple face detection: Round 1
- 0950-1010: Round 2: What did you notice?
- 1010-1030: Debrief: What did we learn?
- 1030-1045: Coffee
- 1045-1055: Pixel arithmetic
- 1055-1105: A Universal Set of Building Blocks
- 1105-1125: A Better Set of Building Blocks?
- 1125-1145: Towards an Optimal Basis
- 1145-1200: Broadcast debrief (via Zoom)
- 1200-1220: Course logistics
- 1220-1230: Day 1 survey

Welcome to QEA Module One! In this module, you will develop software to recognize your face among everyone in QEA (hello, new late-night security). It all functions through applying some beautiful mathematics and using computational tools. Let’s first imagine how a computer “sees" an image as numbers.

1.2 Facial recognition- "seeing" via numbers

Round 1: From images to numbers [30 mins]

At your table you will find a smiley face. Imagine converting this face into a form that a computer can understand (i.e., numbers). A grid is superimposed on the face for your reference.

Goal Design a method that enables a "computer" to (approximately) reproduce the face from a list of numbers and an algorithm that you define. The numbers can be grouped within the list, but your list should contain numbers only. An example of a group of numbers is [2,4] or [0, 100, 14]. You will create the algorithm (or, equivalently, the instructions) that tell the computer what to do with your list of numbers.

When you’ve defined your group’s method,

- Generate the list of numbers that represents your face using your method.
- Make a set of instructions (your "algorithm") on your portable white board using a BLACK marker so that another group can recreate your image from your list of numbers.

- Trade instructions with another group.
- Create the other group's face from their algorithm on the blank grid.
- Record any challenges you encounter on their portable whiteboard using a RED marker.
- Exchange back your original materials and debrief on what you've learned about your method at your table.

Round 2 [20 mins]

Goal Adjust your method to be able to distinguish the new faces that you've just been given. The 8x8 grid is shown for reference; you are not restricted to this grid.

Discuss the following and record your answers on your portable whiteboard:

- How does your method need to transform the original image in order to "see" the detail of the face?
- What "demands" does your new method make on the computer compared to the old method? (Remember that the computer is using your algorithm and numbers to represent the images)
- Consider a photo of a human face, in what ways does your numerical method contain inherent limits or biases?

A debrief (in each room) [20 mins]

Coffee Break [15 min]

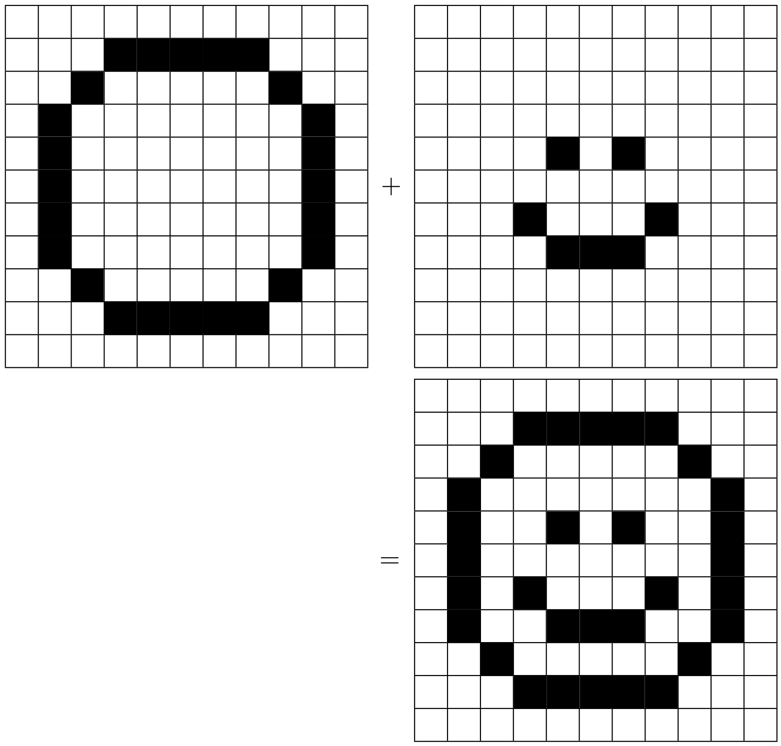
1.3 Facespace

Before the coffee break we thought about various ways to represent an image (e.g., a picture of a face). In this section we're going to narrow in on a particular method of representing images: as a weighted sum of a set of building block images. In this section you'll work through some exercises to scaffold the basic ideas of how this type of representation works and why it is so powerful.

Pixel Arithmetic [10 mins]

Adding is one of the most basic operations in mathematics. While everyone here is familiar with the concept of adding numbers, we can generalize this idea to add together other sorts of entities. We can even think about what it means to add two images together.

As a simple example, let's add the following two images together (we'll explain more precisely how we are defining addition of images once you've seen the result).



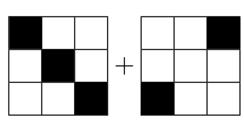
Conceptually, this operation might seem straightforward. Adding two images results in an image that has a black pixel whenever either of the two images has a black pixel at a corresponding position.

More formally, we can think about black pixels as having a value of 255 and white pixels as having a value of 0 (gray pixels would have a value between these two values depending on how dark they are). (A scale from 0 to 255 seems like a weird choice, but there is a very good reason why this is the standard - remember that digital storage uses binary (bit) - how many integers can you represent with an 8-bit number?) To add two images together, all we do is add the corresponding elements at a particular point in the grid! In this way addition on images works much the same as addition of a single number–the only difference is we perform the addition of single numbers multiple times for each position in the grid.

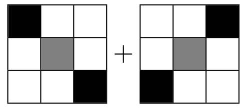
Exercise 1.1

With your tablemates, work through the following pixel arithmetic problems on the board.

1.

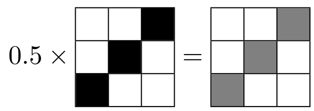


2.



Without too much of a leap, we can also multiply images by a number by simply multiplying each element in the image by that value. We can think of this multiplication operation as “scaling” the image.

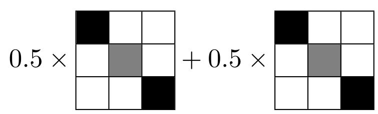
For example,



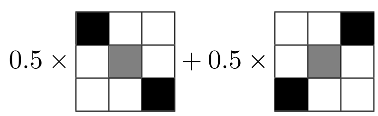
Exercise 1.2

With your tablemates, work through the following pixel arithmetic problems on the board.

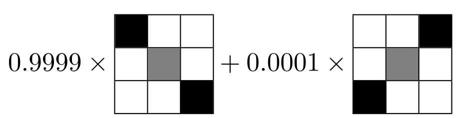
1.



2.



3. (Don’t think about this one too hard. Just draw approximately what this would be)



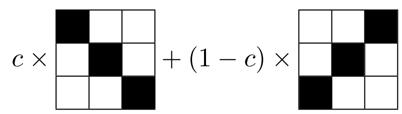
A Universal Set of Building Block Images [10 mins]

Now that we have a sense of how we can add and scale images, let’s think about how we might construct a set of building block images such that we can construct any image as a sum of scaled versions of these building blocks.

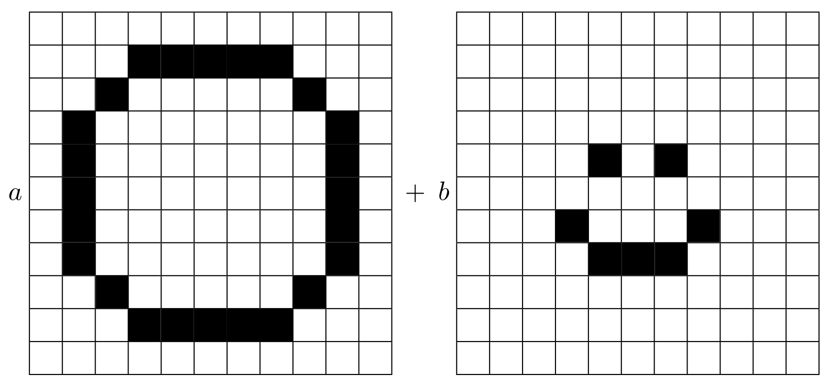
Exercise 1.3

With your tablemates, work through the following problems.

1. What is the range of images that could be constructed by summing over scaled versions of the following building block images? (c is a number between 0 and 1). Another way to think about this is, as you sweep the value of c from 0 to 1, how does the resultant sum of the two images change?



2. What is the range of images that could be constructed by summing over scaled versions of the following building block images? (a and b are both numbers between 0 and 1). Instead of having one knob to turn (as in the previous exercise), you now have two.



In this case we can think of the values a and b as encoding of a particular smiley face. You will deduce the effect that both a and b have on the specific nature of the smiley face.

3. Building on the previous example, come up with your own way of representing a simple face like the one above as the sum of two or more scaled building block images. This is intended to be fun, so be creative! It's up to you what sort of faces that your method is capable of representing.

4. You probably noticed from the previous three exercises that not all possible images can be constructed by adding scaled versions from a small set of building block images. Suppose you wanted to be able to represent *any* possible 3 pixel by 3 pixel image of a face. While there are many possible ways to do this, for simplicity each of your building block images should only have a single black pixel (the rest should be white). At the board, define a set of building block images that lets you represent any possible 3 pixel by 3 pixel face in this manner. How many building block images did you need to represent all possible 3 pixel by 3 pixel faces?

Are there any images that can't be represented as a sum of scaled versions from your building block images? How many building block images would you need if you wanted to encode all possible 5 pixel by 5 pixel faces? What about n pixels by n pixels?

A Better Set of Building Blocks? [20 mins]

At the end of the previous section you showed how can represent any possible image as a sum of scaled single-pixel images. This is a very powerful idea, but we can take it even farther. Before we continue, let's think about some of the ways in which this way of representing face images is not so great.

Exercise 1.4

Suppose you wish to represent 19 pixel by 19 pixel images of faces using the scheme you devised in the previous set of exercises (as a sum of scaled, single-pixel images). Here is an example of what such a face might look like.

1. If you think of the representation of each image as the scaling factor that you apply to each of your single-pixel images, how many numbers do you need to specify this one face image (you answered almost this exact question in the previous part, so don't overthink this).
2. How many numbers would you need to represent a 19 pixel by 19 pixel image of a flower? How many numbers would you need to represent a completely random 19 pixel by 19 images (one with no special structure)?
3. Suppose someone gives you one of the numbers needed to encode a particular face? Without looking at the face image itself, how much information (e.g., age, identity, sex, gender, etc.) could you determine about the face just from that one number?

As you probably deduced in the previous exercise, a major drawback of the encoding we worked out previously is that each scaling factor doesn't really tell us that much useful information about each face (and as a result we need a lot of these numbers to specify a particular face). It turns out that we can fix a lot of these shortcomings through more carefully choosing our set of building block images. Reframing problems by choosing a different set of building blocks is going to be one of the key ideas in this module.

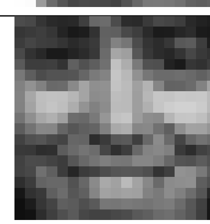
Come to the front of the room and grab a piece of paper with a 6 by 4 grid of face-like images along with a set of transparent face-like images held by a binder clip. Take these materials back to your table. Layout the piece of paper on your table. Also in the envelope you should have a set of transparent versions of those same building block images. Layout the transparent building block images so that they align with the appropriate printed building block image. The bottom building block should go in the upper left corner of the printed sheet. As a sanity check, make sure the textured side of the transparency is facing up (one side will be smooth and the other textured). Be very careful when laying out your images as it is hard to get them back in the right order.

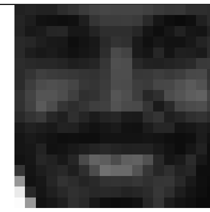
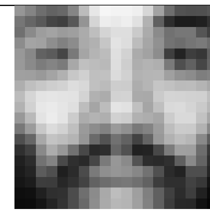
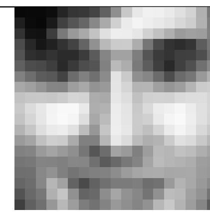
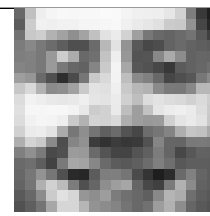
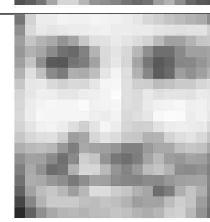
What you see before you is a very carefully chosen set of building block images. You should notice that each row represents a different building block image and each column represents a different scaled version of that same building block. Today, we won't be going into detail about *how* we determined these particular building blocks but we will be having you experiment with them in order to understand, at a conceptual level, some of their properties.

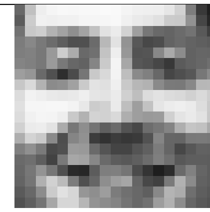
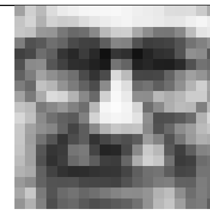
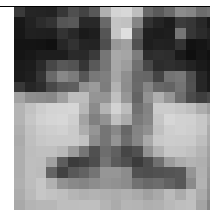
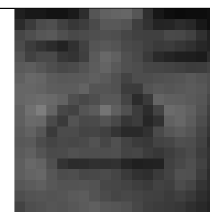
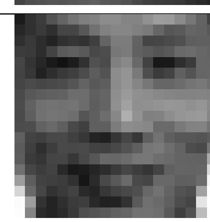
- You can add these scaled building block images by simply stacking multiple transparencies on top of each other and placing them on a white background (make sure to keep them aligned). We've found that using your thumb and index finger and pinching the middle of the transparency is a good way to pick it up (they are pretty sturdy).

- Along with these building block images, we have determined optimal encodings for a bunch of different faces. At your table, pick a few of these faces and try assembling them (you should probably put the transparencies back after assembling each face so you can keep better track of the transparencies).

  Note: that each column in the table corresponds to one of the building block images (row of your transparencies). Higher numbers in the table correspond to choosing the darker (more saturated) versions of each building block image. If a 0 appears for a particular building block, don't include that building block at all to construct a particular face.

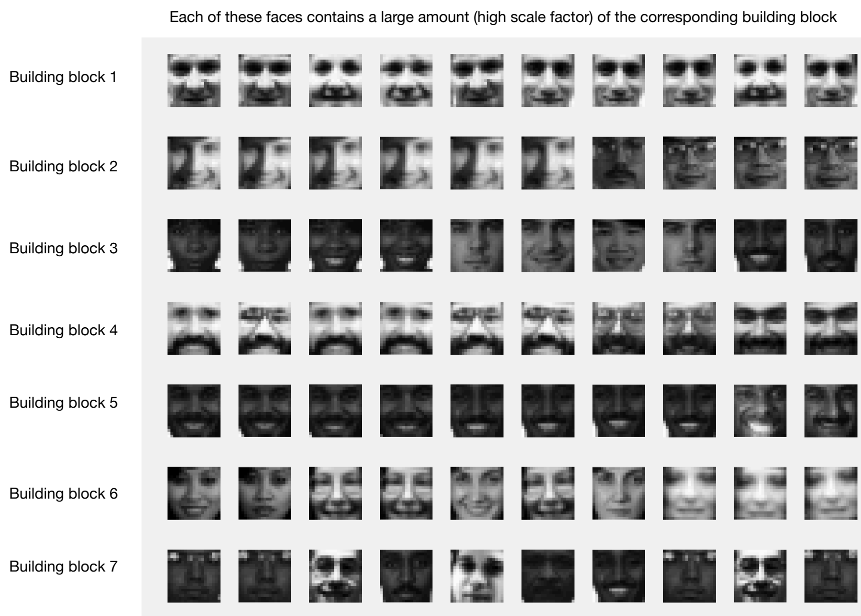
Intensity 1	**Intensity 2**	**Intensity 3**	**Intensity 4**	**Intensity 5**	**Intensity 6**	**face image**
3	3	0	0	2	2	
0	2	3	2	3	1	
3	0	1	4	1	1	

1	4	1	2	3	4	
0	0	3	3	2	1	
2	0	3	0	2	0	
2	0	1	3	0	1	
1	2	0	1	0	0	
1	1	3	0	1	2	

2	0	1	3	0	1	
3	2	0	2	0	1	
2	1	3	0	3	1	
0	3	0	1	3	3	
2	3	0	1	2	1	

- How many numbers do you now need to encode a 19 pixel by 19 pixel face?
- Can you encode any possible face with this set of building blocks?
- How well does this set of building blocks work for encoding these faces? Does it seem to work equally well across all faces? Which faces does it work well on (i.e., they can accurately be reconstructed from the building blocks) and which faces does it work poorly on?
- Looking at the building blocks themselves, what does each building block seem to represent? In other words, as you increases the amount of a particular building block, what features or qualities does that impart on the resulting face. To help you think this through, below we have a grid of faces where

each row corresponds with one of the six building block images and each of the faces in the row contains a large amount of that particular building block image in its encoding.



Towards an Optimal Basis [20 mins]

Exercise 1.5

In this question, we want you to think about process rather than particular techniques for solving this problem. If you have questions on what we mean by this, let us know.

Suppose someone has hired you as a consultant to create a method to encode 19 pixel by19 pixel images of faces (similar to the ones you just experimented with) as a sum of scaled versions of just 10 building block images.

1. What questions would you want to ask the person that hired you in order to do a good job on this project? (i.e., what information do you need to know?)
2. What might be some qualities of a good set of building block images? (e.g., how would they look? what sort of dimensions of variability would they have?)
3. What sort of data might you need to collect in order to inform the set of building blocks you will ultimately deliver (this data could be images or it could be other quantitative or qualitative data)?
4. How might you determine whether your method is working (these could be quantitative measurements or qualitative observations of your system)?

5. Are there any other steps might you want to take to complete the project?

6. We will be digging into the various dimensions of the use of facial recognition technology in society later in this module, but for now we want to get you thinking about two particular components of that. Many face processing technologies work best on white males (e.g., check out the Gender Shades project). One possible explanation for this phenomenon is overt bias on the part of the creators of these technologies. Instead, for the sake of this exercise, let's suppose that the differences in performance are actually the result of subtle, unconscious bias in any number of decisions that the technology creators made during the design process. A second problem that plagues face processing algorithms is that they seem to work great when evaluated in the settings that the technology designers had in mind when they built the technology, but often work poorly when deployed in the real world. Looking back on the steps you listed above, flag steps that might have the potential to introduce bias into your system (e.g., having your system work better on one group of people than another or having it fail in a particular use case). It's okay if you don't know where bias might creep in, the purpose of this exercise is to get you asking questions rather than reaching conclusions.

Chapter 2

Night 1: Introduction to Matrices

Overview and Orientation

In this night assignment, we will learn some of the foundational material about matrices and matrix operations.

Learning Objectives

Concepts

- Define a vector, a matrix and an array
- Describe the meaning of the dimensions of a vector, a matrix, and an array
- Give at least one interpretation of matrix-vector multiplication
- Calculate the product of a matrix-vector multiplication for 2D and 3D matrices
- Understand dimensionality-requirements for matrix-vector multiplication and predict resulting dimensions
- Define and recognize the following special matrices: Identity, diagonal, square, rectangular, symmetric

MATLAB skills

- Determine the dimensions of a vector, matrix, or array variable
- Perform operations (addition, multiplication, transposition) on matrices
- Extract desired subarrays or matrices from arrays

Suggested Approach

- First you should quickly scan through the assignment, see what is being asked, and assess the extent to which you already know how to do things. Spend no more than 30 minutes or so doing this.
- You should then read the assignment more closely, try out problems, and if appropriate, look at some of the other resources that are suggested. Don't spend more than 1 hour poking around at stuff online unless it is really being productive: it's easy to spend a lot of time there without accomplishing much.
- Then start doing the problems in earnest, and/or spend focused time with suggested resources.
- Once you've spent a total of 3-4 hours working on the assignment, you should check your progress. Are you on track to finish within about 7-8 hours? Do you feel confident that you can do the stuff that's left? If not, this is when you should ask for help. This means talk to a colleague, or talk to a ninja, or track down an instructor, or send an email to an instructor.

- You should turn in a PDF document with answers to all the numbered questions below. For the MATLAB assignments, please export your work to pdf. Please carefully label the problem number in your MATLAB script.

Resources to read and watch

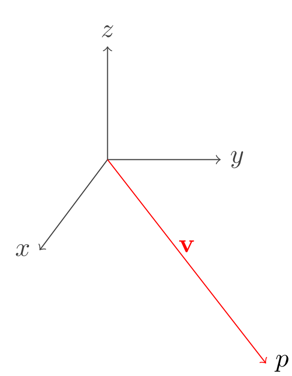
There are lots of books about Linear Algebra and lots of useful videos on the web. Here are some specific recommendations:

- Introduction to Linear Algebra, by Strang
- Linear Algebra, by Lay
- Linear Algebra, by Cherney, Denton, Thomas, Waldron
- Homebrew videos
  - Matrices operating on vectors
  - Matrices operating on vectors (example)
  - Matrices operating on matrices
- Videos from others
  - Vectors, the very basics
  - 3Blue1Brown's YouTube series on Linear Algebra

2.1 Linear Algebra, Vectors, and Matrices

In your concept-maps for eigenfaces, most, if not all of you would have included something about linear algebra, matrices, and vectors. These topics are used very heavily in many different areas, including in data analysis. For the next couple of weeks, you will spend a good deal of time learning about these things and how to apply them.

Linear Algebra and Vectors



Consider the point $p = (1, 2, -1)$ in 3-dimensional space. We can associate a position vector $\mathbf{v}$ with this point, which is the vector from the origin to this point,

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}.$$

Likewise, we can think of every vector as defining a point, if we assume that the vector emanates from the origin. So, for example, the vector

$$\mathbf{v} = \begin{bmatrix} 3 \\ -2 \\ 0 \\ 1 \end{bmatrix}$$

is identified with the point $(3, -2, 0, 1)$ in 4D. Often times we will mix and match these ideas and say things like: the vector (x, y, z). What we really mean when we say this is: the point (x, y, z) can be treated as the position vector

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The vector $\mathbf{v}$, as represented above, is called a column vector. We can also have row vectors such as the following

$$\mathbf{u} = \begin{bmatrix} p & q & r \end{bmatrix}.$$

The operation of converting a column vector to a row vector or vice-versa is called taking the *transpose* of the vector and is denoted with a superscript T. For example, the transpose of the row vector $\mathbf{u}$ from above is

$$\mathbf{u}^T = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.1}$$

and the transpose of the vector $\mathbf{v}$ from above is

$$\mathbf{v}^T = \begin{bmatrix} x & y & z \end{bmatrix}. \tag{2.2}$$

We can take the product of a row vector with a column vector using the following formula

$$\mathbf{uv} = \begin{bmatrix} p & q & r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = px + qy + zr \tag{2.3}$$

If we start with two column vectors

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \text{ and } \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

of length n (i.e., they are n=dimensional), then we can take the *d*ot product

$$\mathbf{v} \cdot \mathbf{w} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n.$$

In some sense, the dot product is a measure of how aligned two vectors are. Here's the key formula:

$$\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\| \|\mathbf{w}\| \cos\theta$$

where θ is the angle between $\mathbf{v}$ and $\mathbf{w}$ and

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}^{1/2}$$

is the length of the vector $\mathbf{v}$ in n-dimensional space.

Exercise 2.1

1. Assume $\mathbf{v}$ and $\mathbf{w}$ are two vectors of unit length, i.e., $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$. Using the formula above, what angle between $\mathbf{v}$ and $\mathbf{w}$ maximizes the dot product? Using the formula above, what angle between $\mathbf{v}$ and $\mathbf{w}$ minimizes the dot product?

2. Compute $\mathbf{v} \cdot \mathbf{w}$ where

$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ -4 \\ 6 \end{bmatrix}, \text{ and } \mathbf{w} = \begin{bmatrix} -2 \\ 0 \\ 1 \\ 3 \end{bmatrix}$$

We'll learn more about the dot product as we go. For now, notice that the dot product equals the product of the transpose of one with the other

$$\mathbf{v} \cdot \mathbf{w} = \mathbf{v}^T \mathbf{w}. \tag{2.4}$$

Vectors can also be used to represent many things, such as data. Linear algebra provides a powerful set of tools to manipulate and analyze this data.

Exercise 2.2

For instance, you may have a three-dimensional vector $\mathbf{f}$ whose entries represent the numbers of different fruits you have in your refrigerator. For example, the first entry could be the number of oranges, the second the number of grapefruits and the third could be the number of apples. When organized in this manner, you can use products of row and column vectors to compute the number of different fruits there are. For instance, suppose that

$$\mathbf{f} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \tag{2.5}$$

i.e. you have 1 orange, 2 grapefruits, and 3 apples in your fridge.

1. Find a row vector $\mathbf{t}$ so that the product $\mathbf{tf}$ tells you the total number of fruits in your refrigerator.

2. Find a row vector $\mathbf{c}$ such that the product $\mathbf{cf}$ tells you the total number of *citrus* fruits in your refrigerator.

3. Suppose that in the genetically engineered future, all apples weigh 100 g, all grapefruits weigh 250 g and all oranges weigh 120 g. Find a row vector $\mathbf{w}$, such that the product $\mathbf{wf}$ tells you the total weight of fruits in your refrigerator.

If you wanted to know the vitamin C content of the fruits in your fridge, you could formulate a similar vector to compute it.

In the questions above, you took *linear combinations* of the entries of the vector $\mathbf{f}$ which gave you the desired quantity. *Linear algebra is the study of linear functions.*

Introduction to matrices

Matrices are a set of numbers organized in a two-dimensional array. Matrices are a compact way to represent linear combinations. Matrices can also be used in a number of different ways, such as to represent data. When we multiply a matrix by a vector, it results in a new vector. Therefore, when we say "a matrix operates on a vector", we mean that the matrix multiplies the vector. Notation-wise, we use bold upper-case letters, e.g. $\mathbf{A}$, to represent a matrix and bold lower-case letters to represent a vector, e.g. $\mathbf{v}$.

For instance, you may define a two-dimensional matrix $\mathbf{G}$ with two rows and three columns as follows

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{2.6}$$

Matrices and vectors come in different shapes and sizes and we refer to their shape and size by the number of rows and columns they have. A general matrix $\mathbf{A}$ has m rows and n columns, and we refer to this as an $m \times n$ matrix. Vectors are then examples of matrices: row vectors have a single row, i.e., they are $1 \times n$ matrices; and column vectors have a single column, i.e., they are $m \times 1$ matrices.

Matrices can only multiply vectors of a certain size and produce vectors of a certain size: an $m \times n$ matrix can only operate on a column vector of size $n \times 1$, and will produce an output vector which is a column vector of size $m \times 1$. (Likewise, matrices can only multiply other matrices of a certain size: an $m \times n$ matrix can only act on a matrix of size $n \times k$, and will produce an output matrix of size $m \times k$.) These basic properties will become clearer when we look at an example.

Consider the 3×2 matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \\ 0 & 4 \end{bmatrix}$$

and the input vector $\mathbf{v}$

$$\mathbf{v} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}.$$

The output vector $\mathbf{w}$ is computed as follows

$$\mathbf{w} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} (2)(-2) + (1)(1) \\ (3)(-2) + (-1)(1) \\ (0)(-2) + (4)(1) \end{bmatrix} = \begin{bmatrix} -3 \\ -7 \\ 4 \end{bmatrix}$$

There are two main ways to think about this multiplication. The most common view is to treat each entry of the new vector as a dot product between a row of the matrix and the column vector. So, for example, the first entry in the output vector is the dot product of two vectors

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -2 \\ 1 \end{bmatrix} = -3$$

The second approach is to view the output vector as a linear combination of the columns of the matrix. The entries in the original vector are used as multiplication weights on each column of the matrix, i.e.

$$(-2) \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} + (1) \begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix} = \begin{bmatrix} -3 \\ -7 \\ 4 \end{bmatrix}$$

We encourage you to use both approaches when you think about multiplication.

Exercise 2.3

Recall the matrix $\mathbf{G}$ defined in equation (2.6) and the vector $\mathbf{f}$ defined in Exercise 2.2, which kept track of the number of fruit of different types. What does the vector $\mathbf{Gf}$ represent?

Exercise 2.4

If a matrix multiplies a spatial vector, the resulting vector is *transformed* by the matrix, resulting in a new vector.

1. Please draw the spatial vector

$$\mathbf{v} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.7}$$

2. Please draw the vector $\mathbf{w} = \mathbf{A}\mathbf{v}$, where $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.8}$$

3. What happened to $\mathbf{v}$ when you multiplied by $\mathbf{A}$?
4. Please draw the vector $\mathbf{u} = \mathbf{B}\mathbf{v}$, where $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{bmatrix} \tag{2.9}$$

5. What happened to $\mathbf{v}$ when you multiplied by $\mathbf{B}$?
6. Please draw the vector $\mathbf{t} = \mathbf{R}\mathbf{v}$, where $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.10}$$

7. What happened to $\mathbf{v}$ when you multiplied by $\mathbf{R}$?
8. Please draw a new spatial vector

$$\mathbf{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{2.11}$$

9. Please draw the vector $\mathbf{s} = \mathbf{R}\mathbf{w}$
10. What does multiplying *any* vector by $\mathbf{R}$ do?

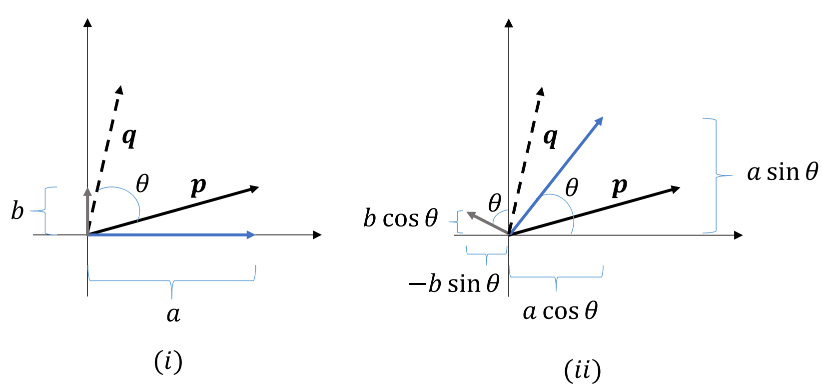


Figure 2.1: Rotation of vectors

You may have guessed that $\mathbf{R}$ defined above, rotates a vector counter-clockwise by θ. This is indeed true, and $\mathbf{R}$ is called a *rotation matrix* as it transforms vectors by rotating them. To understand why $\mathbf{R}$ is a rotation matrix, consider Figure 2.1 (i). Suppose that we wish to rotate the vector $\mathbf{p}$ counter-clockwise by θ, which will result in the vector $\mathbf{q}$. From the figure, we see that

$$\mathbf{p} = \begin{bmatrix} a \\ b \end{bmatrix}, \tag{2.12}$$

and $\mathbf{p}$ is the sum of the gray and blue vectors. If we now rotate the blue and gray vectors counter-clockwise by θ, we see that $\mathbf{q}$ is the sum of the rotated versions of the blue and gray vectors, as shown in Figure 2.1 (ii). By using trigonometry, we see that the blue vector in Figure 2.1 (ii) is

$$\begin{pmatrix} a\cos\theta \\ a\sin\theta \end{pmatrix} \tag{2.13}$$

and the gray vector in Figure 2.1 (ii) is

$$\begin{pmatrix} -b\sin\theta \\ b\cos\theta \end{pmatrix} \tag{2.14}$$

Therefore, $\mathbf{q}$ is given by

$$\mathbf{q} = \begin{pmatrix} a\cos\theta \\ a\sin\theta \end{pmatrix} + \begin{pmatrix} -b\sin\theta \\ b\cos\theta \end{pmatrix} = + \begin{pmatrix} a\cos\theta - b\sin\theta \\ a\sin\theta + b\cos\theta \end{pmatrix} = \mathbf{R}\mathbf{p}. \tag{2.15}$$

General Notation

As we mentioned earlier, $m \times n$ matrices can multiply $n \times 1$ vectors and produce $m \times 1$ vectors. Consider a generic $m \times n$ matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

where the ij-th entry of this matrix, a_{ij} defined above, is the entry corresponding to the i-th row and j-th column. You can multiply an $n \times 1$ vector $\mathbf{v}$ by this matrix. Define the vector $\mathbf{v}$ as follows,

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

Now define another vector $\mathbf{w}$ which is the product of $\mathbf{A}$ and $\mathbf{v}$, i.e., $\mathbf{w} = \mathbf{A}\mathbf{v}$. If we define

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

then the i-th entry of $\mathbf{w}$, is given by the following sum

$$w_i = a_{i1}v_1 + a_{i2}v_2 \cdots a_{im}v_m = \sum_{j=1}^{n} a_{ij}v_j$$

Other Matrix operations

Besides multiplication, a number of other operations can be done using matrices including addition, subtraction, inversion, transposition, etc. We will explore more of these and their associated properties in the next section. All of these operations make matrices a very powerful tool in the study of many different systems which can be represented as linear transformations, or combinations.

2.2 Matrix Operations in MATLAB

Exercise 2.5

In the command window, you can type in commands and press enter. Try the following commands and see what they do.

```
1+1
a=1+1
a
% you can start a comment with "%"
b=2;% this will appear as a variable in your workspace, but the semicolon ...
    suppresses the output
c=3,d=4,e=5;% use commas, semicolons, or shift+enter between commands that you ...
    want to execute together
1+2-(3*4/5)^6
clear a
a% should give you an error because a is not defined anymore
clear all
clc% only if you want to clear your workspace!
```

To practice matrix operations, let's define a matrix and some vectors using MATLAB as follows:

```
>> A = [2 1; 3 -1; 0 4]
```

Note that the semi-colon ends a row and begins a new row. You can also use returns between rows—try it! Square brackets enclose the matrix. To define the column vector $\mathbf{v}$ in MATLAB you can type the following command:

```
>>  v = [-2; 1]
```

whilst to define the row vector $\mathbf{u}$ in MATLAB you can type the following command

```
>>  u = [2 -3 1]
```

Notice that in this case each component of the vector is separated by a space - you could also separate them with a comma.

Exercise 2.6

Using the definitions for **A**, **v**, and **u** from above, please predict the output of the following commands and then solve them using MATLAB.

```
A*v
u*A
A(1:2,:)*v
u*A(:,2)
```

For Night 1, you will also need to plot things. In MATLAB, you can use plot(xv,yv) to create a scatter plot.

```
yv=[1 7 4 5 3 9 2 4]
xv=[1 3 4 6 8 9 11 14]
plot(xv,yv)
```

If you want to know more about how to use a function like plot, use "help." Create the plot above, then type "help plot" into the command window and try to change something about your plot, such as using points instead of a line or adding axis labels.

Finally, you need to know how to use a for loop to repeat a set of commands a number of times. Here's an example for loop that makes a vector that's a sequence of squares: (Try it!)

```
for n=1:3% n is the index variable, which counts from 1 to 3 (call it whatever you want)
v(n)=n^2% assigns the nth component of v to the value n^2 and prints out v
% loop repeats, adding 1 to n each time, until i gets to 3
end% needed to end the loop!
```

Exercise 2.7

Write a for loop that creates the following matrix:

```
M_squares=[1 1;2 4;3 9;4 16]
```

2.3 Elementary Matrix Operations, Properties, and Terminology

In this part of the assignment, you will learn a number of basic operations and properties of matrices which can then be used in applications. Admittedly, most of these exercises are a little dry, but they will be useful in the very near future, we promise!

Matrix-Vector Multiply

Here, you will work on examples of matrices multiplying vectors to get yourselves comfortable with matrix operations in MATLAB. First, let's define the matrix **A** using MATLAB as follows

```
>> A = [2 1; 3 -1; 0 4]
```

Note that the semi-colon ends a row and begins a new row. To define the column vector **v** in MATLAB you can type the following command:

```
>>  v = [-2; 1]
```

whilst to define the row vector **u** in MATLAB you can type the following command

```
>>  u = [2 -3 1]
```

Notice that in this case each component of the vector is separated by a space - you could also separate them with a comma.

Exercise 2.8

Using the definitions for **A**, **v**, and **u** from above, please solve the following using MATLAB. Do the answers match what you expect? (Not all of these may be defined!)

1. `A*v`
2. `u*A`
3. `A*u`
4. `v*A`
5. `A(1:2,:)*v`
6. `u*A(:,2)`
7. `A(:, 2:4)*v`
8. `u*A(1,:)`

Addition, subtraction, scalar multiplication and transpose of matrices

We can add matrices of the same size, and subtract them from one another. Both operations result in matrices of the same size and shape. The addition and subtraction operations are done element-wise. For instance the difference of the two matrices can be calculated as below

$$\mathbf{A} = \begin{bmatrix} 3 & 4 & 1 \\ 3 & 1 & 1 \end{bmatrix} \tag{2.16}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \end{bmatrix} \tag{2.17}$$

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} (3-1) & (4-2) & (1-3) \\ (3-2) & (2-1) & (1-1) \end{bmatrix} \tag{2.18}$$

$$= \begin{bmatrix} 2 & 2 & -2 \\ 1 & -1 & 0 \end{bmatrix} \tag{2.19}$$

Multiplying a matrix by a scalar simply scales each entry of the matrix by the scale factor. For instance

$$3\mathbf{A} = \begin{bmatrix} 9 & 12 & 3 \\ 9 & 3 & 3 \end{bmatrix} \tag{2.20}$$

The transpose of a vector, denoted by the superscript ^{T} turns a column vector into a row vector, and vice versa. For matrices, the transpose replaces the rows with the columns (or vice-versa). For example,

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 7 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 \\ 3 & 7 \\ 5 & 6 \end{bmatrix} \tag{2.21}$$

Since the columns are replaced with the rows, the shape of the matrix changes when you transpose it. The following property of transposes will be useful moving forward. Consider a matrix $\mathbf{A}$ and a vector $\mathbf{v}$. Then

$$(\mathbf{A}\mathbf{v})^T = \mathbf{v}^T\mathbf{A}^T \tag{2.22}$$

Exercise 2.9

Using $\mathbf{A}$ and $\mathbf{B}$ previously defined, evaluate $4\mathbf{A} - 5\mathbf{B}$

Exercise 2.10

If the matrix $\mathbf{A}$ has dimensions of 4×5, what are the dimensions of $\mathbf{A}^T$?

Exercise 2.11

If the matrix $\mathbf{A}$ is 4×5 (i.e., $\mathbf{A}$ has dimensions 4×5) and the vector $\mathbf{v}$ is 5×1, what are the dimensions of $\mathbf{Av}$ and $(\mathbf{Av})^T$?

Exercise 2.12

How do you find the transpose of a vector or matrix in MATLAB?

Matrix-Matrix Multiply

Matrices can be multiplied together to produce other matrices. In general, when you multiply a matrix $\mathbf{A}$ with another matrix $\mathbf{B}$, you need the matrix on the left side of the product to have the same number of columns as the number of rows in the matrix on the right side. In other words if $\mathbf{A}$ is $m \times n$, and $\mathbf{B}$ is $p \times q$, you need $n = p$ for the product $\mathbf{C} = \mathbf{AB}$ to be defined. The product results in a new matrix $\mathbf{C}$ which is $m \times q$. The q columns of the product matrix $\mathbf{C}$ are precisely the q vectors that would result from multiplying $\mathbf{A}$ with the vectors formed by the columns of $\mathbf{B}$.

Consider the following matrices

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 5 \\ -2 & 3 \end{bmatrix}.$$

The product of the two $\mathbf{C} = \mathbf{AB}$ is computed as follows

$$\mathbf{C} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} (2)(1) + (1)(-2) & (2)(5) + (1)(3) \\ (3)(1) + (-1)(-2) & (3)(5) + (-1)(3) \end{bmatrix} = \begin{bmatrix} 0 & 13 \\ 5 & 12 \end{bmatrix}$$

As a second example consider the matrices $\mathbf{A}$ and $\mathbf{B}$ defined below, and let the product $\mathbf{C} = \mathbf{AB}$.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 2 \\ 4 & 1 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} (1)(1) + (2)(2) & (1)(4) + (2)(3) \\ (3)(1) + (2)(2) & (3)(4) + (2)(3) \\ (4)(1) + (1)(2) & (4)(4) + (1)(3) \end{bmatrix} = \begin{bmatrix} 5 & 10 \\ 7 & 18 \\ 6 & 19 \end{bmatrix}$$

As mentioned above, one way of envisioning matrix multiplication is if we consider the columns of input matrix $\mathbf{B}$ as a set of column vectors, we can multiply these column vectors one at a time by the matrix $\mathbf{A}$, and the resulting vectors will be the corresponding columns of the output matrix $\mathbf{C}$, i.e.

$$\mathbf{AB} = \mathbf{A}[\mathbf{B}_1, \mathbf{B}_2, \ldots] = [\mathbf{AB}_1, \mathbf{AB}_2, \ldots]$$

where $\mathbf{B}_1$ is the first column of matrix $\mathbf{B}$ etc.

Consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix}$$

Exercise 2.13

Find the matrix product $\mathbf{AB}$.

Exercise 2.14

Find the matrix product $\mathbf{BA}$

Note that these two products are NOT equal. In general, matrix multiplication, unlike scalar multiplication, is NOT commutative. In other words, in general $\mathbf{AB} \neq \mathbf{BA}$. However, the distributive property IS valid for matrices: $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}$ so long as we keep the order of the multiplication the same $(\mathbf{B}+\mathbf{C})\mathbf{A} = \mathbf{BA}+\mathbf{CA}$. Recall the definition of matrix addition: if two matrices are of the same size then they can be added and each entry of the new matrix is the sum of the entries of the original matrices, e.g.

$$\begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix} + \begin{bmatrix} 4 & -2 \\ -3 & -1 \end{bmatrix} = \begin{bmatrix} 9 & -5 \\ -4 & -2 \end{bmatrix}$$

In addition to matrices $\mathbf{A}$ and $\mathbf{B}$ defined above, consider the matrix

$$\mathbf{C} = \begin{bmatrix} -5 & -1 \\ -3 & 2 \end{bmatrix}$$

Exercise 2.15

Calculate $\mathbf{A}(\mathbf{B}+\mathbf{C})$.

Exercise 2.16

Calculate $\mathbf{AB} + \mathbf{AC}$. Is it equal to your previous answer?

Finally, since matrix multiplication is defined, there is no reason not to multiply a matrix by itself. This only works if it is a square matrix. (Think about why this is true.) Using $\mathbf{A}$ and $\mathbf{B}$ from above, evaluate the following expressions

Exercise 2.17

1. $\mathbf{A}^2$
2. $\mathbf{B}^3$

Special Types of Matrices

Exercise 2.18

There are lots of matrices that are special. Use a trusted linear algebra reference to define the following types of matrices, and provide an example of each:

1. Square Matrix
2. Rectangular Matrix
3. Diagonal Matrix
4. Identity Matrix
5. Symmetric Matrix

2.4 *Matrices as transformation operators*

When matrices operate on (i.e., multiply) spatial position vectors, the vector which results is another spatial position vector. The original spatial position has been 'transformed' into another position. In particular, there are specific matrices which accomplish specific desired transformations. These are used in many different disciplines.

Identity and Scaling Operations

The matrix

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

when multiplying the vector

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.25}$$

will reproduce the same vector, i.e. $\mathbf{I}\mathbf{v} = \mathbf{v}$. For this reason, the matrix $\mathbf{I}$ above is called an identity matrix. Identity matrices in higher dimensions are defined the same way, i.e., a 4-dimensional identity matrix is a 4×4 matrix with 1s on the diagonal and zeros everywhere else, i.e.,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.26}$$

Exercise 2.19

1. Another important and simple operation is to be able to take a vector and scale (increase or decrease its length) it by an overall multiplicative factor while maintaining its direction. Consider the vector

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Thinking about how the identity matrix acts on this vector, propose a 3×3 matrix which scales this vector by a factor of 3 to the vector

$$3\mathbf{v} = \begin{bmatrix} 3x \\ 3y \\ 3z \end{bmatrix}.$$

In other words, find a 3×3 matrix $\mathbf{M}$ such that $\mathbf{M}\mathbf{v} = 3\mathbf{v}$ for any vector $\mathbf{v}$.

2. What if you want to scale the x component differently than the y component? Write down the 3×3 matrix which scales the x component by 3 and the y component by 5 and leaves the z component the same.

3. Write down the 3×3 matrix which scales the x component by a, the y component by b, and the z component by c.

2.5 *Data in Matrices and Vectors*

Most of the examples you saw up to now in this assignment involved vectors which represent spatial positions, and most of the matrices you encountered represent transformations of the spatial vectors. But, as you saw with the example involving fruits, vectors can also be used to store data. So can matrices.

For instance, you may have the following matrix

$$\begin{bmatrix} 41 & 35 & 37 & 43 \\ 49 & 40 & 48 & 61 \end{bmatrix} \tag{2.27}$$

whose first row represents the forecasted high temperature in Needham for the next 4 days (as of the day this was written) and the second row represents the forecasted high temperatures for Washington DC. By representing this data in matrix form, you can do a number of operations to help extract useful information from the data.

Exercise 2.20

For this exercise, you will work with historical temperature data for the cities of Boston, Providence, Washington DC and New York.

1. Download the file `temps.mat` from canvas and load the data in it into MATLAB using `» load temps.mat`. You should now have access to a matrix `T` which contains daily average temperatures from 1995 to 2015 for the cities of Boston, Providence, Washington DC and New York (we are not telling you in what order yet). By using MATLAB's `size` function, determine the dimensions of this matrix. Are the temperatures for each city contained in the rows or the columns of this matrix?

2. The data provided is given in Fahrenheit, and suppose you wish to convert it to Celsius using matrix operations (note that there are a number of ways of doing this, but we are focusing on using matrices here). The formula for converting a Fahrenheit temperature to Celsius is to first subtract 32 from the Fahrenheit temperature, and multiply the result by $\frac{5}{9}$.

   a) Define a matrix of the same shape as `T` with all its entries equalling 32, and call this matrix B. You will find MATLAB's `ones` function, which generates a matrix filled with 1's, useful here.
   b) Define a square, diagonal matrix of the appropriate dimensions which when multiplying another matrix scales all its entries by $\frac{5}{9}$. You should call this matrix A. You will find MATLAB's `eye` function, which generates an identity matrix, useful here.

c) Using your answers to the previous parts and appropriate matrix operations, please provide 1 line of MATLAB code which generates a new matrix Y which contains the temperature data in Celsius.

3. Lets go back to Fahrenheit for the rest of the assignment. Extract the temperatures for each city into 4 different vectors `t1, t2, t3, t4`, and check that the dimensions of these vectors are as expected.

4. Using MATLAB's mean function, which computes the average of the values in a vector, and guess, based on geography, which of the vectors corresponds to the temperature for which city.

5. What are the maximum and minimum temperatures for Boston in the 20 years for which you have data?

6. On the same axes, plot graphs for the daily temperatures for the four cities for the last year for which you have data. Use MATLAB's `legend, xlabel, ylabel` functions to label the graphs.

7. Suppose that a genie told you that you can guess the temperature of New York, which we call T_n, using the temperatures of Boston, Providence, and Washington DC, which we respectively call T_b, T_p and T_w. From the matrix T, extract a 3×365 matrix of daily temperatures for the last year (for which you have data) in Boston, Providence and Washington DC.

8. The genie says that a good approximation for the temperature on a given day in New York is given by

$$T_n \approx 0.2235 T_b + 0.4193 T_p + 0.3856 T_w. \tag{2.28}$$

Formulate a matrix equation which uses the matrix from the previous part and the formula from the genie to guess the daily temperature in New York for the last year. Apply this equation in MATLAB.

9. On the same axes, plot your prediction for the temperature in New York from the previous part, and the true temperature data which you extract from T. Is the prediction close?

In the course of this module, you will learn how to come up with the coefficients we provided here using historical data. (No, we don't actually have a genie.)

2.6 *Conceptual Quiz*

Please figure out the answer to these questions and mark your answer in Canvas. You can retake the quiz, as needed.

1. $\mathbf{A}$ is a 3×4 matrix and $\mathbf{B}$ is a 4×2 matrix. What is the size of $\mathbf{AB}$?

A. 2×3
B. 3×1
C. 3×2
D. The product is not defined.

2. What is the result of the following matrix product

$$\begin{bmatrix} 1 & 2 & -3 \\ 2 & -2 & 4 \\ 3 & -4 & 5 \end{bmatrix} \begin{bmatrix} 4 & 2 \\ -3 & -3 \\ 1 & 1 \end{bmatrix}$$

A.
$$\begin{bmatrix} -5 & -7 \\ 18 & 14 \\ 24 & 23 \end{bmatrix}$$

B.
$$\begin{bmatrix} -5 & -7 \\ 18 & 14 \\ 29 & 23 \end{bmatrix}$$

C.
$$\begin{bmatrix} -5 & 18 & 24 \\ -7 & 14 & 23 \end{bmatrix}$$

D.
$$\begin{bmatrix} -5 & -7 & 3 \\ 18 & 14 & 6 \\ 24 & 23 & 9 \end{bmatrix}$$

3. Match the following items (* means any number):

1. Rectangular Matrix	A. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
2. Diagonal Matrix	B. $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$
3. Identity Matrix	C. $\begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix}$
4. Symmetric Matrix	D. $\begin{bmatrix} * & * \\ * & * \\ * & * \end{bmatrix}$

4. Which of the following matrices will scale the length of any 2-D vector by $\frac{1}{2}$?

A.

$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

B.

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

C.

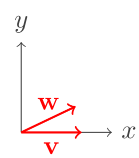
$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$$

D.

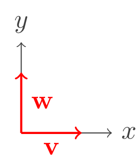
$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

5. All of the following vectors are unit length. In which picture is $\mathbf{v} \cdot \mathbf{w}$ the largest?

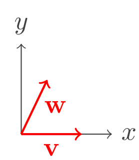
A.



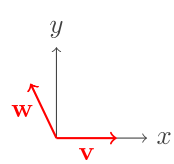
B.



C.



D.



Solution 2.1

1. Using the formula, $\mathbf{v} \cdot \mathbf{w} = \cos(\theta)$. So, when $\theta = 0$, (i.e., the vectors point in the same direction) the dot product is maximized and when $\theta = \pi/2$ (i.e., the vectors are perpendicular) the dot product is minimized.

2. The dot product is
$$\mathbf{v} \cdot \mathbf{w} = -2 + 0 - 4 + 18 = 12$$

Solution 2.2

1. Let $\mathbf{t} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. Then $\mathbf{tf} = 1 + 2 + 3 = 6$, the total number of fruits in your refrigerator.

2. Let $\mathbf{c} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$. Then $\mathbf{cf} = 1 + 2 + 0 = 3$, the total number of citrus fruits in your refrigerator.

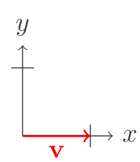
3. Let $\mathbf{w} = \begin{bmatrix} 120 & 250 & 100 \end{bmatrix}$. Then $\mathbf{wf} = 120 + 500 + 300 = 920$, the total weight of the fruits in your refrigerator.

Solution 2.3

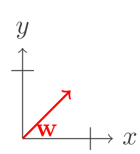
The vector $\mathbf{Gf}$ is a 2×1 vector whose first entry represents the total number of fruits and second entry represents the number of citrus fruits.

Solution 2.4

1. The vector $\mathbf{v}$ is

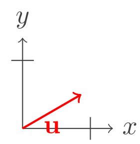


2. First, we compute
$$\mathbf{w} = \mathbf{A}\mathbf{v} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix},$$
which is visually represented as

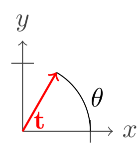


3. Multiplying $\mathbf{v}$ by $\mathbf{A}$ rotated the vector counterclockwise by 45 degrees.

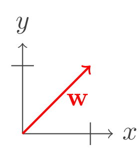
4. First we compute
$$\mathbf{u} = \mathbf{B}\mathbf{v} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}$$
which is visually represented as



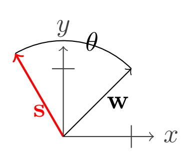
5. Multiplying $\mathbf{v}$ by $\mathbf{B}$ rotated the vector counterclockwise by 30 degrees.
6. First we compute
$$\mathbf{t} = \mathbf{R}\mathbf{v} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$
which is visually represented as



7. Multiplying $\mathbf{v}$ by $\mathbf{R}$ rotated the vector counterclockwise by θ degrees.
8. The vector $\mathbf{w}$ is



9. First we compute
$$\mathbf{s} = \mathbf{R}\mathbf{w} = \begin{bmatrix} \cos\theta - \sin\theta \\ \sin\theta + \cos\theta \end{bmatrix}$$
which is visually represented



10. Multiplying any vector by $\mathbf{R}$ rotates it by θ.

Solution 2.7

```
for n=1:4
M_squares(n,1) = n; %assigns the nth row, 1st column the value n
M_squares(n,2) = n^2; %assigns the nth row, 2nd column the value n^2
end
```

Solution 2.8

1. `[-3; -7; 4]`
2. `[-5 9]`
3. Does not work because the inner matrix dimensions must agree and here we have a 3×2 matrix multiplied by a 1×3 matrix
4. Does not work because the inner matrix dimensions must agree and here we have a 2×1 matrix multiplied by a 3×2 matrix
5. `[-3; -7]`
6. 9
7. Does not work because the index exceeds matrix dimensions. It is trying to access columns 2-4 of a two column matrix.
8. Does not work because the inner matrix dimensions must agree and here we have a 1×3 matrix multiplied by a 1×2 matrix.

Solution 2.9

$$4\mathbf{A} - 5\mathbf{B} = \begin{bmatrix} 7 & 6 & -11 \\ 2 & -6 & -1 \end{bmatrix} \tag{2.23}$$

Solution 2.10

The dimensions of $\mathbf{A}^T$ are 5×4.

Solution 2.11

$\mathbf{Av}$ is 4×1 and $(\mathbf{Av})^T$ is 1×4.

Solution 2.12

You use the apostrophe: $(\mathbf{A})^T$ is `A'` in Matlab.

Solution 2.13

$$\mathbf{AB} = \begin{bmatrix} -14 & 2 \\ -3 & -3 \end{bmatrix}$$

Solution 2.14

$$\mathbf{BA} = \begin{bmatrix} -10 & 11 \\ 2 & -7 \end{bmatrix}$$

Solution 2.15

$$\mathbf{A}(\mathbf{B}+\mathbf{C}) = \begin{bmatrix} -16 & 12 \\ -12 & 3 \end{bmatrix}$$

Solution 2.16

It is the same answer, as expected, since you can distribute matrices.

Solution 2.17

1. $$\mathbf{A}^2 = \begin{bmatrix} 4 & 4 \\ 0 & 9 \end{bmatrix}$$

2. $$B^3 = \begin{bmatrix} 152 & -72 \\ -24 & 8 \end{bmatrix}$$

Solution 2.18

1. A square matrix is one that has size $n \times n$, e.g.,

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}.$$

2. A rectangular matrix is one that has size $m \times n$ where n is not equal to m, e.g.,

$$\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix}.$$

3. A diagonal matrix is one whose only non-zero elements are on the diagonal from upper left to lower right, e.g.,

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 8 \end{bmatrix}.$$

4. The identity matrix is a square matrix with all zeroes except along the diagonal from the upper left to lower right, where the entries are all 1, e.g.,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

5. A matrix is symmetric if it is square and equal to its own transpose, i.e. $A = A^T$, e.g.,

$$\begin{bmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix}.$$

Solution 2.19

1.
$$\mathbf{M} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

2.
$$\mathbf{M} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. We can generalize the result:
$$\mathbf{M} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

Solution 2.20

1. After loading the temperatures you can see that they are stored inside a matrix called T which has 4 rows and 7670 columns, so presumably the temperature for each city is stored in a row.

2. a) We can define a matrix B of the same size as T but filled with 1s by typing » `B = ones(4,7670)` and then we can multiply it by 32 by typing » `B = 32*B`. Alternatively we could include the 32 from the start by typing » `B = 32*ones(4,7670)`

   b) There are two ways to do this: we can multiply either on the left or right.

      - Option 1: To multiply a 4×7670 matrix on the left, we need a 4×4 matrix, which we would create by typing » `A = eyes(4)*5/9`.
      - Option 2: To multiply a 4×7670 matrix on the left, we need a 7670×7670 matrix, which we would create by typing » `A = eyes(7670)*5/9`.

c) Now that we have the pieces in place we could simply type » `Y = A*(T-B)` (or `Y = (T-B)*A` if you chose the right multiply option). We can actually make this a lot simpler because MATLAB will create matrices of the correct size on the fly, so the following will work » `Y = (T-32)*5/9`. Normally we would expect `T-32` to be a problem because we are subtracting a scalar from a matrix, but MATLAB simply assumes that we wish to subtract 32 from every element in the matrix.

3. We can extract the first temperature by typing the following » `t1 = T(1,:)` - this simply grabs all of the elements in the first row, so that `t1` should be a row vector of size 1 by 7670. We create the other vectors in a similar way.

4. We can take the mean of the first city by typing » `mean(t1)` and we get 51.7667. The other means respectively are 51.9140, 58.4365, and 55.9451. A little bit of geography suggests that the cities are ordered as follows: Boston, Providence, DC, New York.

5. We can compute the maximum by typing » `max(t1)` and we get 90.7. The minimum is 0.7.

6. We are only supposed to grab the last year (365 days) so for Boston we would type » `plot(t1(end-364:end))`, or we could use the actual size of the vector.

7. Boston, Providence, and DC are stored in the first three rows. We'll extract their data and store it in a new matrix S by typing » `S = T(1:3,end-364:end)`, which grabs the first three rows and the last 365 entries.

8. We can define Tn by typing » `Tn = 0.2235*S(1,:) + 0.4193*S(2,:) + 0.3856*S(3,:)` since the city temperatures are stored in the each of the three rows of the matrix S.

9. Graphically they look pretty good. We can also examine the data a little more closely by looking at the difference between the predicted temperature and the actual temperature - it fluctuates with a mean of roughly 8.8115e-04, a maximum of 7.5334, and a minimum of -6.8966. Compared to the actual temperatures this implies that the prediction is never any worse than roughly 10%.

Chapter 3

Day 2: Matrix Transformations

3.1 *Schedule*

- 0900-0915: Debrief
- 0915-0945: Synthesis
- 0945-1030: 2D Rotations
- 1030-1045: Coffee
- 1045-1130: 3D Rotations
- 1130-1200: Reflections and Shearing
- 1200-1220: Review and Preview
- 1220-1230: Survey

3.2 *Debrief*

- With your table-mates, identify a list of key concepts/take home messages/things you learned in the assignment. Try to group them in categories like "Concepts", "Technical Details", "Matlab", etc.
- Try to resolve your confusions with your table-mates and by talking to an instructor.

3.3 *Synthesis*

Exercise 3.1

These are fundamental ideas about matrices and it is important to complete these. They should be done by hand.

1. What is the difference between a scalar, a vector, a matrix, and an array?
2. What are the rules for adding matrices?
3. When can two matrices be multiplied, and what is the size of the output?
4. What is the distributive property for matrix multiplication?
5. What is the associative property for matrix multiplication?
6. What is the commutative property for matrix multiplication?

Exercise 3.2

These are synthesis problems. It would be helpful to complete these. They should be done by hand.

1. Consider the matrix $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. Show that $\mathbf{A}^2$ commutes with $\mathbf{A}$.
2. Use the distribution law to expand $(\mathbf{A} + \mathbf{B})^2$ assuming that $\mathbf{A}$ and $\mathbf{B}$ are matrices of appropriate size. How does this compare to the situation for real numbers?
3. Show that $\mathbf{D} = \begin{bmatrix} 4 & -2 \\ 3 & -3 \end{bmatrix}$ satisfies the matrix equation $\mathbf{D}^2 - \mathbf{D} - 6\mathbf{I} = \mathbf{0}$.

Exercise 3.3

These are challenge problems. Pick one of them to wrestle with. It is not important to complete these. They should be done by hand.

1. The matrix exponential is defined by the power series
$$\exp \mathbf{A} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!}$$
Assume $\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$. Find a formula for $\exp \mathbf{A}$.
2. The real number 0 has just one square root: 0. Show, however, that the 2×2 zero matrix has infinitely many square roots by finding all 2×2 matrices $\mathbf{A}$ such that $\mathbf{A}^2 = \mathbf{0}$.
3. Use induction to prove that $\mathbf{A}^n$ commutes with $\mathbf{A}$ for any square matrix $\mathbf{A}$ and positive integer n.

3.4 2D Rotation Matrices

We're going to think about how to use rotation matrices to rotate a geometrical object. In doing so we will solidify fundamental concepts around matrix multiplication and start to explore the notion of "inverse". For clarity we will first work in 2D. Recall that the rotation matrix $\mathbf{R}(\theta)$:

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

will rotate an object counterclockwise **about the origin** through an angle of θ.

Exercise 3.4

This is a hands-on, conceptual problem involving the multiplication of 2D rotation matrices.

1. Place an object on your table, and imagine that the origin of an xy-coordinate system is at the center of your object with $+z$ pointing upwards.
2. Rotate it counterclockwise by 30 degrees, and then again by another 60 degrees. What is it's orientation now? How would you get there in one rotation instead? What does this suggest about the multiplication of rotation matrices?
3. What happens if you first rotate it by 60 degrees, and then by 30 degrees? What does this suggest about the commutative property of 2D rotation matrices?

Exercise 3.5

This is an algebra problem involving the multiplication of 2D rotation matrices.

1. Use some algebra to show that 2D rotation matrices commute, i.e. $\mathbf{R}(\theta_1)\mathbf{R}(\theta_2) = \mathbf{R}(\theta_2)\mathbf{R}(\theta_1)$.
2. Use some algebra to show that $\mathbf{R}(\theta_1)\mathbf{R}(\theta_2) = \mathbf{R}(\theta_1 + \theta_2)$. You will need to look up some trig identities.

Exercise 3.6

Now, consider a rectangle of width 2 and height 4, centered at the origin. For clarity, this means that the corners of the rectangle have coordinates $(1, 2)$, $(-1, 2)$, $(-1, -2)$, and $(1, -2)$.

1. Plot these four points by hand and connect them with lines to complete the rectangle.
2. Now, using the appropriate rotation matrix, transform each of the corner points by a rotation through 30 degrees counterclockwise (recall that the sin and cos of 30 degrees can be expressed

exactly). Compute and plot the resulting points by hand and connect them with lines. Does the resulting figure look like you'd expect?

3. Do it in MATLAB. Create and plot the original 4 points, create the rotation matrix, transform each of the four original points using the rotation matrix, and plot the resulting points. Does this look right? *Reminder:* `plot(1,2,'x')` *puts a mark at the point (1,2). Matlab: the functions cos and sin expect radians, while cosd and sind expect degrees.*

4. Operating on individual points with the rotation matrix is cool, but we can be much more efficient by operating on all 4 points at the same time. Write down the matrix whose columns represent the four corners of the rectangle. Then write down the matrix multiplication problem we can solve to transform the rectangle from above all at once. Create these matrices in MATLAB to perform the rotation in a single operation. Plot the resulting matrix to confirm your transformation! *Some MATLAB tips:* `plot(X,Y)` *creates a line plot of the values in the vector Y versus those in the vector X. So if you wanted to plot a line from the origin (0,0) to the point (1,2), you would do this: plot([0 1],[0 2]).* The command `axis([-xlim xlim -ylim ylim])` sets the axes of the current plot to run from `-xlim` to `xlim` and from `-ylim` to `ylim`

5. What is the area of the rectangle before and after the rotation?

6. What matrix should you use to undo this rotation? Define it in MATLAB and check.

7. Show on the board that the product of this matrix with the original rotation matrix is the identity matrix. For clarity, let's give this matrix the symbol $\mathbf{R}^{-1}$. It is the matrix that inverts the original operation and is known as the *inverse* of the matrix $\mathbf{R}$.

3.5 3D Rotations

We can extend the idea of 2D rotations to 3D rotations. The simplest approach is to think of 3D rotations as a composition of rotations about different axes. First let's define the rotation matrices for counterclockwise rotations of angle θ about the x, y and z axes respectively.

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{3.1}$$

$$\mathbf{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.2}$$

$$\mathbf{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

For example, to first rotate a vector $\mathbf{v}$ counterclockwise by θ about the x axis followed by counterclockwise by ϕ about the z axis, you need to do the following

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \mathbf{v} \tag{3.4}$$

We will next look at some sequence of physical rotations and relate them to these rotation matrices.

Exercise 3.7

Hold a closed book in front of you, with the top of the book towards the ceiling ($+z = (0, 0, 1)$ direction) and the cover of the book pointed towards you ($+x = (1, 0, 0)$ direction), which leaves the opening side of the book pointing towards your right ($+y = (0, 1, 0)$) and the spine toward the left.

1. Rotate the book by 90 degrees counter-clockwise about the x-axis, then from this position, rotate the book by 90 degrees counter-clockwise about the z-axis. Which direction is the cover of the book facing now?
2. Return to the starting position. Now rotate the book by 90 degrees counter-clockwise about the z axis, and then from this position, rotate the book by 90 degrees counter-clockwise about the x axis. Which direction is the cover of the book facing now? Is it the same as in part a?
3. An operation "commutes" if changing the order of operation doesn't change the result. Do 3D rotations commute?
4. The cover of the book is originally pointed towards $(1, 0, 0)$. Multiply this vector with the appropriate sequence of rotation matrices from above to reproduce your motions from part a. Do you end up with the correct final cover direction?
5. Multiply the $(1, 0, 0)$ vector with the appropriate sequence of rotation matrices to reproduce the motions from part b. Do you end up with the correct final cover direction?

6. Multiply the result of the previous part by the appropriate sequence of rotation matrices to return to the original $(1, 0, 0)$ vector.
7. From either of your answers to part d or part e, try, instead of operating on the $(1, 0, 0)$ vector sequentially with one rotation matrix and then the other, take the product of the two rotation matrices first, and then multiply $(1, 0, 0)$ with the resultant matrix. Does this reproduce your answer?
8. Based on your answers to the previous parts, show that $(\mathbf{R}_z\mathbf{R}_x)^{-1} = \mathbf{R}_x^{-1}\mathbf{R}_z^{-1}$. This is a general property of matrix inverses – it works for all square, invertible matrices, not just rotation matrices!

3.6 Reflection and Shearing

In this activity we will meet reflection and shearing matrices, which will allow us to explore transformation matrices in general.

Reflection

Exercise 3.8

What do the following *reflection* matrices do? Think about it first, draw some sketches and then test your hypothesis in MATLAB using the rectangle with vertices $(0, 0)$, $(2, 0)$, $(2, 1)$, and $(0, 1)$. How much does the area of your basic rectangle change, if at all? What is the inverse of each?

1. $$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$
2. $$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$
3. $$\begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$$

Shearing

Exercise 3.9

What do the following *shearing* matrices do? Think about it first, draw some sketches and then test your hypothesis in MATLAB with the rectangle with vertices $(0,0)$, $(2,0)$, $(2,1)$, and $(0,1)$. How much does the area of your basic rectangle change, if at all? What is the inverse of each?

1. $$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

2. $$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

3. $$\begin{bmatrix} 1 & 2k \\ 0 & 1 \end{bmatrix}$$

4. $$\begin{bmatrix} 1 & 0 \\ 2k & 1 \end{bmatrix}$$

Review and Preview

Solution 3.1

1. Scalars, vectors, and matrices are examples of arrays. A 0-dimensional array can be thought of as a scalar. A 1-dimensional array is a vector. A 2-dimensional array is a matrix.
2. The matrices have to be the same size and addition is element-wise.
3. The matrices have to compatible (inner dimensions agree), and the output is dictated by the outer dimensions, i.e. $(n \times m)(r \times s) = (n \times s)$.
4. Distributive property: $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
5. Associative property: $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$
6. Commutative property: Two matrices commute if $\mathbf{AB} = \mathbf{BA}$ but this is not always true.

Solution 3.2

1. You need to show that $\mathbf{A}^2\mathbf{A} = \mathbf{A}\mathbf{A}^2$ for this particular matrix. You can do it by multiplying.
2. Using the distributive property you can see that $(\mathbf{A} + \mathbf{B})^2 = (\mathbf{A} + \mathbf{B})(\mathbf{A} + \mathbf{B}) = \mathbf{A}^2 + \mathbf{AB} + \mathbf{BA} + \mathbf{B}^2$
3. If you plug $\mathbf{D}$ and $\mathbf{D}^2$ into the equation you should find that the result is a zero matrix.

Solution 3.3

1. The matrix exponential is defined by the power series $\exp \mathbf{A} = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \ldots$. Notice that this $\mathbf{A}$ is diagonal and $\mathbf{A}^2 = \begin{bmatrix} 2^2 & 0 \\ 0 & 3^2 \end{bmatrix}$ and the exponential becomes $\exp \mathbf{A} = \begin{bmatrix} 1 + 2 + 2^2/2! + \ldots & 0 \\ 0 & 1 + 3 + 3^2/2! + \ldots \end{bmatrix}$. If you have seen power series before then you will recognise that $\exp \mathbf{A} = \begin{bmatrix} \exp 2 & 0 \\ 0 & \exp 3 \end{bmatrix}$.
2. You can define a general two by two matrix $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, find $\mathbf{A}^2$, set each of the entries equal to zero and find constraints on the entries a, b, c, d.
3. You need to show that $\mathbf{A}^n\mathbf{A} = \mathbf{A}\mathbf{A}^n$ for any square matrix $\mathbf{A}$ and any positive integer n by induction. First you show it is true for $n = 1$ and $n = 2$. Then assume it is true for some $n = k$, and prove that it must be true for $n = k + 1$. You use the fact that $\mathbf{A}$ commutes with itself and the associative property, i.e $\mathbf{A}^2\mathbf{A} = (\mathbf{AA})\mathbf{A} = \mathbf{A}(\mathbf{AA}) = \mathbf{AA}^2$.

Solution 3.4

1. Okay, I placed my book on the table.

2. You could get there by rotating once by 90 degrees. This suggests that the product of two rotation matrices of angles θ_1 and θ_2 is a rotation matrix of $\theta_1 + \theta_2$, i.e. $\mathbf{R}(\theta_1)\mathbf{R}(\theta_2) = \mathbf{R}(\theta_1 + \theta_2)$.

3. You end up in the same orientation so it doesn't matter the order. This suggests that the order of multiplication doesn't matter so that two rotation matrices must commute.

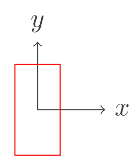
Solution 3.5

1. You could multiply out two rotation matrices with angle θ_1 and θ_2 in the two different orders and you will observe that the output is the same because real numbers commute, i.e. $\cos\theta_1 \cos\theta_2 = \cos\theta_2 \cos\theta_1$.

2. If you multiply two matrices together you will get the following expression in the first row and first column, $\cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2$. You will find a trig identity which reduces this to $\cos(\theta_1 + \theta_2)$. Similar reductions take place for the other elements.

Solution 3.6

1. The rectangle is



2. The rotation matrix is

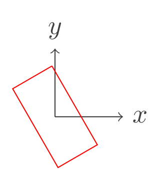
$$\mathbf{R} = \begin{bmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

Applying this to each point, we get

$$\mathbf{R}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}-2}{2} \\ \frac{1+2\sqrt{3}}{2} \end{bmatrix}, \ \mathbf{R}\begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}+2}{2} \\ \frac{1-2\sqrt{3}}{2} \end{bmatrix},$$

$$\mathbf{R}\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{-\sqrt{3}-2}{2} \\ \frac{-1+\sqrt{3}}{2} \end{bmatrix}, \ \mathbf{R}\begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} \frac{-\sqrt{3}+2}{2} \\ \frac{-1-\sqrt{3}}{2} \end{bmatrix}.$$

And the rotated figure looks like,



3. There are lots of ways to do this point by point. Here is an example of how to transform the bottom right point:

```
>> BR = [1;-2]
>> plot(BR(1,:),BR(2,:),'b*')
>> rotmatrix = [cosd(30) -sind(30); sind(30) cosd(30)]
>> nBR = rotmatrix*BR
>> plot(nBR(1,:),nBR(2,:),'r*')
```

4. There are lots of ways to do this. Here is an example where we include the first point twice so that the points can easily be connected with lines:

```
>> pts = [1 -1 -1 1 1;2 2 -2 -2 2]
>> npts = rotmatrix*pts
>> plot(pts(1,:),pts(2,:),'b'), hold on
>> plot(pts(1,:),pts(2,:),'r')
>> axis([-3 3 -3 3])
>> axis equal
```

5. The area of the rectangle is the same before and after rotation: 8 square units.

6. To undo this rotation you could simply rotate it by 30 degrees clockwise, using the matrix

$$\mathbf{R}^{-1} = \begin{bmatrix} \cos 30 & \sin 30 \\ -\sin 30 & \cos 30 \end{bmatrix}.$$

7. The product of $\mathbf{R}^{-1}$ and $\mathbf{R}$ is

$$\mathbf{R}^{-1}\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos^2\theta + \sin^2\theta & 0 \\ 0 & \cos^2\theta + \sin^2\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where we have used the trig identity $\cos^2\theta + \sin^2\theta = 1$.

Solution 3.7

1. The cover is now facing toward the $+y$ axis (the positive part of the y axis).

2. The cover is now facing the $+z$ axis. This is different than in part a.

3. Since the answers for the first two parts are different, 3D rotations do not commute.

4. Let $\mathbf{v}$ be the vector that represents the initial direction of the cover of the book,

$$\mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Rotation by 90 degrees counterclockwise around the x axis is given by

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

so that the new vector becomes

$$\mathbf{R}_x\mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Rotation by 90 degrees counterclockwise around the z axis is given by

$$\mathbf{R}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so that the new vector becomes

$$\mathbf{R}_z\mathbf{R}_x\mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

which is the correct final direction.

5. Using the matrices from above,

$$\mathbf{R}_x\mathbf{R}_z\mathbf{v} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

6. To rotate 90 degrees clockwise around the x axis we use the matrix

$$\mathbf{R}_x^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

and to rotate 90 degrees clockwise around the z axis we use the matrix

$$\mathbf{R}_z^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then we can return the vector $(0, 0, 1)$ to its original position $(1, 0, 0)$ by

$$\mathbf{R}_z^{-1}\mathbf{R}_x^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

7. We can multiply the rotation matrices together and perform a single matrix multiplication. For part d, the relevant matrix product is

$$\mathbf{R}_z\mathbf{R}_x = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and we see that

$$\mathbf{R}_z\mathbf{R}_x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

as expected.

8. We can see from the previous parts that

$$(\mathbf{R}_z\mathbf{R}_x)^{-1} = \mathbf{R}_x^{-1}\mathbf{R}_z^{-1}.$$

In other words, when you take the inverse, the order of operations must swap!

Solution 3.8

1. This matrix reflects everything over the y-axis. In the figure below, the original blue rectangle becomes the orange rectangle. The area of the rectangle stays the same.

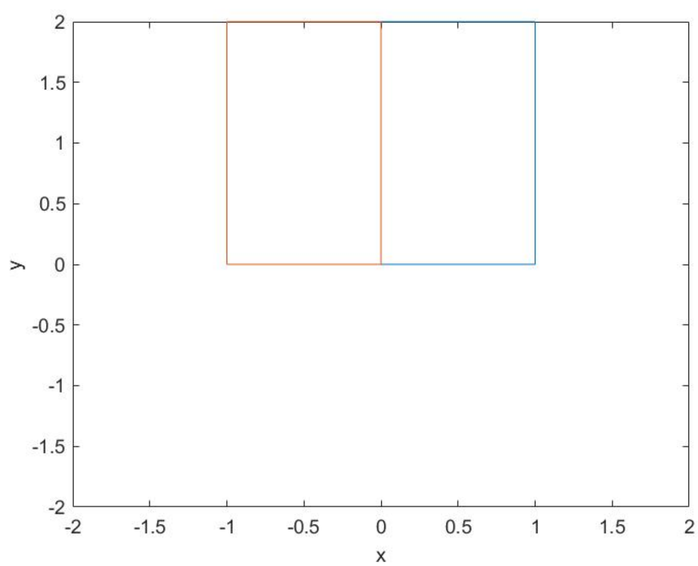


Figure 3.1: Reflection over y-axis.

2. This matrix reflects everything over the x-axis. In the figure below, the original blue rectangle becomes the orange rectangle. The area of the rectangle stays the same.

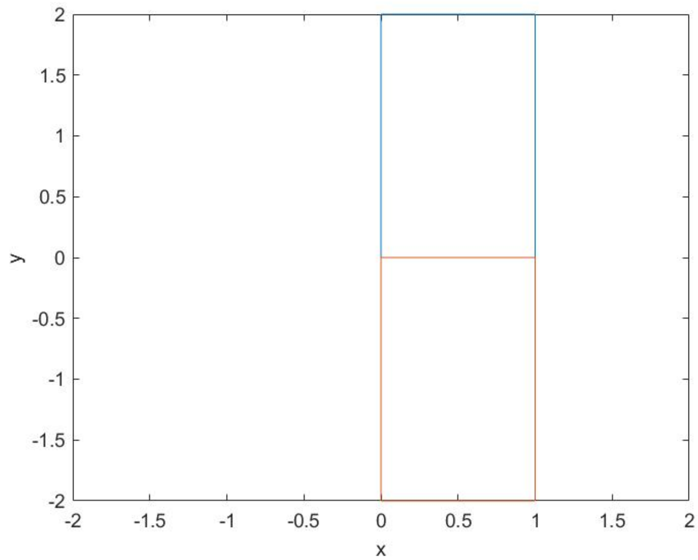


Figure 3.2: Reflection over x-axis.

3. For example, let $\theta = 30$ degrees. Then the rectangle is reflected along the line that is 30 degrees counterclockwise from the x-axis. In the figure below, the original blue rectangle becomes the orange rectangle.

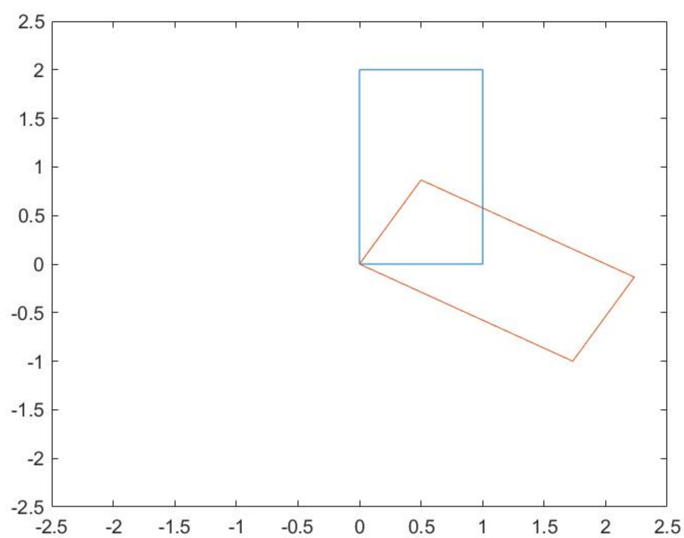


Figure 3.3: Reflection over 30 degree line.

Notice that, if we plug in $\theta = 90$, we get the matrix from part 1, which reflects over the x-axis (i.e., 90 degree line) and, if we plug in $\theta = 0$, we get the matrix from part 2, which reflects over the y-axis (i.e., the 0 degree line).

Solution 3.9

1. This shearing matrix pulls the points along horizontal lines and the strength of the pull is proportional to the y coordinate. In the figure below, the blue rectangle is sheared to become the orange rectangle:

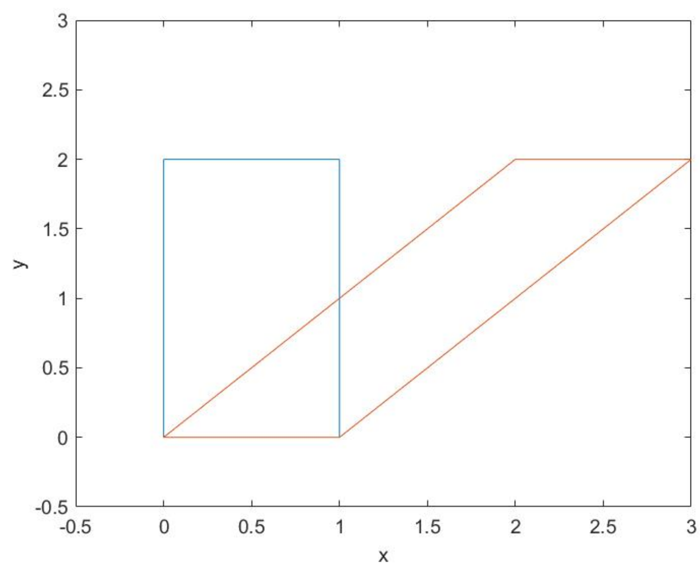


Figure 3.4: Shearing in x direction.

The area of the rectangle does not change. The inverse is

$$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

2. This shearing matrix pulls the points along vertical lines and the strength of the pull is proportional to the x coordinate. In the figure below, the blue rectangle is sheared to become the orange rectangle:

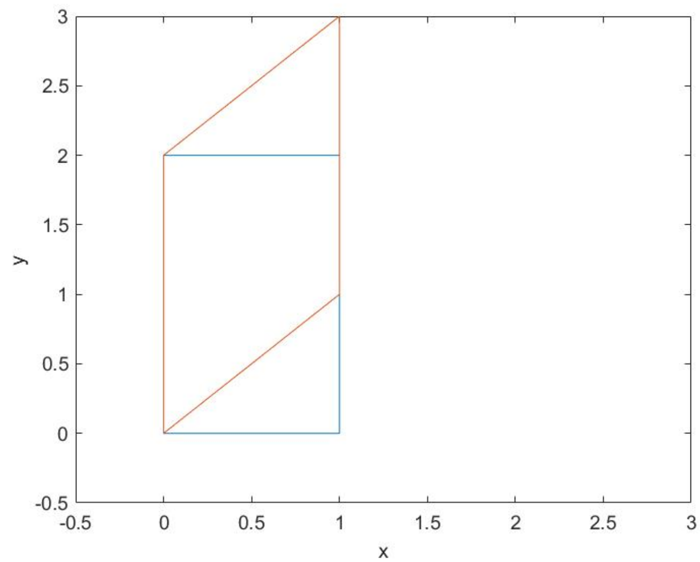


Figure 3.5: Shearing in y direction.

The area of the rectangle does not change. The inverse is

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

3. This shearing matrix pulls the points along horizontal lines and the strength of the pull is proportional to the y coordinate and the constant k (the bigger the k, the stronger the pull). In the figure below, with $k = 2$, the blue rectangle is sheared to become the orange rectangle:

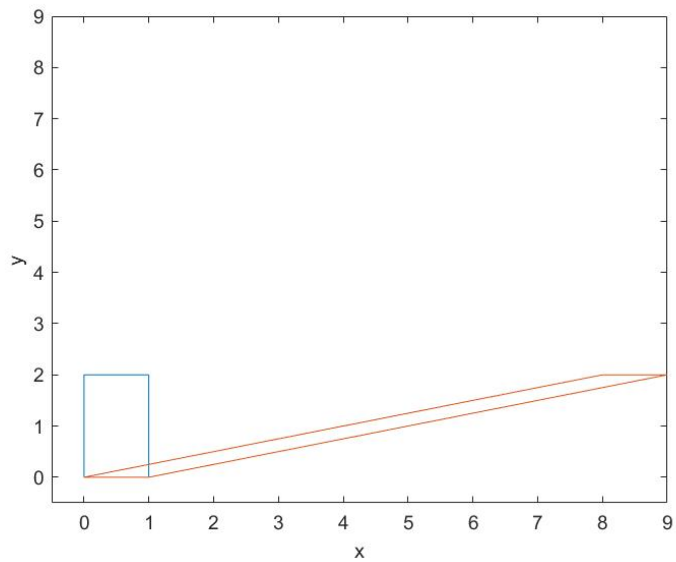


Figure 3.6: Shearing in x direction with $k = 2$.

The area of the rectangle does not change. The inverse is

$$\begin{bmatrix} 1 & -4 \\ 0 & 1 \end{bmatrix}.$$

4. This shearing matrix pulls the points along vertical lines and the strength of the pull is proportional to the x coordinate and the constant k (the bigger the k, the stronger the pull). In the figure below, with $k = 2$, the blue rectangle is sheared to become the orange rectangle:

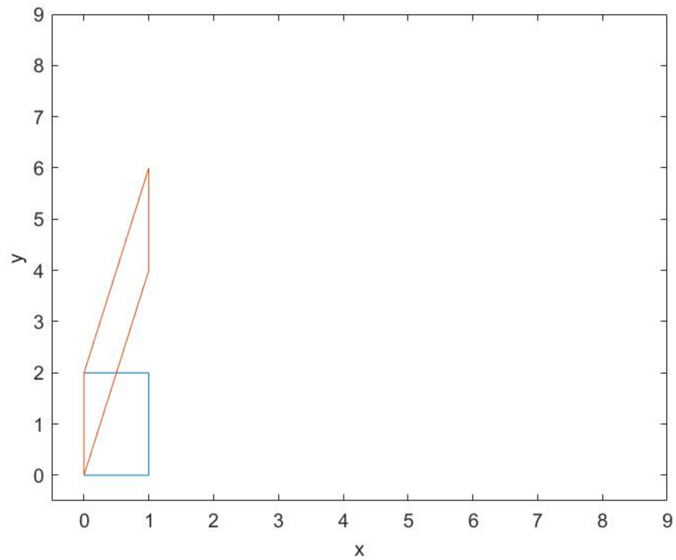


Figure 3.7: Shearing in y direction with $k = 2$.

The area of the rectangle does not change. The inverse is

$$\begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix}.$$

Chapter 4

Night 2: Matrix Operations

Overview and Orientation

Learning Objectives

Concepts

- Compute the determinant of a 2×2 matrix
- Know the relationship between the determinant of a matrix and whether the matrix is invertible
- Find the inverse of a 2×2 matrix by hand
- Use computational tools to find the inverse of an $n \times n$ matrix
- Design a 2 or 3-dimensional matrix that will scale a vector by given amounts in the x, y or z direction
- Design a 3-dimensional matrix that will translate a 2-D vector by given amounts in x and y

MATLAB skills

- Represent a set of points in 2-D space (i.e., pairs of x, y values) as column vectors
- Transform a set of 2-D points (i.e., the outline of a shape) using a matrix to rotate and translate the original
- Multiply matrices and find their inverses
- Compute the determinant of a matrix

Suggested Approach

See Night 1 assignment for our general suggested approach to night assignments and a list of linear algebra resources.

4.1 Determinant of a Matrix

The determinant of a square matrix is a property of the matrix which indicates many important things, including whether a matrix is invertible or not. We will see more of this when we see matrix inverses shortly. The determinant of a matrix $\mathbf{G}$ is denoted a few different ways.

$$\det(\mathbf{G}) = |\mathbf{G}| \tag{4.1}$$

Consider a generic 2×2 matrix $\mathbf{G}$:

$$\mathbf{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The formula for the determinant of a 2×2 matrix is quite straightforward:

$$\det(\mathbf{G}) = ad - bc \tag{4.2}$$

For example, for the following 2×2 matrix,

$$\begin{aligned} \det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right) &= \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \\ &= (1)(4) - (2)(3) = -2 \end{aligned} \tag{4.3}$$

Exercise 4.1

Return to the transformation matrices in the day assignment and calculate the determinant for the following:

1. The generic 2×2 rotation matrix

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2. The matrix which reflects over the y axis

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

3. The matrix which shears in the horizontal direction

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Exercise 4.2

1. What do the following matrices do? Think about it first, draw some sketches and then test your hypothesis in MATLAB. How much does the area of your basic rectangle change, if at all?

a)

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

b)

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

2. Is it possible to "undo" the matrices above? Why or why not?

Exercise 4.3

1. What are the determinants of the two matrices from the previous exercise, Exercise 4.2?
2. Generalizing from Exercise 4.1 and Exercise 4.2, what's the relationship between the determinant of a matrix and the result of transforming a rectangle by that matrix?

Finding the determinant of an $n \times n$ matrix, where $n > 2$, is a bit more computationally intensive. If you want to learn how to do the procedure by hand, check out this Khan Academy video. For this course, we simply recommend you use the `det` function in MATLAB.

4.2 Matrix Inverses

Inverse of 2×2 Matrices

In class you worked with rotation matrices and transformations that were compositions of simpler rotations, and you learned how to invert them. When you multiply a vector by any matrix (not just ones that are associated with simple spatial transformations), you transform the original vector into a new vector. More generally (than rotations), you can *often* undo the linear transformation (just like you did with the rotation matrix). Undoing this linear transformation is a linear transformation itself! Therefore the act of undoing a linear transformation can be formulated with a matrix multiply.

Exercise 4.4

Consider the following matrices and vector. (Don't try to interpret these as intuitive geometrical operations; we're just using them to explore the determinant.) Work out the following problems in

MATLAB.

$$\mathbf{P} = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \tag{4.4}$$

$$\mathbf{Q} = \begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -2 & 1 \end{bmatrix} \tag{4.5}$$

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \tag{4.6}$$

1. Find $\mathbf{w} = \mathbf{Pu}$.
2. Find $\mathbf{Qw}$. How is this related to $\mathbf{u}$?
3. Find $\mathbf{QP}$. Does the answer look familiar?
4. Find $\mathbf{PQ}$.
5. Find the determinant of $\mathbf{P}$. In MATLAB, you can compute the determinant of any (not just 2×2) matrix using the `det` function.
6. Find the determinant of $\mathbf{Q}$.

A matrix $\mathbf{B}$ is said to be the inverse of the matrix $\mathbf{A}$ if, and only if, $\mathbf{BA} = \mathbf{I}$ and $\mathbf{AB} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For 2×2 matrices, the inverse (if it exists) is given by the following

$$\mathbf{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4.7}$$

$$\mathbf{G}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{4.8}$$

The last equation should indicate to you that the inverse of the matrix $\mathbf{G}^{-1}$ is only defined if $ad - bc \neq 0$. Sweet mother of linear algebra, $ad - bc$ is our buddy the determinant. More generally, any square matrix can be inverted if and only if its determinant is non-zero.

Now let's practice calculating inverses, some of their properties, and how we may use them.

Exercise 4.5

All matrices $\mathbf{A}$ and $\mathbf{B}$ which have inverses have the following properties

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$
$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$$

1. Using the above properties, please compute the following by hand.

a) If

$$\mathbf{P} = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \tag{4.9}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \tag{4.10}$$

$$\tag{4.11}$$

find $(\mathbf{PB})^{-1}$. Recall that you already know the inverse of $\mathbf{P}$ from earlier.

b) For $\mathbf{P}$ as defined above, find

$$(\mathbf{P}^T)^{-1} \tag{4.12}$$

2. Use the inverse formula to calculate the inverses for the first three matrices in Exercise 4.1. Confirm your answers by multiplying the inverse with the original matrix.

   a) By hand, write an equation relating $\mathbf{n}$ and $\mathbf{d}$, using a matrix-vector product.
   b) By hand, calculate how many oranges and apples you have.
   c) Why do you think this type of problem is often called an inverse problem?

Note that solving matrix-vector equations like above can be done without explicitly computing the matrix inverse which is computationally expensive. (A nod to our future friend, left matrix divide or backslash divide.)

Inverse of $n \times n$ Matrices

For higher-dimensional matrices, e.g. $n \times n$ matrices for $n > 2$, the matrix inverse is defined in the same way. Suppose you have an $n \times n$ matrix $\mathbf{A}$ and an $n \times n$ matrix $\mathbf{B}$. Then $\mathbf{B}$ is the inverse of $\mathbf{A}$ if and only if $\mathbf{BA} = \mathbf{I}$ and $\mathbf{AB} = \mathbf{I}$. The following are some properties of inverses of matrices

- Only square matrices are invertible, i.e., only square matrices have inverses.
- A matrix has an inverse only if its determinant is non-zero.

There are a number of different procedures to compute the inverse of higher-dimensional matrices, but we will not be going into the details of their computation here. You can look them up if you are interested, or need to in the future. In MATLAB, you can compute the inverse of a matrix using the `inv` function.

Exercise 4.6

1. Consider the example with the fruits that you worked out earlier. Now, in addition to apples

and oranges, suppose you also had an unknown number of pears which each weigh 3 oz, and cost \$3. Additionally, suppose that the total weight of the fruits is 45 oz, and you paid a total of \$21 for the fruit.

a) If possible find the numbers of oranges, apples and pears. If not, please explain why.

b) Suppose that you additionally know that you have a total of 14 fruits. Can you formulate and solve a matrix-vector equation to find out the numbers of oranges, apples and pears you have?

c) What is the determinant of the matrix you have set up to solve this?

2. The fruit vendors bought the pricing algorithm from Uber. Oranges are still \$2, pears are now only \$1.50, and (due to an influx of teachers) apples are now surging at \$1.50 each. Their weights stay the same. You return to the market, and again purchase 14 fruits, which have the same total weight and total cost.

   a) Can you formulate and solve a matrix-vector equation to find out the numbers of oranges, apples and pears you have?

   b) What is the determinant of the matrix you have set up to solve this?

   c) Debrief at your table about what this means.

4.3 *Transformation Matrices, Continued*

Scaling

Returning to two dimensions. In the Night 1 assignment, you also learned about scaling matrices. Recall that the scaling matrix $\mathbf{S}$ scales the x-component by s_1 and the y-component by s_2

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}.$$

Let's assume for the moment that $s_1 = 2$ and $s_2 = 1/3$. Working with the rectangles defined in class whose corners have coordinates $(1, 2)$, $(1, -2)$, $(-1, 2)$, and $(-1, -2)$ complete the following activities:

Exercise 4.7

1. Predict what would happen if you operate on the rectangle with $\mathbf{S}$.

2. Write a MATLAB script to carry out this operation and check your prediction.

3. How does the area of the rectangle change?

4. What matrix should you use to *undo* this scaling? Show that the product of this matrix with the original scaling matrix is the *identity* matrix.

5. Define it in MATLAB and check. Again, this is the *inverse* matrix and we give it the symbol $\mathbf{S}^{-1}$.

6. In MATLAB, change the value of s_2 to 1 and find the product of the new $\mathbf{S}$ and your rectangle. How does the area of the rectangle change? Change the value of s_2 back to 1/3.

7. Predict what would happen if you operate on the original rectangle with $\mathbf{SR}$, where $\mathbf{R}$ is the rotation matrix. How about $\mathbf{RS}$? Implement both of these in MATLAB and check.

8. How would you *undo* each of these operations ($\mathbf{SR}$ and $\mathbf{RS}$)? How is the inverse of the product related to the individual inverses, i.e. what is the relationship between $(\mathbf{SR})^{-1}$ and $\mathbf{S}^{-1}$ and $\mathbf{R}^{-1}$? What about $(\mathbf{RS})^{-1}$?

Translation

It would be really useful if, in addition to scaling and rotating our objects, we could translate them. Let's start by thinking about vectors and then we will figure out how to represent translation as a matrix operation.

Consider an initial vector $\mathbf{v}$ and a translation vector $\mathbf{t}$. The new translated vector is simply $\mathbf{v}+\mathbf{t}$. For example, if you start with the initial vector $\mathbf{v}=\begin{bmatrix}x\\y\end{bmatrix}$ and translate it using the vector $\begin{bmatrix}2\\3\end{bmatrix}$ then the new vector is just $\begin{bmatrix}x+2\\y+3\end{bmatrix}$. More generally, if the translation vector is $\begin{bmatrix}t_x\\t_y\end{bmatrix}$ then the new vector will be $\begin{bmatrix}x+t_x\\y+t_y\end{bmatrix}$.

Wouldn't it be handy if we could define translation as a matrix operation? Yes, indeed it would be, we hear you say. Here is the standard method: add another entry to the original vector, and set it equal to 1, i.e., $\mathbf{v}=\begin{bmatrix}x\\y\\1\end{bmatrix}$. Now define the translation matrix as

$$\mathbf{T}=\begin{bmatrix}1&0&t_x\\0&1&t_y\\0&0&1\end{bmatrix}.$$

Exercise 4.8

1. Show that $\mathbf{Tv}$ accomplishes the process of translation (if you ignore the third entry in the new vector). What is the final vector?

2. Predict what would happen if you operate on our old friend the rectangle with the translation

matrix defined by $t_x = 2$ and $t_y = 3$.

3. Write a MATLAB script to carry out this operation and check your prediction. How has the area of your rectangle changed?

4. What matrix should you use to *undo* this translation? Show on paper that the product of this matrix with the original translation matrix is the *identity* matrix. Define it in MATLAB and check. Again, this is the *inverse* matrix and we give it the symbol $\mathbf{T}^{-1}$.

5. Choose a rotation matrix $\mathbf{R}$. Predict what would happen if you operate on the original rectangle with $\mathbf{TR}$. How about $\mathbf{RT}$? Implement both of these in MATLAB and check. How would you undo each of these operations? (You will first have to adjust your definition of $\mathbf{R}$ so that it is the correct size.)

6. Predict what would happen if you operate on the original rectangle with $\mathbf{STR}$. How about $\mathbf{TRS}$? How would you *undo* each of these operations? (You will first have to adjust your definition of $\mathbf{S}$ so that it is the correct size.)

7. How would you generalize translation to 3D?

Putting it all together: Dancing Animals

In this activity you will animate a circus act. (No real or imaginary animals will be injured in this performance.) Here is what we would like you to do:

Exercise 4.9

1. Decide on an animal.

2. Decide on a circus act that consists of a set of translations, rotations (think back to Day 2), shearings, and/or scalings in some order. Storyboard this idea and imagine the resulting animation.

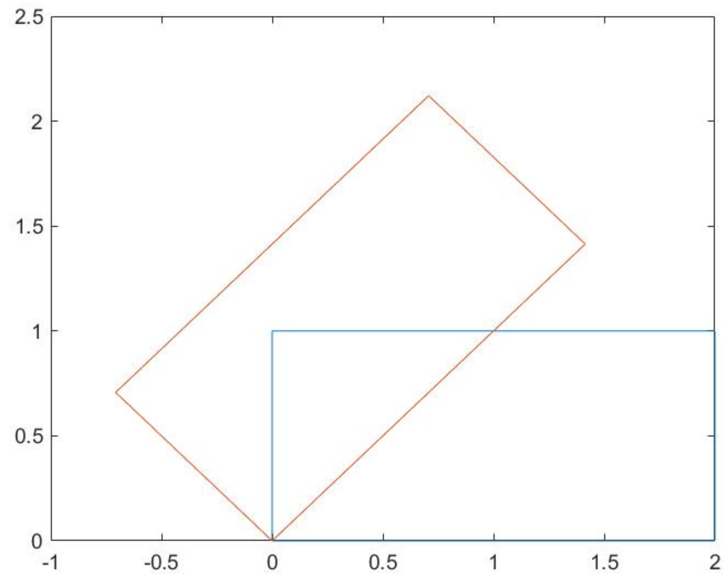
3. Propose a set of points that defines the outline and relevant features of your animal. You may find `ginput` useful. Define the points in MATLAB and plot your animal.

4. Create a script that makes your animal dance (in 2-D, unless you really want to go 3-D). You may want to make use of the `pause` and `drawnow` commands.

5. Now use your sequence of operations and animate your animal! In class you will have the opportunity to show off your dancing animal!

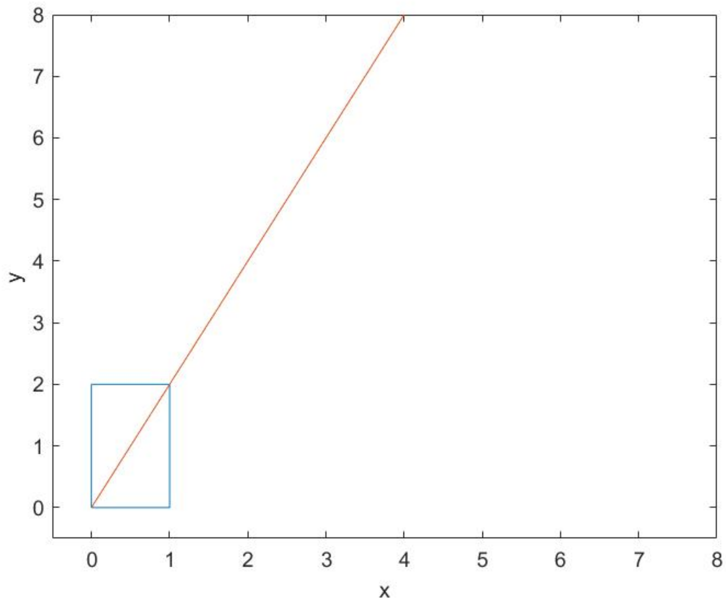
4.4 Conceptual Quiz

1. The orange shape is the result of applying a matrix $\mathbf{M}$ to the blue rectangle.



   What is the determinant of $\mathbf{M}$?

2. The orange shape is the result of applying a matrix $\mathbf{M}$ to the blue rectangle.



What is the determinant of $\mathbf{M}$?

3. The determinant is multiplicative, i.e, $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$. Let $\mathbf{M}$ be a matrix such that $\det(\mathbf{M}) = \frac{1}{3}$. What's $\det(\mathbf{M}^{-1})$? (Hint: $\det(\mathbf{I}) = 1$.)

4. Let R be a rectangle with area 1. Apply the scaling matrix $\mathbf{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$. What is the area of $\mathbf{S}R$?

   A. $\frac{s_1 s_2}{2}$
   B. 1
   C. $s_1 s_2$
   D. $s_1 + s_2$

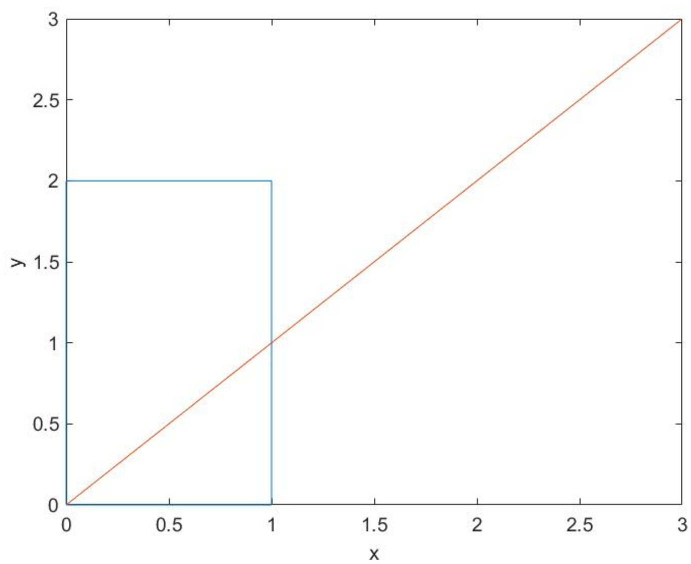
5. True or false: Any shearing matrix $\mathbf{S}$ and any rotation matrix $\mathbf{R}$ commute, i.e., $\mathbf{RS} = \mathbf{SR}$.

Solution 4.1

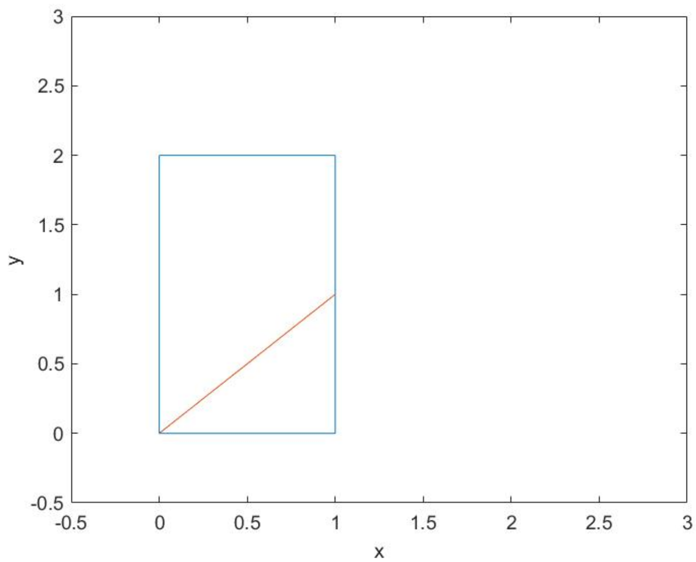
1. The determinant is 1. (Recall that $\cos^2\theta + \sin^2\theta = 1$.)
2. The determinant is -1.
3. The determinant is 1.

Solution 4.2

1. Each of the figures below shows the basic blue rectangle and the orange rectangle, which is the result of applying the transformation.



a)

b) 

2. It is not possible to undo these matrix transformations. Since everything is squished onto the same line, we would not be able to distinguish the original vectors.

Notice that, in the above matrices, the first row is a constant multiple of the second row. In other words, the matrix looks like $\begin{bmatrix} a & b \\ ca & cb \end{bmatrix}$ for some constant c. If we apply a matrix of this form to a point in 2D space reprsented by the vector $\begin{bmatrix} x \\ y \end{bmatrix}$, then the result will be $\begin{bmatrix} z \\ cz \end{bmatrix}$, where $z = ax + by$. In other words, the resulting point will always fall on the line $y = cx$.

Solution 4.4

1.
$$\mathbf{w} = \mathbf{Pu} = \begin{bmatrix} 7 \\ 17 \end{bmatrix}$$

2.
$$\mathbf{Qw} = \mathbf{QPu} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

3.
$$\mathbf{QP} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

   which is the identity matrix

4. The determinate of $\mathbf{P}$ is 2.

5. The determinate of $\mathbf{Q}$ is $\frac{1}{2}$.

Solution 4.5

1. a)

$$(\mathbf{PB})^{-1} = \begin{bmatrix} -17/2 & 7/2 \\ 5 & -2 \end{bmatrix}$$

   b)

$$(\mathbf{P}^T)^{-1} = \begin{bmatrix} 3/2 & -2 \\ -1/2 & 1 \end{bmatrix}$$

2.

$$\left(\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\right)^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$\left(\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\right)^{-1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

Solution 4.6

1. a) It's not possible to find the numbers of oranges, apples, and pears. We have the equation

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 3 \end{bmatrix} \begin{bmatrix} n_0 \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \end{bmatrix},$$

   but we cannot take the inverse of a 2×3 (non-square) matrix.

   b) Now we have the equation

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_0 \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \\ 14 \end{bmatrix}.$$

   So by taking the inverse of the 3×3 matrix we find that $n_0 = 3, n_a = 9$ and $n_p = 2$.

   c) The determinant of the matrix is 2.

2. a) The equation becomes

$$\begin{bmatrix} 2 & \frac{3}{2} & \frac{3}{2} \\ 4 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_0 \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \\ 14 \end{bmatrix}.$$

   But the matrix is not invertible, so we cannot solve for the number of fruit.

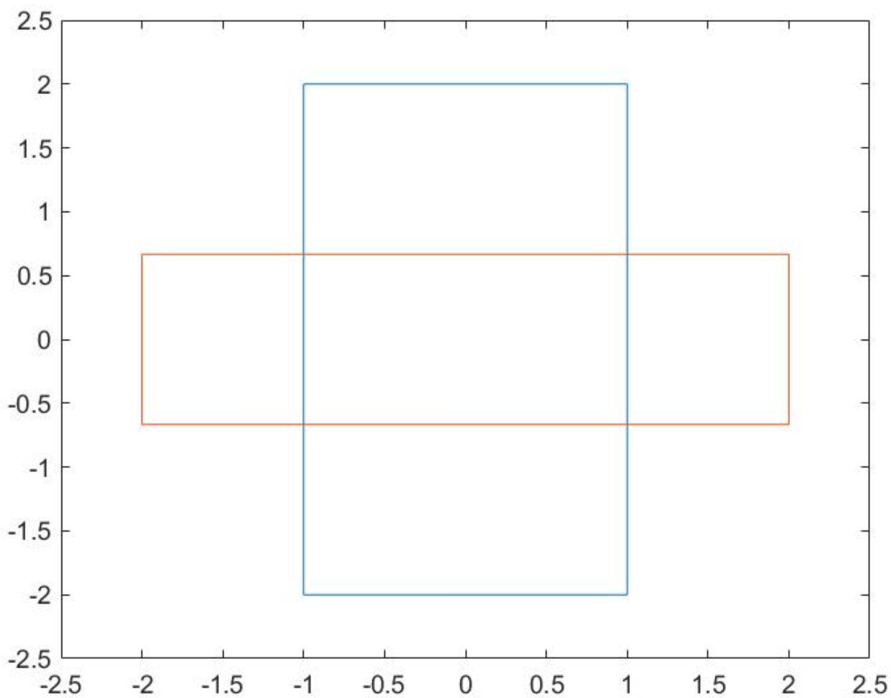
   b) The determinant of the matrix is 0.

   c)

Solution 4.7

1. The length of the rectangle would double in the x direction and be reduced to 1/3 the length in the y direction.

2. First we define the corners of the rectangle as the columns in a matrix
   » `points=[1 1 -1 -1; 2 -2 -2 2]`
   and we define the scaling matrix
   » `S=[2 0; 0 1/3]`. Then we simply multiply them
   » `scaledpoint=S*points`.
   Plotting them, here is the original rectangle in blue and the scaled rectangle in orange



3. The area is reduced from 8 units^2 to 5.33 units^2, or 2/3 of the original area.

4. To undo the process we use the inverse of the $\mathbf{S}$ matrix, or $\mathbf{S}^{-1}$ would be used.

$$\mathbf{S}^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 3 \end{bmatrix}.$$

   You should check that $\mathbf{S}^{-1}\mathbf{S} = \mathbf{S}\mathbf{S}^{-1} = \mathbf{I}$.

5. We define the inverse matrix » `Sinv=[0.5 0; 0 3]` and check that » `S*Sinv` and `Sinv*S` both produce the identity matrix.

6. The area of the rectangle doubles.

7. When the original rectangle is operated on with

$$\mathbf{SR}$$

, the resulting image will be a horizontally stretched parallelogram. When the original rectangle is operated on with $\mathbf{RS}$, the resulting image will be the scaled rectangle from the previous exercise only rotated 60 degrees counter-clockwise.

8. $(\mathbf{SR})^{-1} = \mathbf{R}^{-1}\mathbf{S}^{-1}$ or $(\mathbf{RS})^{-1} = \mathbf{S}^{-1}\mathbf{R}^{-1}$

Solution 4.8

1. $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix}$

2. The rectangle would be moved 2 to the right and 3 up.

3. The area of the rectangle does not change.

4. $$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

5. If the original rectangle is operated on by $\mathbf{TR}$, the rectangle would first be rotated with respect to the origin and than translated. If the original rectangle is operated on by $\mathbf{TR}$, the rectangle would first be translated and then rotated. As rotation happens with respect to the origin, the 2 operations will not result in the same rectangle.

   To undo the operation $\mathbf{TR}$, the resulting figure should be operated on by $\mathbf{R}^{-1}\mathbf{T}^{-1}$. To undo the operation $\mathbf{RT}$, the resulting figure should be operated on by $\mathbf{T}^{-1}\mathbf{R}^{-1}$.

6. If the original rectangle is operated on with $\mathbf{STR}$, the resulting image will be of the rectangle rotated 60 degrees around the origin, translated 2 to the right and 3 up and then scaled by $\mathbf{S}$. If the original rectangle is operated on with $\mathbf{TRS}$, the resulting image will be the scaled rectangle rotated 60 degrees around the origin and then translated 2 to the right and 3 up.

   To undo $\mathbf{STR}$, the resulting figure should be operated on by $\mathbf{R}^{-1}\mathbf{T}^{-1}\mathbf{S}^{-1}$. To undo $\mathbf{TRS}$, the resulting figure should be operated on by $\mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{T}^{-1}$.

7. $$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Chapter 5

Day 3: Linear Independence, Span, Basis, and Decomposition

5.1 Schedule

- 0900-0930: Debrief and Dancing Animal Demos
- 0930-1000: Synthesis
- 1000-1030: Mini-Lecture: Linear Independence, Span, Basis, Decomposition
- 1030-1045: Coffee
- 1045-1210: Technical Details: Linear Independence, Span, Basis, Decomposition
- 1210-1220: Preview

5.2 Debrief and Dancing Animal Demos

- Please discuss your overnight work with your table-mates, create a set of key concepts, and a set of ideas that you are still confused by.
- Be prepared to demo your dancing animal!

5.3 Synthesis

Exercise 5.1

You should do all of these.

1. Assume the matrix $\mathbf{D}$ represents a geometrical object. What is the correct matrix expression if we want to rotate it first ($\mathbf{R}$), then scale it ($\mathbf{S}$), and finally translate ($\mathbf{T}$) it?
   A. $\mathbf{DRST}$
   B. $\mathbf{TSRD}$
   C. $\mathbf{RSTD}$
   D. $\mathbf{DTSR}$
2. What would be the correct expression in order to undo the transformation in the previous problem?
3. $\mathbf{A}$ and $\mathbf{B}$ are square, invertible matrices of the same size. Which of the following are **always** true (no matter the entries in $\mathbf{A}$ and B?

A. $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$
B. $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
C. $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$
D. $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$
E. $\mathbf{A}+\mathbf{B} = \mathbf{B}+\mathbf{A}$
F. $\mathbf{AB} = \mathbf{BA}$
G. $\det(\mathbf{AB}) = \det(\mathbf{A}) + \det(\mathbf{B})$
H. $(\mathbf{AB})^T = \mathbf{A}^T\mathbf{B}^T$
I. $(\mathbf{AB})^{-1} = \mathbf{A}^{-1}\mathbf{B}^{-1}$

5.4 *Linear Independence, Span, and Decomposition*

Exercise 5.2

Consider two column vectors

$$\mathbf{a}_1 = \begin{bmatrix}1\\1\\0\end{bmatrix}, \ \mathbf{a}_2 = \begin{bmatrix}1\\2\\0\end{bmatrix} \tag{5.1}$$

Both these vectors lie on the xy-plane since their z components are zero. Define a new vector $\mathbf{a}_3 = c_1\mathbf{a}_1 + c_2\mathbf{a}_2$, where c_1 and c_2 are arbitrary variables. Therefore $\mathbf{a}_3$ is a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$.

1. Does $\mathbf{a}_3$ also lie on the xy-plane?
2. Next, define a 3×3 matrix $\mathbf{A}$ whose columns are $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$. Show that the product of $\mathbf{A}$ and any 3×1 vector always lies on the xy-plane.

Exercise 5.3

Next, we will do a similar problem, but in MATLAB. Consider the following matrix:

$$\mathbf{B} = \begin{bmatrix}1 & 1 & 3\\1 & 2 & 4\\1 & 1 & 3\end{bmatrix} \tag{5.2}$$

The third column of this matrix equals the second column plus twice the first column. Hence these three vectors lie on some plane (not the xy-plane as in the previous part).

1. Open up MATLAB and using the `quiver3` command together with `hold on`, please plot the vectors corresponding to the three columns of $\mathbf{B}$, e.g., to plot the first column, type » `quiver3(0,0,0, 1,1,1);` in MATLAB.
2. Using the "rotate 3D" function on the MATLAB figure window, rotate the figure around so that it appears as if all three arrows overlap. This should indicate that the vectors lie on a plane.
3. Using `det` compute the determinant of matrix $\mathbf{B}$. Does this make sense?

The fundamental property here is that the columns of the $\mathbf{A}$ and $\mathbf{B}$ matrices are not *linearly independent*. We shall next define the idea of linearly independent vectors more formally.

- A finite set $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ of vectors in $\mathbf{R}^n$ is said to be *linearly dependent* if there exist scalars $c_1, c_2, \dots, c_m$ which are not all zero, such that

$$c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \ldots + c_m\mathbf{x}_m = \mathbf{0}.$$

Note that $\mathbf{R}^n$ here refers to the set of all n-dimensional vectors that are made up of real numbers. (For example, $\mathbf{R}^1$ is the real line and $\mathbf{R}^2$ is the plane.) For any value of n, $\mathbf{R}^n$ is an example of a *vector space* - we will meet different examples of vector spaces in the future. We can also express this equation using a matrix $\mathbf{A}$, whose columns are $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_m$.

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_m \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \mathbf{0}\,. \tag{5.3}$$

If a non-zero solution exists to $\mathbf{Ac} = \mathbf{0}$ then the set of vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ is linearly dependent. In the case of a square matrix ($n = m$), the vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_m$ are linearly dependent if and only if the $\det(\mathbf{A}) = 0$. Otherwise, the only way to satisfy the equation above is if $c_1 = c_2 = \cdots = c_m = 0$. Figure 5.1 illustrates two examples of three vectors that are in 3D space, but are linearly dependent, since in each case, all three vectors are on a plane.

Figure 5.1: Linearly dependent vectors in $\mathbf{R}^3$. (from Wikimedia Commons).

- The set of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ is *linearly independent* if it is not linearly dependent. In other words, the set of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ is linearly independent if

$$c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \ldots + c_m\mathbf{x}_m = \mathbf{0} \tag{5.4}$$

  only when $c_1 = c_2 = \cdots = c_m = 0$. In other words, if the only solution to $\mathbf{Ac} = \mathbf{0}$ is $\mathbf{c} = \mathbf{0}$, the set of vectors made up of the columns of $\mathbf{A}$ is linearly independent. For a square matrix this means the set is linearly independent if and only if $\det(\mathbf{A}) \neq 0$.

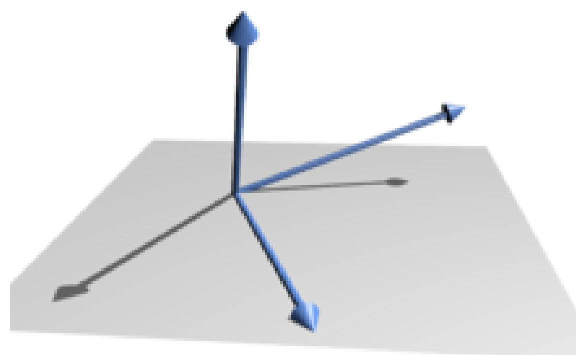


Figure 5.2: Linearly independent vectors in $\mathbf{R}^3$. (from Wikimedia Commons).

- The *span* of S is the set of all linear combinations of its vectors. In other words, the span of the set S is the set of all possible vectors of the form

$$c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \ldots + c_m\mathbf{x}_m$$

  The *span* is usually denoted by $span(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$.

- A finite set $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ of vectors is said to form a basis of a vector space V, if the vectors in S are linearly independent, and every point in V can be expressed as a linear combination of the vectors in the set S. Hence, if a set of vectors S is linearly independent those vectors form a *basis* of the set which is the span of those vectors.

Let's solidify our understanding of linear dependence, bases and span by working on a few problems by hand.

Exercise 5.4

1. Determine which of the following sets of vectors are linearly independent.

   a) $\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\2\end{bmatrix}$

   b) $\begin{bmatrix}1\\3\\0\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}$

   c) $\begin{bmatrix}1\\2\\3\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}3\\4\\3\end{bmatrix}$

   d) $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$ and $\mathbf{s}$, where the vectors are all 3-dimensional.

   e) $\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}3\\3\end{bmatrix}$

2. In words, describe the span of the vectors $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}1\\-1\end{bmatrix}$.

3. In words, describe the span of the vectors $\begin{bmatrix}1\\1\\0\end{bmatrix}$, $\begin{bmatrix}2\\3\\0\end{bmatrix}$ and $\begin{bmatrix}1\\-1\\0\end{bmatrix}$ which are all in 3-dimensional Euclidean space.

Orthogonality

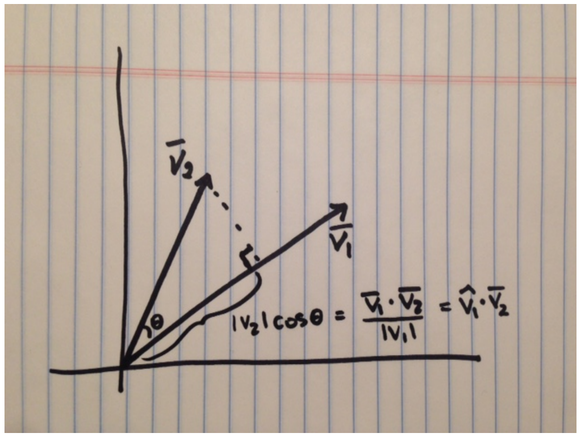


Figure 5.3: Projection

By trigonometry, if we have two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ which have an angle of θ between them, the component of $\mathbf{v}_2$ which lies along the direction of $\mathbf{v}_1$ is $|\mathbf{v}_2|\cos\theta$. Since the dot product of the two vectors can be expressed as $|\mathbf{v}_1||\mathbf{v}_2|\cos\theta$, this component (referred to as the projection) can be written as $\mathbf{v}_1 \cdot \mathbf{v}_2/|\mathbf{v}_1|$. If the projection is zero, the vectors are *orthogonal*, and $\mathbf{v}_1 \cdot \mathbf{v}_2 = 0$. If the vectors are unit length, in addition to being normal, the vectors are said to be *orthonormal*. Additionally, if a basis set is made up of orthonormal vectors, it is known as an orthonormal basis.

A square matrix with columns of unit vectors which are orthogonal to each other is known as an orthogonal matrix. An orthogonal marix $\mathbf{A}$ has the property that $\mathbf{A}^T = \mathbf{A}^{-1}$.

Exercise 5.5

Which of the following pairs of vectors are orthogonal or orthonormal?

1. $\begin{bmatrix}1\\2\\3\end{bmatrix}, \begin{bmatrix}-3\\2\\1\end{bmatrix}$

2. $\begin{bmatrix}1\\0\\-3\end{bmatrix}, \begin{bmatrix}3\\2\\1\end{bmatrix}$

3. $\begin{bmatrix}\frac{2}{\sqrt{13}}\\\frac{-2}{\sqrt{13}}\end{bmatrix}, \begin{bmatrix}\frac{-3}{\sqrt{13}}\\\frac{3}{\sqrt{13}}\end{bmatrix}$

Decomposition

Suppose we have a set (collection) of m basis vectors $\{\mathbf{v}_i\}$ which are normalized ($|\mathbf{v}_i| = 1$), mutually orthogonal ($\mathbf{v}_i^T\mathbf{v}_j = 0$ unless $i = j$) and span our space (every point can be written as some linear combination of the vectors $\{\mathbf{v}_i\}$). How do we actually find the linear combination which is equal to a given vector in our space?

Let's say we have a vector $\mathbf{w}$ which we are interested in expressing as a linear combination of our set of orthonormal vectors $\{\mathbf{v}_i\}$. We can write this linear combination as

$$\mathbf{w} = \sum_{i=1}^{m} c_i\mathbf{v}_i \tag{5.5}$$

and our problem is now to find the coefficients c_i in this expression.

The obvious option is to pack the basis vectors $\mathbf{v}_i$ into the columns of a matrix $\mathbf{A}$, and find solutions of

$$\mathbf{Ac} = \mathbf{w}$$

Since the columns of $\mathbf{A}$ are formed from basis vectors they are linearly independent and a non-zero solution exists and can be determined by the usual methods.

However, our basis vectors form an orthogonal set (collection) which permits a more direct calculation. Consider a particular vector $\mathbf{v}_k$ in our basis set, and let's take the dot product between $\mathbf{v}_k$ and our vector $\mathbf{w}$:

$$\mathbf{v}_k^T\mathbf{w} = \mathbf{v}_k^T\sum_{i=1}^{m} c_i\mathbf{v}_i \tag{5.6}$$

Distributing the dot product into the summation we have:

$$\mathbf{v}_k^T\mathbf{w} = \sum_{i=1}^{m} c_i\mathbf{v}_k^T\mathbf{v}_i \tag{5.7}$$

But from orthogonality we know that the dot product of any two different vectors in our orthonormal set is zero, so all terms in the sum where $k \neq i$ are zero. This leads to the following simplification

$$\mathbf{v}_k^T\mathbf{w} = c_k\mathbf{v}_k^T\mathbf{v}_k \tag{5.8}$$

In addition, since our set of vectors is normalized, we know that $\mathbf{v}_k^T\mathbf{v}_k = 1$, leaving us with

$$\mathbf{v}_k^T\mathbf{w} = c_k \tag{5.9}$$

This gives us a very nice, simple way of decomposing a vector into a linear combination of the vectors within our basis set. The dot product of each basis vector with our target vector will result in the coefficient of that term in the linear decomposition.

Exercise 5.6

1. There are many (in general, an infinite number) of bases for a given set V. Hence, we can describe elements in the set V as linear combinations of vectors from different bases. Consider

the following two basis sets which form bases for 2-dimensional space.

- $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and
- $\mathbf{u}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, $\mathbf{u}_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Express the vector $\mathbf{w} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ as a linear combination of the first basis set (i.e., a sum of scaled versions of each vector in the basis set). Repeat for the second. Please make two different drawings of $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$, one expressed as a sum of scaled vectors in the first basis set and another for the vectors from the second basis set. Please label the lengths of each vector in the set.

2. Suppose that you wish to write the vector $\mathbf{w} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$ as a linear combination of the vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \ \mathbf{v}_2 = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \text{ and } \mathbf{v}_3 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

Please write a matrix equation to find the coefficients of the linear combination, and solve for the coefficients using MATLAB if possible.

3. Representing vectors using different bases is a very powerful technique that we will keep coming back to in this class (in both semesters). Vectors described in different bases can give us insight that may not be so obvious when viewed in the original basis. Representing vectors in different bases can also be used for dimensionality reduction, which is an important technique that is used to speed up computations and compress data in a number of different fields. Here we will consider a problem of lossy data compression using a change of basis. Lossy compression refers to methods of representing data more efficiently, but with a loss of accuracy. Examples of lossy data compression include jpg images, and mp3 audio files. If care is taken in lossy compression, the effects of the data loss can be kept at acceptable levels (this is of course subjective and dependent on the application). We will start with a toy example and then move to more complicated ones in subsequent homework problems. Consider a set of four 2-dimensional data variables stored in the following vectors:

$$\mathbf{d}_1 = \begin{bmatrix} 2.2 \\ 1.2 \end{bmatrix}, \mathbf{d}_2 = \begin{bmatrix} 1 \\ 0.6 \end{bmatrix}, \mathbf{d}_3 = \begin{bmatrix} 1.5 \\ 0.7 \end{bmatrix}, \mathbf{d}_4 = \begin{bmatrix} 1.7 \\ 0.8 \end{bmatrix} \tag{5.10}$$

a) In MATLAB, plot the data using points (without lines connecting them) by typing `plot([2.2 1 1.5 1.7],[1.2 0.6 0.7 0.8], 'o');` You will find that these points lie close to the line through the origin with slope 1/2.

b) Define a unit vector that points in the direction $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and call it $\mathbf{u}_1$. Find another unit vector that is orthogonal to $\mathbf{u}_1$ and call it $\mathbf{u}_2$. These vectors form a basis in 2 dimensional space.

c) Rather than storing the original data, we are now going to express the original data in terms of the new basis that we have defined. To do that, write $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ and $\mathbf{d}_4$, as a linear combination of $\mathbf{u}_1$ and $\mathbf{u}_2$. You can use MATLAB here to find the coefficients.

d) In this toy example, we are going to "compress" our data by only keeping the coefficients corresponding to $\mathbf{u}_1$. i.e. we will discard the coefficient corresponding to $\mathbf{u}_2$. Suppose that we wish to recover approximations to $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4$, from the four coefficients. These approximations, which you should denote by $\tilde{\mathbf{d}}_1, \cdots \tilde{\mathbf{d}}_4$, are all scaled versions of $\mathbf{u}_1$. In your axes from part a, please plot the points corresponding to $\tilde{\mathbf{d}}_1, \cdots \tilde{\mathbf{d}}_4$. Do you think they make good approximations?

e) We can describe how well our compressed data represents our original data. One way to do this is to calculate the difference between our original and compressed data, and call this error vector $\mathbf{f_i} = \mathbf{d}_i - \tilde{\mathbf{d}}_i$. Now, compute the size of this error using $norm(\mathbf{f_i})$ for $i = 1, 2, 3, 4$. Then, summarize the error by finding the root-mean-square (RMS) error between your approximations and the true data points. The RMS function squares the errors, takes the mean, and then takes the square root. This quantity is a single number that can be used to measure how well or poorly your compressed data represents your original data. You may find MATLAB's `norm` and `rms` functions helpful here.

This toy example illustrates that we can sometime be more efficient (albeit at the cost of some accuracy) in representing (or computing) data when it is expressed in certain bases.

Solution 5.2

1. Yes, a linear combination of two vectors which lie in the xy-plane will also lie in the xy-plane.
2. Let $\mathbf{A}$ be the matrix
$$\mathbf{A} = \begin{bmatrix} 1 & 1 & c_1 + c_2 \\ 1 & 2 & c_1 + 2c_2 \\ 0 & 0 & 0 \end{bmatrix}$$
and let $\mathbf{v}$ be an arbitrary 3×1 vector
$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$
Then the product
$$\mathbf{Av} = \begin{bmatrix} x + y + (c_1 + c_2)z \\ x + 2y + (c_1 + 2c_2)z \\ 0 \end{bmatrix}$$
lies in the xy-plane

Solution 5.3

1. Type the following into MATLAB:
```
» quiver3(0,0,0,1,1,1)
» hold on
» quiver3(0,0,0,1,2,1)
» quiver3(0,0,0,3,4,3)
```
2.
3. The determinant of $\mathbf{B}$ is zero. Recall that a matrix is not invertible if and only if the determinant is zero. This matrix is not invertible since it collapses all vectors to a plane.

Solution 5.4

1. a) They are linearly independent since they span $\mathbf{R}^3$.
   b) They are linearly dependent since the first vector is equal to the second vector plus two times the third vector.
   c) They are linearly dependent since the third vector is equal to the first vector plus two times the second vector.
   d) They are linearly dependent. You can have a maximum of n linearly independent vectors in $\mathbf{R}^n$.
   e) They are linearly independent since they do not lie on the same line.
2. The span of these two vectors is all over $\mathbf{R}^2$, i.e., a plane.

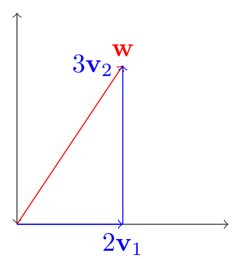
3. The span of these three vectors is the xy-plane in $\mathbf{R}^3$.

Solution 5.5

1. The dot product of these two vectors is non-zero, so they are not orthogonal.
2. The dot product of these two vectors is zero, so they are orthogonal.
3. The dot product of these two vectors is zero, so they are orthogonal. Furthermore, each vector is unit length, so they are orthonormal.

Solution 5.6

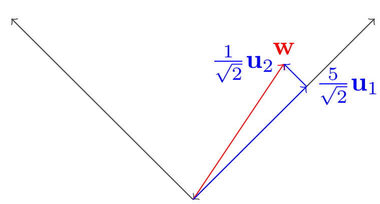
1. It's clear that $2\mathbf{v}_1 + 3\mathbf{v}_2 = \mathbf{w}$. We visualize this as



To write $\mathbf{w}$ as a linear combination of the basis vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ requires a bit more work. We can set up the matrix equation

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

and solve to learn that $\frac{5}{\sqrt{2}}\mathbf{u}_1 + \frac{1}{\sqrt{2}}\mathbf{u}_2 = \mathbf{w}$. We can visualize this as



2. First, we create a matrix in MATLAB whose columns are the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$,
» `V=[1 3 1; 1 1 2; 1 2 2]`
and the vector $\mathbf{w}$,
» `w=[1; 2; 4]`.
Let $\mathbf{c}$ be the vector of coefficients. We have the equation $\mathbf{V}\mathbf{c} = \mathbf{w}$, so to solve for $\mathbf{c}$ we compute $\mathbf{c} = \mathbf{V}^{-1}\mathbf{w}$. In MATLAB, we use » `inv(V)*w`. This tells us that $\mathbf{w} = -10\mathbf{v}_1 + 2\mathbf{v}_2 + 5\mathbf{v}_3$.

3. a)

   b) We define » `u1=[2; 1]` and » `u2=[-1; 2]`. There are other choices for $\mathbf{u}_2$, but they are all constant multiples of this choice, e.g., » `u2=[-2;4]`.

   c) Create a 2×2 matrix with $\mathbf{u}_1$ and $\mathbf{u}_2$ as the columns,
      » `U=[2 -1; 1 2]`
      and a 2×4 matrix the vectors $\mathbf{d}_i$ as the columns
      » `D=[2.2 1 1.5 1.7; 1.2 0.6 0.7 0.8]`.
      Then compute
      » `inv(U)*D`
      to get the matrix of coefficients. This tells us that

$$\mathbf{d}_1 = 1.12\mathbf{u}_1 + 0.04\mathbf{u}_2,\ \mathbf{d}_2 = 0.52\mathbf{u}_1 + 0.04\mathbf{u}_2,$$

$$\mathbf{d}_3 = 0.74\mathbf{u}_1 - 0.02\mathbf{u}_2,\ \text{and}\ \mathbf{d}_1 = 0.84\mathbf{u}_1 - 0.02\mathbf{u}_2.$$

Chapter 6

Night 3: Linear Systems of Algebraic Equations

Learning Objectives

Concepts

- Determine for a system of 3 or fewer unknowns whether it has a unique solution, no solution or infinite solutions.
- Create a set of linear equations from a narrative about how the unknown variables are related to given data.
- Represent a system of linear equations with matrix, vector notation
- Solve a linear system of equations

MATLAB skills

- Compute the determinant of a matrix
- Solve systems of linear equations of the form $\mathbf{Ax} = \mathbf{b}$ using all three methods: inverse matrix, linsolve, or backslash operator.

Suggested Approach

See Night 1 for suggested approaches to the assignment and list of resources.

6.1 Determinants and Invertibility

You have already encountered the determinant in class: the determinant of a square matrix is a property of the matrix which among other things indicates whether a matrix is invertible or not: if the determinant of a square matrix is zero, it is non-invertible. As a reminder:

The determinant of a matrix $\mathbf{G}$ is denoted a few different ways.

$$\det(\mathbf{G}) = |\mathbf{G}| \tag{6.1}$$

For a generic 2×2 matrix $\mathbf{G}$

$$\mathbf{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

the formula for the determinant is quite straightforward:

$$\det(\mathbf{G}) = ad - bc \tag{6.2}$$

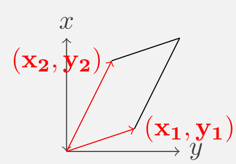
For example, for the following 2×2 matrix,

$$\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right) = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = (1)(4) - (2)(3) = -2 \tag{6.3}$$

You already considered the determinant of some transformation matrices, now let's consider what the determinant is really telling us about a general matrix.

Exercise 6.1

1. Let $\mathbf{A}$ be a 2×2 matrix
$$\mathbf{A} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}.$$
We can think of the columns of $\mathbf{A}$ as two vectors beginning at the origin and ending at the points (x_1, y_1) and (x_2, y_2), respectively. These vectors form a parallelogram, as shown here:



Show that the magnitude (i.e., absolute value) of $\det(\mathbf{A})$ is equal to the area of a parallelogram formed by the column vectors of the matrix $\mathbf{A}$.

2. What is the determinant of $\mathbf{A}$ if its column vectors are on the same line? Graphically, what happens to the parallelogram?

From this, you should get the feeling for the fact that the determinant is a measure of how co-linear the columns of $\mathbf{A}$ are: or in other words, how linearly independent the two columns are. The determinant therefore lets us know quickly if a linear system of algebraic equations has a solution, as illustrated in the following example.

Exercise 6.2

Consider the following matrix whose columns lie on the same line: the second column is simply twice the first column.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \tag{6.4}$$

1. What is $\det(\mathbf{A})$?
2. Find all the solutions to $\mathbf{Ax} = \mathbf{0}$.
3. For which vectors $\mathbf{b}$ does $\mathbf{Ax} = \mathbf{b}$ have a solution? Why are there only certain $\mathbf{b}$ vectors that lead to solutions to $\mathbf{Ax} = \mathbf{b}$?

While the formula for the determinant of a 2×2 matrix is quite straightforward, the procedures for computing the determinant of larger matrices is more difficult, but they are well known and well documented. Fortunately, MATLAB has the `det` function which computes the determinant.

6.2 Linear Systems of Algebraic Equations: Formulation and Definition

In previous classes, you've encountered a bunch of exercises where you had to operate on a vector to find another vector:

$$\mathbf{Ax} = \mathbf{b}, \tag{6.5}$$

where $\mathbf{A}$ and $\mathbf{x}$ were known, and your job was to find $\mathbf{b}$. While this is fun and, as you saw above in the rectangle exercise, can be useful, there is another related problem which is easily as important. It involves the same equation, but now you know $\mathbf{A}$ and $\mathbf{b}$ and need to find the vector $\mathbf{x}$. As we will discuss here, this problem captures the concept of a Linear System of Algebraic Equations.

One key idea in building models is the step of abstraction: going from some real-world situation to an abstracted model for the system (e.g., a set of differential equations). There are two important aspects of building such a model: first, deciding what to include or ignore, and second, deciding how to mathematically represent those things you choose to include.

One particularly common kind of mathematical framing is a set of linear algebraic equations, which can be represented by a matrix equation. A general system of m linear algebraic equations in n unknown variables $x_1, x_2, \ldots, x_n$ takes the form

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
\ldots &= \ldots \\
a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n &= b_m
\end{aligned}$$

where $a_{11}, a_{12}, \ldots, a_{mn}$ are known as coefficients and $b_1, b_2, b_3, \ldots, b_m$ are constants. We can write this using matrices and vectors in the form

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ is the $m \times n$ *coefficient matrix*, $\mathbf{x}$ is the $n \times 1$ unknown vector, and $\mathbf{b}$ is a $m \times 1$ constant vector which is known. In other words,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Note that "linear" here means linear in terms of the unknown variables, e.g., if x is an unknown there are only terms like ax, and no terms like $\sin(x)$, x^2, $1/x$, etc. It is often the case that you might have *coefficients* that appear to be non-linear; for example, in solving physics problems, you might have coefficients that depended on trig functions of angles, such as $(L\cos\theta)F_x$, which is is linear in F_x but not linear in θ). Be careful to be clear about what you're solving for when you decide whether something is linear or non-linear.

6.3 Using Matrix Inverses to Solve Linear Systems

Over the last week, you have worked with rotation matrices, and transformations that were compositions of simpler rotations, and learned how to invert them. When you multiply a vector by any matrix (not just ones that are associated with simple spatial transformations), you transform the original vector $\mathbf{x}$ into a new vector $\mathbf{b}$.

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

More generally (than rotations), you can *often* undo the linear transformation (just like you did with the rotation matrix). Undoing this linear transformation is a linear transformation itself! Therefore the act of undoing a linear transformation can be formulated with a matrix multiply.

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$
$$\Rightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

This reduces our linear system of algebraic equations problem to the problem of finding the inverse of our matrix $\mathbf{A}$. Note this is only possible if $\mathbf{A}$ is square and *invertible*.

When solving a system of equations, at least half of the battle is typically getting your system abstracted to the point that it can be thought of as a system of linear equations. The following are a set of problems. You don't need to solve these problems – you just need to formulate them as linear algebra problems.

An Investment Example

In this section we will focus on deciding whether and how you can abstract the system to a mathematical model that can be written as a matrix equation.

Exercise 6.3

Suppose that the following table describes the stock holdings of three of the QEA instructors. Also suppose that on a given day the value of the Apple, IBM and General Mill's stock are \$100, \$50 and

$20 respectively.

	Apple	IBM	General Mills
Jeff	100	100	100
Emily	100	200	0
John	50	50	200

1. *Here's your first linear algebra formulation question:* What is the total value of the holdings for each professor on the day in question? Can you formulate this as a matrix expression? If so, what is it? If not, why not?

2. Now, suppose that you do not know how many shares of each stock are owned by the instructors. However, you know that the total value of the stocks for each instructor for three consecutive days is as given in the following table

	Jeff	Emily	John
Day 1	$1500	$2600	$950
Day 2	$1600	$2810	$1020
Day 3	$1400	$2550	$1000

You also know that the price of each stock on each of the three days was as follows:

	Apple	IBM	General Mills
Day 1	$100	$50	$20
Day 2	$110	$50	$22
Day 3	$100	$40	$30

Now here's the second formulation question: how many stocks of each company does each professor own? Can you formulate this as a matrix equation? If so, what are the matrices/vectors? If not, why not?

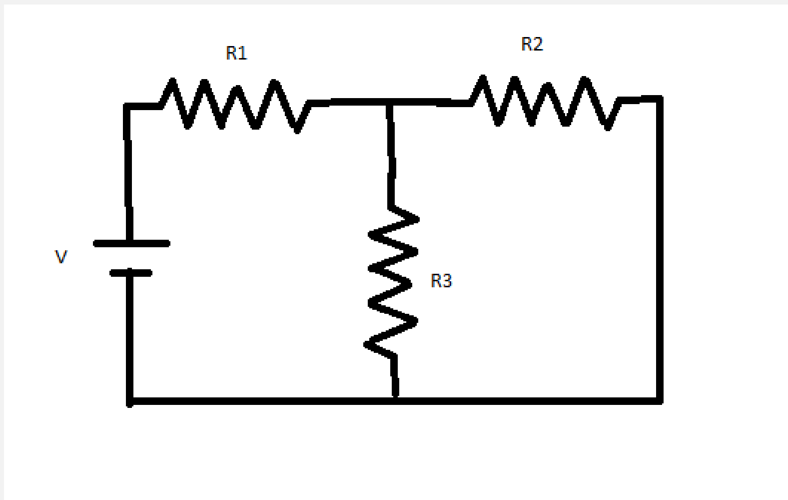
An Electrical Example

Remembering your circuit analysis back from ISIM, recall that Kirkhoff's laws:

- Kirkhoff's Voltage Law says that the sum of all the voltage drops around any loop of a circuit must sum to zero. (Batteries contribute a voltage increase of V, resistors contribute a voltage drop of IR.)
- Kirkhoff's Current Law says that the sum of all current going into and out of any junction of wires in the circuit must be zero.

Exercise 6.4

In the following circuit, consider that there is a current I_1 going through resistor R_1, a current I_2 going through resistor R_2 and a current I_3 going through resistor R_3. Find a linear algebra expression for the vector of our three unknown currents.



6.4 Types of Linear Systems and Types of Solutions

Consider the linear system of algebraic equations expressed in matrix-vector form as,

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

If $\mathbf{b} = \mathbf{0}$ the system of linear algebraic equations is *homogeneous* and if $\mathbf{b} \neq \mathbf{0}$ the system is *non-homogeneous*. As mentioned before, we've already dealt with systems like this before when we were transforming geometrical objects, but in that case we already knew $\mathbf{x}$ and we were simply multiplying by $\mathbf{A}$ in order to get $\mathbf{b}$. Here, we are considering the so-called *inverse* problem, and trying to find $\mathbf{x}$ given $\mathbf{A}$ and $\mathbf{b}$. However, let's back up and consider some small examples to explore the solution possibilities a little.

Elimination of Variables

In high school you probably learned some basic techniques for solving small linear systems of algebraic equations. Consider the following linear system of algebraic equations,

$$2x_1 + 3x_2 = 6 \tag{6.6}$$
$$4x_1 + 9x_2 = 15 \tag{6.7}$$

The basic technique, called *elimination of Variables*, proceeds as follows: First, solve equation (2) for x_1

$$x_1 = 3 - \frac{3}{2}x_2 \tag{6.8}$$

Now substitute this expression for x_1 into equation (3)

$$4(3 - \frac{3}{2}x_2) + 9x_2 = 15$$

Now we simplify this equation

$$\begin{aligned} 12 - 6x_2 + 9x_2 &= 15 \\ \Rightarrow 3x_2 &= 3 \end{aligned}$$

and solve for x_2 to give $x_2 = 1$. Now we substitute this solution back into equation (2) or (4) to determine $x_1 = \frac{3}{2}$. The original linear system of algebraic equations therefore has a unique solution, $\mathbf{x} = \begin{bmatrix} 1 \\ 3/2 \end{bmatrix}$.

However, not all linear systems of algebraic equations have a unique solution. For example, the system

$$x_1 + 2x_2 = 1 \tag{6.9}$$
$$2x_1 + 4x_2 = 2 \tag{6.10}$$

has an infinite number of solutions because equation (6) is just a multiple of equation (5). Solving equation (5) for x_1 gives

$$x_1 = 1 - 2x_2$$

and choosing an arbitrary value of $x_2 = \alpha$ gives

$$\begin{aligned} x_1 &= 1 - 2\alpha \\ x_2 &= \alpha \end{aligned}$$

or in vector form

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

This defines an infinite number of solutions since α is any real number. What do you notice about each part of this vector?

It's also possible that a linear system of algebraic equations has no solution. For example, the system

$$x_1 + 2x_2 = 1 \tag{6.11}$$
$$2x_1 + 4x_2 = 1 \tag{6.12}$$

has no solution. Solving equation (8) for x_2 gives

$$x_2 = \frac{1}{4} - \frac{1}{2}x_1$$

and replacing into equation (7) gives

$$x_1 + 2(\frac{1}{4} - \frac{1}{2}x_1) = 1$$

which on simplification gives

$$\frac{1}{2} = 1$$

which hopefully we all agree is incorrect. We assumed that there was a solution, performed elimination and substitution and found a statement that contradicts our assumption: no solution therefore exists.

Exercise 6.5

1. Using the technique of elimination of variables described above, determine which values of h and k result in the following system of linear algebraic equations having (a) no solution, (b) a unique solution, and (c) infinitely many solutions?

$$\begin{aligned} x_1 + hx_2 &= 1 \\ 2x_1 + 3x_2 &= k \end{aligned}$$

2. Using the technique of elimination of variables described above, determine whether the following linear systems of algebraic equations have zero, one, or infinitely many solutions. If solution(s) exist, determine the actual solution(s).

a)

$$\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ x_2 + x_3 &= 2 \\ x_1 - 2x_3 &= 4 \end{aligned}$$

b)

$$\begin{aligned} x_1 + x_2 + x_3 &= -6 \\ 2x_1 + x_2 - x_3 &= 18 \\ x_1 - 2x_3 &= 4 \end{aligned}$$

c)

$$\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ 2x_1 + x_2 - x_3 &= 10 \\ x_1 - 2x_3 &= 4 \end{aligned}$$

Solving a linear system of algebraic equations in MATLAB

Exercise 6.6

In the last class, you worked with an example of fruits in your refrigerator, and we asked you questions like how to calculate the total weight of the fruits, how many fruits there are, etc. We can use matrix operations to calculate *inverse problems* as well, as this question illustrates. Suppose that you know that you have apples and oranges in the fridge and that in the genetically engineered future, the weights of all apples are 3oz and all oranges are 4oz. Because of inflation in this genetically engineered future, the price of each apple is \$1 and the price of each orange is \$2. Suppose that you also know that you paid \$13 total for your fruit and the total weight of the fruit is 33 oz. We can use this information and tools we have developed to figure out how many apples and oranges we have. Let n_o and n_a be the numbers of oranges and apples in your fridge respectively, and that you don't know what these numbers are. Define the following vectors

$$\mathbf{n} = \begin{bmatrix} n_o \\ n_a \end{bmatrix} \tag{6.13}$$

$$\mathbf{d} = \begin{bmatrix} 13 \\ 33 \end{bmatrix} \tag{6.14}$$

1. Write an equation relating $\mathbf{n}$ and $\mathbf{d}$, using a matrix-vector product.
2. Calculate how many oranges and apples you have.
3. Why this kind of problem is often called an inverse problem?

Exercise 6.7

1. Consider the example with the fruits that you worked out earlier. Now, in addition to apples and oranges, suppose you also had an unknown number of pears which each weigh 3 oz, and cost \$3. Additionally, suppose that the total weight of the fruits is 45 oz, and you paid a total of \$21 for the fruit.
   a) If possible find the numbers of oranges, apples and pears. If not, please explain why.
   b) Suppose that you additionally know that you have a total of 14 fruits. Can you formulate and solve a matrix-vector equation to find out the numbers of oranges, apples and pears you have?
   c) What is the determinant of the matrix you have set up to solve this?
2. The fruit vendors bought the pricing algorithm from Uber. Oranges are still \$2, pears are now only \$1.50, and (due to an influx of teachers) apples are now surging at \$1.50 each. Their

weights stay the same. You return to the market, and again purchase 14 fruits, which have the same total weight and total cost.

a) Can you formulate and solve a matrix-vector equation to find out the numbers of oranges, apples and pears you have?
b) What is the determinant of the matrix you have set up to solve this?

3. Recall the example with fruits from class: Suppose that you have a total number of 14 apples, oranges and pears in your fridge. Suppose that each apple costs \$1, each orange costs \$ 2 and each pear costs \$3. Assume also that the weights of every apple is 3 oz, every orange is 4 oz and every pear is 3 oz. Additionally, suppose that the total weight of the fruits is 45 oz, and you paid a total of \$21 for the fruit.

   a) Formulate (or look up your formulation from class) and write down (but don't solve it yet) a matrix-vector equation to find out the numbers of oranges, apples and pears you have.
   b) Solve this equation to find the numbers of apples, oranges and pears using the following approaches (they will of course give you the same results, but we want you to get familiar with using the different operations here).
      i. Using MATLAB, compute the inverse of the matrix in part a and use it to find the numbers of apples, oranges and pears.
      ii. Use MATLAB's `linsolve` function to find the numbers of apples, oranges and pears.
      iii. Use MATLAB's $\backslash$ operator to find the numbers of apples, oranges and pears.

6.5 Conceptual Quiz

1. Select the matrices which are invertible.

   a) $\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$

   b) $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$

   c) $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

   d) $\begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix}$

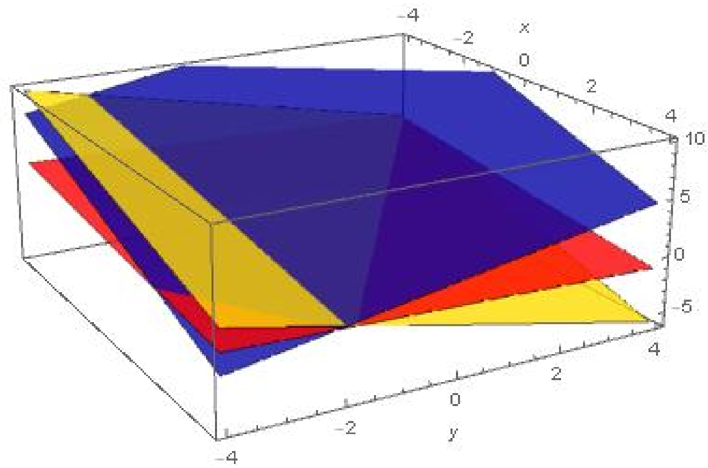
   e) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

2. Let (a, b, c) be the point of interesection for the following three planes, pictured below:

$$z = 2 - x - y$$

$$z = (31 - 6x + 4y)/5$$

$$z = (13 - 5x - 2y)/2$$



What is a?

3. How many solutions does the following system of equations have?

$$x + y = 9$$

$$x - z = 2$$

$$y + z = 7$$

A. Zero
B. One
C. Two
D. Infinitely many

4. What is the area of a parallelogram whose vertices are $(0, 0)$, $(2, 4)$, $(5, 1)$ and $(7, 5)$?

5. Solve the following system of linear equations

$$x - y = 2$$

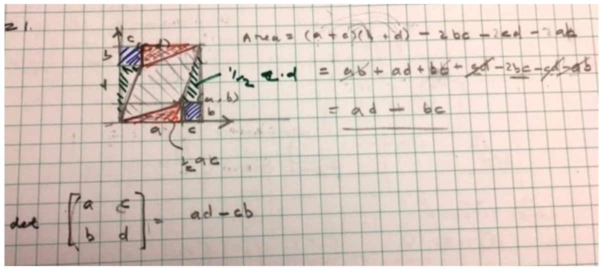
$$3x + z = 11$$

$$y - 2z = -3$$

What is the value of y?

Solution 6.1

1.

2. 

3. The determinant is equal to 0, or det($\mathbf{A}$)=0.

Solution 6.2

1. det($\mathbf{A}$)=(1)(4)-(2)(2)=0

2. There are infinitely many solutions of the form $-x_1 = 2x_2$.

3. Solutions are of the form $\mathbf{b} = \begin{bmatrix} k \\ 2k \end{bmatrix}$ where k is a constant.

Solution 6.3

1. This can be formulated as $\mathbf{Ax} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} 100 & 100 & 100 \\ 100 & 200 & 0 \\ 50 & 50 & 200 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} 100 \\ 50 \\ 20 \end{bmatrix}, \text{ and } \mathbf{b} = \begin{bmatrix} d_{jd} \\ d_{et} \\ d_{jg} \end{bmatrix}.$$

   Doing the matrix multiplication shows that Jeff has $d_{jd} = 17000$, Emily has $d_{et} = 20000$, and John has $d_{jg} = 11500$.

2. There are several ways to do this. Perhaps the simplest is to compute each person's stock holding individually. To do this, we let $\mathbf{A}$ be a matrix with the stock prices

$$\mathbf{A} = \begin{bmatrix} 100 & 50 & 20 \\ 110 & 50 & 22 \\ 100 & 40 & 30 \end{bmatrix},$$

   let $\mathbf{b}_{jd}$ be a vector representing the value of Jeff's stocks on each day,

$$\mathbf{b}_{jd} = \begin{bmatrix} 1500 \\ 1600 \\ 1400 \end{bmatrix},$$

and let $\mathbf{x}_{jd}$ be a vector representing Jeff's stock holdings (i.e., the first entry tells us how many stocks of Apple he has, the second entry is IBM, and the third is General Mills). This gives the equation $\mathbf{A}\mathbf{x}_{jd} = \mathbf{b}_{jd}$. By inverting A we can solve for $\mathbf{x}_{jd}$. Then we repeat this procedure for each of the other instructors.

But... we can do it quicker! Form a 3×3 matrix $\mathbf{X}$ whose columns are made the vectors $\mathbf{x}_{jd}, \mathbf{x}_{et}$, and $\mathbf{x}_{jg}$. Then form a 3×3 matrix $\mathbf{B}$ whose columns are made of the vectors $\mathbf{b}_{jd}, \mathbf{b}_{et}$, and $\mathbf{b}_{jg}$. This gives the equation $\mathbf{AX} = \mathbf{B}$. Inverting A, we can solve for $\mathbf{X}$:

	Jeff	Emily	John
Apple	10	20	5
IBM	10	10	5
General Mills	0	5	10

Solution 6.4

$$\begin{bmatrix} R_1 & 0 & R_3 \\ 0 & -R_2 & R_3 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix}$$

Solution 6.5

1. a) h=3/2, k≠2,
   b) h≠3/2,
   c) h=3/2, k=2
2. a) $x = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix}$
   b) No Solution
   c) Infinite Solutions

Solution 6.6

1. $\begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} n_o \\ n_a \end{bmatrix} = \begin{bmatrix} 13 \\ 33 \end{bmatrix}$
2. $\begin{bmatrix} n_o \\ n_a \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$
3. In this case we know the result $\mathbf{b}$, and are working backwards to find the number of apples and oranges. We also use a matrix inverse to find the result.

Solution 6.7

1. a) No, you have three unknowns and only two equations.

b) Yes, you now have three equations and three unknowns.

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \\ 14 \end{bmatrix}$$

$$\begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 2 \end{bmatrix}$$

c) $det(\mathbf{A}) = 2$

2. a)

$$\begin{bmatrix} 2 & 1.50 & 1.50 \\ 4 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \\ 14 \end{bmatrix}$$

This forumation cannot be solved because A is not invertible.

b) $det(\mathbf{A}) = 0$

3. a)

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 21 \\ 45 \\ 14 \end{bmatrix}$$

b) i.

$$\begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 2 \end{bmatrix}$$

ii.

$$\begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 2 \end{bmatrix}$$

iii.

$$\begin{bmatrix} n_o \\ n_a \\ n_p \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 2 \end{bmatrix}$$

Chapter 7

Day 4: Linear Systems of Algebraic Equations

7.1 Schedule

- 0900-0915: Debrief
- 0915-1000: Synthesis
- 1000-1030: Applications of LSAE
- 1030-1045: Coffee
- 1045-1115: Applications of LSAE
- 1115-1200: Concept Map for Eigenfaces

7.2 Debrief

- Please discuss your overnight work with your table-mates, create a set of key concepts, and a set of ideas that you are still confused by.

7.3 Synthesis

We will increasingly use a computational tool like MATLAB to compute determinants, matrix inverses, and the solutions to linear systems of algebraic equations. In this synthesis section we will explore the theoretical foundation of these algorithms - the so-called LU decomposition.

Gaussian Elimination

The basic process of *elimination of variables* can be formalized and is known as Gaussian Elimination. Here will briefly introduce it but you can consult other sources, such as the *Gaussian Elimination* page at WolframMathWorld for more details.

Rather than writing equations, we can cast a LSAE in matrix form and perform *Gaussian Elimination* on the augmented matrix $[\mathbf{A}\ \mathbf{b}]$.

For example, the linear systems of algebraic equations

$$\begin{aligned} 2x_1 + 3x_2 &= 6 \\ 4x_1 + 9x_2 &= 15 \end{aligned}$$

can be written as the following augmented matrix

$$\begin{bmatrix} 2 & 3 & 6 \\ 4 & 9 & 15 \end{bmatrix}$$

Thinking now in terms of rows, we replace the second row with row 2 - 2 row 1 to give

$$\begin{bmatrix} 2 & 3 & 6 \\ 0 & 3 & 3 \end{bmatrix}$$

This matrix is now in so-called *echelon* form: we can find the solution to the original LSAE by first solving the equation implied by the last row and then back-substituting into the equation implied by the previous row.

Exercise 7.1

1. Set up the augmented matrix for the following example (you will recognise this from the last assignment)

$$\begin{aligned} 2x_1 + x_2 &= 13 \\ 4x_1 + 3x_2 &= 33 \end{aligned}$$

and perform *Gaussian Elimination* to reduce the augmented matrix to *echelon form*. Interpret the resulting system and determine the solution(s).

LU Decomposition

The steps used to solve a LSAE using Gaussian Elimination can also be used to *decompose* a matrix into a product of two matrices: a *lower-triangular* matrix $\mathbf{L}$ and an *upper-triangular* matrix $\mathbf{U}$. Here we will briefly introduce it but you could consult other sources, such as the *LU Decomposition* page at WolframMathWorld for more details.

In Gaussian Elimination we execute a set of row operations. In our ongoing example, we replaced row 2 with the result of row 2 - 2 row 1. This action can be neatly represented in terms of a matrix operation. Let's multiply the original matrix equation $\mathbf{Ax} = \mathbf{b}$ with the transformation matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$$

to form $\mathbf{MAx} = \mathbf{Mb}$. Note that this transformation leaves row 1 of $\mathbf{A}$ unchanged, and it replaces the row 2 with row 2 - 2 row 1. The product $\mathbf{MA}$ is therefore an *upper-triangular* matrix $\mathbf{U}$

$$\mathbf{U} = \begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix}$$

and the LSAE is now expressed as $\mathbf{Ux} = \mathbf{Mb}$. If we now multiply this expression by $\mathbf{M}^{-1}$ we obtain

$$\mathbf{M}^{-1}\mathbf{Ux} = \mathbf{b}$$

The inverse of $\mathbf{M}$ is straight-forward to write down because it "undoes" the row operations

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Notice that this matrix is just a *lower-triangular* matrix $\mathbf{L}$. The LSAE now reads

$$\mathbf{LUx} = \mathbf{b}$$

We have therefore *decomposed* the original matrix $\mathbf{A}$ into the product of $\mathbf{L}$ and $\mathbf{U}$,

$$\mathbf{A} = \mathbf{LU}$$

How does this help, you might be asking? First of all, knowing the decomposition of $\mathbf{A}$ into $\mathbf{LU}$ allows us to solve the original LSAE $\mathbf{Ax} = \mathbf{b}$. Here is how.

Let's define a new vector $\mathbf{y} = \mathbf{Ux}$. Then the original LSAE can be expressed as

$$\mathbf{Ly} = \mathbf{b}$$

which is straight-forward to solve by *forward-substitution* because $\mathbf{L}$ is *lower-triangular*,

$$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \end{bmatrix}$$

and the solution for $\mathbf{y}$ is $y_1 = 6$, $y_2 = 3$. We can now solve $\mathbf{Ux} = \mathbf{y}$ for $\mathbf{x}$ using *forward-substitution* because $\mathbf{U}$ is *upper-triangular*,

$$\begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

and the solution for $\mathbf{x}$ is $x_1 = 1$, $x_2 = 3/2$.

Second of all, and more importantly, knowing the decomposition of $\mathbf{A}$ into $\mathbf{LU}$ allows us to solve any LSAE involving $\mathbf{A}$. Need to solve the LSAE with a different $\mathbf{b}$? No problem, just use the $\mathbf{LU}$ decomposition that you already computed and away you go. No need to redo all the steps of *Gaussian Elimination* just because $\mathbf{b}$ changed. Need to solve a LSAE for lots of different $\mathbf{b}$'s? No problem, just use the $\mathbf{LU}$ decomposition that you already computed and away you go. Finally, if you want to compute the inverse or determinant of a matrix this is easy too using LU decomposition as we show next.

There is an algorithm in MATLAB, *lu*, which does LU decomposition for you, but you should not necessarily expect to get the same $\mathbf{L}$ and $\mathbf{U}$, even for this example. (There are a variety of ways to define the $\mathbf{L}$ and $\mathbf{U}$ matrices, but this is beyond the scope of this section.)

Exercise 7.2

1. Consider the appropriate matrix from the last exercise and perform *LU Decomposition.* Check your answer by confirming that $\mathbf{A} = \mathbf{LU}$. (Please note that you perform LU decomposition on the original matrix $\mathbf{A}$, not the augmented matrix.)

Determinant

The basic algorithm for computing a determinant of $\mathbf{A}$ is to first perform LU decomposition, and make use of the following property:

The determinant of an upper-triangular or lower-triangular matrix is just the product of the diagonal entries.

We already met another property of determinants, namely that the determinant of a product is just the product of the determinants. Therefore, $det(\mathbf{A}) = det(\mathbf{L})det(\mathbf{U})$, each of which is just the product of the diagonal entries.

Exercise 7.3

1. Consider the appropriate matrix from the last exercise and find the determinant using the LU decomposition previously determined. Check your answer using *det* in MATLAB.

Inverse

The basic algorithm for computing the inverse of $\mathbf{A}$ is to first perform LU decomposition, and make use of the following idea. $\mathbf{B}$ is the inverse of $\mathbf{A}$ if it satisfies the following property

$$\mathbf{AB} = \mathbf{I}$$

The columns of $\mathbf{B}$ are just the solutions of a LSAE with a different $\mathbf{b}$. For example, in the two by two case we can solve

$$\mathbf{Ax} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and then

$$\mathbf{Ax} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and if we fill the columns of $\mathbf{B}$ with the solution to these LSAE we will have constructed the inverse. Since we already have the LU decomposition of $\mathbf{A}$ we simply solve each case using the technique already presented.

For example, the first column of $\mathbf{B}$ is determined as follows: First we solve $\mathbf{Ly} = \mathbf{b}$

$$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

to give $y_1 = 1$ and $y_2 = -2$. Now we solve $\mathbf{Ux} = \mathbf{y}$

$$\begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

and the solution for $\mathbf{x}$ is $x_1 = 3/2$, $x_2 = -2/3$. This is the entries in the first column of the inverse. Repeating this process for $\mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ will give the second column of the inverse which now reads

$$\mathbf{A}^{-1} = \begin{bmatrix} 3/2 & -1/2 \\ -2/3 & 1/3 \end{bmatrix}$$

Exercise 7.4

1. Consider the appropriate matrix from the previous exercise and find the inverse using the LU

decomposition previously determined. Check your answer using *inv* in MATLAB.

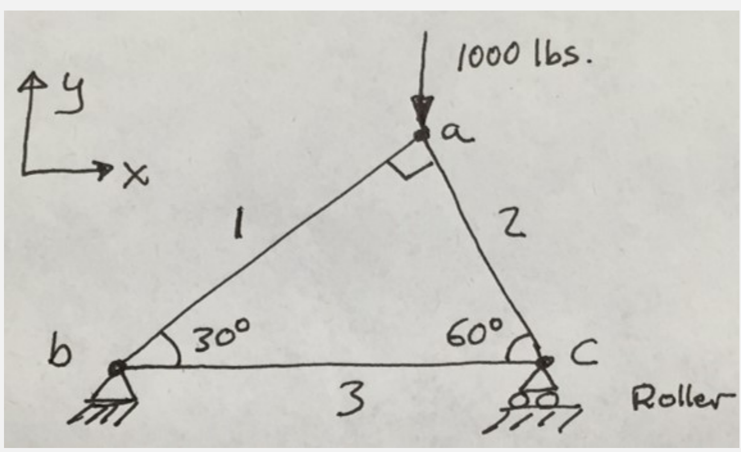
7.4 Applications of LSAE

Choose at least one of the following problems involving linear systems of algebraic equations.

Truss Analysis

Exercise 7.5

Systems of linear equations often come up in engineering when evaluating the strength and stability of structures under load. A *truss* is a simplified model of a structure. It consists of a collection of straight, rigid elements or sections that are long compared to the dimensions of their cross-section. Sections are connected only at their ends through frictionless, pin joints (Remember them? They can only constrain translation but not rotation, i.e., they can only apply force but not moments to a section). This means that sections of a truss are either in tension or compression (axial forces along its length). The roller can be assumed to be frictionless, and thus only exerts normal force. In analyzing trusses it is often assumed that the weight of the sections (dead load) is relatively small, and can therefore be neglected. The method of joints is a classic technique for determining the forces acting on all of the sections of a truss that is in static equilibrium. Here are the steps:



1. Draw a free body diagram for every pin in the truss. Note that the forces acting at the pin have to be in the directions implied by the things the pin is attached to!
2. Write out the equations of static equilibrium, $\sum \vec{F} = 0$, for every one of the pins. Note that some of your forces will be known forces (e.g., external loads), and some will be unknown reaction forces.

3. Express these equations in the matrix form $\mathbf{Ax} = \mathbf{b}$.

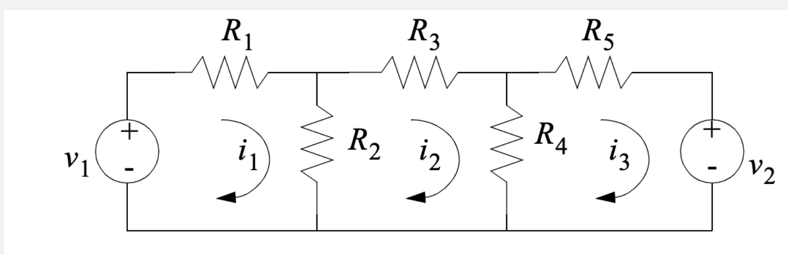
4. Evaluate whether the system is statically determinate or not. Note the connection to types of solutions to linear equations here: if you look at the form of $\mathbf{A}$, you should be able to tell whether the system is statically determinate!

5. Find the solution.

Circuit Analysis

Exercise 7.6

Systems of linear equations naturally arise in circuit analysis, although very few courses on circuits use these anymore. They do, however, form the backbone of circuit design software tools. You've met the relevant physical ideas/models before which are based on Kirchoff's circuit laws: the sum of currents into any node must be zero, and the sum of voltages around any loop must be zero. For the circuit shown in the figure:



1. Set up the linear system of algebraic equations required to solve for the three unknown currents (assuming that the resistors and the voltage sources are known.)

2. Find the solution if all of the resistors are 1 ohm, v_1 is 5 volts, and v_2 is -6 volts.

Chemical Analysis

Exercise 7.7

The complete combustion of propane, C_3H_8, with oxygen, O_2 yields carbon dioxide, CO_2, and water, H_2O. Based on conservation of mass, this reaction can be written as

$$a(C_3H_8) + b(O_2) \rightarrow c(CO_2) + d(H_2O)$$

Determine the coefficients in the combustion equation. Note that you will need to learn how to "balance" a chemical reaction.

7.5 *Concept Map for Eigenfaces*

For the facial recognition project we will be primarily focusing on an early facial recognition software algorithm, Eigenfaces, which is still used for face detection, and introduces some other concepts that are extremely important in both facial recognition and other tasks.

We would like you to spend some time developing an understanding of what you know, and what you don't know about facial recognition using Eigenfaces. A good way to do this is to break down the concept until you get to the point that you have terms that you *do* know:

1. Write the key term at the top or in the center. Circle it, since you don't know it.
2. Research it, and identify terms that are immediately associated with it. Write them down and connect them.
3. Circle new terms you don't understand, and break these down too.

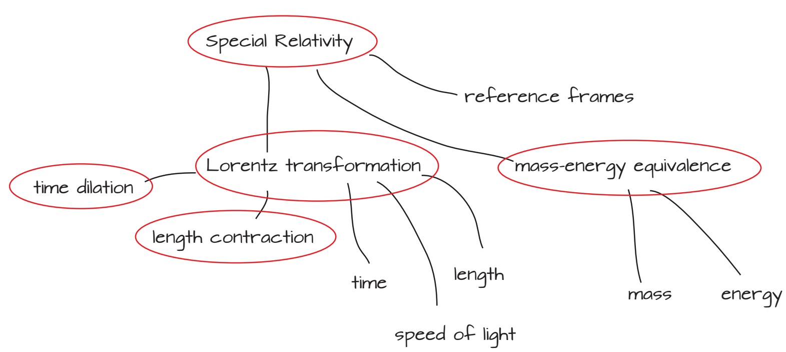


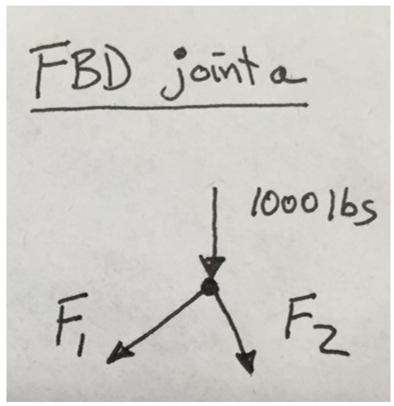
Figure 7.1: If you were trying to break down special relativity, a *portion* of your breakdown might look like this...

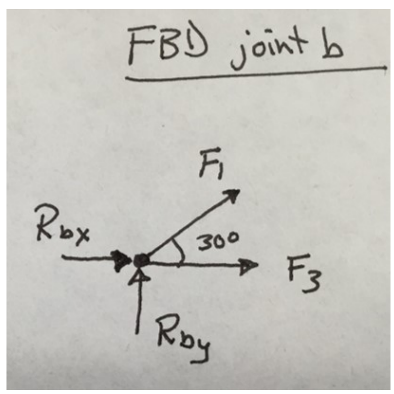
Once you've done your breakdown, try to make the following lists individually:

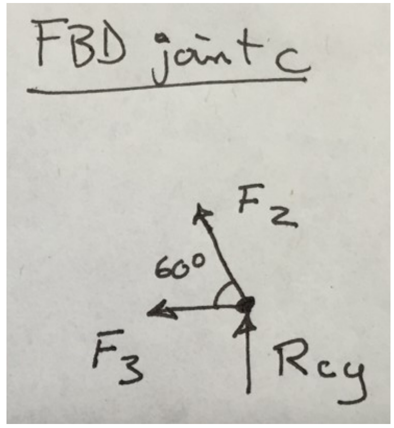
1. Relevant fundamental mathematical terms that I don't know
2. Relevant fundamental mathematical terms that I do know
3. Ideas specific to facial or image recognition that I don't know
4. Ideas specific to facial or image recognition that I do know

Solution 7.5

1. Note that the pin joint, b, cannot move in space so the ground must apply unknown reaction forces in both the x and y directions. The pin joint, c, is attached to a roller so it is free to move in the x direction but cannot move in the y direction. Therefore, the ground only applies a reaction force (unknown) in the y direction. For the entire truss, there is one known applied external force (1000 lbs) and six unknown axial and reaction forces (F_1, F_2, F_3, R_{bx}, R_{by}, and R_{cy}).







2. For each joint, $\sum F_x = 0$ and $\sum F_y = 0$. Thus we have the following six equations:

$$\begin{aligned}
-F_1 \cos 30 + F_2 \cos 60 &= 0 \\
-F_1 \sin 30 - F_2 \sin 60 &= 1000 \\
R_{bx} + F_3 + F_1 \cos 30 &= 0 \\
R_{by} + F_1 \sin 30 &= 0 \\
-F_3 - F_2 \cos 60 &= 0 \\
R_{cy} + F_2 \sin 60 &= 0
\end{aligned}$$

3. These equations can be written as $\mathbf{Ax} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} -\cos 30 & \cos 60 & 0 & 0 & 0 & 0 \\ -\sin 30 & -\sin 60 & 0 & 0 & 0 & 0 \\ \cos 30 & 0 & 1 & 1 & 0 & 0 \\ \sin 30 & 0 & 0 & 0 & 1 & 0 \\ 0 & -\cos 60 & -1 & 0 & 0 & 0 \\ 0 & \sin 60 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ R_{bx} \\ R_{by} \\ R_{cy} \end{bmatrix}$$

,

$$\mathbf{b} = \begin{bmatrix} 0 \\ 1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4. We have 6 unknowns, and 6 equations, so this is a determinate situation.

5.

$$\mathbf{x} = \begin{bmatrix} -500 \\ -866.03 \\ 433.01 \\ -5.6843e^{-14} \\ 250 \\ 750 \end{bmatrix}$$

Section 3 is the only one in tension (F_3 is positive). 1 and 2 should be in compression, and it makes sense that 2 has more compression than 1 because they support the same load, but 2 is more vertical. The horizontal reaction force is 0 because there is no net horizontal force on the system, and the sum of the vertical reaction forces is $1000\,lbf$ as we expect.

Solution 7.6

1.

$$\mathbf{A} = \begin{bmatrix} -R_1 & -R_2 & 0 & 0 & 0 \\ 0 & R_2 & -R_3 & -R_4 & 0 \\ 0 & 0 & 0 & R_4 & R_5 \\ 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix}$$

,

$$\mathbf{b} = \begin{bmatrix} -V_1 \\ 0 \\ V_2 \\ 0 \\ 0 \end{bmatrix}$$

2.

$$\mathbf{x} = \begin{bmatrix} 3.8750 \\ 1.1250 \\ 2.7500 \\ -1.6250 \\ -4.3750 \end{bmatrix}$$

Solution 7.7

$$C_3H_8 + 5O_2 \longrightarrow 3CO_2 + 4H_2O$$

Chapter 8

Night 4: Facial Recognition, Image Manipulation and Decomposition

Learning Objectives

Concepts

- Describe how a vector can be used to represent a data set.
- Explain how a matrix is used to represent multiple data sets.
- Explain what is meant by vectorizing a grayscale image.
- Predict the size of a vectorized image, given its pixel dimensions and color (gray or color).

MATLAB skills

- Convert a color image into a grayscale image.
- Convert an image to a matrix and back again

8.1 Ethics, Artificial Intelligence, and Facial Recognition

Face recognition is a technology with many possible applications. In just the past dozen or so years, the technology has gone from the stuff of science fiction to something that we interact with everyday (e.g., auto-tagging of images uploaded to social media). In this part of the assignment we are going to ask you to take a deep dive into how this technology manifests itself in the real world—often with mixed consequences for society.

This section is structured into three parts. First, we'll have you read about some of the issues that have been raised around face recognition technology (and more generally face analysis technology). Next, you'll read some frameworks that have been proposed to help mitigate the potential harm and maximize the benefits that might otherwise come from releasing poorly tested and biased AI systems. Finally, we'll have you branch out from face recognition technology to AI in general to examine which applications of the technology you think have the potential to most positively impact the world. You will discuss and synthesize your findings in class on Thursday, so make sure to take some sort of notes on what you read (there are also some specific prompts to respond to below).

Face Recognition Technology

Exercise 8.1

For a good overview of the issues, we'd like you to read Joy Buolamwini's written testimony that she then presented orally . You can pick whether you read the testimony or watch the video, although one nice thing about the written testimony is that it cites a lot of sources that you can read for more

information.
Based on this reading, generate a list of surprising insights (e.g., spurred by key quotes) that you gained. Also generate at least one discussion question.

Frameworks and Guidelines for Responsible Machine Learning

Exercise 8.2

Face recognition technology falls under the umbrella of machine learning. Machine learning is a field concerned with creating technologies that enable computers to learn to perform tasks automatically from experience (e.g., recognizing someone's identify from a picture of their face)—often by ingesting large training sets of labeled data. Sparked by a recognition that machine learning technologies were causing unanticipated harm in the real world, a lot of attention has been paid in recent years (both in industry and academia) to issues of fairness, accountability, and transparency. Here are two frameworks that have been created.

- Principles for Accountable Algorithms
- Google's Inclusive ML

Based on this reading, generate a list of surprising insights (e.g., spurred by key quotes) that you gained. Also generate at least one discussion question.
To get a sense of all of the conversations taking place around this topic, check out ACM's FAccT network of events.

Beyond Face Recognition

One thing that is important to mention at this point in the module is that while we are learning linear algebra and data analysis techniques within the context of face recognition, what you are learning can be applied to innumerable applications and fields of study. Even if we just stay within the realm of artificial intelligence, what you are learning now (and will learn later in the course) is the bedrock of many AI algorithms that are used in all sorts of applications. When learning about all of the issues that a technology like face recognition has, we find that students can sometimes have a tendency to move towards a nihilistic perspective on technology as a whole (e.g., all technology is bad / harmful). Critiquing technology and its role and effect in society is absolutely vital for *any* engineer. However, we contend that trying to understand how technology can be developed in a way that minimizes harm while maximizing benefit (e.g., the frameworks from the previous section) or by applying technology to problems or domains that have great potential for positive impact is also crucial. In this section, we are asking you to look into applications of image analysis (or artificial intelligence more generally) that have the potential for great positive impact on society.

Exercise 8.3

Find an article or paper about an application of artificial intelligence (it could be specifically about image or face analysis, but it need not be) that you think has the potential for great positive impact on society. Come to class ready to summarize the application and why you think it has the potential for positive impact. Unpack the notion of positive impact by specifying what the benefits (or downsides) would be of the application and who would reap them.
If you need some inspiration, here are some starting points (we are not claiming these are necessarily unambiguously positive, but they may provide some good starting points for your search).

- Automated diagnosis of cancer from medical images
- Automated, personalized education
- Optimizing energy use with artificial intelligence (more generally "Computational Sustainability")
- Sensing for driverless cars (e.g., pedestrian detection, road sign reading)
- Recognition and reading of text in a camera feed for people who are blind
- Automated wildlife monitoring via image analysis
- This one is kind of cheating. Olin 2nd year Austin Vesiliza put together a list of links to AI for social good projects that you might use for inspiration.

8.2 *Manipulating Images with Matrices*

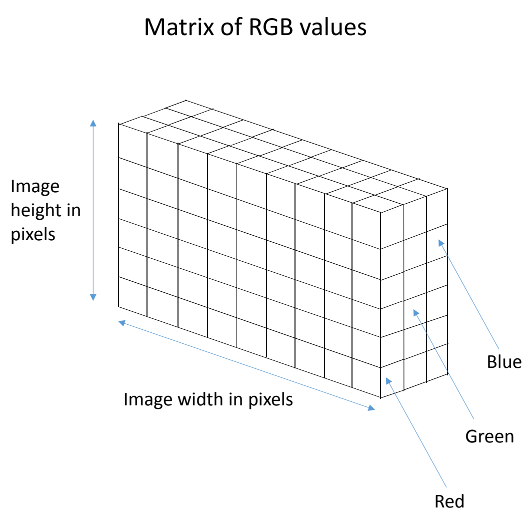


Figure 8.1: Anatomy of an RGB image array.

Exercise 8.4

Our next example is of an image pre-processing step that many of you would eventually do using built-in MATLAB functions before running your face detection algorithm.

1. Read an image file using MATLAB, and convert it to double precision numbers (the data format that MATLAB uses by default for vectors and matrices) using the following code:

```
X = imread('giraffe.jpg');
```

   (If you get an error, try re-typing the apostrophes.)

2. Color images are stored in a 3-dimensional array (as opposed to matrices, which are 2-dimensional arrays) in MATLAB. Compare this to the smiley face image you saw in class which was a matrix whose entries are the gray-scale values. Here, instead of gray-scale values, the color information is stored in Red, Green and Blue entries of the three-dimensional array. Therefore, each pixel in the image is associated with three different values which indicate how much of Red, Green and Blue are present in that pixel. This array is illustrated in Figure 8.1.

   You can see the dimensions of this array using the following.

```
size(X)
```

3. Display the image using

```
imagesc(X);
```

   The image may be squashed; if you would like it not be be squashed, type `axis equal` into the command window.

4. What will the dimensions of the matrix with the grayscale representation of this image be?

5. We will now use matrix manipulations to turn the image into a grayscale image. The RGB array can be separated into three slices, one for each color. For example, the red slice is all the data in the the first layer of the array:

```
X_red=X(:,:,1);
```

   Converting a pixel to grayscale can be accomplished by taking a linear combination of the red, green and blue values of that pixel which are weighted by 0.2989, 0.5870 and 0.1140 respectively. Use these weights to create a linear combination of the red, green, and blue slices.

6. Verify if this was done correctly by displaying the image using the following commands.

```
imagesc(grayscaleX); colormap('gray'); axis equal
```

8.3 Further Examples on Decomposition

Exercise 8.5

1. In this problem, we are going to express the temperature data for four cities we encountered earlier using a given set of basis vectors. Load some sample temperature data in MATLAB by typing » `load temperatures_and_bases.mat`. Type `whos` at the MATLAB prompt to see all your variables. You should have a matrix T which has the temperature data for 1 year for the cities of Boston, New York, Washington DC and Providence in that order. Use the `size` command to determine how the data are organized in this matrix. You should also have four vectors $\mathbf{u}_1 \cdots \mathbf{u}_4$ which a genie has provided to you.

   a) Verify that the vectors $\mathbf{u}_1, \cdots \mathbf{u}_4$ are all mutually orthogonal, and that they have unit length.
   b) Set up and solve the linear algebra problem in order to express each column of the temperature matrix T as a linear combination of $\mathbf{u}_1 \cdots \mathbf{u}_4$. Check that you can undo this operation and retrieve the original data.
   c) Now let's reconstruct an approximation to the original temperature data, using only the vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$, What is the rms error for this approximation?
   d) Compare the rms error for the previous scheme to a simpler scheme where in order to compress the data, we simply discard the temperature of Providence. When we want to reconstruct the data, we simply approximate the temperature in Providence by the temperature in Boston.

   Once again, we have to disappoint you by letting you know that there is no genie! There is just data. In the coming weeks, we are going to find out how to find bases vectors that can be useful for dimensionality reduction for a given set of random data, given some training data. This will be particularly useful in speeding up computations where instead of doing computations on all the dimensions of the data we have, we perform computations on fewer dimensions.

2. We will finally be dealing with images of faces. We are going to compress these face images in a similar way as the temperature data (we give you the bases). Here, the data have really

high dimensions (each pixel is a dimension). The bases that we give you (matrix $\mathbf{U}$) doesn't span the entire high dimensional space (so there will be lossy compression).

a) Load the file `face_bases.mat` in MATLAB. You will see a 3-dimensional array `test_images`, of dimensions $256 \times 256 \times 424$, and a matrix U of dimensions 65536×424. The `test_images` array contains 424 grayscale images.Each image is 256×256 pixels.

b) Select any one image from the set of 424 and call it T. Display this image using » `imagesc(T); colormap('gray')`. This image is currently represented as a 256×256 matrix of grayscale values. We will find it very convenient to work with vectors instead of matrices representing an image. Therefore, to make our lives simple, we will take the data for an image which is stored in a matrix and store it in a vector.

   We are going to *vectorize* this image by stacking its columns one on top of another to create a single vector that is $(256)^2 \times 1$, i.e. 65536×1 which will be a lot easier to work with. This operation can be accomplished in MATLAB as follows: » `Tstacked = reshape(T, 65536,1);`. When you need to recover the unstacked version of the image, you can undo-the stacking as follows: » `Tunstacked = reshape(Tstacked, 256, 256)`.

c) The matrix U contains a set of 424 65536×1 linearly independent vectors provided by the genie. Approximate the `Tstacked` vector as a linear combination of the first 10 of columns of U, and call this vector `Tapprox10`. `Tapprox10` should be a 65536 $\times$ 1 vector, and you will only have 10 weight values to find this approximation. See how well this approximation works by reshaping `Tapprox10` into a 256×256 matrix and displaying it using `imagesc` and `colormap('gray')`.

d) Now repeat the previous exercise with the first 50 columns of U and then again with the first 100 columns of U.

You should observe that the more columns of U you use, the better the approximation. Note here that we are trying to approximate a 65536 dimensional vector using 10, 50 and 100 numbers. Therefore, you should not expect the approximation to work super well, but with 100 columns of U, you should be able to recognize the picture. At a later date we will quantify the fidelity of the approximation.

Note that more sophisticated image compression algorithms use methods that rely on special properties of images and human vision in order to a achieve high degree of compression.

8.4 Data: Many Measurements of the Same Thing

One of the simplest forms of data is a set of data which represents many measurements of nominally the same thing. Depending on what the goal is of our analysis, this might encompass measurements of the same quantity across many different situations, or many instances of the same situation.

Visualizing Measurements of the Same Thing

It's usually a good idea to *look* at data before you start calculating things associated with it.

You've surely encountered these ideas before, but for the sake of completeness, we'll highlight a couple of ideas here. If you have a large number of data points (say, for example, that you measured the heights of a bunch of different people), you might choose to simply plot the data versus the person number – the index. Note here that the data is plotted as individual points, since each point represents a measurement. Ideally we might also include error bars here to indicate our uncertainty in a given measurement, but for now, let's leave that out.

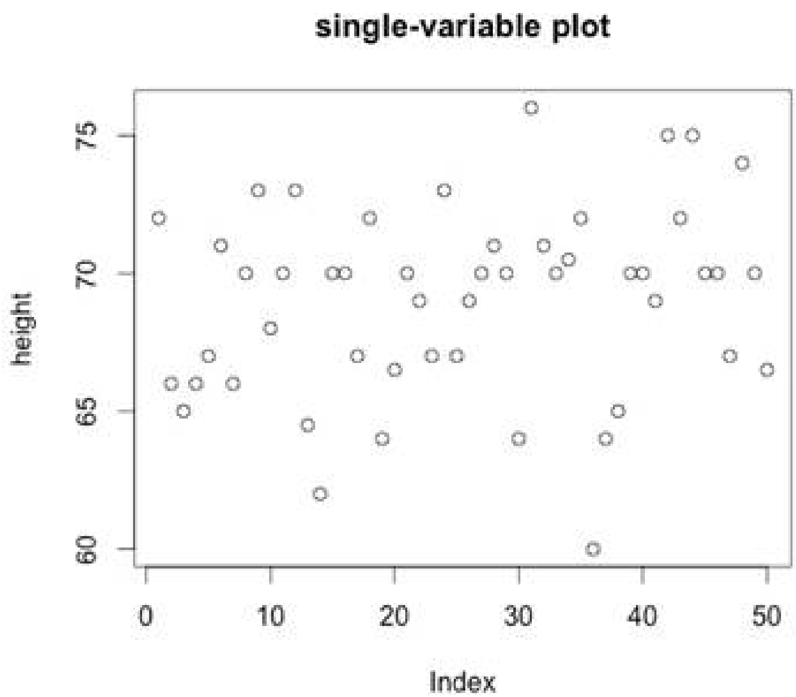


Figure 8.4: An example of a single variable plot

Alternatively, you could also visualize many measurements of the same thing by creating a *histogram*. This is a representation of how many measurements fall into different "bins": the height of a given bar is the number of samples that fall within the range associated with the bar. For example, in the figure, you can see that about 20 million people made between 0 and $5000 in 2008. You've likely seen this kind of thing before as well: it's not an uncommon way to represent test scores.

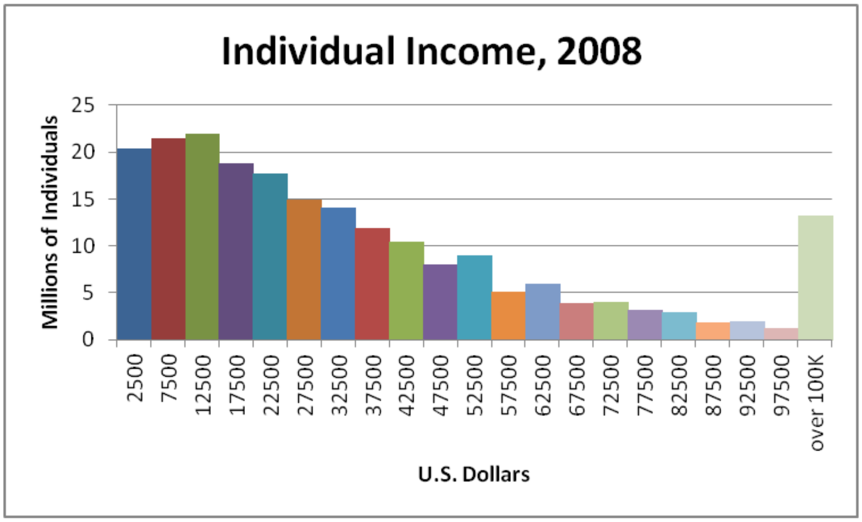


Figure 8.5: An example of a histogram.

Note, of course, that how a histogram *looks* depends on what you choose for the bins - both how many there are, and where they are centered!

Common Figures of Merit for the Same Thing

While looking at the data is certainly helpful, we can also extract or calculate a couple of important figures of merit of the data. The first is the average, or *mean* of the data, given by summing all the elements in the dataset $\{d_i\}$ and dividing by the number N of elements in the set:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{8.1}$$

Note that if our data is a continuous function $f(x)$ over a range of the independent variable x as opposed to a set of discrete points, we can express the same thing as an integral:

$$\mu = \frac{\int_{range} f(x)dx}{\int_{range} dx} \tag{8.2}$$

The average captures the center or 'expected value' of the distribution of data. In addition to this, it is often helpful to capture the spread of the data around this average. There are a few different metrics which are used for this. A simple one is the *variance* from the mean, σ^2: the average of the squared difference between each data point and the mean.

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(d_i - \mu)^2 \tag{8.3}$$

Please note that this definition normalizes using $N-1$, but you will often see alternative definitions which normalize using N. Another commonly encountered measure is the *standard deviation*, which is simply the square root of the variance from the mean:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d_i - \mu)^2} \tag{8.4}$$

Exercise 8.6

1. Look at the single variable plot in Figure 8.4 above. Estimate the value of the mean and the value of the standard deviation. What are the units of each?

2. Look at the histogram plot in Figure 8.5 above. Estimate the value of the mean and the value of the standard deviation.

3. What is the mean and standard deviation of this data set (Do this in your head!)

$$\{1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3\}$$

4. Begin by considering the simple dataset of the high temperatures in Needham for ten days in March:

$$T = \{57, 61, 46, 43, 46, 46, 54, 46, 46, 55\} \tag{8.5}$$

   a) By hand, create a histogram of this data. What size bin makes sense? What bin centering makes sense?
   b) By hand, compute the mean temperature over these ten days. If you look at the data, does this mean make sense?
   c) By hand, compute the variance and standard deviation of the temperature over these ten days. If you look at the data histogram does this make sense?
   d) This dataset has a flaw: it has a small number of datapoints. What do you see as the possible effects of having such a small sample?

5. Now consider the larger dataset below of the approximated heights of the Olin faculty, measured in inches. In MATLAB, create a vector which has this dataset as the entries.

$$\begin{aligned} H = \{&63, 66, 71, 65, 70, 66, 67, 65, 67, 74, 64, 75, 68, 67, 70, 73, 66, 70, 72, 62, 68,\\ &70, 62, 69, 66, 70, 70, 68, 69, 70, 71, 65, 64, 71, 64, 78, 69, 70, 65, 66, 72, 64\}\end{aligned}$$

   a) Computationally histogram this data. What size bin makes sense? What bin centering makes sense? Try a few different combinations. See MATLAB function `histogram`.
   b) Computationally, find the mean, standard deviation, and variance of this dataset. See MATLAB functions `mean`, `std`, and `var`.
   c) Does the mean, standard deviation, and variance make sense given the histogram of the data?

8.5 Brightness and Contrast

The brightness and contrast of images is controlled by scaling the histogram of the pixel values. Try this out!

Note: for displaying images in this part, make sure to NOT use imagesc: imagesc is specifically setup to auto-scale the image to use the full range from 0 to 255. Just use the command 'image'.

Exercise 8.7

1. Load an image of your choice into MATLAB using the `imread` command. (Make sure you are in the correct directory for the image or give it the complete path). Display the image using the 'image' command.

2. If your image is a color image, convert it into grayscale by using the `rgb2gray` command.

3. Create a vector of the intensities in your image: use the `reshape` command to create a giant column vector in which the first n elements are the first column of the image, the next n are the second column, etc.

4. Make a histogram of the intensity values in your image. Note that the default variable type for image data is uint8 (8-bit unsigned integer) which is an integer that ranges from 0 to 255. Does your image use the entire range of values from 0 to 255? What is the minimum pixel value used? What is the maximum?

5. Find the mean of the intensities in your image data. Find the standard deviation. Is the intensity data well-centered on the available range? The location of the intensity data in the range determines the brightness of the image. How does the standard deviation compare to the available range? Does the intensity data span a good portion of the available range? This affects the contrast.

6. To adjust the brightness of your image, you can scale all of the intensity values by a multiplicative factor down (towards darker values) or up (towards brighter values). Based on looking at the histogram, should your image be brightened? Dimmed? Why?

7. To adjust the contrast, you make a linear mapping of the existing range onto the full 0 to 255 range. In other words, if you think of the current intensity value as your independent variable x, and the new intensity value as the dependent variable y, a contrast adjustment is defined by a function $y = f(x)$. Propose an equation for a line which gives you the "best" range of y's, given the input intensity values in the image. You should be able to justify this based on the histogram of the image. Note that any values of y that end up below 0 should be interpreted as 0, and any values over 255 should be interpreted as 255.

8. Implement brightness and contrast adjustment:

   a) Load a picture of a face.
   b) Analyze the intensity histogram.
   c) Calculate the adjusted face by applying both brightness and contrast adjustments to make it as “good” as possible.
   d) Create a figure that includes four subplots: the original image, the original intensity histogram, the new image, and the new intensity histogram.

9. What would happen if the function for contrast adjustment was not linear? Why might you choose a non-linear function for this mapping?

Solution 8.4

1.

2. The size of the array is $740 \times 740 \times 3$.

3. You should see the following picture:

Figure 8.2: Giraffe

4. The gray-scale version of this image is represented by a 740×740 matrix.

5. Create the matrix **grayscaleX** which represents the grayscale version of this image using » `grayscaleX = 0.2989*X(:,:,1) + 0.5870*X(:,:,2) + 0.1140*X(:,:,3)`.

6. You should see the following image:

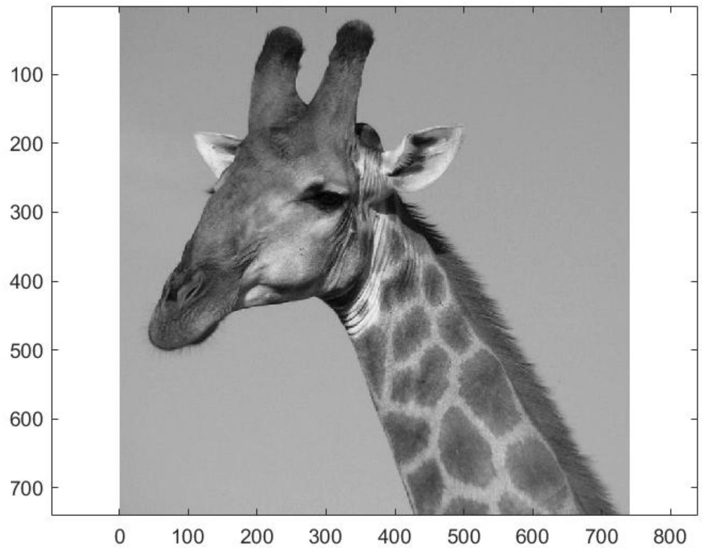


Figure 8.3: Gray Giraffe

Solution 8.5

1. a) To check that `u1` and `u2` are orthogonal, type » `transpose(u1)*u2`. The result should be zero. Check the other five pairs of vectors.

   To check that `u1` has unit length, type » `transpose(u1)*u1`. The result should be one. Check the other three vectors.

   b) First we need to construct a matrix U with columns each of the $\mathbf{u}_i$
   » `U=[u1 u2 u3 u4],`
   and convert T to the basis of $\mathbf{u}_i$ vectors by multiplying » `Tu=U*T.` You can recover the original data with » `inv(U)*Tu`.

   c)

   d)

2. a)

   b)

   c) Isolate the first 10 columns of U using » `U10=U(:,1:10);`. Then determine the weights for each of these column vectors using
   » `Tweights10=transpose(Tstacked)*U10;`
   and then take the linear combination
   » `Tapprox10=U10*transpose(Tweights10);`. Then we unstack the vector into a matrix
   » `Tapprox10unstacked=reshape(Tapprox10,256,256);` and display the image.

d)

Solution 8.6

1. Assuming that the “heights” plotted are heights of randomly selected humans, then the unit for the mean and standard deviation is inches.

2.

3. Since half the digits are 1 and the other half are 3, the mean will be the average of 1 and 3, so $\mu = 2$. Looking at the formula for standard deviation, we can see that $d_i - \mu = 1$ for each data point, so $\sigma = \sqrt{20/19}$.

4. a)
   b) We compute $\mu = 50$.
   c) We compute $\sigma = 6.15$.
   d)

5. a) By simply entering » `histogram(H)`, MATLAB automatically chooses bins of size one.
   b) Using MATLAB we find that $\mu = 68.1429$, $\sigma = 3.5721$ and $\sigma^2 = 12.7596$.
   c)