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]

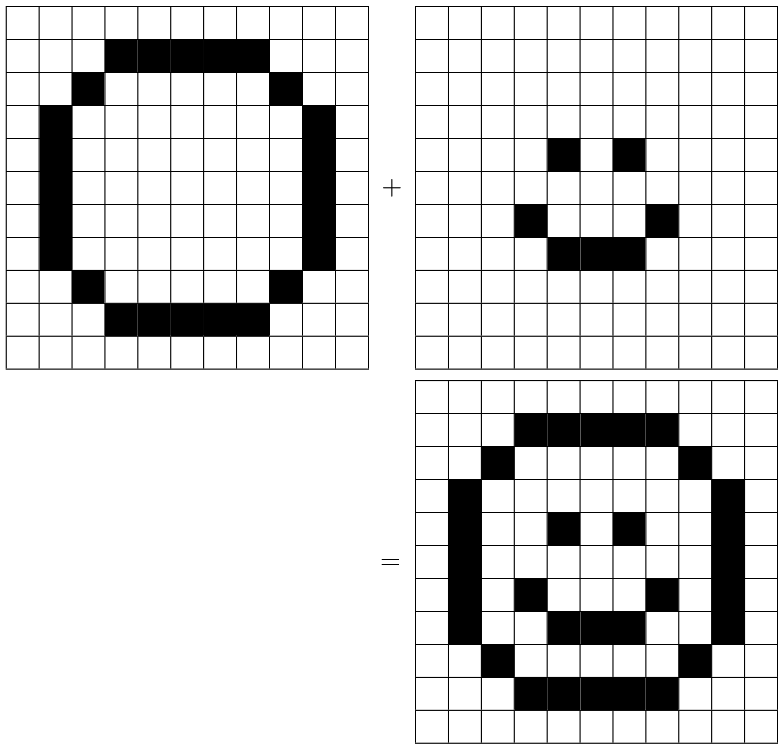
1.3 Facespace

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).



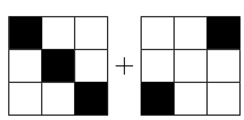
Conceptually, this operation might seem straightforward. Adding two images results in an image that has a black pixel whenever either of the two images has a black pixel at a corresponding position.

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.

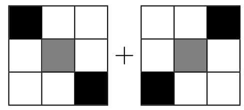
Exercise 1.1

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

1.

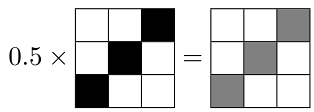


2.



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.

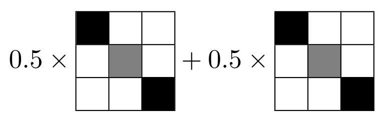
For example,



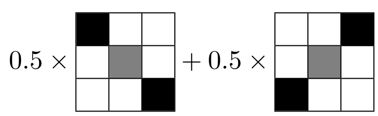
Exercise 1.2

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

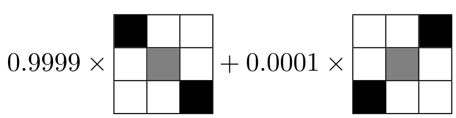
1.



2.



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



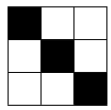
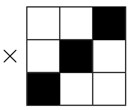
A Universal Set of Building Block Images [10 mins]

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.

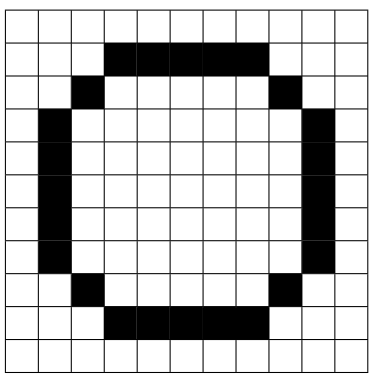
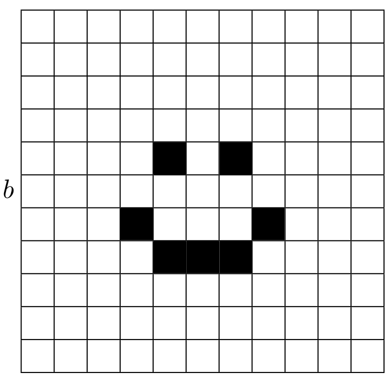
Exercise 1.3

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?

   $c \times$  $+ (1 - c) \times$ 

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.

   a  $+\ b$ 

   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?

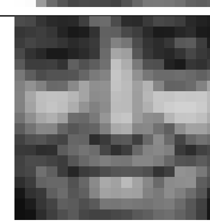
As you probably deduced in the previous exercise, a major drawback of the encoding we worked out previously is that each scaling factor doesn't really tell us that much useful information about each face (and as a result we need a lot of these numbers to specify a particular face). It turns out that 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.

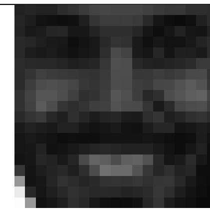
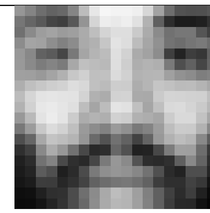
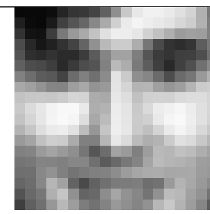
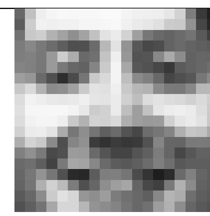
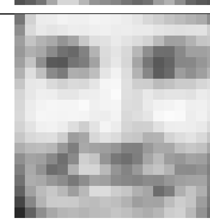
Come to the front of the room and grab a piece of paper with a 6 by 4 grid of face-like images along with a set of transparent face-like images held by a binder clip. Take these materials back to your table. Layout the piece of paper on your table. Also in the envelope you should have a set of transparent 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.

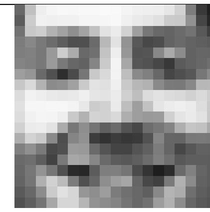
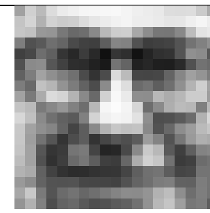
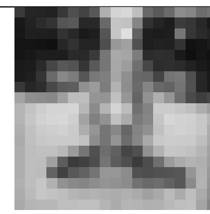
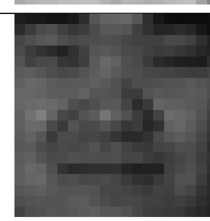
What you see before you is a very carefully chosen set of building block images. You should notice that each row represents a different building block image and each column represents a different scaled version of that same building block. Today, we won't be going into detail about *how* we determined 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.

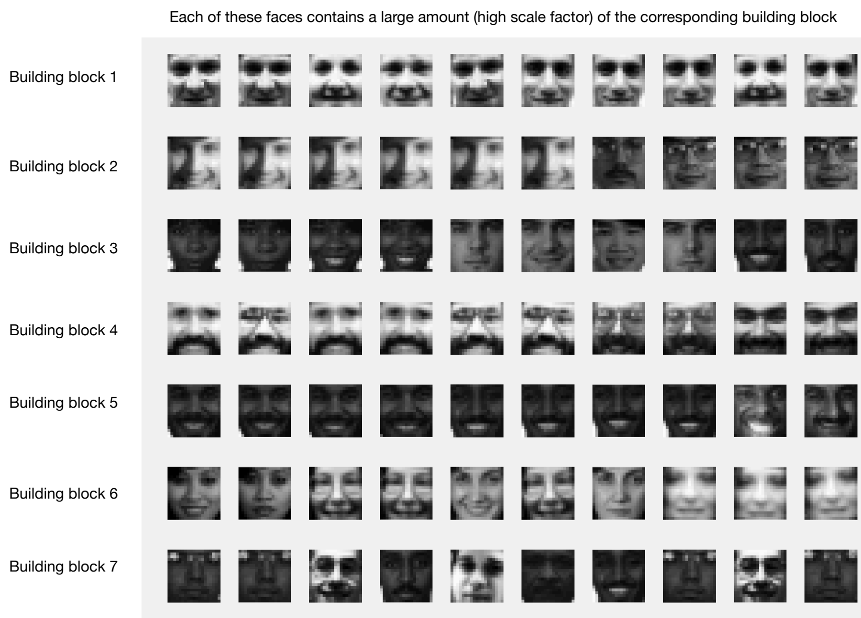
Intensity 1	**Intensity 2**	**Intensity 3**	**Intensity 4**	**Intensity 5**	**Intensity 6**	**face image**
3	3	0	0	2	2	
0	2	3	2	3	1	
3	0	1	4	1	1	

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

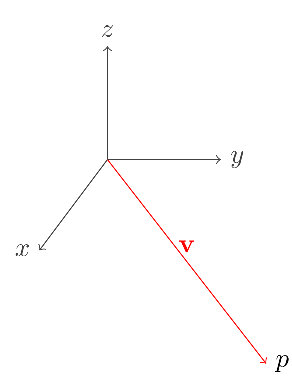
There are lots of books about Linear Algebra and lots of useful videos on the web. Here are some specific recommendations:

- 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

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

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?

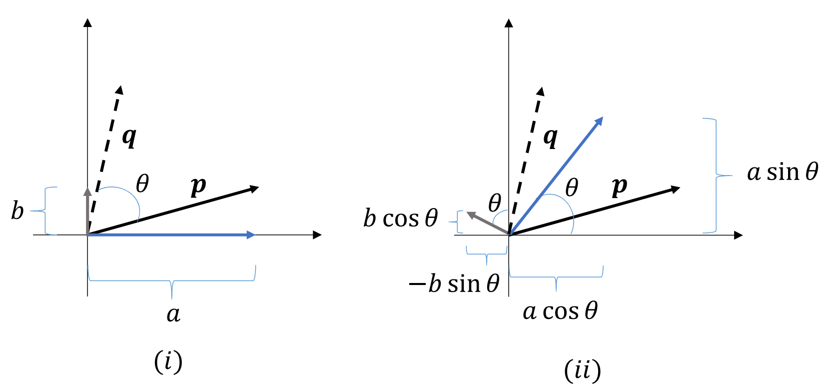


Figure 2.1: Rotation of vectors

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 41.4 (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 41.4 (ii). By using trigonometry, we see that the blue vector in Figure 41.4 (ii) is

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

and the gray vector in Figure 41.4 (ii) is

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

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

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

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., $\mathbf{w} = \mathbf{Av}$. 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 $\mathbf{A}$, $\mathbf{v}$, and $\mathbf{u}$ 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 $\mathbf{A}$ 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 $\mathbf{v}$ 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 2.8

Using the definitions for $\mathbf{A}$, $\mathbf{v}$, and $\mathbf{u}$ 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. $\mathbf{Iv} = \mathbf{v}$. 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 $\mathbf{Mv} = 3\mathbf{v}$ 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

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

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.

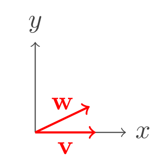
$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$$

D.

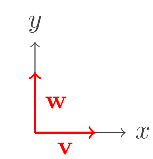
$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

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

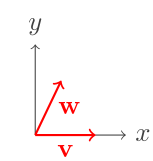
A.



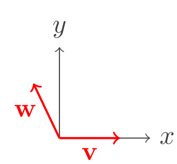
B.



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.

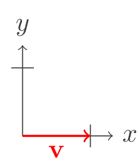
3. Let $\mathbf{w} = \begin{bmatrix} 120 & 250 & 100 \end{bmatrix}$. Then $\mathbf{wf} = 120 + 500 + 300 = 920$, the total weight of the fruits in your refrigerator.

Solution 2.3

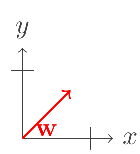
The vector $\mathbf{Gf}$ is a 2×1 vector whose first entry represents the total number of fruits and second entry represents the number of citrus fruits.

Solution 2.4

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



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

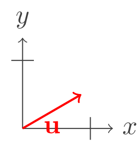


3. Multiplying $\mathbf{v}$ by $\mathbf{A}$ rotated the vector counterclockwise by 45 degrees.

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

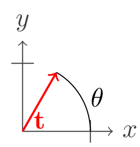


5. Multiplying $\mathbf{v}$ by $\mathbf{B}$ rotated the vector counterclockwise by 30 degrees.

6. First we compute

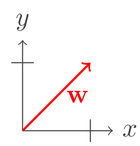
$$\mathbf{t} = \mathbf{R}\mathbf{v} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

which is visually represented as



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

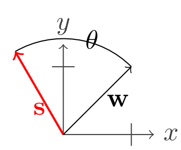
8. The vector $\mathbf{w}$ is



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?

Exercise 3.7

Now, let's do it in MATLAB.

1. Create and plot the original 4 points:$(1, 2)$, $(-1, 2)$, $(-1, -2)$, and $(1, -2)$. Then create the matrix that rotates them by 30 degrees counterclockwise, 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.*

2. 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`

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

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

5. 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 3.8

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 1. 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 2. 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 4 or part 5, 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 3.9

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 3.10

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 $\mathbf{A}^2\mathbf{A} = \mathbf{AA}^2$ 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 $\mathbf{A}^n\mathbf{A} = \mathbf{AA}^n$ 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.

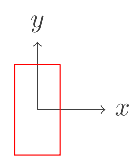
Solution 3.5

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

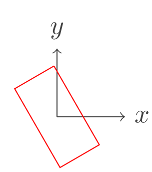
$$\mathbf{R} = \begin{bmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

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,



Solution 3.7

1. 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*')
```

2. 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
```

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

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

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

5. 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 3.8

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 3.9

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.

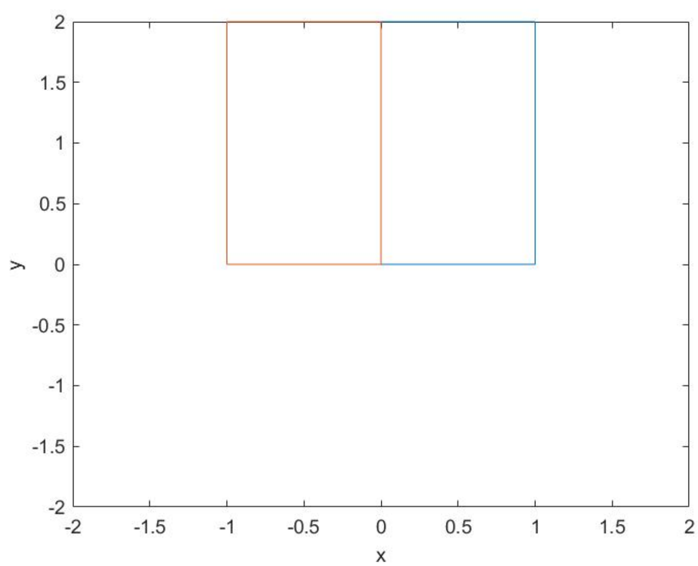


Figure 3.1: Reflection over y-axis.

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.

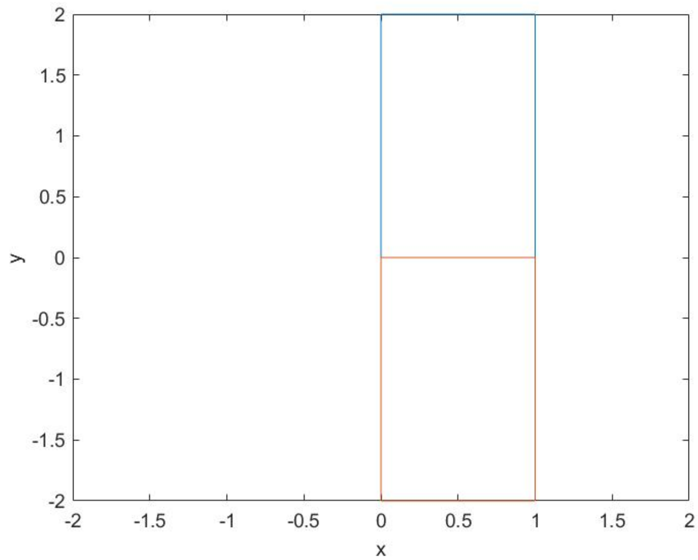


Figure 3.2: Reflection over x-axis.

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.

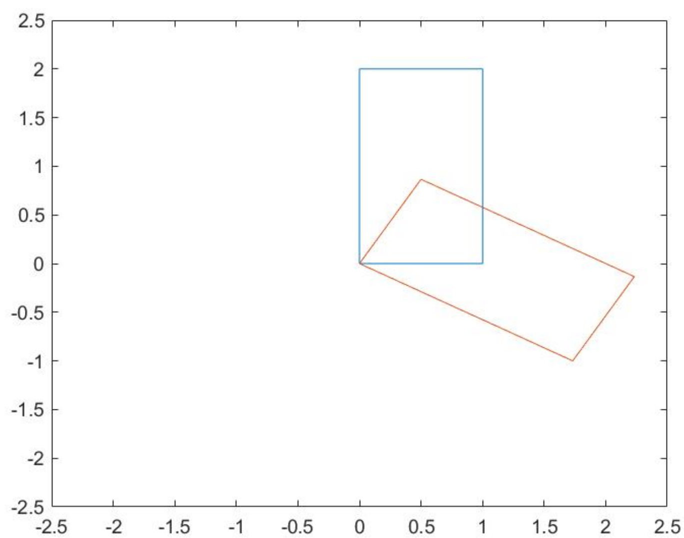


Figure 3.3: Reflection over 30 degree line.

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 3.10

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:

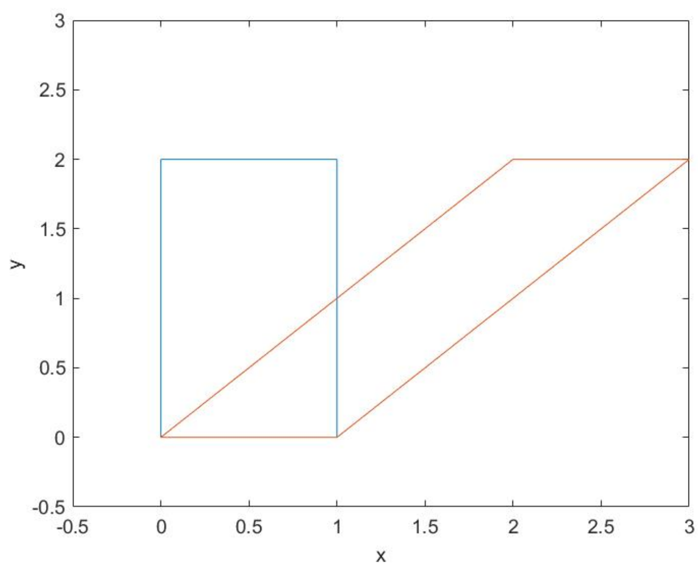


Figure 3.4: Shearing in x direction.

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:

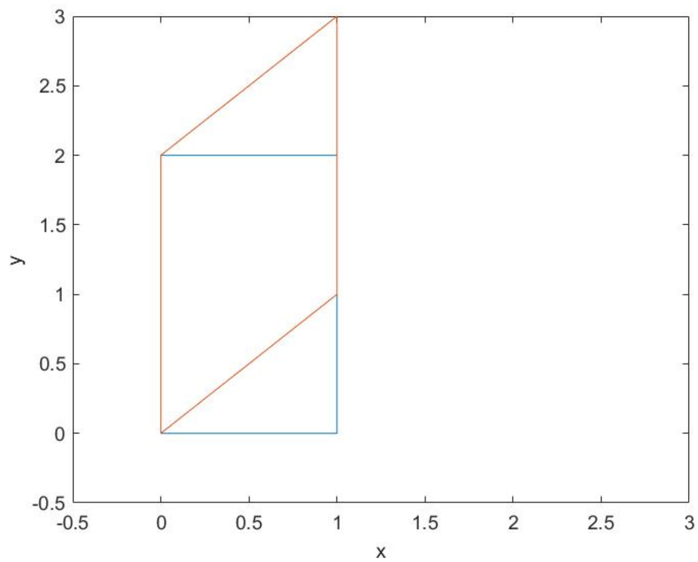


Figure 3.5: Shearing in y direction.

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:

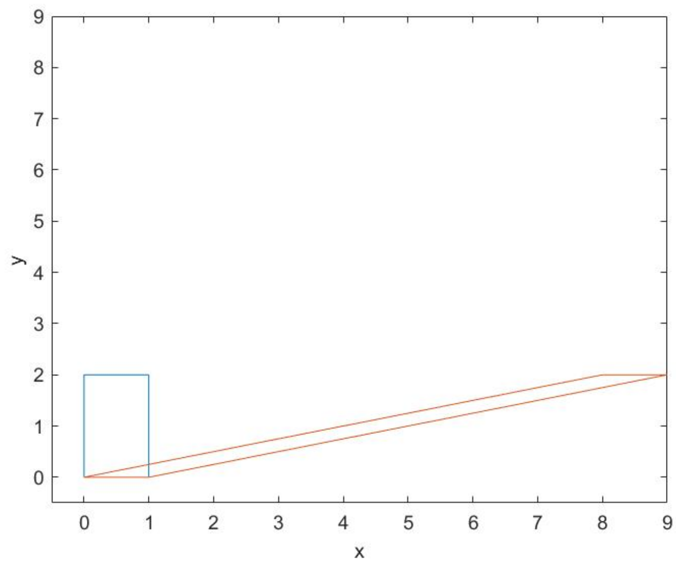


Figure 3.6: Shearing in x direction with $k = 2$.

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:

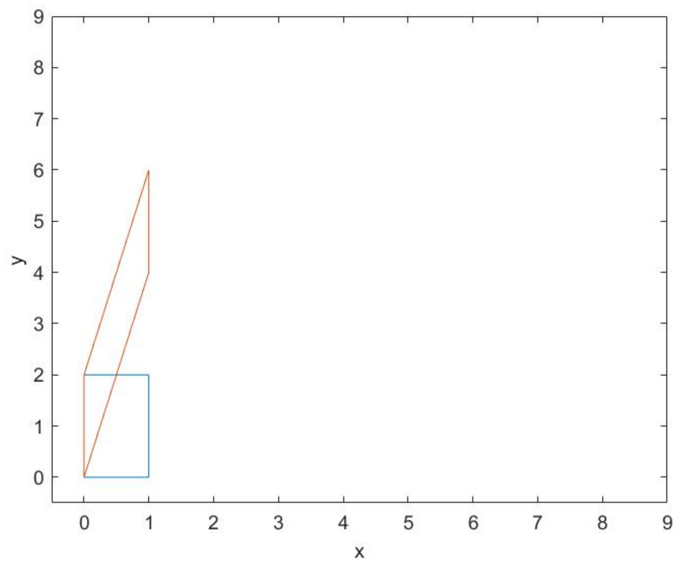


Figure 3.7: Shearing in y direction with $k = 2$.

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 $\mathbf{w} = \mathbf{P}\mathbf{u}$.
2. Find $\mathbf{Q}\mathbf{w}$. How is this related to $\mathbf{u}$?
3. Find $\mathbf{Q}\mathbf{P}$. Does the answer look familiar?
4. Find $\mathbf{P}\mathbf{Q}$.
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.

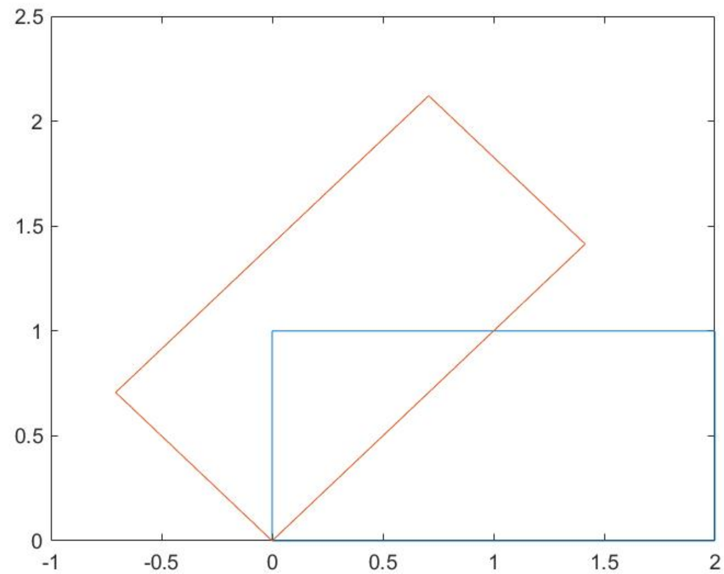
3. Propose a set of points that defines the outline and relevant features of your animal. You may find `ginput` useful. Define the points in MATLAB and plot your animal.

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!

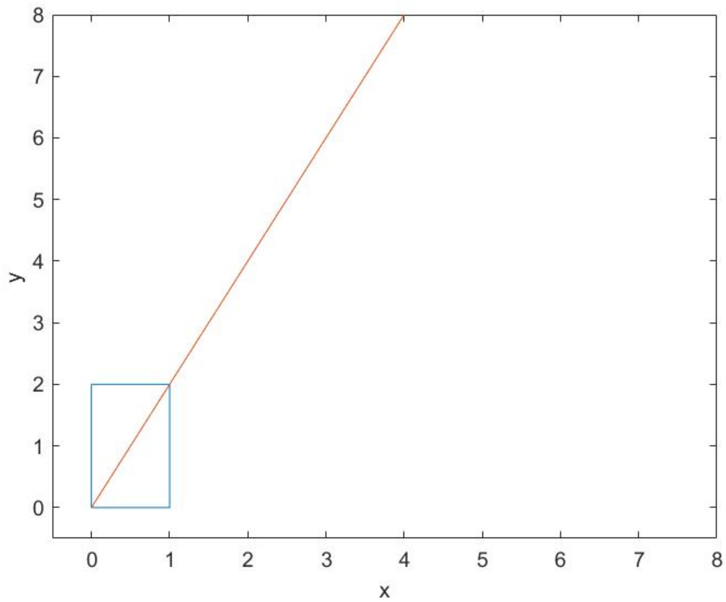
4.4 Conceptual Quiz

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. $s_1 + s_2$

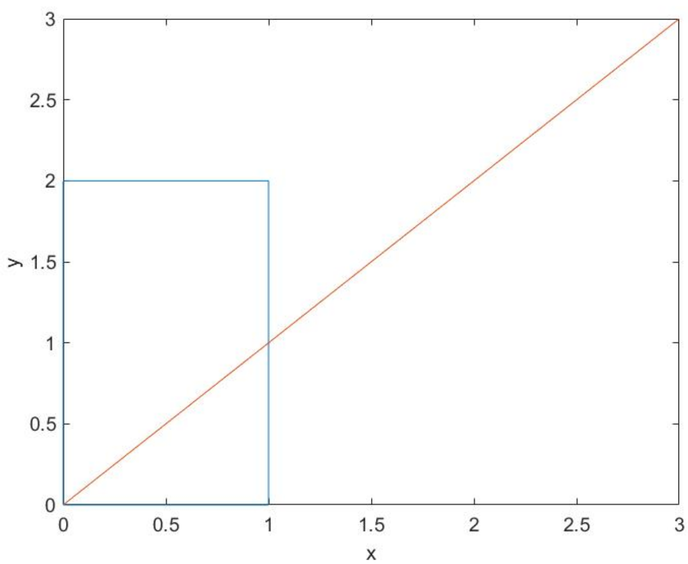
5. True or false: Any shearing matrix $\mathbf{S}$ and any rotation matrix $\mathbf{R}$ commute, i.e., $\mathbf{RS} = \mathbf{SR}$.

Solution 4.1

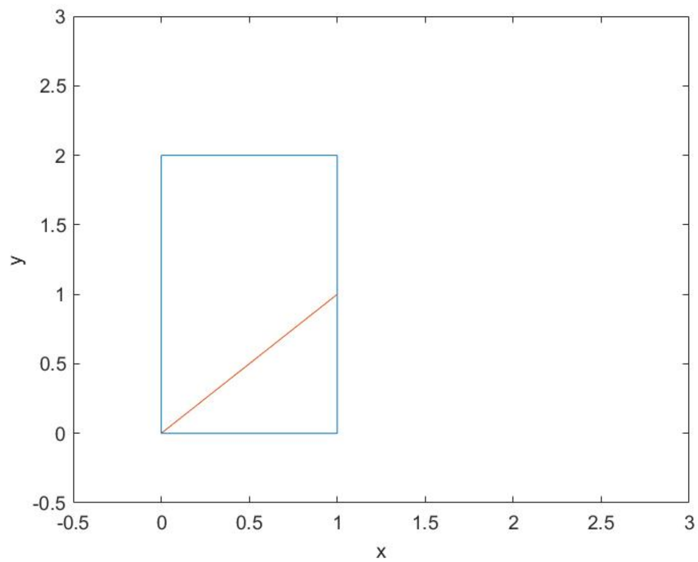
1. The determinant is 1. (Recall that $\cos^2\theta + \sin^2\theta = 1$.)
2. The determinant is -1.
3. The determinant is 1.

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. $$\mathbf{w} = \mathbf{P}\mathbf{u} = \begin{bmatrix} 7 \\ 17 \end{bmatrix}$$

2. $$\mathbf{Q}\mathbf{w} = \mathbf{Q}\mathbf{P}\mathbf{u} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

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

   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.

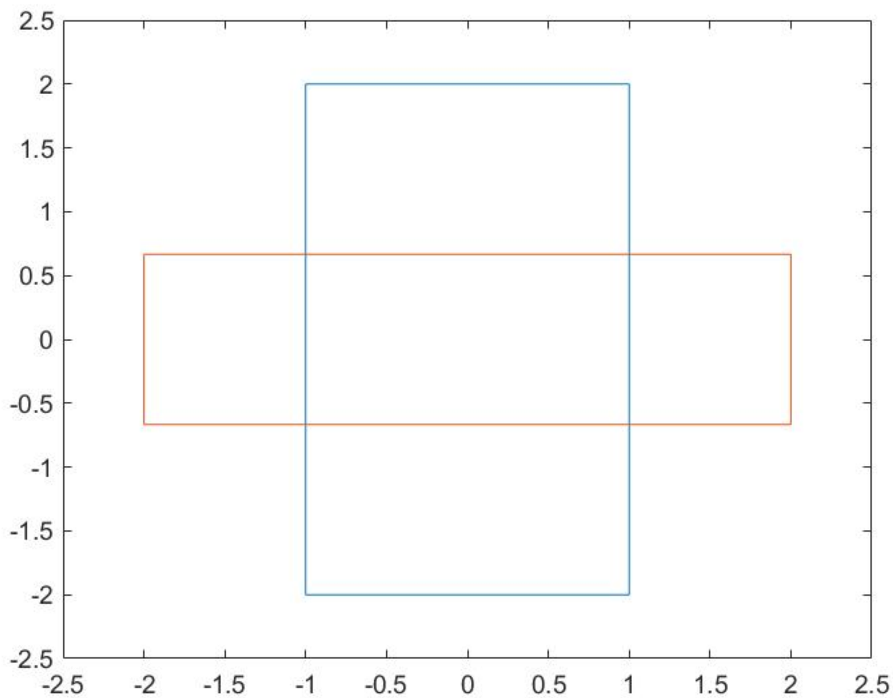
b) The determinant of the matrix is 0.

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 $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ of vectors in $\mathbf{R}^n$ is said to be *linearly dependent* if there exist scalars $c_1, c_2, \ldots, c_m$ 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 $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_m$.

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

If a non-zero solution exists to $\mathbf{A}\mathbf{c} = \mathbf{0}$ then the set of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ 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 $\det(\mathbf{A}) \neq 0$.

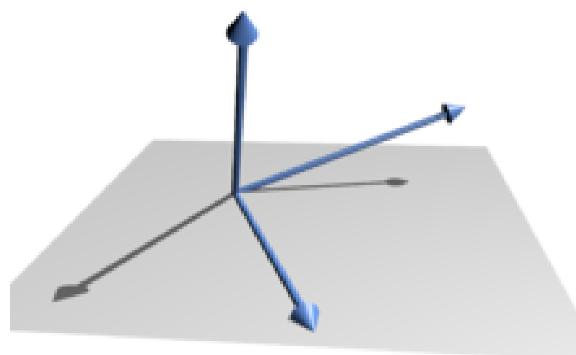


Figure 5.2: Linearly independent vectors in $\mathbf{R}^3$. (from Wikimedia Commons).

- 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

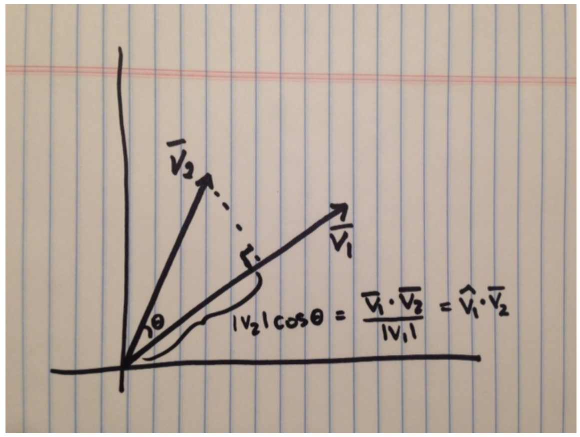


Figure 5.3: Projection

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.

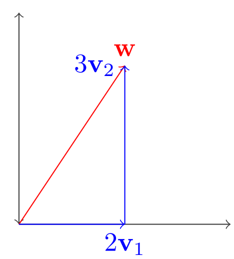
3. The span of these three vectors is the xy-plane in $\mathbf{R}^3$.

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

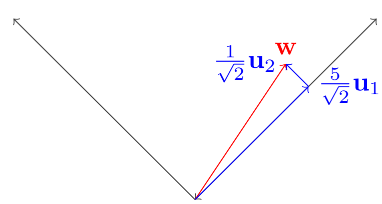
1. It's clear that $2\mathbf{v}_1 + 3\mathbf{v}_2 = \mathbf{w}$. We visualize this as



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,

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

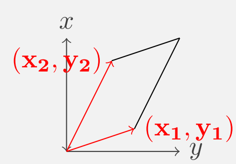
You already considered the determinant of some transformation matrices, now let's consider what the determinant is really telling us about a general matrix.

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 $\mathbf{b}$.

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

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.

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

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?

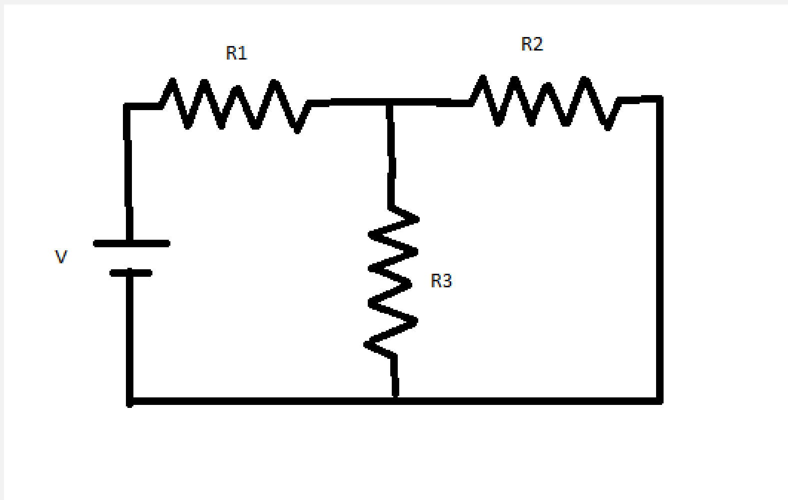
An Electrical Example

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

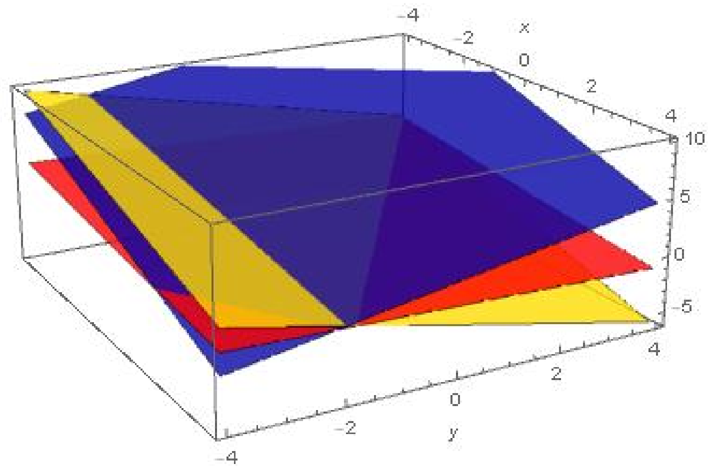
   e) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

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

$$x - y = 2$$

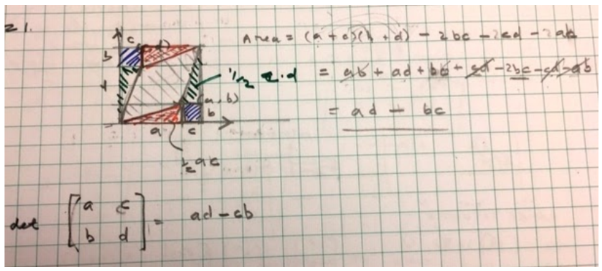
$$3x + z = 11$$

$$y - 2z = -3$$

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. Rearrange the equations to linear form $y = mx + b$. If the lines are identical, there are infinitely many solutions; if the lines are parallel, but don't overlap, there are zero solutions; if the lines are not parallel, there is one solution.

   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.

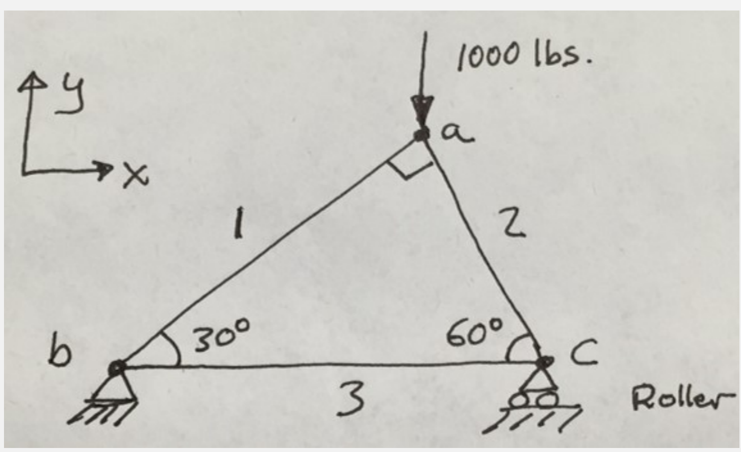
7.4 Applications of LSAE

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!

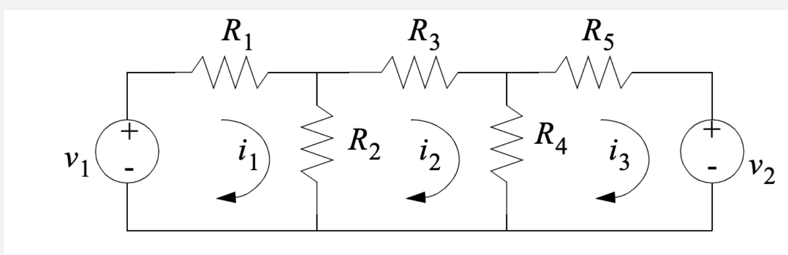
2. Write out the equations of static equilibrium, $\sum \vec{F} = 0$, for every one of the pins. Note that some of your forces will be known forces (e.g., external loads), and some will be unknown reaction forces.

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.

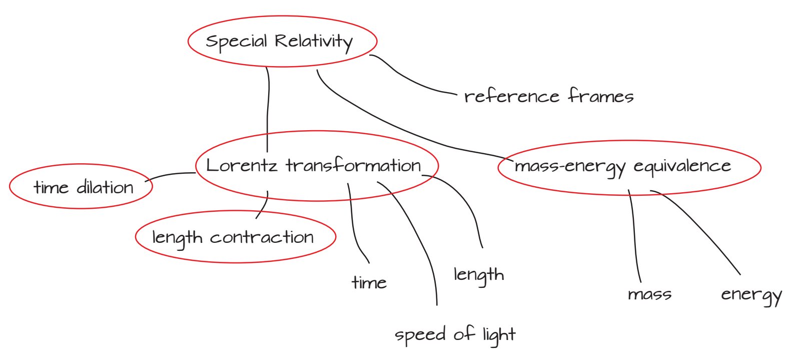


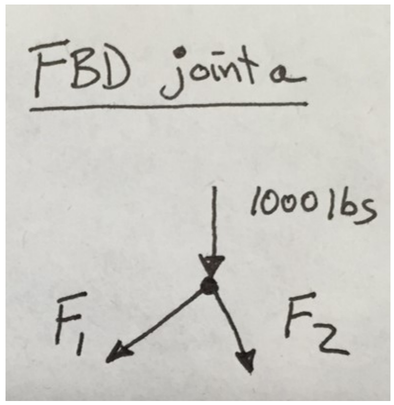
Figure 7.1: If you were trying to break down special relativity, a *portion* of your breakdown might look like this...

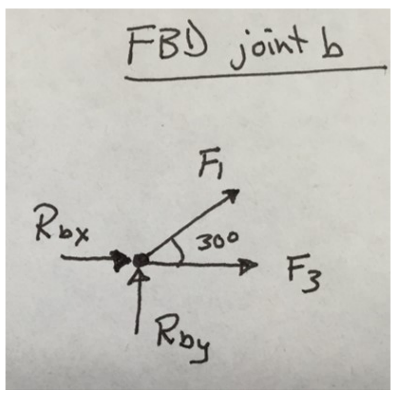
Once you've done your breakdown, try to make the following lists individually:

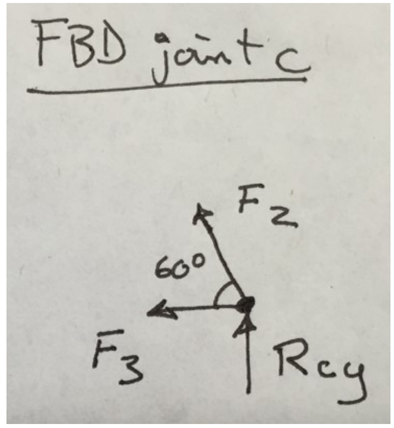
1. Relevant fundamental mathematical terms that I don't know
2. Relevant fundamental mathematical terms that I do know
3. Ideas specific to facial or image recognition that I don't know
4. Ideas specific to facial or image recognition that I do know

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

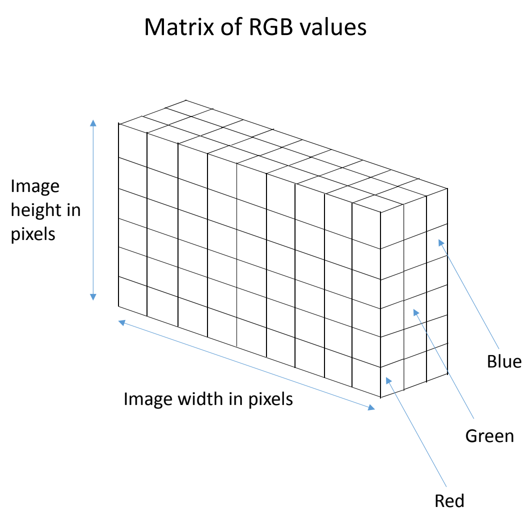


Figure 8.1: Anatomy of an RGB image array.

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 65536×1 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.

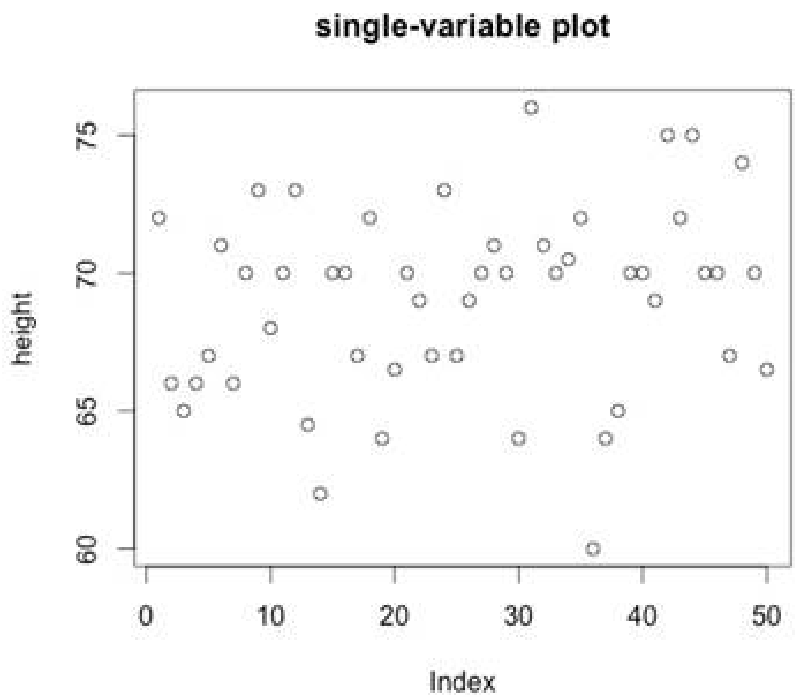


Figure 8.4: An example of a single variable plot

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.

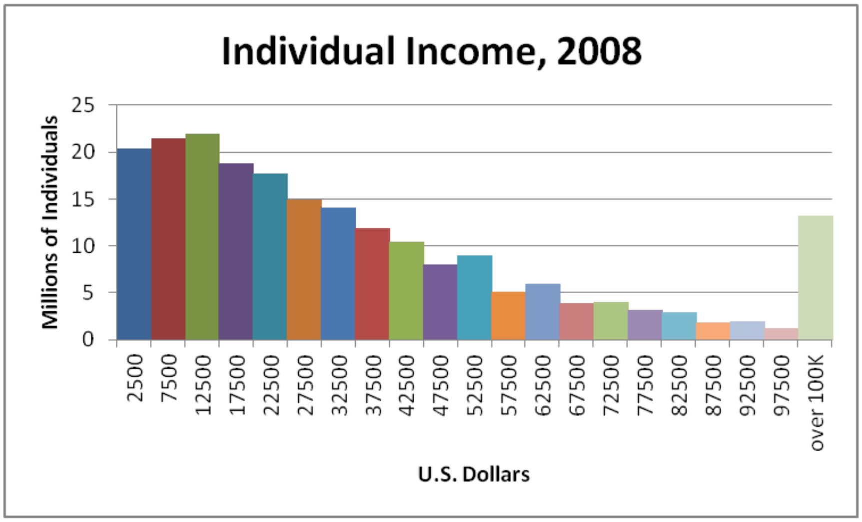


Figure 8.5: An example of a histogram.

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:

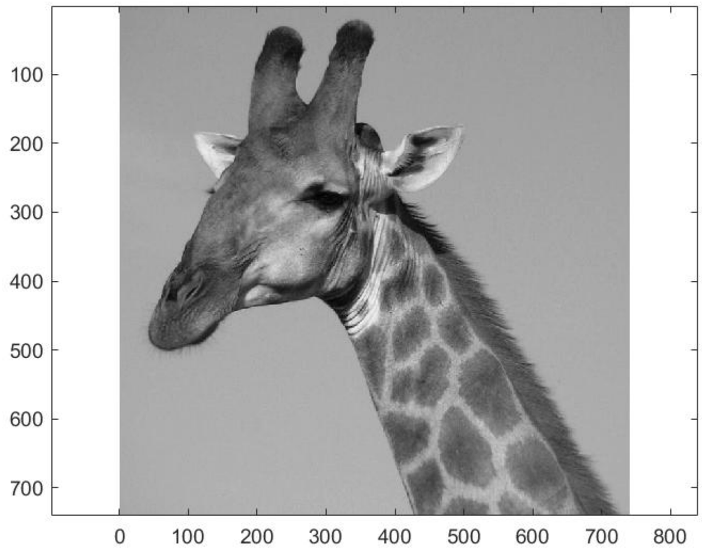


Figure 8.3: Gray Giraffe

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)

Chapter 9

Day 5: LSAE, Brain data, and C&E

9.1 Schedule

- 0900-0915: Table Dynamics Discussion
- 0915-1010: Debrief and Synthesis
- 1010-1025: Coffee Break
- 1025-1035: Correlation
- 1035-1135: Brain Data with Samantha Michalka!
- 1135-1220: AI and Society Discussion
- 1220-1230: Day Survey

9.2 Table Dynamics Survey Discussion

9.3 Debrief and Synthesis

- Please discuss your overnight work with your table-mates, and get help with the ideas that you are still confused by.

In the next set of exercises we will explore a common method for finding "the" solution of a linear system of algebraic equations ($\mathbf{Ax} = \mathbf{b}$) in the case where there are more equations that unknowns (more rows than columns). We will first need to synthesise some previous ideas about the span of vectors.

Range of $\mathbf{A}$

We discussed earlier the concept of the **span** of a collection of vectors. Recall that the span of a collection of vectors is the set of all linear combinations of the vectors. Now we will apply this concept to the columns of a matrix:

Definition: The Range of a matrix $\mathbf{A}$ *is the span of its columns.*

Exercise 9.1

Describe in words the Range of the following matrices:

1. $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

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

3. $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

Exact Solution to $\mathbf{Ax} = \mathbf{b}$

When does a linear system of algebraic equations, $\mathbf{Ax} = \mathbf{b}$, have a solution? Since the product $\mathbf{Ax}$ is a linear combination of the columns of $\mathbf{A}$, then $\mathbf{Ax} = \mathbf{b}$ will have a solution if and only if $\mathbf{b}$ is in the Range of $\mathbf{A}$. Think about that, and complete the following exercise.

Exercise 9.2

Which of the following linear systems of algebraic equations will have a solution? Think about it from an equation perspective and the Range of $\mathbf{A}$ perspective.

1. $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

2. $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

3. $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 3 \\ 7 \\ 11 \end{bmatrix}$

4. $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 3 \\ 7 \\ 5 \end{bmatrix}$

Approximate solution to $\mathbf{Ax} = \mathbf{b}$

You should have found that some of these systems do not have a solution in the usual sense, i.e. there is no vector $\mathbf{x}$ which makes the equation $\mathbf{Ax} = \mathbf{b}$ true. We might refer to such a solution as an **exact** solution. We will now consider an **approximate** solution, i.e. a vector $\mathbf{x}$ which approximately satisfies

$\mathbf{Ax} = \mathbf{b}$. We will consider a particular approximation based on **orthogonal projection** now, and later in this module we will look at this approximation from a different perspective where it is known as the **Least-Squares** approximation. We met orthogonal projection earlier in the module when we spoke about vector components and basis vectors.

Exercise 9.3

Hold your hand up in front of you, and think about it as occupying a location in 3D.

1. Point to the location on the front wall that is closest to your hand.
2. Point to the location on the floor that is closest to your hand.
3. Have one of your table-mates hold a mobile white board at some angle. Now point to the location on the white board that is closest to your hand (you might have to imagine a larger mobile white board).
4. What is the relationship between the "pointing" vector and the surface being pointed at?

Now let's put this in the context of solving $\mathbf{Ax} = \mathbf{b}$.

- If $\mathbf{b}$ is not in the Range of $\mathbf{A}$ then we will define an approximate solution by orthogonal projection of $\mathbf{b}$ onto the Range of $\mathbf{A}$.
- The "pointing" vector from $\mathbf{b}$ to the relevant point in the Range of $\mathbf{A}$ is $\mathbf{Ax} - \mathbf{b}$.
- Since the Range of $\mathbf{A}$ is defined by the span of the columns of $\mathbf{A}$ then the "pointing" vector must be orthogonal to **every** column of $\mathbf{A}$.
- This implies that $\mathbf{A}^T(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}$. (Think about why this must be true).
- Re-arranging this equation leads to $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. The matrix $\mathbf{A}^T\mathbf{A}$ is a square matrix (which we will meet again and again this module).
- This is a linear system with equal numbers of equations and unknowns and can therefore be solved using our usual techniques. Did you get that? You should re-read this paragraph a few times. To summarize:

The approximate solution to $\mathbf{Ax} = \mathbf{b}$ based on orthogonal projection can be obtained by solving

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

*This solution is also known as the **least-squares** solution because it minimises the distance between $\mathbf{b}$ and the Range of $\mathbf{A}$ (more about this later in the module).*

Warning: *Do not think about $\mathbf{x}$ defining a coordinate system that $\mathbf{b}$ lives in! When you draw a picture you should think about the space that the columns of $\mathbf{A}$ live in. We are projecting $\mathbf{b}$ onto a basis defined by the columns of $\mathbf{A}$. The solution vector $\mathbf{x}$ is better thought of as a set of "weights" or "coordinates" with respect to this basis.*

Exercise 9.4

1. Consider the linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. (You've already thought about this earlier).
   a) Sketch the Range of $\mathbf{A}$ and locate the point in the Range that is closest to $\mathbf{b}$.
   b) Multiply both sides of $\mathbf{Ax} = \mathbf{b}$ by $\mathbf{A}^T$ and solve the resulting linear system.
2. Consider the linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 3 \\ 7 \\ 5 \end{bmatrix}$. (You've already thought about this earlier).
   a) Sketch the Range of $\mathbf{A}$ and locate the point in the Range that is closest to $\mathbf{b}$.
   b) Multiply both sides of $\mathbf{Ax} = \mathbf{b}$ by $\mathbf{A}^T$ and solve the resulting linear system.

Solving $\mathbf{Ax} = \mathbf{b}$ in Matlab

In many ways Matlab makes life easy for us. There is a single command in order to solve a linear system $\mathbf{Ax} = \mathbf{b}$

```
>> x = A\b
```

although it can also be used by typing

```
>> x = mldivide(A,b)
```

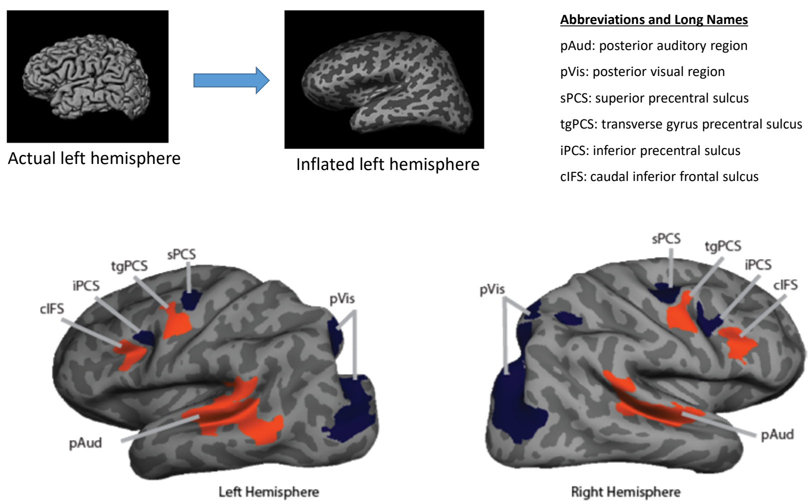
If there are more rows than columns then Matlab finds the approximate solution we discussed above. If there are equal numbers of rows and columns then Matlab computes a solution by LU decomposition. If there are less rows than columns then Matlab computes one of the infinite number of solutions - the solution it computes is not an approximation but it does select the solution that minimizes the length of the solution vector.

9.4 Correlation (activity sheet will be printed available at your table)

9.5 Analyzing brain data with special guest star Sam Michalka!

In this section, you are going to find the correlation between 12 brain regions and use this to support the hypothesis that these regions are part of two different “networks.” Brain regions are considered to be part of a network if the signals in the regions are correlated with each other. Here, we are going to look at the correlations of really slow changes over time (several seconds).

We have already collected the data using functional magnetic resonance imaging (fMRI), which looks at changes in the amount of oxygen in the blood in the brain. These changes in oxygen levels are related to large populations of neurons firing and changes slowly and with several seconds of delay. Here, we are going to refer to the signal that the fMRI detects as “activity” in the brain (this is a gross oversimplification of what’s going on, but is enough to approach this analysis with some guidance. Feed Sam coffee if you want to know more.)

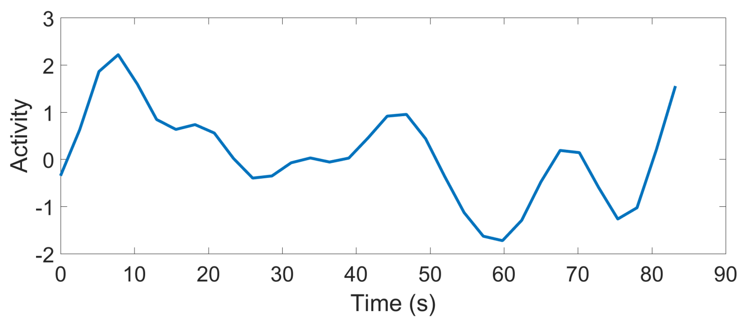


The fundamental idea behind what we are looking for is as follows:

- The brain can be divided into different regions (physical areas of the brain that are thought to serve some shared purpose). The brain is approximately symmetrical and has a left and a right hemisphere.
- We have identified 12 brain regions to investigate: 6 in the left hemisphere and 6 in the right hemisphere. They all have long fancy names (such as left superior precentral sulcus) and abbreviations (Left_sPCS). Due to the symmetry of the brain, there are matching names in each hemisphere (Right_sPCS and Left_sPCS both exist).
- The 12 brain regions that were selected were chosen because a previous experiment indicated that these regions are part of two different networks: one network that is more involved in processing auditory (sound) information and one network that is more involved in processing visual information.

- Our goal is to look at the correlation between the activity in each of these regions and the other 11 regions. Based on our first study, we hypothesize that the regions in blue will be correlated with other regions in blue to form a "visual network" and the regions in orange will be correlated with other regions in orange to form an "auditory network".

We have given you the data from one person (the actual study had more people). Also, we are not going to worry about statistical testing here; we are just going to look at the correlations in relation to each other.



Let's analyze some brain data! The plot below shows the first 80-ish seconds of data from the "right posterior auditory region" or "Right_pAud".

Exercise 9.5

1. **Consider your data and hypothesis.** Before you transform into a Matlab-mastermind, let's take a minute to think and work on the whiteboard.

    a) Roughly sketch a signal like the one shown in the plot above and label it "A". Don't worry about the details of this signal, just draw some bumps at approximately the same places.

    b) Sketch a line for a signal that has a high (but not perfect) positive correlation with "A". Label it "B". (A different color for each signal would be nice, if convenient.)

    c) Sketch a line for a signal that has a high (but not perfect) negative correlation with "A". Label it "C".

    d) Sketch a line for a signal that is uncorrelated with "A". Label it "D".

    e) Make a rough estimate of the correlation between each of these pairs of signals. Organize these estimates in a 4 x 4 table and add labels.

    f) Discuss alternate signals that you could have drawn for B, C, and/or D.

2. **Load and explore the data.** Whenever you encounter a new data set, it's helpful to look at the variables and make a few quick plots.

    a) Load the fmridata.mat file.

    b) Make sure you have the proper variables.

- There should be 12 brain region vectors, each starting with the word "Left" or "Right" (e.g., `Left_sPCS`). Each value in the vector contains a measurement at a particular point in time. The data were collected for a period of 11 minutes and 5.6 seconds. Measurements were taken every 2.6 seconds. A timepoint refers to one of those 2.6 second increments. How many timepoints do you expect? Does this match your vector length?
- The variable `braindata` contains all 12 brain region vectors organized into an array with dimensions `timepoints` $\times$ `brain regions`.
- The variable `names` contains the names of the 12 brain regions in an order that matches `braindata`.
- The variable `uncleandata` is similar to `braindata` and can be ignored for now.

c) Generate a plot with time on the x-axis and "activity" (the data values) on the y-axis. Represent each brain region as a line. You will first need to generate a new vector called `time` that represents the appropriate time increments (see above). Do your data seem reasonable? How do you know?

3. **Focus on a few pairs of regions.** The plot of all brain regions over all time points had a lot of information, so let's zoom in a bit and tie things back to the question at hand. Are the "auditory" (orange) regions correlated with other "auditory" regions? And are they correlated with the "visual" (blue) regions? Let's use one "auditory" regions called `Left_pAud` to investigate. The region `Left_pAud` is in the left hemisphere of the brain and should be colored orange in the image of the brains above.

   a) Let's begin looking the relationship between `Left_pAud` and `Right_pAud`.
   b) Look at the brain image to determine if `Right_pAud` is an "auditory" (orange) or "visual" (navy blue) type of region. Do you expect the activity of this region to be correlated with `Left_pAud`?
   c) Plot the signals from these two regions (with time on the x-axis). You may want to zoom in to the first 100 seconds to get another view. Guess the correlation between these regions.
   d) Calculate the correlation between between the two regions using the built-in MATLAB function `corr` (remember, you can get more information on a MATLAB command by typing, for example `doc corr` or `help corr` into the MATLAB command prompt).
   e) Is this correlation what you expected? How does it fit into our investigation of the correlation between "auditory" and "visual" regions?
   f) Repeat the steps above to look at the relationship between `Left_pAud` and `Left_sPCS`.
   g) Repeat the steps above to look at the relationship between `Left_pAud` and `Left_tgPCS`.

4. **Bonus fun: Generalize your analysis to calculate and display the correlations between all pairs of brain regions.** We looked at a few specific pairs of brain regions, but now we want to investigate our hypothesis by looking at all possible pairs of brain regions.

a) Create a matrix that contains the values of the correlation between each brain region. This is the same idea as the table of correlations that you wrote on the board. (Hint: Use the `braindata` matrix instead of the individual brain region vectors.) You can confirm that your values are correct by comparing to the correlations that you calculated above.

b) Display this matrix using `imagesc`. The following code may be helpful in labeling the plot:

```
>> colorbar; colormap('jet'); caxis([-1 1]);
% Set color bar parameters
>> xticks(1:12); xtickangle(-45); xticklabels(names);
% Label x axis
>> yticks(1:12); yticklabels(names);
% Label y axis
```

c) Discuss your observations from this plot. Refer to the figure with the brain image to determine which brain regions are “auditory” regions and which are “visual” regions? What patterns to you observe about their correlations?

5. **More Bonus fun: Explore the effects of preprocessing and outliers.** If you have extra time, check out the data in *uncleandata*. This data has the same structure as *braindata*, but contains some “bad” time points (208, 209, 210 , 232) caused by a blip in the recording equipment (this is a real problem that people face, though this example is extreme).

a) Plot this data versus time. How does this compare to the “clean” data from `braindata`?

b) Using the `uncleandata`, recreate the correlation matrix figure that you created using imagesc(), which shows the correlations between each pair of regions. How does this compare to your original version of this figure? What is important about these differences?

c) Try to clean the data yourself. There are many possible ways to deal with these “bad” timepoints, so you can choose one to investigate. Some options:

- Remove the timepoints completely.
- Replace the values at these time points with the mean of the good timepoints.
- Replace the values at these time points with a randomly selected value from the good timepoints.
- Another strategy of your choosing.

d) Explain which strategy you chose and why. These are decisions that scientists and engineers have to make. There are pros and cons of each. It’s really important to consider these and document the analysis decisions that you make.

e) Discuss with your partner/table or reflect on your own: what are some of the potential ethical implications of "cleaning" data?

f) Plot the activity in all of the brain regions over time for your “cleaned” data.

g) Plot the correlation matrix figure for your “cleaned” data.

h) What similarities and differences do you observe between the correlation matrix for the "unclean" and "clean" data? How do you interpret these differences? What do you take away from this for future work?

9.6 AI and Society Discussion

Framing (5 minutes)

Today we'll be talking about a constellation of issues that arise when AI technology, like facial recognition, is deployed in society. As the historian Melvin Kranzberg famously remarked, "Technology is neither good nor bad; nor is it neutral." As you saw in the reading from the night assignment, the effect of AI technology in society intersects a number of sensitive issues around race, class, and gender. Due to intersection of AI and these sensitive issues, it helps to take a few minutes to consider some guidelines for having fruitful discussions at your tables.

- Check out this poster put together by some Oliners with suggestions for having conversations on sensitive topics.
- The readings provide common information and framing, which we find is very helpful to finding common ground when discussing issues that individuals may relate to in very different ways.
- As you may be relatively new to these ideas, consider adopting a mindset of identifying key questions rather than necessarily coming to conclusions.
- When talking about the effect of a technology on a group that has been historically oppressed, you should be particularly sensitive in these discussions if you are not a member of this group. Be conscious of the ways in which your words might be experienced by those who may have faced a history of discrimination due to being a member of this group.

Unpacking the Readings (15 minutes)

Write down key concepts and clear up points of confusion on the readings.

- Joy Buolamwini's written testimony on bias in facial recognition technology (you may have watched this instead).
- Principles for Accountable Algorithms
- Google's Inclusive ML

Share Your Positive Application of AI (10 minutes)

Go around and share the application of AI that you think has the potential for great positive impact on society. Say a little bit about what you learned and how you think it would have a positive impact (e.g., in what ways and for whom).

What Did You Take Away? (15 minutes)

Please have one person take notes on this in some electronic format so they can submit it as part of their day survey

As a table, discuss what you took away from the readings and your discussion thus far. Here are some dimensions that you might want to explore.

1. What parts or quotes from the readings were most surprising / impactful to you?
2. Were you surprised by your reaction to reading any of the material (e.g., felt unexpectedly angry, sad, indifferent)?
3. What are the big questions that have been raised for you (these could be things that were already on your radar or new ones entirely)? These questions could relate to our society as a whole, your role as a citizien within society, your role as an Olin student, your future career path, etc.).
4. How do these readings intersect with knowledge you've gained from other contexts (e.g., in other courses or in your daily life experience)?

Solution 9.1

1. The column is a two-dimensional vector. The span is a line (slope = 1) in 2D space.
2. The columns are linearly-independent three-dimensional vectors. Their span is therefore a plane in 3D space. Since all the z-entries are zero, the plane is actually the xy-plane.
3. The columns are linearly-independent three-dimensional vectors. Their span is therefore a plane in 3D space. The plane is defined by the column vectors.

Solution 9.2

1. The Range of $\mathbf{A}$ is all multiples of $\begin{bmatrix}1\\1\end{bmatrix}$. Since $\mathbf{b}$ is a multiple of this vector then there is a solution. From an equation point of view, the solution is simply $x = 5$.
2. The Range of $\mathbf{A}$ is all multiples of $\begin{bmatrix}1\\1\end{bmatrix}$. Since $\mathbf{b}$ is not a multiple of this vector then there is no solution. From an equation point of view this makes sense because we are demanding that $x = 2$ and $x = 3$ at the same time.
3. The Range of $\mathbf{A}$ is a plane in 3D. Since $\mathbf{b}$ is the sum of the columns it must be in the Range of $\mathbf{A}$ and so there is a solution. From an equation point of view there are two linearly-independent equations in two unknowns.
4. The Range of $\mathbf{A}$ is a plane in 3D. Since $\mathbf{b}$ is not in this plane there is no solution. From an equation point of view this makes sense because trying to solve the equations results in an inconsistency.

Solution 9.3

In each case the "closest" point is the location where the "pointing" vector meets the surface at right angles, i.e. they are orthogonal.

Solution 9.4

1. Consider the linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} = \begin{bmatrix}1\\1\end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix}2\\3\end{bmatrix}$. (You've already thought about this earlier).
   a) Sketch the Range of $\mathbf{A}$ and locate the point in the Range that is closest to $\mathbf{b}$. (The Range is a straight line and the point is the orthogonal projection onto this line.)
   b) Multiply both sides of $\mathbf{Ax} = \mathbf{b}$ by $\mathbf{A}^T$ and solve the resulting linear system. (You should find that $x = 5/2$).
2. Consider the linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} = \begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix}3\\7\\5\end{bmatrix}$. (You've already thought about this earlier).

a) Sketch the Range of $\mathbf{A}$ and locate the point in the Range that is closest to $\mathbf{b}$. (The Range is a plane in 3D and the point is the orthogonal projection onto this line.)

b) Multiply both sides of $\mathbf{Ax} = \mathbf{b}$ by $\mathbf{A}^T$ and solve the resulting linear system. (You should find that $x = -3$ and $y = 7/2$.)

Chapter 10

Night 5

Learning Objectives

Concepts

- Describe the physical significance of the mean and standard deviation of a data set.
- Describe the physical significance of correlation, anti-correlation or non-correlation of two variables.
- Approximate the mean and standard deviation from a histogram of the data.
- Interpret the meaning of a pair of images that has a Pearson Correlation Coefficient of: about 0; or 0.5; or 0.9.
- Interpret the physical/mathematical meaning of the diagonal and off-diagonal elements in a 2 x 2 correlation matrix, $\mathbf{C} = \mathbf{A}^T\mathbf{A}$, if given the equation for the Pearson Correlation Coefficient.

MATLAB skills

- Compute the dot product of two vectors
- Set up the appropriate matrices to compute the correlation coefficient between two variables.

10.1 Correlation

Now let's consider that we measure two different associated quantities and want to test whether these are linearly correlated (if one goes up, the other also goes up), anti-correlated (if one goes up the other goes down) or uncorrelated (the behavior of one cannot be predicted by watching the behavior of the other). (Please note that correlation has nothing to do with causality!). There are many different measures of correlation, but we will discuss here one of the most common, the Pearson Correlation Coefficient.

For a pair of associated datasets $X = \{x_i\}$ and $Y = \{y_i\}$, each with n elements, we define the Pearson Correlation Coefficient to be:

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{(N-1)\sigma_x\sigma_y} \tag{10.1}$$

where μ_x, μ_y, σ_x and σ_y are the means and standard deviations of the datasets. Essentially, for each pair of values, we take the product of the variations from the mean, then sum these products up over all pairs of values and normalize by the expected variation as characterized by the standard deviation. If the two values are consistently always on the same side of the mean, then each term in the sum will contribute positively, and the total value will be close to one, indicating positive correlation. If the two values are consistently on the opposite sides of the mean, then each term in the sum will contribute negatives, and the total value will be close to negative one, indicating anticorrelation. If, for every pair, it is just as likely that the two values will be on opposite sides of the mean as on the same side of the mean, then the sum will go to zero, and the two values are uncorrelated.

Consider the following data:

	A	B	C	D	E	F	G	H	I	J
3		Poverty	Infant Mort	White	Crime	Doctors	Traf Deaths	University	Unemployed	Income
4	Alabama	15.7	9.0	71.0	448	218.2	1.81	22.0	5.0	42,666
5	Alaska	8.4	6.9	70.6	661	228.5	1.63	27.3	6.7	68,460
6	Arizona	14.7	6.4	86.5	483	209.7	1.69	25.1	5.5	50,958
7	Arkansas	17.3	8.5	80.8	529	203.4	1.96	18.8	5.1	38,815
8	California	13.3	5.0	76.6	523	268.7	1.21	29.6	7.2	61,021
9	Colorado	11.4	5.7	89.7	348	259.7	1.14	35.6	4.9	56,993
10	Connecticut	9.3	6.2	84.3	256	376.4	0.86	35.6	5.7	68,595
11	Delaware	10.0	8.3	74.3	689	250.9	1.23	27.5	4.8	57,989
12	Florida	13.2	7.3	79.8	723	247.9	1.56	25.8	6.2	47,778
13	Georgia	14.7	8.1	65.4	493	217.4	1.46	27.5	6.2	50,861
14	Hawaii	9.1	5.6	29.7	273	317.0	1.33	29.1	3.9	67,214
15	Idaho	12.6	6.8	94.6	239	168.8	1.60	24.0	4.9	47,576

Exercise 10.1

1. Look over the data. By eye, which columns look correlated? Anticorrelated? Uncorrelated?
2. Choose your two favorite columns of data from this dataset. Input these into vectors in Matlab. For each of these vectors, subtract off the mean, and then divide out the standard deviation.
3. With these vectors, how would you directly compute the correlation coefficient between them? Go ahead and do this in MATLAB, and reflect on your result.

A note of warning. Correlation does not imply causation.

Exercise 10.2

To drive this point home, visit the Spurious Correlation Website. Follow the link at the bottom of the site to discover and plot a spurious correlation of your very own.

10.2 Correlation: The Idea, the Math, and the MATLAB

Here we are going to consider a matrix like the one you constructed above, based on the datasets, which has elements of the correlation coefficients. Let's first consider a matrix $\mathbf{B}$ which has the datasets as columns:

$$\mathbf{B} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{pmatrix}$$

If we subtract out the means and divide out the standard deviation and a factor of square root of $N-1$, we get the matrix $\mathbf{A}$:

$$\mathbf{A} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} \frac{x_1-\mu_x}{\sigma_x} & \frac{y_1-\mu_y}{\sigma_y} \\ \frac{x_2-\mu_x}{\sigma_x} & \frac{y_2-\mu_y}{\sigma_y} \\ \frac{x_3-\mu_x}{\sigma_x} & \frac{y_3-\mu_y}{\sigma_y} \\ \dots & \dots \end{pmatrix}$$

where μ_x, μ_y and σ_x, σ_y are the mean and standard deviations of each column, and N is the number of samples (rows). Now we can write the *correlation matrix* as $\mathbf{C} = \mathbf{A}^T\mathbf{A}$ which has elements of the self and cross correlations between the datasets.

Exercise 10.3

1. Before we start to use this idea, let's think through it a bit...
   a) What is the size of the matrix $\mathbf{C}$?
   b) What are the elements on the *diagonal* of this matrix?
   c) What are the elements of the off-diagonal? What is element C_{12} of this matrix? What is element C_{21}? What do you notice? Is this always going to be true? What about if we had three datasets? What can you say about the elements C_{13} vs C_{31}?
   d) If you create a data matrix that has completely identical columns of data, what should the correlation matrix look like?
   e) If you create a data matrix that has completely uncorrelated datasets, what should the correlation matrix look like?
2. Now let's actually implement this. For the sake of getting you more comfortable manipulating matrices in MATLAB, do the following at the command line, and see if you can predict what each term will do before you type it in. For each command, explain what MATLAB is doing.

```
>> a = [(1:10)',rand(10,1)]
>> o = ones(size(a,1),1)
>> m = o*mean(a)
>> s = o*std(a)
>> b = (a-m)./s
>> c=(1/(size(a,1)-1))*b'*b
```

3. What would happen if you used this matrix for a instead:

```
>> a = [(1:10)',(1:10)'/2 -(1:10)' ]
```

   Make a prediction, and try it out.
4. Now learn how to use the MATLAB function `corrcoef`, which will compute the correlation coefficients using the original matrix.

10.3 From Data to Dimensions

Thus far we've made a distinction between vectors as representing points in space, and vectors as representing data (e.g., a list of intensity values for pixels). But if you wanted to, there's no reason you couldn't think of a vector of n data points as representing a point in n dimensional space (and, in fact, it would be very powerful *to*). If you did this, you could define all kinds of interesting things. For example, you could ask about the magnitude of the vector (i.e., the size of a data point), the distance from one vector (i.e., data point) to another, the dot product of two vectors (i.e., data points), etc.

Exercise 10.4

1. As a thought experiment, think about the following questions.
   a) Imagine you have grayscale images that are 100×100 pixels. If you represent each image as a vector, how large is the vector space?
   b) If the intensity of pixel ij is a_{ij}, come up with an expression for the magnitude of the vector that describes the image $\mathbf{a}$.
   c) Come up with an expression that represents the distance between images $\mathbf{a}$ and $\mathbf{b}$.
   d) What does $\mathbf{a}^T\mathbf{b}$ tell you? What vector operation is this?
2. Now that you've thought through this about, load three face images, and calculate the distance between each pair (i.e., take the difference between the two images, square each element, then sum up all the squared elements and take the square root of the sum). Do your answers make sense?

10.4 Correlation in Facial Recognition

Kinds of Correlation in an Image Set

If we think about pictures now, we can think about two different correlations: the correlation between a given pair of pixels (across all the pictures in a data set), and the correlation between pictures (across all the pixels in those images). In order to compute an accurate correlation coefficient, you need to have multiple data points in each set being correlated, e.g., many pixels in each picture being correlated, or many pictures across which a pair of pixels (pixel locations) can be correlated.

Think about what each of these correlations *means*. What would a high correlation between a given pair of images mean? What about a high correlation between a given pair of pixels (e.g., the upper-left-most pixel and the upper-right-most pixel)? It might help to open a few face images or draw some face sketches to think about.

Exercise 10.5

Consider six grayscale pictures, each with a resolution of m x n pixels.

1. What is the size of the data matrix containing these six pictures as the columns?
2. What is the expression for the correlation matrix between the pictures? What size is this correlation matrix?
3. What is the expression for the correlation matrix between different pixels? Pay careful attention to the mean and standard deviation you are using. What is the size of this correlation matrix?
4. People's faces are approximately left-right symmetric. How would you expect this to affect the entries in the correlation matrix between different pixels?

Test your understanding...

- Pull in **six images** from the class data matrix (`test_images` variable in `face_bases.mat` file) and put them in a variable called `faces`. Each of these images should come from different people - there are 8 images per person stored in the data matrix.
- Use the `resample` command to bring them down to a smaller resolution (e.g., 25 x 25) using
  `dfaces = imresize(faces,[25 25]);`
  his should be a 25 x 25 x 6 matrix.
- Now reshape them appropriately to create a matrix in which each column is a (reshaped) face using
  `rdfaces = reshape(dfaces,size(dfaces,1) * size(dfaces,2),size(dfaces,3));`

Exercise 10.6

1. Find the correlation between six different images. Which images have the highest correlation?
2. Now find the correlation between pixels across images. Try taking a single column of this matrix and reshape that column into an image. What does that image tell you? You may want to repeat this reshape and visualization for columns 1, 25, 400, and 625 to get a feel for what is happening.

10.5 Diagnostic Quiz

Please see Canvas for the questions.

Solution 10.3

1. a) Since $\mathbf{A}$ has size $N \times 2$, we know $\mathbf{A}^T$ has size $2 \times N$. Then $\mathbf{C} = \mathbf{A}^T\mathbf{A}$ has size 2×2.
   b) Each element of the diagonal will be 1.
   c) The elements $\mathbf{C}_{12} = \mathbf{C}_{21}$ is the correlation between the two columns of data. Regardless of size, the correlation matrix will be symmetric.
   d) If the data matrix had identical columns of data, the correction matrix would be all 1s.
   e) If the data is uncorrelated, the off-diagonal entries in the correlation matrix will be all 0s, so it will be the identity matrix. (Note: real data is unlikely to have 0 correlation, just by accident, so it will just have numbers that are close to 0.)
2. You should end up with a correlation matrix between the two columns of `a`.
3. Again, you end up with a correlation matrix between the two columns of `a`, but since the numbers are non-random and negatively correlated, you get the matrix $\mathbf{c} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$.
4. You should use “help” or “doc” to find information on how to use this function.

Solution 10.4

1. Each pixel is represented by a number. So the vector representing an image has 10000 entries and thus lives in 10000-dimensional space.
2. The magnitude of the vector is given by
$$\|\mathbf{a}\| = \left(\sum_{i,j} a_{i,j}^2 \right)^{1/2}.$$
3. The distance between two images $\mathbf{a}$ and $\mathbf{b}$ is given by
$$\|\mathbf{a} - \mathbf{b}\| = \left(\sum_{i,j} (a_{i,j} - b_{i,j})^2 \right)^{1/2}.$$
4. The dot product $\mathbf{a}^T\mathbf{b}$ tells you how similar the two vectors are.

Solution 10.5

1. Each picture is represented by mn data points. So the data matrix containing these six pictures as columns is $mn \times 6$.
2. To find the correlation we can either use the MATLAB code from Exercise 10.3 or the command `corrcoef`. The correlation matrix will be 6×6.

3. To find the correlation between pixels, we need to take the transpose of our data matrix. This new data matrix will be $6 \times mn$, since we have mn variables (each pixel) and 6 observations (within in picture). The correlation matrix will then be $mn \times mn$.

4. High correlation between pixels equidistant from centerline.

Solution 10.6

1. To find the correlation matrix between the six images, enter
   » `corrcoef(rdfaces)`.

2. To find the correlation across pixels, enter
   » `pixels=corrcoef(transpose(rdfaces));`. We can reshape the first column of this matrix and convert it into an image using
   » `pixels1=reshape(pixels(:,1),25,25);`
   » `imagesc(pixels1).`
   The (i, j) entry of this image tells us how similar the top-left pixel is to the (i, j) pixel. (Note: the pixels are "numbered" 1–625 going down the first column, then down the second column, and so on.) Repeating this reshape and visualization procedure on column 400 will give you the correlation between pixel 400 and each of the other pixels, for example.

Chapter 11

Day 6: AI Discussion, Smile Detection and Eigenthings

11.1 Schedule

- 0900-0915: Debrief
- 0915-1000: AI and Society Discussion
- 1000-1030: Smile Detection—Concepts
- 1030-1045: Coffee
- 1045-1115: Machine Learning
- 1115-1145: Smile Detection—Implementation
- 1145-1210: Eigenthings
- 1210-1230: Review/preview

11.2 Debrief and Synthesis [15 mins]

- Please discuss your overnight work with your table-mates, and get help with the ideas that you are still confused by.

11.3 AI and Society Discussion [45 mins]

Framing (5 minutes)

Today we'll be talking about a constellation of issues that arise when AI technology, like facial recognition, is deployed in society. As the historian Melvin Kranzberg famously remarked, "Technology is neither good nor bad; nor is it neutral." As you saw in the reading from the night assignment, the effect of AI technology in society intersects a number of sensitive issues around race, class, and gender. Due to intersection of AI and these sensitive issues, it helps to take a few minutes to consider some guidelines for having fruitful discussions at your tables.

- Check out this poster put together by some Oliners with suggestions for having conversations on sensitive topics.
- The readings provide common information and framing, which we find is very helpful to finding common ground when discussing issues that individuals may relate to in very different ways.
- As you may be relatively new to these ideas, consider adopting a mindset of identifying key questions rather than necessarily coming to conclusions.
- When talking about the effect of a technology on a group that has been historically oppressed, you should be particularly sensitive in these discussions if you are not a member of this group. Be conscious of the ways in which your words might be experienced by those who may have faced a history of discrimination due to being a member of this group.

Unpacking the Readings (15 minutes)

Write down key concepts and clear up points of confusion on the readings.

- Joy Buolamwini's written testimony on bias in facial recognition technology (you may have watched this instead).
- Principles for Accountable Algorithms
- Google's Inclusive ML

Share Your Positive Application of AI (10 minutes)

Go around and share the application of AI that you think has the potential for great positive impact on society. Say a little bit about what you learned and how you think it would have a positive impact (e.g., in what ways and for whom).

What Did You Take Away? (15 minutes)

Please have one person take notes on this in some electronic format so they can submit it as part of their day survey

As a table, discuss what you took away from the readings and your discussion thus far. Here are some dimensions that you might want to explore.

1. What parts or quotes from the readings were most surprising / impactful to you?
2. Were you surprised by your reaction to reading any of the material (e.g., felt unexpectedly angry, sad, indifferent)?
3. What are the big questions that have been raised for you (these could be things that were already on your radar or new ones entirely)? These questions could relate to our society as a whole, your role as a citizen within society, your role as an Olin student, your future career path, etc.).
4. How do these readings intersect with knowledge you've gained from other contexts (e.g., in other courses or in your daily life experience)?

11.4 Smile Detection—Concepts [30 mins]

In this session we are going to use our toolbox of linear algebra skills to "detect" whether or not a person is smiling in a photograph. The approach that we will take is very common in **machine learning** - we will use a dataset to **train** our algorithm, and we will use a different dataset to **test** our algorithm. We will first develop the conceptual framework and then implement the approach in MATLAB.

The Big Idea

Let's assume that we have 100 training photos of faces, each consisting of a 5 by 5 grid of pixels. Let's pack these into a matrix $\mathbf{A}$ with 100 rows and 25 columns, i.e. every row is a different face and every column is a different pixel.

Let's also assume that we have already classified every training face as "smiling" or "not-smiling". Let's create a column vector $\mathbf{b}$ with 100 rows (corresponding to each face) which has either 1 (smiling) or 0 (not-smiling).

Let's now develop a linear system of algebraic equations by trying to express the vector $\mathbf{b}$ as a linear combination of the columns of $\mathbf{A}$, i.e.

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

Notice that the vector $\mathbf{x}$ is a column vector with 25 rows - one row for each pixel. Since there are more rows than columns we know that an **exact** solution does not exist, so we will find the **approximate** solution by orthogonal projection, i.e. we will solve

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

Now that we have the vector $\mathbf{x}$, let's use it to detect whether a test image is smiling. Assuming that the test image is packed into a single row vector $\mathbf{t}$ (with 25 columns) then the product

$$\mathbf{t}\mathbf{x}$$

will return a scalar. If this scalar is close to "1" then we predict the face is smiling. If this scalar is close to "0" then we predict the face is not smiling.

Exercise 11.1

In this exercise you will be carefully reading and interpreting this big idea. We are including these questions as a scaffold, pointing out interesting features along the way.

1. Read "The Big Idea" again!
2. Interpret what it means to write down the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ and give a meaning to the vector $\mathbf{x}$.
3. Interpret the product $\mathbf{A}^T\mathbf{A}$ and the product $\mathbf{A}^T\mathbf{b}$.
4. The vector $\mathbf{x}$ does not satisfy $\mathbf{A}\mathbf{x} = \mathbf{b}$ exactly. What does the expression $\mathbf{A}\mathbf{x} - \mathbf{b}$ tell you?
5. How would you decide whether your "trained" algorithm was worth using on a test dataset?
6. Assume you had 40 test images with 25 pixels each and that you pack them into a matrix $\mathbf{T}$ with 40 rows and 25 columns. Write down the matrix-vector product you would use for smile detection on this test dataset.
7. How would you measure the accuracy of your predictions if we also provided you with the data on whether each test image was smiling or not?

11.5 Machine Learning in general [30 mins]

Machine learning is an interdisciplinary field concerned with the idea that rather than preprogramming machines to solve tasks explicitly, instead we can program them to learn to solve tasks through experience. In order to connect this idea to what you've done thus far, first we'll introduce a somewhat more formal definition of machine learning.

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.*
>
> — *Tom Mitchell*

Let's take the smile detection problem that you just framed in the preceding section of the document.

Symbol from Mitchell's Definition	Smile Detection
E	5 by 5 grids of pixels with corresponding labels as to whether or not the person in the grid is smiling
T	Given a new 5 by 5 grid of pixels, predicting whether the person in the image is smiling.
P	Accuracy on predicting whether a person is smiling (e.g., percent correct)

The last part of the definition states that in order to say a program is "learning", it should "improve with experience E." In the case of smile detection, what this means is that given more images with corresponding indication of whether each person is smiling, our program should get better at predicting whether a face is smiling. Later in this document when you implement smile detection, you'll see that the framing of smile detection as an LSAE problem absolutely meets this definition (and does a surprisingly good job at it too)!

Exercise 11.2

In this exercise you and your table-mates will be working to frame various machine learning problems as systems of linear equations. In doing so, you should answer the following questions.

- What is the thing you want to predict (i.e., what is your $\mathbf{b}$ in $\mathbf{A}\mathbf{x} = \mathbf{b}$).
- What quantities are you using to make your prediction (i.e., what do each of the rows and columns in the matrix $\mathbf{A}$ represent)?
- While $\mathbf{x}$ would be determined by solving $\mathbf{A}\mathbf{x} = \mathbf{b}$, come up with a guess as to what $\mathbf{x}$ would be (don't worry about coming up with numbers. Instead, try to identify the sign of each element of $\mathbf{x}$ and whether its magnitude is large or small relative to the other entries of $\mathbf{x}$).
- How would you measure how well your system works? For example, for smile detection we might apply our learned model $\mathbf{x}$ to new data and see how often it correctly predicts the facial expression (smiling vs. not smiling) of the person in each image.

- What sorts of issues of bias might you have to worry about in this system?

1. AirBnB has a smart pricing option that lets folks who list their properties on the site have the price for those listings determined automatically. How might you frame the creation of this smart pricing tool as solving a system of linear equations? We suggest that you follow the steps outlined at the start of this exercise. You might want to do a quick AirBnB search if you are not familiar with the site.

2. Netflix suggests content to a user that they are likely to enjoy based on their viewing history as well as the viewing histories of others. In the past (although we think possibly not anymore), they used to also incorporate ratings data (1 to 5 stars) of particular movies from both the user receiving the recommendation and other users on the site. How might you frame the creation of this recommendation system in terms of solving a system of linear equations? We suggest that you follow the steps outlined at the start of this exercise. You might want to do a quick Netflix search if you are not familiar with the site.

11.6 Smile Detection—Implementation [30 mins]

Please download the file `smiles.mat` from the canvas site. If you load this file in MATLAB, you will then have access to the following variables in your workspace.

```
train_data       - a 3D array containing 19685 24 x 24 pixel
                   images of faces
smile_flag_train - a vector of the same length as the number
                   of images in train_data, with 1s indicating
                   which images are smiling
test_data        - 500 24 x 24 pixel images of faces
smile_flag_test  - a vector of the same length as the number
                   of images in test_data, with 1s indicating
                   which images are smiling
```

The ‘train data’ and the associated ‘smile flag train’ are the sets of data you should use to develop your mathematical model. The ‘test data’ and its associated ‘smile flag test’ are the sets of data you should use to test your algorithm when you are finished!

Exercise 11.3

For this exercise we recommend that you use our walkthrough notebook. The notebook has embedded solutions or you can try it with minimal scaffolding using the suggested process below. Even if you decide not to use the walkthrough notebook, it’s worth running the embedded solutions to pickup some techniques for visualizing your smile detector model. In any case, as you move into actually implementing the smile detector, we recommend you work as pairs at your table. Working in a pair will allow you to have someone to bounce ideas off of and also make sure you can see the

laptop easily.
Only if you decide NOT to use the walkthrough notebook, you should consider following the procedure below to implement the smile detector.

1. Sketch out a set of steps you would take in order to implement smile detection. (Just words here - no code. e.g. we will have to pack all images into a single matrix)
2. Turn this set of steps into MATLAB pseudo-code. Identify important coding elements without implementing, e.g. we will use **reshape** to pack the given dataset into a matrix.
3. Review the documentation for MATLAB functions that will be used and be clear on how to use them before implementation, e.g. » help reshape
4. Methodically implement smile detection in MATLAB, testing as you go.

11.7 Introduction to Eigenthings [25 mins]

We are now going to learn the secret of the genie ...

Eigenvalues and Eigenvectors: Definition and Notation

Consider a square $n \times n$ matrix $\mathbf{A}$. A vector $\mathbf{v}$ is said to be an eigenvector of $\mathbf{A}$ with corresponding eigenvalue λ if $\mathbf{v}$ is not a vector of all zeros, and

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}. \tag{11.1}$$

If we treat $\mathbf{A}$ as a transformation matrix then $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ if it is simply scaled when acted on by the matrix $\mathbf{A}$. In other words, $\mathbf{v}$ does not change direction when acted upon by $\mathbf{A}$. In general, an $n \times n$ matrix has exactly n eigenvalues (although some of these may be repeated and some of these may be complex!). Note that any scalar multiple of an eigenvector of a matrix is also an eigenvector of that matrix - it's only the direction of the eigenvector that matters.

In the next overnight assignment we are going to develop formal techniques for finding the eigenvalues and eigenvectors of matrices. For now, we are going to focus on concepts and developing some intuition.

Exercise 11.4

1. Show that $\mathbf{v} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ is an eigenvector of the following matrix by computing the product $\mathbf{A}\mathbf{v}$, and find the corresponding eigenvalue

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \tag{11.2}$$

2. On the same axes, plot the vector representing $\mathbf{v} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ and $\mathbf{Av}$. Does the plot confirm that this is an eigenvector?
3. On the same axes, plot the vector representing $\mathbf{u} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{bmatrix}$ and $\mathbf{Au}$. Is this is an eigenvector of $\mathbf{A}$?

Eigenvalues and eigenvectors of a diagonal matrix

Recall from our earlier work that the matrix

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

scales vectors by a factor of 2 in the x-direction and by a factor of 3 in the y-direction. Thus a vector that had a non-zero component only in the x direction will be scaled by a factor of 2 when transformed by this matrix. In other words, $\lambda_1 = 2$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and that $\lambda_2 = 3$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Let's check if the first one is true:

$$\mathbf{Av}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 2\mathbf{v}_1$$

Therefore $\lambda_1 = 2$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Exercise 11.5

Confirm that $\lambda_2 = 3$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ by computing the product $\mathbf{Av}_2$.

Based on this example, we can heuristically guess that the eigenvalues of an $n \times n$ diagonal matrix are the entries on the diagonal. The n eigenvectors each have a single 1 in them, with the remaining entries being zero.

Exercise 11.6

What are the eigenvalues and eigenvectors of the following diagonal matrices

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

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

Exercise 11.7

1. Assume that $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is an eigenvector with eigenvalue 3. Construct an appropriate matrix A with this eigenvalue and eigenvector by first rotating $\mathbf{v}$ onto the x-axis, scaling it by 3, and then rotating back.

Exercise 11.8

What is one eigenvector of the following matrix?

$$\mathbf{R} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{11.6}$$

11.8 Review and Preview [20 minutes]

Solution 11.1

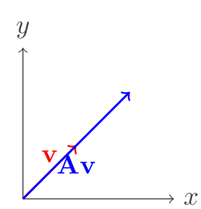
1. Read, read, read ....
2. We are trying to take a linear combination of the data in order to predict whether each image is smiling or not. The vector $\mathbf{x}$ is the magic set of weights we have to use. Its size is the same as the number of pixels, so maybe it should look like a mask that we can place over an image to tell us whether it is smiling.
3. The product $\mathbf{A}^T\mathbf{A}$ is like a pixel to pixel correlation matrix, except we haven't scaled the data matrix $\mathbf{A}$. The product $\mathbf{A}^T\mathbf{b}$ is the sum of the images that are smiling.
4. The expression $\mathbf{A}\mathbf{x} - \mathbf{b}$ tells us the error in predicting whether a training image is smiling or not.
5. We could add up how often the predictor is correct and divide by the number of images to get an estimate of the accuracy. We would decide on a cut-off before we used it on a test dataset.
6. It is simply $\mathbf{T}\mathbf{x}$.
7. As before. Determine how many we got correct and average it.

Solution 11.3

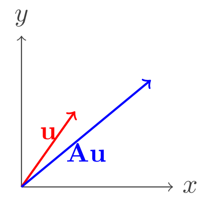
You can use the solutions that are embedded in the walkthrough notebook.

Solution 11.4

1. Compute that
$$A\mathbf{v} = \begin{bmatrix} \sqrt{2} \\ \sqrt{2} \end{bmatrix} = 2\mathbf{v}$$
and so the corresponding eigenvalue is $\lambda = 2$.
2. As we can see in the picture below, both $\mathbf{v}$ and $\mathbf{A}\mathbf{v}$ point in the same direction, which confirms $\mathbf{v}$ is an eigenvalue of $\mathbf{A}$



3. As we can see in the picture below, $\mathbf{u}$ and $\mathbf{A}\mathbf{u}$ point in different directions, so $\mathbf{u}$ is not an eigenvector of $\mathbf{A}$



Solution 11.5

We compute that

$$\mathbf{A}\mathbf{v}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 3\mathbf{v}_2.$$

Solution 11.6

1. The eigenvalues are $\lambda_1 = -3$, $\lambda_2 = -1$ and $\lambda_3 = 4$ and the corresponding eigenvectors are $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.
2. The eigenvalues are $\lambda_1 = 2$, $\lambda_2 = 4$ and $\lambda_3 = 0$ and the corresponding eigenvectors are $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

Solution 11.7

1. First rotate it using a 45 degree clockwise rotation matrix

$$\mathbf{R}(-45) = \begin{bmatrix} cosd(-45) & -sind(-45) \\ sind(-45) & cosd(-45) \end{bmatrix} \tag{11.3}$$

Now that it is along the x-axis we can scale it by 3 using the scaling matrix

$$S = \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \tag{11.4}$$

We then rotate it back using $\mathbf{R}(45)$. Multiplying these matrices together gives

$$\mathbf{A} = \begin{bmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix} \tag{11.5}$$

Solution 11.8

The vector $\mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ is an eigenvector of $\mathbf{R}$ because it is the rotation axis and therefore remains unchanged on rotation.

Chapter 12

Night 6: Eigenvalues and Eigenvectors

Learning Objectives

Concepts

- Compute the eigenvalues and eigenvectors of a 2×2 matrix by hand
- Compute the eigenvalues and eigenvectors of an $n \times n$ matrix using MATLAB
- Describe the geometric meaning of eigenvalues and eigenvectors
- Use eigenvectors to compute and interpret directions of variation in data

MATLAB skills

- Compute the eigenvectors and eigenvalues of a given matrix
- From a given dataset, set up the relevant matrices and compute the covariance matrix of the dataset.

What is this about?

The big ideas of this assignment are eigenvectors and eigenvalues. Recall that when you multiply a vector by a matrix, the resulting vector usually points in a different direction. An eigenvector of a square matrix is a vector which does not change direction when multiplied by that matrix. It can only change in length. The eigenvalue corresponding to this eigenvector is the scale factor that is applied to that eigenvector as a result of the matrix multiplication. Therefore, the eigenvector of a matrix points in a special direction – its a direction that is not modified by the linear transformation associated with that matrix. This is an idea that we will keep coming back to in a number of different ways throughout QEA (including next semester). The ideas contained here can be applied in many ways (many of which we won't get to until next semester) such as

- Directions of greatest variation in data.
- Natural co-ordinates of systems.
- Frequency response of filters.
- Analysis of dynamical systems.

Reference Material

- Eigenvalues and Eigenvectors by 3Blue1Brown (watch first 14 mins)
- Paul's Online Notes. Review : Eigenvalues and Eigenvectors
- Intro to eigenvectors by PatrickJMT

- Calculating eigenvalues and eigenvectors of a 2×2 matrix by PatrickJMT.

12.1 Calculating Eigenvalues and Eigenvectors of Matrices

Recall from class that λ is an eigenvalue of a matrix $\mathbf{A}$ with corresponding eigenvector $\mathbf{v}$ if $\mathbf{Av} = \lambda\mathbf{v}$. Geometrically, this means that the matrix $\mathbf{A}$ doesn't change the direction of $\mathbf{v}$, it simply scales it by a factor of λ.

Given a square matrix, how can we find its eigenvalues and eigenvectors? In class, we calculated these by hand for the special case of diagonal matrices, and now we will move to generic 2×2 matrices. For general square matrices which are larger than 2×2, we will use MATLAB's `eig` to compute the eigenvalues and eigenvectors.

Finding eigenvalues

So far we've dealt with matrices for which it is possible to think your way to the eigenvalues. For general matrices, this is rarely the case, and we need a method that is foolproof. The method most widely adopted involves the determination of an algebraic equation for the eigenvalues, usually known as the *characteristic* equation. For this reason, eigenvalues are often known as characteristic values.

Let's start with an example. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 18 & -2 \\ 12 & 7 \end{bmatrix}$$

The definition of an eigenvalue and eigenvector imply that we are seeking λ and $\mathbf{v}$ which satisfy

$$\mathbf{Av} = \lambda\mathbf{v}.$$

We subtract $\lambda\mathbf{v}$ from both sides

$$\mathbf{Av} - \lambda\mathbf{v} = \mathbf{0}$$

and then factor the left hand side to give

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

For this example we have

$$\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 18-\lambda & -2 \\ 12 & 7-\lambda \end{bmatrix}.$$

We are only interested in $\mathbf{v}$ that are nonzero, i.e., $\mathbf{v}$ is not the vector of all zeroes. (This is because $\mathbf{v} = \mathbf{0}$ is always a solution to $\mathbf{Av} = \lambda\mathbf{v}$ for any $\mathbf{A}$ and any λ, so it's not very interesting or informative.) Assuming $\mathbf{v}$ is nonzero implies that the matrix $(\mathbf{A} - \lambda\mathbf{I})$ is not invertible. Why? If $(A - \lambda\mathbf{I})$ were invertible, then we could rearrange the equation to get

$$\mathbf{v} = (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{0} = \mathbf{0}$$

which contradicts our assumption that $\mathbf{v} \neq \mathbf{0}$. THerefore, $(\mathbf{A} - \lambda\mathbf{I})$ is not invertible.

Since $(\mathbf{A} - \lambda\mathbf{I})$ is not invertible, it must have determinant zero. In other words,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0.$$

In our example, this implies that

$$\det(\mathbf{A} - \lambda\mathbf{I}) = (18 - \lambda)(7 - \lambda) + 24 = 0.$$

This is called the characteristic equation:

$$(18 - \lambda)(7 - \lambda) + 24 = 0$$

or, rearranged,

$$\lambda^2 - 25\lambda + 150 = 0.$$

The characteristic equation is a polynomial with the variable λ that arises by setting the determinant of $(\mathbf{A} - \lambda\mathbf{I})$ equal to zero. The solutions to this polynomial give the eigenvalues λ. In our example, the polynomial can be factored

$$(\lambda - 15)(\lambda - 10) = 0$$

so that gives eigenvalues $\lambda_1 = 10$ and $\lambda_2 = 15$. (We could use the quadratic formula if necessary.)

Let's retrace our steps: If λ is either 10 or 15, then the determinant of $(\mathbf{A} - \lambda\mathbf{I})$ is zero. This implies that $(\mathbf{A} - \lambda\mathbf{I})$ is not invertible, so we can look for nonzero solutions $\mathbf{v}$ to $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$ and those $\mathbf{v}$ are eigenvectors associated to the eigenvalue λ.

In summary, here's the general procedure for finding the eigenvalues of a matrix:

1. Rearrange $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ to get $(\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0}$.
2. Compute the determinant of $(\mathbf{A} - \lambda\mathbf{I})$.
3. Since the matrix is not invertible, we set that determinant equal to zero: $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. This gives a polynomial in λ, known as the characteristic equation.
4. Solve the polynomial for the roots λ. Those are the eigenvalues.

Exercise 12.1

1. You already know that the eigenvalues of a diagonal matrix are just the entries on the diagonal. Using the above procedure, confirm that

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

has eigenvalues $\lambda_1 = 2$ and $\lambda_2 = -3$.

2. Notice that one of the eigenvalues is positive and one is negative. The eigenvector associated with $\lambda_1 = 2$ is $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and the eigenvector associated with $\lambda_2 = -3$ is $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Plot $\mathbf{v}_1, \mathbf{v}_2, \mathbf{A}\mathbf{v}_1$ and $\mathbf{A}\mathbf{v}_2$. What affect does the negative sign in the eigenvalue have? In other words, what is the difference between a negative and positive eigenvalue?

It's worth noting that eigenvalues come in more flavors than positive or negative. They can also be complex numbers. For now, will focus on matrices with real eigenvalues, but if you're curious about the complex case, you can learn about it in this worksheet (ignore the first page).

Eigenvalues in terms of the trace and determinant

The steps we took just in the previous sections will work for any matrix, so let's apply it to the most general 2×2 matrix

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

Again we seek λ and $\mathbf{v}$ so that

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

or equivalently

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

Non-zero solutions for $\mathbf{v}$ exist when

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

so in the general case the characteristic equation is

$$(a - \lambda)(d - \lambda) - bc = 0$$

Expanding and simplifying gives

$$\lambda^2 - (a + d)\lambda + (ad - bc) = 0$$

which is a second-order polynomial in λ with two coefficients. Notice that the last one is just $det(\mathbf{A})$ and the middle one involves the sum of the diagonal entries of $\mathbf{A}$, which is known as the trace of $\mathbf{A}$ or $tr(\mathbf{A})$ for short. The characteristic equation is therefore

$$\lambda^2 - tr(\mathbf{A})\lambda + det(\mathbf{A}) = 0$$

Finally, let's consider the solutions of the characteristic equation for a 2x2 matrix. Using the quadratic formula we have

$$\lambda = \frac{tr(\mathbf{A}) \pm \sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}}{2}$$

If you recall all the work you did in school with the solutions to the quadratic, you will notice that there are two solutions as expected, one for each eigenvalue. Furthermore, the solutions may be *complex* if

$$tr(\mathbf{A})^2 - 4det(\mathbf{A}) < 0.$$

Exercise 12.2

Determine the trace and determinant of the following 2x2 matrices and then write down the corresponding characteristic equation. Solve the characteristic equation to find the eigenvalues.

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

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

Exercise 12.3

Optional problem if you're interested in further exploring the relationship between the trace, determinant, and eigenvalues.

1. Use the solutions of the characteristic equation to prove that $\lambda_1 + \lambda_2 = tr(\mathbf{A})$.
2. Use the solutions of the characteristic equation to prove that $\lambda_1 \lambda_2 = det(\mathbf{A})$.
3. Use the solutions of the characteristic equation to prove that the eigenvalues of a symmetric 2×2 matrix are real.

Finding Eigenvectors

In the example in the previous section, we discovered that the eigenvalues of

$$\mathbf{A} = \begin{bmatrix} 18 & -2 \\ 12 & 7 \end{bmatrix}$$

are $\lambda_1 = 10$ and $\lambda_2 = 15$. How do we find the corresponding eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$?

First, let's find the eigenvector corresponding to $\lambda_1 = 10$. Remember that we knew λ_1 was an eigenvalue because it solved the characteristic equation, i.e., $\det(\mathbf{A} - \lambda_1 \mathbf{I}) = \mathbf{0}$. This is important because it implies $(\mathbf{A} - \lambda_1 \mathbf{I})$ is non-invertible, and therefore, there exists a nonzero vector $\mathbf{v}_1$ such that $(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{v}_1 = \mathbf{0}$. But it's not enough just to know that such a vector exists, we want to know exactly what it is.

In our running example, this means we are looking for $\mathbf{v}_1$ such that

$$(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{v}_1 = \begin{bmatrix} 18 - 10 & -2 \\ 12 & 7 - 10 \end{bmatrix} \mathbf{v}_1 = \begin{bmatrix} 8 & -2 \\ 12 & -3 \end{bmatrix} \mathbf{v}_1 = \mathbf{0}.$$

Let's write $\mathbf{v}_1$ in terms of its components

$$\mathbf{v}_1 = \begin{bmatrix} a \\ b \end{bmatrix},$$

to get the equation

$$\begin{bmatrix} 8 & -2 \\ 12 & -3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This gives us two equations

$$8a - 2b = 0 \text{ and } 12a - 3b = 0.$$

But these equations provide the same information: they both imply that $b = 4a$. This is because $(\mathbf{A} - \lambda_1 \mathbf{I})$ is not invertible, so the rows are linear dependent. The system of linear equations implied by $(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{v}_1 = \mathbf{0}$ has infinitely many solutions of the form $\begin{bmatrix} a \\ 4a \end{bmatrix}$ for any a. Letting $a = 1$, we get $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$.

If we let $a = 5$, we would have the eigenvector $\begin{bmatrix} 5 \\ 20 \end{bmatrix}$. This hints at an important fact about eigenvectors: *we only care about an eigenvector's direction, not its length.* So we could have chosen $\mathbf{v}_1$ to be any vector pointing the same direction as $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$ (such as $\begin{bmatrix} 5 \\ 20 \end{bmatrix}$ or $\begin{bmatrix} -2 \\ -8 \end{bmatrix}$). We often speak about "the" eigenvector corresponding to an eigenvalue, but only the direction of the eigenvector is unique, not the length.

Exercise 12.4

Using the basic eigenvalue/eigenvector equation

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$$

show that if $\mathbf{v}$ is an eigenvector for λ, then $c\mathbf{v}$ is also an eigenvector for λ, where c is any constant.

Exercise 12.5

We can always check that λ_1 and $\mathbf{v}_1$ are the corresponding eigenvalue and eigenvector for the matrix $\mathbf{A}$ by plugging them into the equation $\mathbf{A}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1$ and verifying that it holds.

Use this procedure to check that $\lambda_1 = 10$ and $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ are the corresponding eigenvalue and eigenvector for $\mathbf{A} = \begin{bmatrix} 18 & -2 \\ 12 & 7 \end{bmatrix}$.

Exercise 12.6

Continuing the example above, with

$$\mathbf{A} = \begin{bmatrix} 18 & -2 \\ 12 & 7 \end{bmatrix}$$

find the eigenvector that corresponds to the eigenvalue $\lambda_2 = 15$.

More Eigen-stuff

Exercise 12.7

Determine the eigenvalues and eigenvectors of the following 2x2 matrices.

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

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

Exercise 12.8

We have two vectors,

$$\mathbf{n} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{12.1}$$

but

$$\mathbf{z} = \begin{bmatrix} -1 \\ 1.01 \end{bmatrix} \tag{12.2}$$

In other words, the vectors $\mathbf{n}$ and $\mathbf{z}$ point in a very similar direction, but are not perfectly aligned. Now consider a matrix $\mathbf{S}$ given by

$$\mathbf{S} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{12.3}$$

1. On the same axes, plot the vectors $\mathbf{n}$ and $\mathbf{z}$ using MATLAB.
2. Suppose that $\mathbf{n}$ and $\mathbf{z}$ are transformed by $\mathbf{S}$. On the same axes as in the previous part, plot the vectors $\mathbf{Sn}$ and $\mathbf{Sz}$ using MATLAB.
3. Now, we shall see what happens to these vectors under repeated transformations by $\mathbf{S}$. On the same axes as in the previous part, plot the vectors $\mathbf{SSn}$ and $\mathbf{SSz}$ using MATLAB.
4. On the same axes as in the previous part, plot the vectors $\mathbf{SSSn}$ and $\mathbf{SSSz}$ using MATLAB.
5. On the same axes as in the previous part, plot the vectors $\mathbf{SSSSn}$ and $\mathbf{SSSSz}$ using MATLAB.
6. You should find that $\mathbf{n}$ is unaffected by the transformation by $\mathbf{S}$, but $\mathbf{z}$ on the other hand moves farther and farther away. In other words, under repeated transformations by $\mathbf{S}$, $\mathbf{z}$ grew further and further apart from his four friends. Explain what you see in terms of eigenvalues and eigenvectors.

12.2 Properties and Applications of Eigenvalues and Eigenvectors

While most of our work on eigenvalues and eigenvectors has focused on 2D vectors and 2×2 matrices, these ideas extend to higher dimensions as well. The eigenvalues and eigenvectors can be found by solving the characteristic polynomial, or by using the MATLAB `eig` function.

A few words about `eig` are in order. The following command

```
>> [V,D] = eig(A)
```

will return two matrices. The columns of the matrix V are the eigenvectors. D is a diagonal matrix, with the eigenvalues on the diagonal. The first eigenvector is in the first column of V and has a corresponding eigenvalue in the first diagonal entry of D. Each eigenvector is normalized to have a magnitude of 1. The eigenvalues will often "appear" to be sorted according to their size, but this is not necessarily true, and is simply an artifact of the algorithm used to compute them. See the documentation in MATLAB for more details.

Consider an $n \times n$ matrix $\mathbf{A}$. The characteristic polynomial will be a polynomial of degree n in λ, i.e., it will have the form

$$c_n\lambda^n + \cdots + c_1\lambda + c_0 = 0$$

where c_i are constants. This polynomial will have n roots, although some of those roots might be the same (e.g., both roots of the polynomial $\lambda^2 + 2\lambda + 1 = 0$ are -1, so we say $\lambda_1 = -1$ and $\lambda_2 = -1$.) This means that an $n \times n$ matrix has n eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, where its possible that some eigenvalues are equal.

The following are properties of the eigenvalues (some of these are n-dimensional extensions of what you already saw for 2 dimensions).

- $\mathrm{Tr}(\mathbf{A}) = \lambda_1 + \lambda_2 + \cdots + \lambda_n$

- $\det(\mathbf{A}) = \lambda_1 \lambda_2 \cdots \lambda_n$

- $\mathbf{A}$ is invertible if and only if all eigenvalues are nonzero.
- If the eigenvalues are distinct (none are equal) then the corresponding eigenvectors are linearly independent.
- If a matrix is symmetric, i.e., $\mathbf{A} = \mathbf{A}^T$, then its eigenvalues are real and its eigenvectors are orthogonal to each other.

Exercise 12.9

In this problem, you will get some practice seeing some of the properties above in action. First, create a 3×3 matrix in MATLAB, with any values you'd like and call it A. Alternatively, you can ask MATLAB to generate a 3×3 matrix with random entries using `A = randn(3, 3);`.

1. Use MATLAB's `eig` function to get the eigenvalues and eigenvectors of the matrix.
2. Using MATLAB's `trace` function, confirm that the trace equals the sum of the eigenvalues.
3. Using MATLAB's `det` function, confirm that the determinant equals the product of the eigenvalues, and explain why a square matrix is invertible if and only if all its eigenvalues are nonzero.
4. Generate a new matrix $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ which must be symmetric. Find its eigenvalues and eigenvectors using `eig`, and verify that the eigenvectors are orthogonal.

12.3 Eigenvalues and Eigenvectors in Data Analysis

In the last class you worked on examples involving correlation matrices. Here we will look at covariance matrices, which are related to correlation matrices, except that the entries are not normalized by the standard deviations of the variables. You can think of covariance matrices as measuring the relationship between random quantities, but without normalization. Thus, information about how small or large these data values are will still be preserved in the covariance matrix.

Suppose that we have two different data variables x and y (e.g. corresponding to temperatures in Boston and Sao Paolo), with x_i and y_i being different values in the data set we can define a a matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} x_1 - \mu_x & y_1 - \mu_y \\ x_2 - \mu_x & y_2 - \mu_y \\ x_3 - \mu_x & y_3 - \mu_y \\ \vdots & \vdots \\ x_N - \mu_x & y_N - \mu_y \end{pmatrix}$$

where μ_x is the mean of the first column, and N is the number of samples (rows). The covariance matrix of x and y is $\mathbf{R} = \mathbf{A}^T\mathbf{A}$. You can think of the entries of this matrix as storing the un-normalized correlations between the temperatures. Because $\mathbf{R}^T = \mathbf{R}$, this matrix is symmetric, and hence has orthogonal eigenvectors.

The eigenvectors and eigenvalues of $\mathbf{R}$ tell us something about how the data are distributed. The eigenvector corresponding to the largest eigenvalue of $\mathbf{R}$, which is also called the *principal eigenvector* of $\mathbf{R}$ points in the direction with the largest variation in the data. The eigenvector corresponding to the second largest eigenvalue points in the direction orthogonal to the principal eigenvector in which there is the second largest amount of variation in the data, and so on (if you have more than 2 dimensional data). The square-root of the eigenvalues tells you about the amount of variation there is in each of those directions. Of course when you only have two different variables in the data set, the matrix $\mathbf{R}$ has only 2 orthogonal eigenvectors.

To illustrate, consider Figure 12.2 which shows the centered (mean subtracted) temperatures of Boston vs Sao Paolo. We have also plotted the two eigenvectors, scaled by the square-root of their corresponding eigenvalues, to illustrate the relative variation of the data along the directions of the two eigenvectors. Notice that the principal eigenvector is in the direction of greatest variation in the data. Figure 13.1 is a similar plot with the temperatures of Boston and Washington DC instead.

The proof of this is optional and will be introduced in the future.

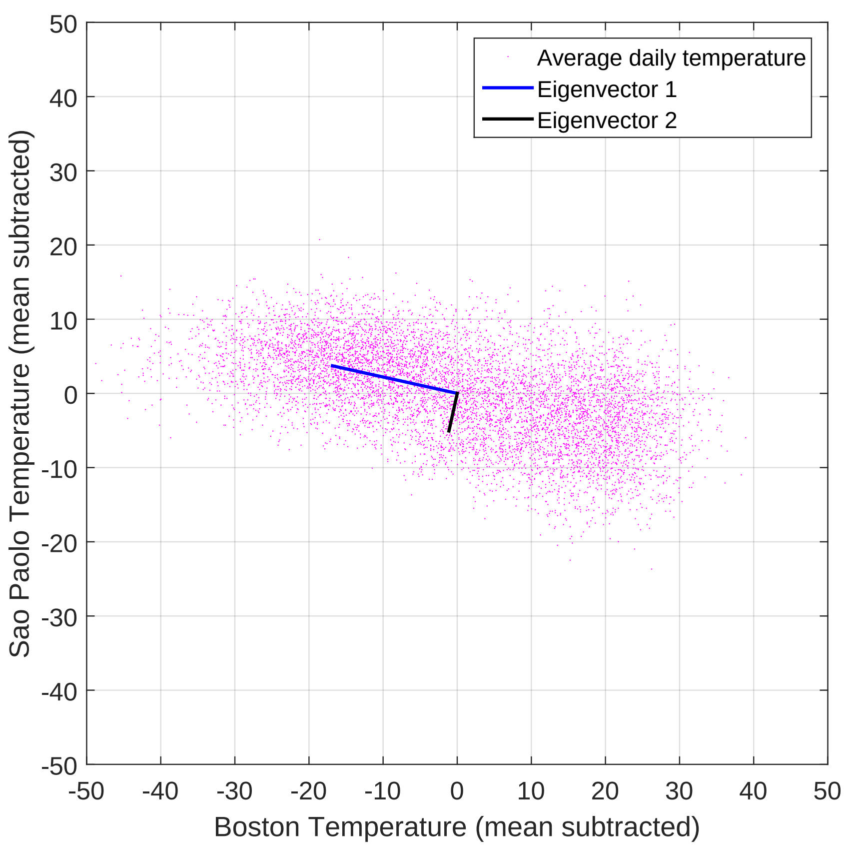


Figure 12.2: Centered average daily temperatures of Boston vs Sao Paolo, with the eigenvectors of the covariance matrix.

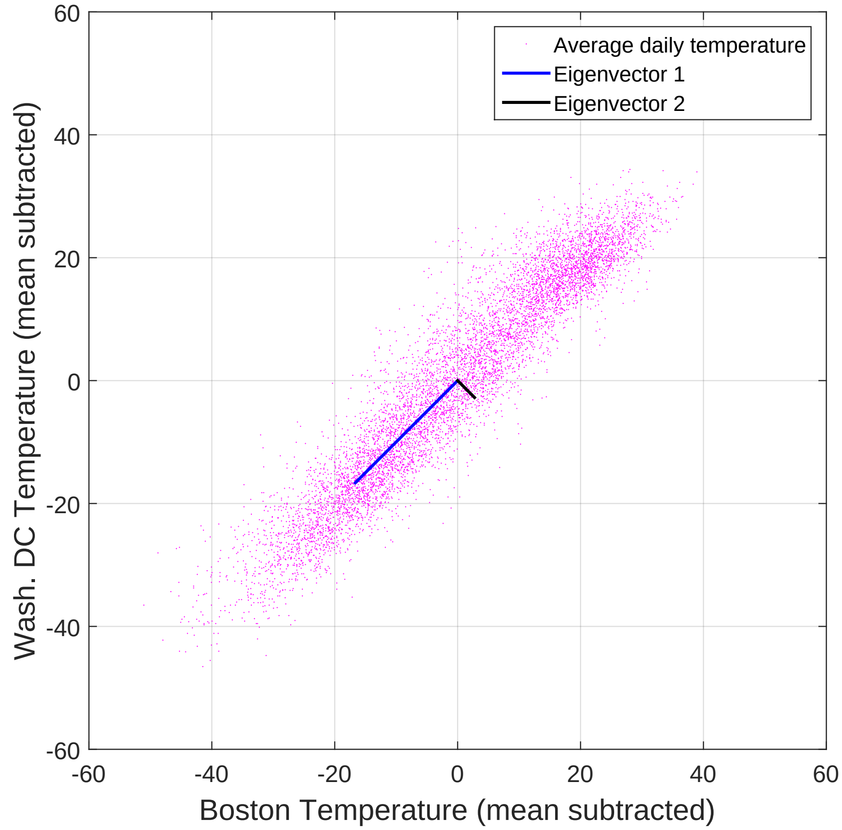


Figure 12.3: Centered average daily temperatures of Boston vs Washington DC, with the eigenvectors of the covariance matrix.

Exercise 12.10

In this next problem, we are going to visualize how the eigenvectors of covariance matrices can tell us about the directions of most variation in 3D data. Load the file `temps_bos_sp_dc.mat` in MATLAB. This file will load 21 years of temperature values for Boston, Sao Paolo and Washington DC. Treat the temperatures of Boston, Sao Paolo, and Washington DC for a given day as a point in a 3D space.

1. Subtract out the mean temperature of each city from the daily temperature data.

2. Make a 3D scatter plot of the data points with the means subtracted out. You will find MATLAB's `plot3` function useful. You may wish to use the `'MarkerSize'` argument

for `plot3` with a marker size of 0.1 or less to make the plots clearer.

3. Construct a covariance matrix for the data and compute its eigenvectors.
4. On the same axes, using `quiver3`, or `plot3`, plot the eigenvectors scaled by the square-root of their corresponding eigenvalues. Use `grid on` to draw grid lines on the axes to improve your visualization.
5. Using the rotate 3D button on the figure window, rotate the image around to see how the eigenvectors tell you about the variation in the data.

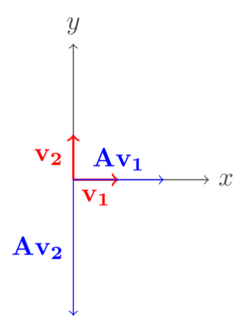
12.4 Diagnostic Quiz

Please see Canvas for the quiz questions.

Solution 12.1

1. First we find
$$\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 2-\lambda & 0 \\ 0 & -3-\lambda \end{bmatrix}.$$
   And then we compute the determinant
$$\det(\mathbf{A} - \lambda\mathbf{I}) = (2-\lambda)(-3-\lambda).$$
   Setting this equal to zero produces the characteristic equation,
$$(2-\lambda)(-3-\lambda) = 0$$
   whose roots are, in fact, $\lambda_1 = 2$ and $\lambda_2 = -3$.

2. Here's a plot of $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{A}\mathbf{v}_1$ and $\mathbf{A}\mathbf{v}_2$:



   When the eigenvalue is negative, the eigenvector is reversed in direction and then scaled.

Solution 12.2

1. Since $\text{tr}(\mathbf{A}) = 2$ and $\det(\mathbf{A}) = -3$, we have
$$\lambda^2 - 2\lambda - 3 = 0$$
   so $\lambda_1 = -1$ and $\lambda_2 = 3$.

2. Since $\text{tr}(\mathbf{A}) = 3$ and $\det(\mathbf{A}) = -4$, we have
$$\lambda^2 - 3\lambda - 4 = 0$$
   so $\lambda_1 = -1$ and $\lambda_2 = 4$.

Solution 12.3

1.

$$\begin{aligned}\lambda_1 + \lambda_2 &= \frac{tr(\mathbf{A}) + \sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}}{2} + \frac{tr(\mathbf{A}) - \sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}}{2} \\ &= \frac{tr(\mathbf{A})}{2} + \frac{tr(\mathbf{A})}{2} = tr(\mathbf{A})\end{aligned}$$

2.

$$\begin{aligned}\lambda_1\lambda_2 &= \left(\frac{tr(\mathbf{A})}{2}\right)^2 + \left(\frac{tr(\mathbf{A})}{2}\right)\left(\sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}\right) \\ &- \left(\frac{tr(\mathbf{A})}{2}\right)\left(\sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}\right) - \left(\frac{\sqrt{tr(\mathbf{A})^2 - 4det(\mathbf{A})}}{2}\right)^2 \\ &= \frac{tr(\mathbf{A})^2}{4} - \frac{tr(\mathbf{A})^2 - 4det(\mathbf{A})}{4} \\ &= det(\mathbf{A})\end{aligned}$$

3. Symmetric means $b = c$ and real λ means:

$$tr(\mathbf{A})^2 - 4det(\mathbf{A}) \geq 0$$

$$(a+d)^2 - 4(ad - bc) \geq 0$$

$$(a+d)^2 - 4(ad-bc) = a^2 + d^2 + 2ad - 4ad + 4bc = a^2 + d^2 - 2ad + 4b^2 = (a-d)^2 + 4b^2$$

Squares of real numbers are positive, so $tr(\mathbf{A})^2 - 4det(\mathbf{A}) \geq 0$ and the eigenvalues λ are real.

Solution 12.4

Using the fact that $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, we see that

$$\mathbf{A}(c\mathbf{v}) = c\mathbf{A}\mathbf{v} = c\lambda\mathbf{v} = \lambda(c\mathbf{v})$$

and therefore $c\mathbf{v}$ is also an eigenvector.

Solution 12.5

First we compute the left-hand side

$$\mathbf{A}\mathbf{v}_1 = \begin{bmatrix} 18 & -2 \\ 12 & 7 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \end{bmatrix}$$

and the right-hand side

$$\lambda_1\mathbf{v}_1 = 10 \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \end{bmatrix}.$$

Fortunately, they are equal.

Solution 12.6

First we compute

$$\mathbf{A} - \lambda_2 \mathbf{I} = \begin{bmatrix} 18-15 & -2 \\ 12 & 7-15 \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ 12 & -8 \end{bmatrix}.$$

Now, letting $\mathbf{v}_2 = \begin{bmatrix} a \\ b \end{bmatrix}$, we are trying to solve

$$\begin{bmatrix} 3 & -2 \\ 12 & -8 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which produces the equations

$$3a - 2b = 0 \text{ and } 12a - 8b = 0.$$

(These equations give the same information since the rows of $(\mathbf{A} - \lambda_2 \mathbf{I})$ are linearly dependent.) This gives $b = \frac{3}{2}a$, so $\mathbf{v}_2 = \begin{bmatrix} a \\ \frac{3}{2}a \end{bmatrix}$ for any value of a. Picking $a = 2$, we have $\mathbf{v}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$.

Solution 12.7

1. $$\lambda = 5, 2$$

   and

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

2. $$\lambda = -1, 4$$

   and

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

Solution 12.8

See Figure 12.1. $\mathbf{n}$ is an eigenvector of $\mathbf{S}$ with an eigenvalue of 1, so it is unchanged by the transformation $\mathbf{S}$. However, $\mathbf{z}$ is not an eigenvector of $\mathbf{S}$, so it changes each time the transformation $\mathbf{S}$ is applied, and the change accelerates as it diverges from the eigenvector $\mathbf{n}$.

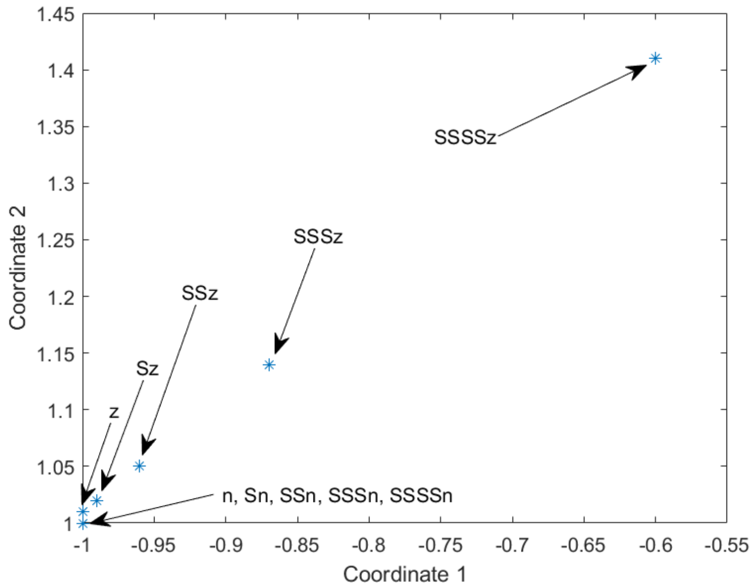


Figure 12.1: Plot for Exercise 9.

Solution 12.9

1. `A=randn(3,3);[V,D]=eig(A)`
2. `trace(A)-sum(diag(D))`
3. `det(A)-prod(diag(D))`

   The product of all eigenvalues equals the determinant of the matrix. If any eigenvalue is zero, the determinant is zero and the matrix is non-invertible; if all eigenvalues are nonzero, the determinant is nonzero and the matrix is invertible.
4. `B=A'*A;[V D]=eig(B);V'*V` gives the identity matrix, showing that the eigenvectors are orthogonal (and of unit length).

Solution 12.10

1. `bn=b-mean(b);sn=s-mean(s);wn=w-mean(w);`
2. ```
   plot3(bn,sn,wn,'.','MarkerSize',0.1)
   xlabel('Boston temperature (mean subtracted)')
   ylabel('Sao Paolo temperature (mean subtracted)')
   zlabel('Wash. D.C. temperature (mean subtracted)'
   ```
   ```

3.
```
A=1/sqrt(length(b)-1)*[bn,sn,wn];
R=A'*A;
[V,D]=eig(R)
```

4.
```
plot3(bn,sn,wn,'.','MarkerSize',0.1)
Vs=V.*sqrt(diag(D))
hold on
plot3([0,Vs(1,1)],[0 Vs(2,1)],[0 Vs(3,1)],'LineWidth',2)
plot3([0,Vs(1,2)],[0 Vs(2,2)],[0 Vs(3,2)],'LineWidth',2)
plot3([0,Vs(1,3)],[0 Vs(2,3)],[0 Vs(3,3)],'LineWidth',2)
grid on
axis equal
```

   See Figure 14.1, which has the first eigenvector clearly aligned with the direction of greatest variation.

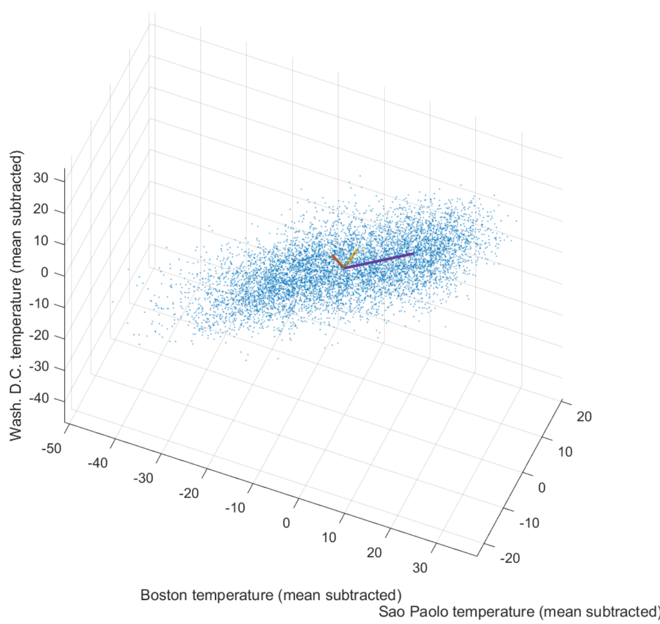


Figure 12.4: Temperatures and eigenvectors.

Chapter 13

Day 7: EVD and PCA

13.1 Schedule

- 0900-0930: Debrief
- 0930-1015: Eigenvalue Decomposition (EVD)
- 1015-1030: Coffee
- 1030-1115: PCA and Maximum Variance
- 1115-1210: Conceptual PCA and PCA blog post
- 1210-1225: Review and Preview
- 1225-1230: Survey

13.2 Debrief [15 mins]

In the last class and in the take-home exercise, you worked on a number of different exercises involving eigenvalues and eigenvectors.

Exercise 13.1

1. With your table, identify a list of key concepts/take home messages/things you learned in the last class and take-home assignment.
2. Try to resolve your confusions with the folks at your table and by talking to an instructor.

13.3 Eigenvalue Decomposition (EVD) [45 mins]

The eigenvalue decomposition, also known as the eigendecomposition, is an operation on matrices in which a square matrix is expressed as a product of matrices made up of its eigenvalues and eigenvectors. It can be used to find inverses and powers of matrices, as well as to derive some important results in data analysis. For instance, in a prior exercise, you saw that the eigenvector corresponding to the largest eigenvalue of a covariance matrix was in the direction of greatest variance in your data set. This property can be proved using the eigendecomposition.

The eigenvalue decomposition is also helpful in dimensionality reduction, which is a process where we can represent higher-dimensional vectors as a linear combination of a smaller number of vectors than dimensions – an example of which you saw in a previous exercise where you represented pictures of peoples faces using a linear combination of vectors. The eigendecomposition is also often used to change coordinate systems.

The Big Idea

Assume that a square $n \times n$ matrix $\mathbf{A}$ has n linearly independent eigenvectors $\mathbf{v}_i$ with corresponding eigenvalues λ_i, i.e.

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \; i = 1, 2, \ldots, n$$

Instead of thinking of these eigenvalues and eigenvector separately, let's package them into matrices as follows:

$$[\mathbf{A}\mathbf{v}_1 \; \mathbf{A}\mathbf{v}_2 \; \ldots \; \mathbf{A}\mathbf{v}_n] = [\lambda_1 \mathbf{v}_1 \; \lambda_2 \mathbf{v}_2 \; \ldots \; \lambda_n \mathbf{v}_n]$$

Properties of matrix multiplication suggests that we can re-write this matrix equation in the form

$$\mathbf{A}[\mathbf{v}_1 \; \mathbf{v}_2 \; \ldots \; \mathbf{v}_n] = [\mathbf{v}_1 \; \mathbf{v}_2 \; \ldots \; \mathbf{v}_n] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}$$

where the last matrix has each eigenvalue on the diagonal. If we now define

$$\begin{aligned} \mathbf{V} &= [\mathbf{v}_1 \; \mathbf{v}_2 \; \ldots \; \mathbf{v}_n] \\ \mathbf{D} &= \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \end{aligned}$$

then the previous equation becomes

$$\mathbf{AV} = \mathbf{VD}$$

Since we assumed that the eigenvectors are linearly independent this implies that the columns of $\mathbf{V}$ are linearly independent which in turn implies that the inverse of $\mathbf{V}$ exists. We can there fore write

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \tag{13.1}$$

where the matrix $\mathbf{V}$ has the i-th eigenvector of $\mathbf{A}$ as its i-th column, and $\mathbf{D}$ is a diagonal matrix with the i-th eigenvalue of $\mathbf{A}$ as its ii-th entry. This expression is known as the *eigendecomposition* of $\mathbf{A}$. In the special case where $\mathbf{A}$ is symmetric, the eigenvalues are real, and the eigenvectors are mutually orthogonal so that

$$\mathbf{V}^{-1} = \mathbf{V}^T,$$

which is a property of $n \times n$ matrices whose column vectors are mutually orthogonal and have a length of 1 (i.e., the column vectors are orthonormal).

Exercise 13.2

1. Consider the following 2×2 matrix $\mathbf{A}$.

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

By hand, compute its eigenvectors and eigenvalues, determine the matrices $\mathbf{V}$, $\mathbf{D}$, and $\mathbf{V}^{-1}$, and confirm that (13.1) is correct. Use MATLAB to confirm your results by computing » [V,D]=eig(A). **Note**: you should normalize each of your eigenvectors to be unit length.

Exercise 13.3

1. The eigendecomposition can be used to change basis as follows. Consider the matrix $\mathbf{A}$ from the previous exercise as a transformation matrix.

   a) How does the matrix $\mathbf{A}$ transform the vector $\mathbf{w} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$? Draw both $\mathbf{w}$ and $\mathbf{Aw}$ on an xy-coordinate plane.

   b) Draw both eigenvectors of $\mathbf{A}$ on this coordinate plane.

   c) Decompose the vector $\mathbf{w}$ as a linear combination of both eigenvectors. You should be able to do this with a matrix-vector multiply. You are expressing the vector in a new basis.

   d) Scale each component by the relevant eigenvalue.

   e) Undo the decomposition to return to the original basis.

   f) What just happened?

Exercise 13.4

One thing that the eigendecomposition helps us compute is how to raise $\mathbf{A}$ to an integer power, without going through the process of repeated multiplication.

1. Using eigendecomposition, show the following is true

$$\mathbf{A}^2 = \mathbf{V}\mathbf{D}^2\mathbf{V}^{-1} \tag{13.2}$$

   and confirm this result using the matrix from earlier the earlier exercise. Note that for any diagonal matrix $\mathbf{D}$, $\mathbf{D}^k$ is another diagonal matrix whose ii-th entry equals the ii-th entry of $\mathbf{D}$ raised to the k-th power. Hence computing $\mathbf{D}^n$ is not computationally difficult - you just raise each diagonal entry to the n-th power.

2. Show that the following is also true

$$\mathbf{A}^n = \mathbf{V}\mathbf{D}^n\mathbf{V}^{-1}$$

13.4 Principal Components Analysis (PCA)

In the night assignment you explored, in a graphical manner, the relationship between the eigenvectors of the covariance matrix and the distribution of the data. For instance, you looked at the daily temperature values in Boston versus Sao Paolo and the daily temperatures in Boston versus Washington D.C.

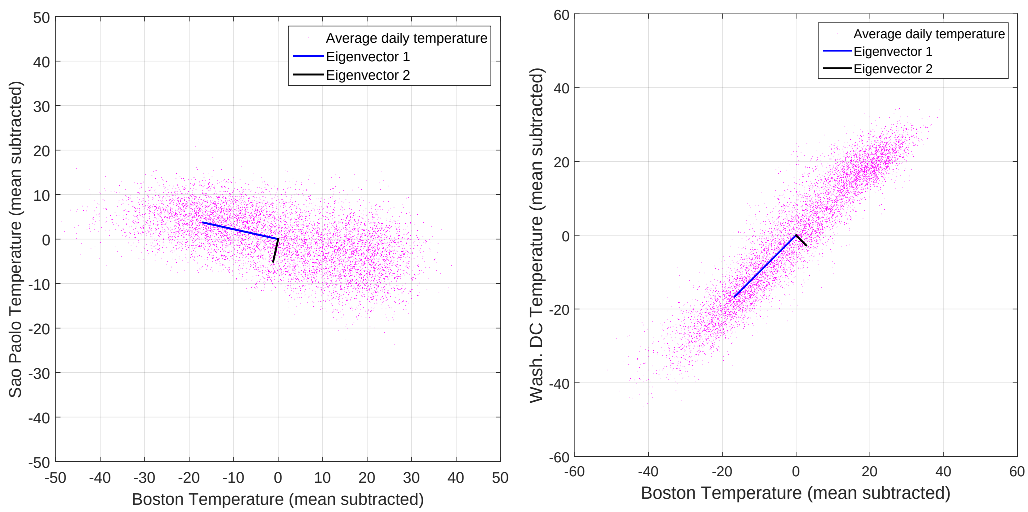


Figure 13.1: Centered average daily temperatures of Boston vs Sao Paolo (left) and Boston vs Washington DC, with the eigenvectors of the covariance matrix.

From visually inspecting these figures we saw that eigenvector 1, which corresponded to the larger of the two eigenvalues, seemed to be pointing in the direction where the data exhibited the most variability (i.e., the data was most spread out along this direction). You also looked at this for a 3D dataset consisting of the temperatures from Boston, Sao Paolo, and Washington DC.

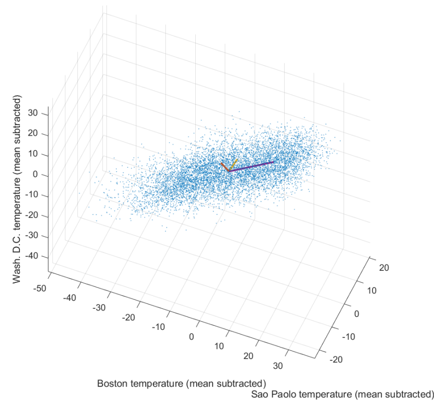


Figure 13.2: Temperatures and eigenvectors for Boston, Sao Paolo, and Washington DC

In this 3D dataset, we see the same phenomenon: that the principal eigenvector points along the direction of maximum variation in the data. It turns out that this phenomenon will hold no matter the dimensionality of the data (it works for 4D datasets, 10D datasets, and even datasets with 1,000s of dimensions)! This fact provides the basis for the principal components algorithm. In PCA, instead of working with the data in its original form, we express it in a basis given by eigenvectors of the covariance matrix that have the largest eigenvalues. We can understand the properties of using this basis through two key properties.

- *Property 1*: the principal eigenvectors of the covariance matrix will maximize the variance of the data when the data is projected onto these vectors (we can think of vectors that capture large variation in the data as representing important properties of the data).
- *Property 2*: the principal eigenvectors of the covariance matrix will allow us, in a particular sense, to optimally compress our data. That is, we will be able to recover the original data with the highest possible accuracy from the projections of the data onto the principal eigenvectors.

The power of PCA lies in its ability to achieve both of the properties described above simultaneously. For this reason, the principal components of a dataset will act as keys to unlocking the secrets lurking in the data! **Today we will be exploring property 1, and in the night assignment you will also be exploring property 2**.

The Principal Eigenvector as the Direction of Maximum Variance

The graphs of the daily temperature data show, graphically, that the principal eigenvector of the covariance matrix corresponds to the direction of maximum variation in the data. In this section we'll be formalizing this result. We've decided to structure this part of the day assignment as an extended exercise where you will be working through the proof of this fact step-by-step. While there are many ways to do this proof, we'll be walking you through one way that will connect well with the ideas we've been exploring in the last week or so of the course. We recommend that you do a part of the proof, check it against the solutions and then move onto the next piece.

Before getting started, let's look at some material from night 6 that shows that the covariance matrix can be computed using matrix multiplication.

Suppose that we have two different data variables x and y (e.g. corresponding to temperatures in Boston and Sao Paolo), with x_i and y_i being different values in the data set we can define a a matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} x_1 - \mu_x & y_1 - \mu_y \\ x_2 - \mu_x & y_2 - \mu_y \\ x_3 - \mu_x & y_3 - \mu_y \\ \vdots & \vdots \\ x_N - \mu_x & y_N - \mu_y \end{pmatrix} \tag{13.3}$$

where μ_x is the mean of the first column, and N is the number of samples (rows). The covariance matrix of x and y is $\mathbf{R} = \mathbf{A}^T\mathbf{A}$. You can think of the entries of this matrix as storing the un-normalized correlations between the temperatures. Because $\mathbf{R}^T = \mathbf{R}$, this matrix is symmetric, and hence has orthogonal eigenvectors.

Let's assume that we are given a dataset with n samples and d dimensions (instead of just 2 dimensions as shown above). We can transform it into the form given in Equation 41.12 by subtracting the mean from each column and dividing the entire matrix by $\sqrt{N-1}$. We now have a mean-centered data matrix $\mathbf{A}$ with n rows and d columns and the covariance matrix of our data is given by $\mathbf{A}^\top\mathbf{A}$.

Exercise 13.5

Our overall goal is to show that if we take a unit vector $\mathbf{u}$, project our mean-centered data onto it (as $\mathbf{Au}$), and examine the variance of the projected data, that this variance is largest when $\mathbf{u}$ is the principal eigenvector of the covariance matrix $\mathbf{A}^\top\mathbf{A}$.

1. First we'll write down an expression for the variance of $\mathbf{Au}$ (we'll write this as $Var[\mathbf{Au}]$) as a matrix multiplication. We'll do this step together (i.e., we'll show you how to do it). For this part of the exercise you should make sure you understand the steps we performed.

   If $\mathbf{A}$ is in the form given in Equation 41.12, then $\mathbf{Au}$ will have 0 mean (since $\mathbf{Au}$ is a linear combination of columns with 0 mean). Using the same logic that led us to conclude that $\mathbf{A}^\top\mathbf{A}$ is the covariance matrix of the data, $(\mathbf{Au})^\top(\mathbf{Au})$ will give us the variance of the data projected onto $\mathbf{u}$ (remember that variance is just a special case of covariance where we are comparing a quantity to itself). It's worth noting that since $\mathbf{Au}$ is a vector, the expression $(\mathbf{Au})^\top(\mathbf{Au})$ is known as the inner product, which is really the same as the dot product (that is, $(\mathbf{Au})^\top(\mathbf{Au}) = \mathbf{Au} \cdot \mathbf{Au}$). Thus, the variance is given by the following equation.

   $$\begin{aligned} Var[\mathbf{Au}] &= (\mathbf{Au})^\top(\mathbf{Au}) \\ &= \mathbf{u}^\top\mathbf{A}^\top\mathbf{Au} \quad \text{note: we are applying the rule that } (\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top \end{aligned}$$

2. Substitute the eigenvalue decomposition, $\mathbf{VDV}^\top$, for the covariance matrix $\mathbf{A}^\top\mathbf{A}$ (since $\mathbf{A}^\top\mathbf{A}$ is symmetric and real, we can substitute $\mathbf{V}^\top$ for the inverse of $\mathbf{V}$ in the eigenvalue decomposition).

3. Define the vector $\mathbf{y} = \mathbf{V}^\top\mathbf{u}$ and substitute it into the expression from part 2.

4. Expand out the expression in part 3 so that it is in terms of the squares of the elements of $\mathbf{y}$ and the diagonal entries of $\mathbf{D}$ in order of largest to smallest.

5. Show that $\mathbf{y}$ is a unit vector by taking the inner product with itself and showing that it is equal to 1 (recall that the inner product is the same as the dot product). *Hint:* $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$ *since* $\mathbf{V}$ *is orthonormal and has* d *linearly independent columns.*

6. Argue that since $\mathbf{y}$ is a unit vector (which implies $\sum_{i=1}^{d} y_i^2 = 1$), that the expression in part 4 is maximized when $y_i = 1$ when i is the index of the principal eigenvector and $y_i = 0$ when i is any other index. To get a feel for why this is true, try writing out a specific case where, perhaps, $\mathbf{y}$ has two or three dimensions.

7. Show that we achieve the value of $\mathbf{y}$ in part 5 (that is where $y_i = 1$ when i is the index of the principal eigenvector and $y_i = 0$ when i is any other index) when $\mathbf{u}$ is the principal eigenvector of $\mathbf{A}^\top\mathbf{A}$.

8. What have you just shown?!? Make sure you have a sense of what you just did (don't get lost in the mathematical symbols).

Beyond the first principal component

We've now gone into depth in understanding the first principal component and its amazing property of maximizing variance. The second principal component is simply going to be the direction that maximizes variance subject to the requirement that it is orthogonal to the first principal component. With a slight modification to your proof you can show that the second principal component will be in the direction of the eigenvector with the second largest eigenvalue. The trend continues for other principal components (i.e., the ith principal component is the eigenvector with the ith largest eigenvalue).

Applications of PCA (thinking it through conceptually)

In this section you're going to be thinking about what the PCA algorithm might do when applied in different domains. The focus of this section will be on trying to understand at a conceptual level what might happen when we apply PCA. In the next section, you'll be reading through an example of applying PCA to some actual data.

Exercise 13.6

For each application, hypothesize what the first principal component might be. That is, for each particular scenario what would the direction be that maximizes the variance of the data projected onto that direction? What might the second principal component be (that is a vector orthogonal to the first that maximizes the variance of the data)?

1. Consider a dataset consisting of ratings from n users of m movies. Let's assume that the ratings are numerical and are on a scale of 1 to 5 (5 being the best). Consider some collection of movies (they could be some specific movies or you could just think of movie genres) and a particular population of users (could be college students, QEA professors, or just the general population). Draw the data matrix $\mathbf{A}$ and label the rows and columns (e.g., with movies or users). In a qualitative sense, make a prediction as to what the first principal component would look like for this dataset. What might the second principal component look like? No numbers... just guess at which dimensions would be positive, negative, or close to 0 for your principal components.

2. Consider a dataset consisting of the prevalence of the flu in various parts of the US. The CDC maintains an animated map of the flu activity over time, which you can (and should) access at `https://www.cdc.gov/flu/weekly/usmap.htm`. To simplify this data, let's think about the number of flu cases in each of the six major major geographical regions of the US.

If we think about our data matrix as consisting of a row for each week of measured flu activity and each column as a region of the US, in a qualitative sense, make a prediction as to what the first principal component would look like for this dataset. What might the second principal component look like? No numbers... just guess at which dimensions would be positive, negative, or close to 0 for your principal components.

Exercise 13.7

With your table-mates, read through this post that shows the application of PCA to understanding the US political leanings (if you are viewing this in DropBox preview and can't click the link, go to `http://bit.ly/37n9qwe`). Before, starting here are some process suggestions.

- Checkin with folks at your table as to how they'd like to go through this document (e.g., read the entire thing individually and come together and ask questions, read it individually but stop after each major section to ask questions, read it aloud as a table).

- If you don't understand something, you can either call over an instructor or note your confusion on the whiteboard and keep going (e.g., if its something that doesn't impede your understanding of the main points in the article).

Solution 13.2

1. The eigenvalues are $\lambda_1 = 1$ and $\lambda_2 = 3$ with corresponding eigenvectors $\mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$ and $\mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$. This gives

$$\mathbf{V} = \frac{1}{\sqrt{2}}\begin{bmatrix}-1 & 1\\1 & 1\end{bmatrix},\ \mathbf{D} = \begin{bmatrix}1 & 0\\0 & 3\end{bmatrix},\ \mathbf{V}^{-1} = \sqrt{2}\begin{bmatrix}-1/2 & 1/2\\1/2 & 1/2\end{bmatrix}.$$

and if you multiply them all together you will get the original matrix $\mathbf{A}$. Running "eig" in MATLAB gives the same eigenvalues and eigenvectors, although every eigenvector could be multiplied by -1. MATLAB may also place your eigenvalues and eigenvectors in a different order.

Solution 13.3

1. The vector becomes $\begin{bmatrix}5\\4\end{bmatrix}$.
2. The eigenvectors were $\mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$ and $\mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$.
3. Decomposing the vector $\mathbf{w}$ as linear combination of the eigenvectors is equivalent to solving

$$\mathbf{Vc} = \mathbf{w}$$

for the vector $\mathbf{c}$. This is the coordinates of the vector $\mathbf{w}$ in the new basis. You should find that $\mathbf{c} = \sqrt{2}\begin{bmatrix}-0.5\\1.5\end{bmatrix}$.

4. We multiply the first component by 1 and the second component by 3 to give $\sqrt{2}\begin{bmatrix}-0.5\\4.5\end{bmatrix}$.
5. In order to undo the change of basis we hit this vector with $\mathbf{V}$ which gives $\begin{bmatrix}5\\4\end{bmatrix}$ as expected.
6. The eigendecomposition can be be thought of as a change of basis followed by a scaling matrix followed by the change back to the original basis.

Solution 13.4

1. Since $\mathbf{A} = \mathbf{VDV}^{-1}$, we know that

$$\mathbf{A}^2 = \mathbf{VDV}^{-1}\mathbf{VDV}^{-1} = \mathbf{VD}^2\mathbf{V}^{-1}.$$

2. Similar reasoning to the previous problem shows that

$$\mathbf{A}^n = \mathbf{VD}^n\mathbf{V}^{-1}$$

Solution 13.5

1. Solution is already given in the problem

2. $$Var[\mathbf{Au}] = \mathbf{u}^\top \mathbf{V}\mathbf{D}\mathbf{V}^\top \mathbf{u}$$

3. $$\begin{aligned} Var[\mathbf{Au}] &= (\mathbf{V}^\top \mathbf{u})^\top \mathbf{D}(\mathbf{V}^\top \mathbf{u}) \\ &= \mathbf{y}^\top \mathbf{D}\mathbf{y} \end{aligned}$$

4. $$\begin{aligned} Var[\mathbf{Au}] &= \mathbf{y}^\top \mathbf{D}\mathbf{y} \\ &= \mathbf{y}^\top \begin{bmatrix} y_1 D_{1,1} \\ y_2 D_{2,2} \\ \vdots \\ y_d D_{d,d} \end{bmatrix} \\ &= \sum_{i=1}^{d} y_i^2 D_{i,i} \end{aligned}$$

5. $$\begin{aligned} \mathbf{y}^\top \mathbf{y} &= (\mathbf{V}^\top \mathbf{u})^\top (\mathbf{V}^\top \mathbf{u}) \\ &= \mathbf{u}^\top \mathbf{V}\mathbf{V}^\top \mathbf{u} \\ &= \mathbf{u}^\top \mathbf{u} \\ &= 1 \end{aligned}$$

6. If we choose $y_i = 1$ where i is the index of the principal eigenvector, then the expression in part 4 will give us $D_{i,i}$. Any other choice of $\mathbf{y}$ will result in some weighted combination of the eigenvalues (the diagonal elements of $\mathbf{D}$) where the weights are all positive and add up to 1. It is easy to see that putting any weight on a non-maximal eigenvalue will result in a lower variance as computed by the expression in part 4.

7. Since $\mathbf{y} = \mathbf{V}^\top \mathbf{u}$, y_i is the dot product of $\mathbf{u}$ and the ith eigenvector, $\mathbf{v_i}$, with $\mathbf{u}$. Since we assume all of the eigenvectors are unit vectors and mutually orthogonal, if we set $\mathbf{u}$ to be the principal eigenvector of $\mathbf{A}^\top \mathbf{A}$, then the dot product of $\mathbf{u}$ and $\mathbf{v_i}$ will be 1 for i corresponding to the principal eigenvector and 0 for all other indices.

8. You just showed that the direction along the principal eigenvector of the covariance matrix maximizes the variance of the projected data. That's pretty cool!

Chapter 14

Night 7: PCA and Eigenfaces

Learning Objectives

Concepts

- Understand the connection between PCA and eigenvalues and eigenvectors.
- Understand how to use PCA to carry out data compression.
- Understand the idea of using Eigenfaces to do facial recognition.

MATLAB skills

- Use "eig" to carry out a PCA.
- Implement facial recognition using PCA.
- Determine the accuracy of PCA for different numbers of principal components.

14.1 Principal Component Analysis Revisited

As we in class, PCA is an algorithm in which we express our original data along the eigenvectors corresponding to the largest eigenvalues of the covariance matrix. We examined the property of PCA that it if we project our data onto these vectors, this will lead to maximizing the variance of the projected data. To refresh your memory further, here is the temperature plot for Boston, Sao Paolo, and Washington DC.

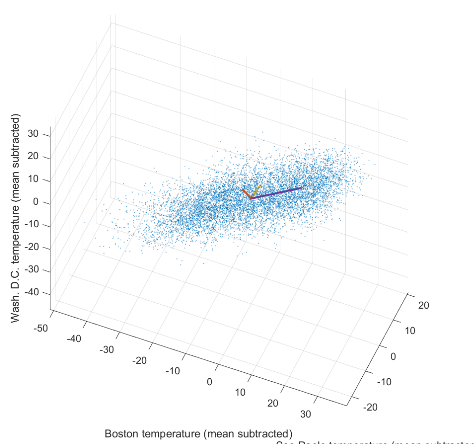


Figure 14.1: Temperatures in three cities and the eigenvectors of the covariance matrix.

We also briefly mentioned a second property of PCA, which is that it can be thought of as an optimal way to compress our data down to a smaller set of numbers. This idea, also known as dimensionality reduction, is going to be a view that we explore in this assignment. If you'd like, here are a few external resources on PCA:

- http://www.cs.otago.ac.nz/
- http://dai.fmph.uniba.sk/courses/ml/sl/PCA.pdf
- https://deeplearning4j.org/eigenvector/linear
- http://www.cerebralmastication.com/2010/09/principal-component-analysis-pca-vs-ordinary-least-squares-ols-a-visual-explination/
- http://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/

PCA in two dimensions

In general, PCA is conducted on data that is mean-centered (i.e., the data has had the mean of each variable subtracted out). To refresh your memory of PCA and scaffold the introduction of the view of PCA as compressing a dataset, let's think about a simple example data set $\mathbf{D}$.

$$\mathbf{D} = \begin{bmatrix} -1 & 3 \\ 1 & 4 \\ 3 & 4 \\ 7 & 5 \\ 10 & 9 \end{bmatrix} \tag{14.1}$$

Exercise 14.1

1. Create a plot of $\mathbf{D}$ as a set of points in the xy-plane.
2. Define a matrix $\tilde{\mathbf{D}}$ which is the mean-centered version of $\mathbf{D}$ and plot $\tilde{\mathbf{D}}$ as a set of points in the xy-plane
3. The principal components ($\mathbf{p_1}$ and $\mathbf{p_2}$) are the eigenvectors of the covariance matrix of the mean-centered $\tilde{\mathbf{D}}$. Compute $\mathbf{p}_1$ and $\mathbf{p}_2$ and plot them on top of the mean-centered data.
4. Compute the projection of your data onto the eigenvector which corresponds to the largest eigenvector, which in this case is $\mathbf{p}_2$. This is the "reduced dimenstionality" version of your data, called $\mathbf{B}$, which only include information about the projection along $\mathbf{p}_2$. (We reduced the 2-dimensional data to 1-dimensional data.) Plot the original data $\mathbf{D}$ and the reduced data $\mathbf{B}$.

Exercise 14.2

1. Can you recreate $\mathbf{D}$ perfectly from $\mathbf{B}$?
2. What would have happened if you had created $\mathbf{B}$ using only information about the values along $\mathbf{p_1}$ instead of $\mathbf{p_2}$?
3. How might you quantify how well you can represent $\mathbf{D}$ in this reduced dimensionality form?
4. If you received a new piece of data, how would you go about representing this as a linear combination of $\mathbf{p_1}$ and $\mathbf{p_2}$?

Data Compression via PCA

In this exercise you will perform a simple data compression exercise, similar to the one you did in a previous Night assignment. You will use temperature data from 3 cities over 10 years, as training data and use it to compress a year's worth of temperature data from 3 cities into a 2×365 matrix. In other words, you will represent 3×365 numbers (daily temperature data from 3 cities over 1 year), using 2×365 values. This compression is lossy, in that you will loose some information. However, by representing the data along the two most significant eigenvectors of the covariance matrix, you can reduce this data loss, because these two directions capture the bulk of the variation in the data set. Please note that while we have laid out the steps you need to take here quite explicitly, it is important for you to fully understand what each step does. You will be using very similar steps in your project.

Exercises

Exercise 14.3

1. Load the file `avg_temperatures_pt2.mat`. You will have 6 data vectors in your workspace. `b_tr, w_tr, s_tr` which represent 10 years of training data for the average daily temperatures in Boston, Washington DC, and Sao Paolo, respectively. The vectors `b_new, w_new, s_new` represent an additional year of data for the three cities – this is the data that you will compress using statistical knowledge of the previous 10 years of data. Create a covariance matrix $\mathbf{R}$ using the 10 years worth of temperature data from Boston, Washington DC and Sao Paolo (in that order).
2. Perform an eigendecomposition of the matrix $\mathbf{R}$, and make a new matrix $\mathbf{V}_p$ which has the 2 eigenvectors corresponding to the 2 largest eigenvalues of $\mathbf{R}$. You should use MATLAB's

`eig` function. Let these eigenvectors be $\mathbf{v}_1$ and $\mathbf{v}_2$.

3. Create centered (i.e. subtract the mean), versions of the new temperature data vectors, and create a 3×365 matrix $\mathbf{T}$ which has the centered temperatures of Boston, Washington DC and Sao Paolo as its rows (in that order). This matrix is a representation of the data you are now going to compress. Let the i-th column of $\mathbf{T}$ be $\mathbf{t}_i$.

4. Take the dot product of each column of the matrix $\mathbf{T}$ (which is a vector of the temperature of Boston, Washington DC and Sao Paolo for a given day) with the two eigenvectors in matrix $\mathbf{V}_p$, and save the values. Let this quantities be called α_{1i} and α_{2i}. In other words,

$$\alpha_{1i} = \mathbf{v}_1^T \mathbf{t}_i$$
$$\alpha_{2i} = \mathbf{v}_2^T \mathbf{t}_i$$

   You can do this using matrix multiplications.

   You should now have 365 different values for α_{1i} and α_{2i}, which are a compressed representation of 3×365 different temperature values. Moreover, these values are the components of the temperature data that lie in the directions of the two eigenvectors of the covariance matrix corresponding to the largest eigenvalues. From what we saw in the previous two classes, these vectors represent the two orthogonal directions in the data that have the most amount of variation, and hence the most "important" directions. Of course, there is a third direction (since the temperature vectors live in a 3-dimensional space), which we are discarding. But since this is the direction in which there is the least amount of variation in the data set, we do not lose too much information.

5. You can now check how well your compression worked, by using the values of α_{1i} and α_{2i} to reconstruct 365 different 3×1 vectors each representing the temperatures for the three cities over the 365 days. Let $\hat{\mathbf{t}}_i$ represent the reconstructed temperature vector on the i-th day. Using what you know about projections onto orthonormal vectors, reconstruct $\mathbf{t}_i$ using α_{1i}, α_{2i}, $\mathbf{v}_1$ and $\mathbf{v}_2$. Repeat this for all 365 days.

6. On the same axes, plot the original and reconstructed temperature for Boston. Repeat this for Washington DC and Sao Paolo. Observe how close the reconstructions are, for the different data sets.

7. How accurately do you think you can represent the data if you used 3 eigenvectors instead of 2?

8. If you feel inspired, repeat the above with temperature data for four different cities, and 2 or 3 different eigenvectors.

While this example can be thought of as a "toy" example where we are representing 3 dimensional data using 2 dimensions, there are many applications for which there may be many more dimensions in the data for which accurate representations can be made using only a few dimensions. Additionally, you should note that such dimensionality reduction techniques are not just useful in compression, but they are also useful in speeding up computation. We can often get away with analyzing data over a small number of important dimensions, and this is an important technique when we deal with large amounts of data. Overall, these class of techniques is called Principal Component Analysis (PCA), since we are performing analysis along a few principal component directions of the data.

14.2 Face Data Compression via PCA

You are now ready to start applying PCA to face data. You have already seen this in a previous class assignment, except in that assignment you had the help of a genie. Load the MATLAB files `classdata_train.mat` and `classdata_test.mat`. These are the training and test datasets with photos of your classmates. The file contains some images and as well as the identity of the person in each image (coded as an integer from 1 to 89).

Remember that the principal eigenvectors of the covariance matrix tell you the directions of greatest variation in a data set and also the directions that optimally compress our data. Your job is to use the training images to build a model for your faces such that you can compress a many-pixeled test face image (pick one from the test image array) using a small set of image vectors (e.g., 10, 20, or 50).

Before you dig into this problem, think through how you would formalize the face data compression as a problem that you can solve with the linear algebra techniques that you've learned so far. There is no exercise to answer, but we want you to think through these steps before going further in the assignment.

- How you will choose your set of image vectors?
- Come up with the steps needed to do the above and write some pseudo code (e.g., load the data, vectorize the images, etc.).
- How could you tell whether your compression algorithm works (these could either be quantitative metrics, like root-mean squared error or qualitative metrics).

14.3 Eigenfaces for Face Recognition

It's time to bring it all together and finally plunge into the prosopagnosia (look it up) problem (or at least build some facial recognition software, but alliteration is fun). You will implement the eigenfaces algorithm to identify photos of your classmates. While it sounds fancy, you have almost all of the pieces needed to understand and implement Eigenfaces (the last necessary piece you will pick up momentarily). Here are the major steps in the Eigenfaces algorithm.

1. Use PCA to compute the k principal components of the training face images (the k eigenvectors with largest eigenvalues).
2. Project the training and test face images onto the k principal components. We'll call this the *facespace* representation of our original images.
3. For each of the test images, compute the closest match between the test image (represented as a k-dimensional vector in facespace) and the training images (again, in facespace). The notion of "closest match" here can be described in a few different ways, but the easiest thing to do is to use

the the Euclidean distance to define how far apart two points are. In this way, you would look for the training point that has the smallest Euclidean distance for a particular test point and predict the identity of the test point to be the same as the identity of this closest training point. This method of classification is known as nearest neighbor classification and it is the one new concept you need to implement Eigenfaces.

Exercise 14.4

Earlier in this assignment you wrote some pseudo code for face compression, which as you can see from the description of Eigenfaces above, gets you most of the way there. Before you actually implement Eigenfaces, we'd like you to extend your pseudocode to cover the whole Eigenfaces algorithm. In addition to the steps of Eigenfaces describe above, you should also think about the steps needed to calculate the accuracy of your system (i.e., how often does it get the person's identity correct).

Exercise 14.5

Implement the eigenfaces algorithm.

1. Your code, which can be a script, function, or livescript, should use the training and test sets of images provided: `classdata_train.mat` and `classdata_test.mat`, respectively.

2. You may want to start by identifying one face from the test set, but by the time you are done, your code should run through all of the test images and report the fraction it guesses correctly.

3. Test different numbers of eigenvectors. How many does it take to guess right most of the time?

4. If you'd like, time how long it takes your code to run. Can you do anything more efficiently to make it run faster? (use `tic` and `toc` in MATLAB for timing)

5. Visualize the first few eigenfaces. Can you interpret what they mean?

6. Generate a figure that depicts the success rate (accuracy at determining the identify of a person in an image) versus the number of eigenfaces used.

7. Generate a figure that depicts the success rate (accuracy at determining the identify of a person in an image) versus the number of eigenfaces used where the training data consists of only images of people not smiling and the test data consists of only images of people smiling (these are located in `classdata_non_smiles.mat` and `classdata_smiles.mat` respectively. Comment on the difference in performance on when using these files versus the data from the previous parts of this problem.

Guidelines:

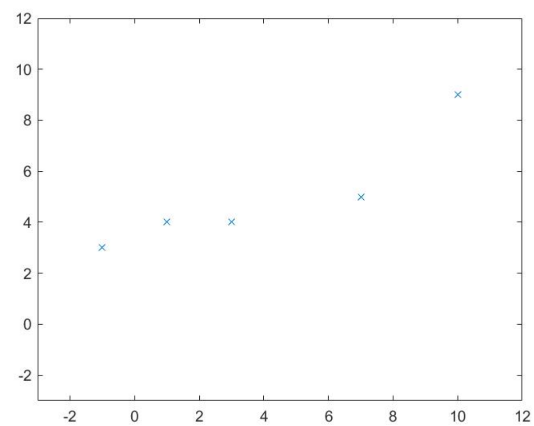
- You should comment your code (use %) so others could read and understand it.
- Don't use the command `pca`, but instead build your algorithm using either the `eig` or `eigs` command (`eigs` computes just a few eigenvectors, which can be faster when you only care about the eigenvectors with large eigenvalues). We want you to think through all the steps involved in your facial recognition program, and that means doing the math "yourself".

(Optional) Extensions (These are not spelled out in much detail. We recommend you talk to a member of the teaching team before trying these (especially the second two).

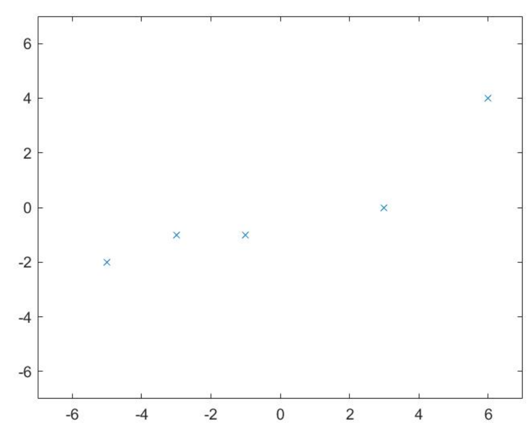
- Analyze the mistakes your algorithm makes (particularly when training on non-smiles and testing on smiles).
- Use Eigenfaces to do smile detection instead of identity recognition.
- Combine Eigenfaces with a classifier other than nearest neighbors (e.g., formulate an LSAE to create a series of one person versus everyone else detectors).
- Get your system working on live video.

Solution 14.1

1.
```
>> D = [-1 3; 1 4; 3 4; 7 5; 10 9]
>> plot(D(:,1),D(:,2),'x')
>> axis ([-3 12 -3 12])
```

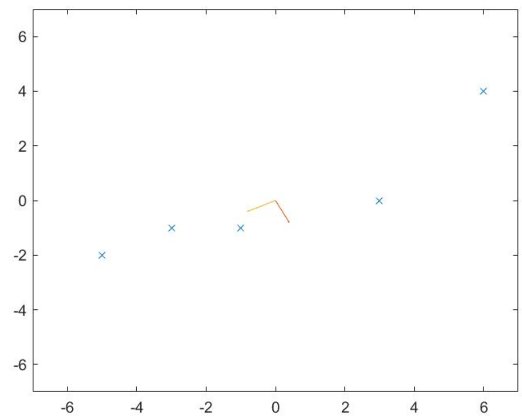


2.
```
>> mumatrix=[mean(D(:,1))*ones(5,1) mean(D(:,2))*ones(5,1)]
>> tildeD=D-mumatrix
>> plot(tildeD(:,1),tildeD(:,2),'x')
>> axis ([-7 7 -7 7])
```

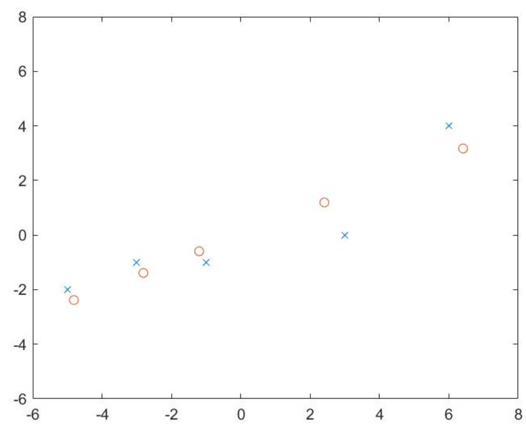


3.
```
>> [Vec,Diam]=eig(tildeD'*tildeD)
```

```
>> hold on
>> quiver(0,0,Vec(1,1),Vec(2,1))
>> quiver(0,0,Vec(1,2),Vec(2,2))
```



4.
```
>> proj=tildeD*Vec(:,2)
>> B=proj*Vec(:,2)'
>> hold on
>> plot(tildeD(:,1),tildeD(:,2),'x')
>> plot(B(:,1),B(:,2),'o')
>> axis ([-6 8 -6 8])
```



Solution 14.2

1. No. If we write a data point $a\mathbf{p}_1 + b\mathbf{p}_2$, then the reduced dimension version is $b\mathbf{p}_2$. It's impossible to recover a, which is the information in the perpendicular direction.

2. We would get the information in the perpendicular direction, which we can interpret as the "error" in reducing the dimension from $\mathbf{D}$ to $\mathbf{B}$.

3. You can use the error $\mathbf{B} - \mathbf{D}$.

4. We can write a new data point $\mathbf{d}$ as

$$(\mathbf{d} \cdot \mathbf{p}_1)\mathbf{p}_1 + (\mathbf{d} \cdot \mathbf{p}_2)\mathbf{p}_2.$$

Solution 14.3

1.
```
>> A = (1/sqrt(7304))*[b_tr-mean(b_tr) w_tr-mean(w_tr) s_tr-mean(s_tr)];
>> R=A'*A
```

2.
```
>> [V,D]=eig(R)
>> Vp=[V(:,2) V(:,3)]
```

3.
```
>> T = [b_new-mean(b_new) w_new-mean(w_new) s_new-mean(s_new)]'
```

4.
```
>> alpha=Vp'*T;
```

Solution 14.5

A reference implementation of Eigenfaces is linked from the Canvas assignment page.

Chapter 15

Day 8: Eigenface Synthesis and Project Kick-Off

Schedule

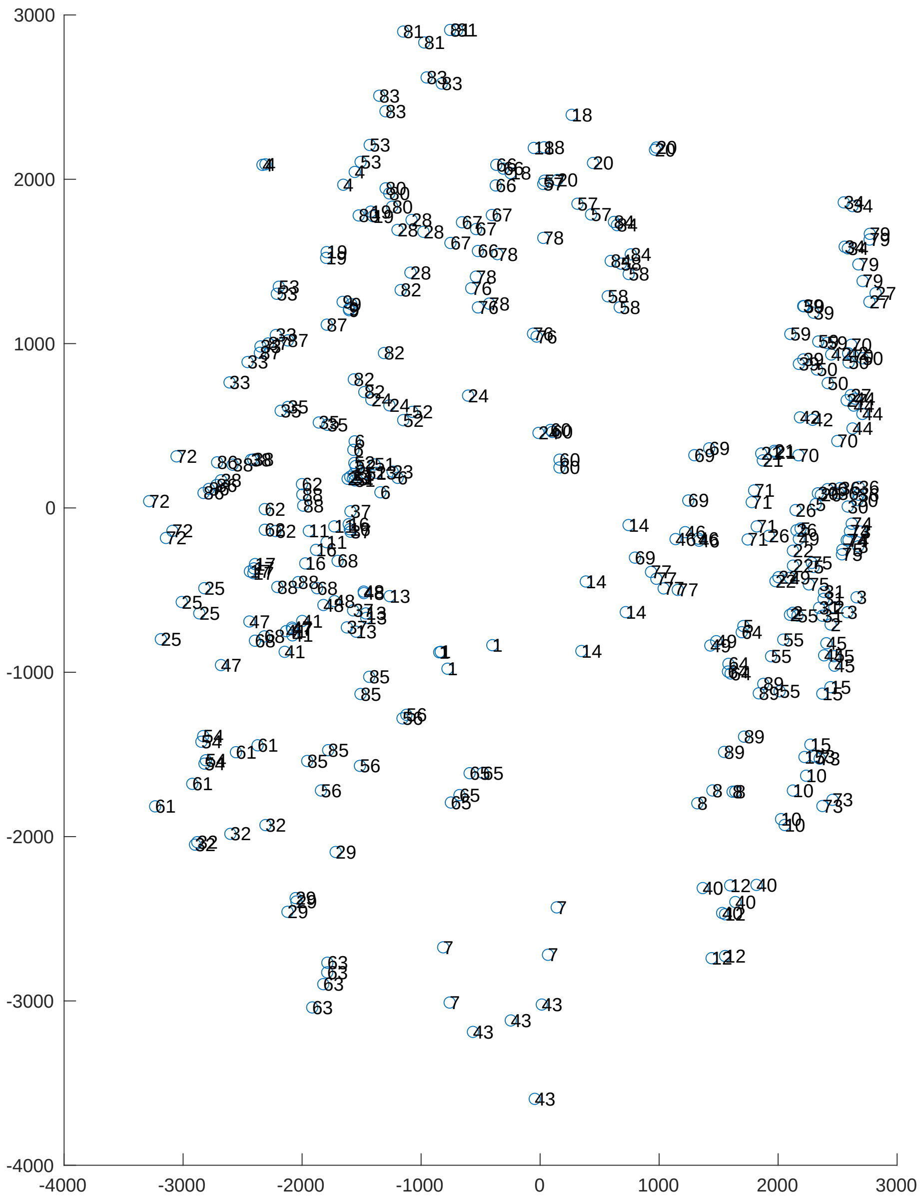
- 0900–1015 Debrief and Synthesis Activity on Eigenfaces
- 1015–1030 Coffee Break
- 1030–1115 Singular Value Decomposition — Theoretical
- 1115–1200 Singular Value Decomposition — In action
- 1200–1220 Review and Preview
- 1220–1230 Survey

15.1 Debrief and Synthesis Activity on Eigenfaces

During the overnight assignment you implemented a facial recognition routine. For the first part of the morning, we want you to work with your table mates to think again about your method and implementation.

An Aside on Nearest Neighbor Classification

One area where folks ran into trouble on the night assignment was the nearest neighbor approach to classification. In nearest neighbor classification you classify a test point, $\mathbf{x_t}$ (think of representing this as a column vector containing a new piece of data like a face image of someone whose identity we don't know), by comparing it to a set of labeled training points $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)$. Each y_i is the thing we are trying to predict (e.g., the identify of the person in a face image). In nearest neighbor classification we check to see which of the n training points is closest to the test point. When we find the one that is closest, we predict the label for $\mathbf{x_t}$ to be the same as the label of the closest point in the training set. To help get the idea across, here is a figure that shows all of your faces projected onto the first two principal components. Also shown are the subject ids for each point. You should be able to see from the figure that points with the same subject id tend to cluster together (i.e., are close together in facespace). This is where the power of nearest neighbor classification comes from.


3000
2000
1000
0
-1000
-2000
-3000
-4000
-4000
-3000
-2000
-1000
0
1000
2000
3000

Eigenfaces Review [45 mins]

We'd like you to take this opportunity to discuss with your table-mates the approach you took to facial recognition and the way you implemented it in MATLAB. The goal here is to think through the method at different levels from conceptual to code, resolving any confusion, and identifying whether any issues are primarily at the conceptual-level, the implementation-level, or the translation space in between. **We highly recommend that you set aside your existing code and focus on developing the approach with your table-mates.** Here are some questions to guide you:

- Will you pre-process the data? If so, how?
- How will you compute the Eigenfaces?
- How will you decide how many Eigenfaces to include? Which Eigenfaces might you leave out?
- How will you identify a face?
- How will you measure the accuracy of your implementation?
- How will you handle false positives?

Exercise 15.1

1. Sketch out on the board your conceptual approach to facial recognition—use words and diagrams here, and think of a set of key frames.
2. Sketch out on the board your mathematical approach to facial recognition—translate the key frames from your conceptual approach to mathematical expressions or equations.
3. Sketch out on the board your implementation approach to facial recognition—translate the key conceptual and mathematical ideas to MATLAB pseudo-code.

Eigenfaces Paper [45 mins]

This is on the next overnight assignment, but it would make sense to start this now if you have the time.

Check out *Eigenfaces for Recognition*, an early paper on eigenfaces, by M. Turk and A. Pentland. You have most of the tools to understand this paper, but the writing style might be unfamiliar (intense!). Spend 1 hour on this paper and then feel free to move on. The first 6 pages of this paper describe the use of eigenfaces in face recognition.

Check out other sources as well. Wikipedia is pretty useful for eigenfaces, and this later paper talks about eigenfaces and an extension called Fisherfaces (not fish faces).

Exercise 15.2

We are asking you read this paper for several reasons. We hope that it highlights and synthesizes all the material you've learned in this module. It will also give you practice reading a technical paper, which is a skill you'll continue to develop over your career.

1. In what ways was your approach to implementing the eigenfaces algorithm similar or different from the authors' approach?
2. In what ways did your understanding of the eigenfaces algorithm change after reading the paper?
3. Were there places in the reading that you "got stuck?" If so, how did you address that?
4. What questions do you have after reading the paper?

15.2 *Singular Value Decomposition (SVD) — Theoretical*

We previously met the Eigenvalue Decomposition (EVD), which we used on square matrices. There is no EVD for rectangular matrices, but there does exist a generalization known as the Singular Value Decomposition, which is one of the most useful matrix decompositions in applied linear algebra. We will begin with some background concepts on singular values and singular vectors, and then explore them in the context of a user-movie rating data matrix. See the following webpage at the American Mathematical Society for a good geometric discussion of the SVD.

The Big Idea

Rectangular matrices don't have eigenvalues and eigenvectors. However, they have a generalisation of these known as singular values and singular vectors.

- The singular values σ_i and singular vectors $\mathbf{u}_i, \mathbf{v}_i$ of an $n \times m$ rectangular matrix $\mathbf{A}$ satisfy the definition

$$\begin{aligned} \mathbf{A}\mathbf{v}_i &= \sigma_i \mathbf{u}_i && (15.1) \\ \mathbf{A}^T\mathbf{u}_i &= \sigma_i \mathbf{v}_i && (15.2) \end{aligned}$$

  The singular vectors $\mathbf{v}_i$ are known as the **right singular vectors** and the singular vectors $\mathbf{u}_i$ are known as the **left singular vectors**.

- There are precisely $r = min(n, m)$ non-zero singular values. The singular vectors $\mathbf{v}_i$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$, and the singular vectors $\mathbf{u}_i$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$. The r non-zero eigenvalues of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are σ_i^2.

- The $n \times m$ matrix $\mathbf{A}$ has a *singular value decomposition* (SVD) of the form

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \qquad (15.3)$$

  where $\mathbf{U}$ is an $n \times r$ orthogonal matrix whose columns are $\mathbf{u}_i$, $\mathbf{\Sigma}$ is an $r \times r$ diagonal matrix with r non-zero entries σ_i, and $\mathbf{V}$ is an $m \times r$ orthogonal matrix whose columns are $\mathbf{v}_i$. Please note that this version of the SVD is called the *reduced* or *economy* SVD - there is a more general form but this is the most useful in a practical setting.

Exercise 15.3

1. Read "The Big Idea" again!
2. Let's assume that $\mathbf{A}$ is a 3×2 matrix. What is the size of $\mathbf{A}^T$? What is the size of $\mathbf{v}_i$ and $\mathbf{u}_i$? What is the size of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$? How many eigenvalues will $\mathbf{A}^T\mathbf{A}$ have? How many eigenvalues will $\mathbf{A}\mathbf{A}^T$ have? What must be true about these eigenvalues according to "The Big Idea"?
3. Show that σ_i^2 and $\mathbf{v}_i$ are the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$ by multiplying Equation (15.1) by $\mathbf{A}^T$ and then using Equation (15.2) to simplify.
4. Show that σ_i^2 and $\mathbf{u}_i$ are the eigenvalues and eigenvectors of $\mathbf{A}\mathbf{A}^T$ by multiplying Equation (15.2) by $\mathbf{A}$ and then using Equation (15.1) to simplify.
5. Take the transpose of Equation (15.2) and justify the use of the term **left singular vector** for $\mathbf{u}_i$.
6. Why is it valid to write
$$\mathbf{A}[\mathbf{v}_1 \dots \mathbf{v}_r] = [\mathbf{u}_1 \dots \mathbf{u}_r] \begin{bmatrix} \sigma_1 & & \\ & \cdot & \\ & & \sigma_3 \end{bmatrix}$$
and why does this imply Equation (15.3)?
7. Why does Equation (15.3) imply that
$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T \tag{15.4}$$
8. An $n \times m$ matrix has nm data values. How many data values do you need to store σ_1, $\mathbf{u}_1$, and $\mathbf{v}_1$? What kind of compression ratio would you have if you only stored the first singular value and the first singular vectors?

15.3 *Singular Value Decomposition (SVD) — In Action*

We're going to be using a LiveScript notebook to go through an example of using SVD to analyze movie ratings.

You'll need to download both the dataset (movielens25m.mat) and the LiveScript (movieLens25m.mlx) before getting started. We have included a PDF of the LiveScript in this document for your referene.

Analyzing Movie Ratings via SVD

In this notebook we're going to be working with a subset the MovieLens25M dataset. The original dataset (https://grouplens.org/datasets/movielens/25m/) contains

- 25 million ratings
- from 162,000 users
- on 62,000 movies

The dataset was generated on November 21st, 2019, so it is pretty current. In this activity, we're going to be using a reduced subset of this data that only includes popular movies (that have been rated at least 1000 times) and users that have rated lots of movies (at least 500). This leaves us with

- 7.1 million ratings
- from 9,663 users
- on 3,790 movies

The goals of this activity are threefold.

1. To work with a different type of data than images or temperatures (here we will be working with ratings). Applying the tools you have learned in this module to different domains will help solidify your learning, help you see connections, and potentially get you excited for your module 1 project.
2. To see how SVD can be used to examine the important trends in your data (since we had lots of practice with using the EVD on the overnight).
3. To have some fun!

To get started, we're going to load the data and display a little bit of the data. Please see the comments in the code for some more information.

```
load('movielens25m.mat');
sizeOfMovies = size(movies)
% the cell array `movies` is 3706 by 3.  Each of the 3706 entries correspopnds to a
% particular movie, and along the second dimension the entries correspond to the movie
% the movie title, and the movie genre
%
% Here we extract the information about the first movie in the dataset
[movieId, movieTitle, movieGenre] = movies{1,:}

ratingsSize = size(ratings)
% the matrix `ratings` is 6040 by 3706 and encodes the rating that a
% particular user (row) gave to a particular movie (column).  The ratings
% are 1, 2, 3, 4, or 5 stars or the special value NaN (not a number) if the
% user didn't rate that particular movie.
%
% Let's look at the ratings that were given to the first movie in the
% dataset, which as we saw is Toy Story.  We can do this using the histc
% function (we'll ignore missing values in this analysis)
possibleRatings = [0.5:0.5:5];
nRatings = histc(ratings(1,:), possibleRatings);
figure;
```

```
bar(possibleRatings, nRatings);
xlabel('Rating');
ylabel('Number of Users');
title(['Ratings for ', movieTitle])
```

Okay, yeah that was a pretty great movie. Let's check out a less good movie, Anaconda. Highly recommended!! Look at this cast https://www.imdb.com/title/tt0118615/fullcredits !!!

```
anacondaIndex = 850;
[movieId, movieTitle, movieGenre] = movies{anacondaIndex,:}
nRatings = histc(ratings(anacondaIndex,:), possibleRatings);
figure;
bar(possibleRatings, nRatings);
xlabel('Rating');
ylabel('Number of Users');
title(['Ratings for ', movieTitle])
```

Cleaning up the Data

As you probably guessed, we're going to be applying SVD to this data. Before we start analyzing this data, we're going to do a few things to make the problem a bit easier to handle. First we're going to have to deal with the fact that we have a bunch of missing values in our ratings matrix (i.e., movies that particular users did not rate). The step of filling in missing values is called **data imputation**. There are many ways to do this, but we've chosen a particularly easy strategy of simply replacing any ratings with the average rating of that particular movie (e.g., if a user didn't rate Toy Story, we would fill it in with the average rating of Toy Story based on the other users in the dataset who actually rated that movie).

```
ratingsFilled = fillmissing(ratings, 'constant', nanmean(ratings));
```

As a final data cleaning step, we're going to subtract out the mean of each row. This will control for the fact that users vary considerably in how the numerical score they assign to movies (e.g., one user's 3 may be more comparable to amother user's 1).

```
ratingsMeanCentered = ratingsFilled - mean(ratingsFilled,2);
```

Framing the Problem Using SVD

Next, let's think about how SVD might help us to analyze this dataset. Suppose we compute the SVD of the matrix `ratingsMeanCentered`. Let's use $\mathbf{u}_1$ to refer to the first left singular vector, $\mathbf{v}_1$ to refer to the first right singular vector, and σ_1 to refer to the first singular value (let's assume that the first pair of singular vectors has the largest singular vector).

Exercise

Before running any other code in this notebook, answer the following questions regarding the first pair of singular vectors.

1. What are the sizes of $\mathbf{u}_1$ and $\mathbf{v}_1$? What do each of the dimensions of $\mathbf{u}_1$ correspond to? How about each dimension of $\mathbf{v}_1$?

2. In 15.3.8 we talked about compressing the original matrix down to m + n + 1 values. If we think of $\mathbf{u_1}, \mathbf{v_1}, \sigma_1$ as the compressed version of ratings data, how would we reconstruct the ratings data using $\mathbf{u_1}, \mathbf{v_1}, \sigma_1$ (you essentially did this already, we're hoping you can recall this fact from earlier and apply it here).
3. We can think of $\mathbf{v_1}$ as encoding the dominant trend that explains the ratings of each movie. For this dataset, what might this correspond to?
4. We can think of $\mathbf{u_1}$ as encoding the dominant trend that explains the ratings by each usere. For this dataset, what might this correspond to? Keep in mind we have already subtracted out the mean of each user. It might be helpful to expand your formula from problem 2 to see how $\mathbf{u_1}, \mathbf{v_1}, \sigma_1$ interact with each other.

Now we're going to compute the SVD. We'll just compute the 10 pairs of left and right singular vectors with the largest singular values.

```
[U, Sigma, V] = svds(ratingsMeanCentered, 10);
```

Examining the Right Singular Vectors

Now that we've computed our singular vectors, let's see if we can make sense of them. It turns out that the right singular vectors (the ones that have to do with movies) are generally more interpretable than the left singular vectors (the ones that have to do with users). We'll start out by looking at each right singular vector.

Exercise

Before running the code, think through the following question with your table-mates.

What might you do in order to make sense of what a particular right singular vector represents? Consider things like examining small or large values, looking for correlations, etc. There's not only one right answer, so throw out some ideas and try to think through what examining a particular aspect of the vector might tell you.

(we'll leave a little space to make it easier not to look at what we did)

Looking at Large and Small Values

One simple way to understand the right singular vectors is to look at the largest and smallest components of each vector. This will tell us which movies are either most strongly (positively) and most strongly (negatively) associated with this component. In the code below, we'll print out the title, genre, and component of the 10 movies that are most positively and most negatively associated with each right singular vector. **Exercise: Based on these outputs, can you tell a story about what the singular vector represents?**

```
for i = 1 : 10
    disp(['Component ', num2str(i)]);
```

```
    getHighAndLowMovies(V(:,i), movies)
end
```

Examining the Left Singular Vectors

Now we're going to check out the left singular vectors.

Exercise

Before running the code, think through the following question with your table-mates.

What might you do in order to make sense of what a particular left singular vector represents? Consider things like examining small or large values, looking for correlations, etc. There's not only one right answer, so throw out some ideas and try to think through what examining a particular aspect of the vector might tell you.

(we'll leave a little space to make it easier not to look at what we did)

Looking at Large and Small Values

Similarly to what we did for the right singular vectors, let's take a look at large (positive) and small (negative) components of each singular vector. Instead of looking at the top 10 and bottom 10, we're instead going to look at the single highest and single lowest component (each of which correspond to a user). For that user, we're going to show a sampling of movies that the user rated (focusing on the top 10 and bottom 10 ratings for that particular user). **Exercise: Given what you know about the corresponding right singular vector, try to make sense of the users that are at either extreme of the left singular vectors.**

```
for i = 1 : 10
    [~, highestUserIndex] = max(U(:,i));
    [~, lowestUserIndex] = min(U(:,i));
    disp('');
    disp(['Component ', num2str(i)]);
    disp('The user with the largest component rated the following movies as high and lo
    getHighAndLowUserRatings(highestUserIndex, movies, ratings)
    disp('The user with the smallest (probably negative) component rated the following
    getHighAndLowUserRatings(lowestUserIndex, movies, ratings)
end
```

Next Steps

To give you a sense of where you might take this in a project, here are some things you might investigate next with this dataset.

1. We didn't really look at how you would use the SVD to make recommendations. It turns out the SVD can be used to come up with good guesses for the missing values in the original ratings matrix (the NaNs) and you can then provide recommendations based tailored for a praticular user.
2. We didn't quantify how well the svd predicted the ratings. In order to do that, you could divide the ratings into a training and test set and see how well your SVD model can predict the test ratings (i.e., a rating set that wasn't used to compute the SVD).
3. We filled in the missing values with the means of each movie, but there are variants of SVD that can handle the missing values directly (they do entail tradeoffs). You could inverstigate how one of those methods would work on this data.

```
function movieExtremes = getHighAndLowMovies(v, movies)
    % return a cell array with the most positive and most negative
    % components of the right singular vector v.
    nHighLow = 10;
    movieExtremes = cell(nHighLow*2, 3);
    [c, indices] = sort(v);
    movieExtremes(1:nHighLow,1) = movies(indices(end-(nHighLow-1):end),2);
    movieExtremes(1:nHighLow,2) = movies(indices(end-(nHighLow-1):end),3);
    movieExtremes(1:nHighLow,3) = num2cell(c(end-(nHighLow-1):end));
    movieExtremes(1+nHighLow:end,1) = movies(indices(1:nHighLow),2);
    movieExtremes(1+nHighLow:end,2) = movies(indices(1:nHighLow),3);
    movieExtremes(1+nHighLow:end,3) = num2cell(c(1:nHighLow));
end

function userRatings = getHighAndLowUserRatings(userIndex, movies, ratings)
    % return a cell array with the most positive and most negative reviews
    % given by the specified user
    nHighLow = 10;
    userRatings = cell(nHighLow*2,2);
    [r, indices] = sort(ratings(userIndex,:));
    % filter out NaNs
    indices = indices(~isnan(r));
    r = r(~isnan(r));
    userRatings(1:nHighLow,1) = movies(indices(end-(nHighLow-1):end),2);
    userRatings(1:nHighLow,2) = num2cell(r(end-(nHighLow-1):end));
    userRatings(1+nHighLow:end,1) = movies(indices(1:nHighLow),2);
    userRatings(1+nHighLow:end,2) = num2cell(r(1:nHighLow));
end
```

Chapter 16

Night 8

16.1 Eigenfaces Paper

Check out *Eigenfaces for Recognition*, an early paper on eigenfaces, by M. Turk and A. Pentland. You have most of the tools to understand this paper, but the writing style might be unfamiliar (intense!). Spend 1 hour on this paper and then feel free to move on. The first 6 pages of this paper describe the use of eigenfaces in face recognition. Check out other sources as well. Wikipedia is pretty useful for eigenfaces, and this later paper talks about eigenfaces and an extension called Fisherfaces (not fish faces).

Exercise 16.1

We are asking you read this paper for several reasons. We hope that it highlights and synthesizes all the material you've learned in this module. It will also give you practice reading a technical paper, which is a skill you'll continue to develop over your career.

1. In what ways was your approach to implementing the eigenfaces algorithm similar or different from the authors' approach?
2. In what ways did your understanding of the eigenfaces algorithm change after reading the paper?
3. Were there places in the reading that you "got stuck?" If so, how did you address that?
4. What questions do you have after reading the paper?

16.2 Beginning the Project

In this project you will extend the work you have already done on facial recognition and feature detection by analyzing the performance of an existing algorithm within a real context. We know that facial recognition and other forms of feature detection algorithms are incredibly powerful, but they are often prone to failure, and those failures can have very real consequences on people's lives. This is a new formulation of this project where we are challenging you to think deeply about the contexts and consequences for facial/feature recognition algorithms. To prepare for tomorrow's in-class ideation activities, we ask you to do two things:

1. Read the project description, which can be found in the next chapter (Chapter 17), and write down any questions you find yourself asking.
2. Fill out this partner survey by midnight tonight, Feb. 19th. We will announce the teams tomorrow morning.

Chapter 17

Faces Project: The Context and Consequences of Feature Recognition, Detection, and Classification

17.1 Overview

This is a project that asks you to extend the work you have already done on facial recognition and feature detection by analyzing the performance and considering the consequences of an existing algorithm within a real(ish) context. This is a fairly new formulation of this project and we are giving you the freedom (and responsibility) to choose an interesting path and follow it judiciously.

> The LinAlgCo owns the rights to all uses of linear algebra. They've recently become aware of the use of linear algebra in *feature recognition, detection, and classification* ("*FeaRDeClass*") algorithms. The company is concerned about the ethical implications of the widespread use of these algorithms. They don't want to tarnish the good name of linear algebra. You have been hired as a consultant to address these concerns.
>
> Specifically, they've asked you to do the following:
>
> 1. Identify a specific context in which linear algebra is being used to do *FeaRDeClass*.
>   (Examples: Smile detection using linear regression, identifying missing persons from photos in social media, unlocking your phone with your face using Eigenfaces, classifying movie preferences...)
> 2. Pose a question about the context in which *FeaRDeClass* is being preformed and the possible ethical implications. You do not have to be able to answer this question, but it should guide your investigations.
>   (Examples: Privacy, bias, misuse, reinforcing negative structures...)
> 3. Answer some part of your question by analyzing the results of a *FeaRDeClass* algorithm. You can use any *FeaRDeClass* algorithm as long as you can explain how it works. You need to do some quantitative analysis, but the specifics are up to you.
>   (Examples: Does the facial recognition have higher accuracy with group X than group Y? Are STEM documentaries more likely to be recommended to men than women?)
>
> Your consulting team is expected to produce a formal report, due to LinAlgCo by Monday March 2nd at 9:00am.

17.2 What we expect you to do

1. Start with some background research on contexts for *FeaRDeClass* and their associated ethical issues. This research will help you to choose what to focus on, and you should also reference this research when discussing the context of your project in the introduction of of your report.

2. Choose an important question related the context and ethics of your chosen *FeaRDeClass*. This should be rooted in a real context, and you do not have to be able to answer it. Break off a small sub-question that you think you can answer in one and a half weeks through an analytical approach that utilizes eigenfaces, another facial recognition algorithm, or linear regression. Consider the ethical consequences of your chosen topic. Under what circumstances could the technology be harmful? Whom might it harm, and how?

3. Plan, execute, and document some analysis (which could include modifying/creating an algorithm) to answer your sub-question.

4. Explain the mathematical algorithm you are using in detail, explaining the various steps and what the purpose of each step of the process is. Use equations!

5. Explain how the results of your analysis inform the question you are trying to answer. Tie the results of your sub-question back into your larger question and chosen context. What can you conclude from the analysis you did? Recommend areas for future investigation.

6. You should understand the metrics against which your programs should be measured. How do you characterize the accuracy of your approach? Against what should you compare this accuracy? How do you quantify the consequences of your approach?

7. Communicate the context, analytical approach, and findings via a formal technical report to the LinAlgCo.

17.3 Resources

1. Your existing eigenfaces algorithm or the example solution posted. Let us know if you need help getting eigenfaces working.

2. The smile detection algorithm, which uses linear regression. Your version or the walkthrough from class can be modified to do something similar.

3. Training and test images for your class and past QEA classes.

4. The 10k faces database. This includes >2,000 images that have been classified in terms of demographics and other info (like whether people are facing the camera) and a software tool to narrow the database by classifiers (e.g., to only smiling men). The downside of this database is there is only one photo of each person.

5. The internet. In addition to doing context/background research, you can go find a different algorithm or face database if you prefer, but be aware that this will take extra time!

6. Your teaching team. Remember that we are here to support your learning! Bounce ideas off of us in office hours. Don't let MATLAB get you down; ask for help early and often.

17.4 Deliverable

You need to produce a written report, but we've broken it into a few sub-assignments to keep you on track and create opportunities for feedback.

1. Due Monday 2/24: An informal document outlining your chosen context, big question, sub-question, and the algorithm you will use to answer your sub-question, plus a plan for what analysis you will do and what kind of results you will get. (You should also get started on the analysis before Monday, but you don't have to turn any results in yet.) This document should serve as an outline for your final report.
2. Due Thursday 4/27: A draft of your written report. The teaching team will give you feedback. The more complete the draft, the more helpful your feedback will be! You should bring a printed version to class.
3. Due Monday 3/2: A final version of your written report.

17.5 Project report

You will generate a professional-looking and edited report to send to LinAlgCo that summarizes and justifies the decisions you have made. The executives at LinAlgCo are familiar with linear algebra and mathematics notation, but you should not assume they know anything about *FeaRDeClass* algorithms in general or the particular one you are studying.

The goal of the report should be to help LinAlgCo understand the context and consequences of the specific *FeaRDeClass* algorithm you have questioned and analyzed. You are NOT writing a story about what you did in the project. Aim for content and clarity, not length.

Structure

The report should have the following six sections (and you might want to break them into subsections):

1. Summary

What will I find in this report?

Open with a one paragraph summary that orients LinAlgCo to what they will find in the report. It should make clear why the report was written, what each section will accomplish, and what the key insights and results are. You should also summarize your recommendations.

2. Introduction

What is this project?

Your introduction should: (1) Provide the background and context for the *FeaRDeClass* that you have chosen to analyze. When and where is it used? By whom? What are the general technological or social issues associated with its use? (2) Explain your algorithm technically. How does it work? Bear in mind your audience. (3) Lay out the general ethical implications of the algorithm that you are investigating. Under what circumstances could the technology be helpful or harmful? Whom might it help or harm, and how? (4) Clearly state, within the broader ethical context, what question or issue you are exploring and and what sub-question you are quantitatively investigating.

3. Methods

How does your approach work, and what did you do with it?
Having introduced the reader to terminology and ideas, this section should lay out the approaches you are using, both in terms of the chosen algorithm and the analysis you are doing with it. Use equations and define all variables.

4. Detailed Findings

What are the main results and consequences of your work?
This section should contain your main results and consequences of your work which you have quantified. This section should contain some clear, informative, labeled, and captioned plots and images that demonstrate your findings. Quantitative results should be clearly connected to the context of the investigation. Why are your findings meaningful? Reflect on the downsides of the technology and the people it could hurt, and suggest some strategies for improvement.

5. Recommendations

What are the key takeaways? Summarize the key findings of the report, situate them in the greater context, and identify areas for future investigation. This section should be concise—they details go in the previous section.

6. References

Provide full citations for sources referenced in the paper. Format doesn't matter here as long as you provide sufficient information about each of your sources.

17.6 Grading rubric

1 pt. Summary presents a clear, high-level overview of paper.

3 pts. Question being investigated is clearly rooted in a real context, as discussed in the introduction and justified with references. Discussion of potential for harm is thorough.

2 pts. Algorithm is clearly explained using equations and words.

2 pts. Analytical approach is clearly explained.

2 pts. Findings are justified with appropriate figures and discussion.

2 pts. A clear connection is drawn between the findings and the original sub-question, greater question/issue, and greater context.

2 pts. Paper is logically organized and writing, figures, and equations are polished.

Chapter 18

Day 9

Schedule

- 0900–0930 Debrief Eigenfaces Paper
- 0930–1030 Project Ideation
- 1030–1045 Coffee Break
- 1045–1225 Project Worktime

18.1 Debrief Eigenfaces Paper

One of the goals of QEA is to develop your abilty to read technical papers and implement the main ideas. Much of engineering practice is based on building on the incredible work of those who came before us, and being able to critically read technical writing is a skill to be developed. Similarly, identifying the particularly helpful, or challenging, portions of a paper can help you improve your own technical writing. Please consider the following prompts:

Exercise 18.1

1. What did the **authors set out to show**? Why did the authors choose this particular goal?
2. What were the **best parts** of the paper? Was the approach to facial recognition easy to follow? What did the authors do particularly well?
3. Identify the **missing pieces** in the paper. These could be missing analyses, missing punch-line graphs, missing equations, etc. Another way to think about this is which of the main conclusions of the paper are you still the most skeptical of. Your skepticism could arise from a lack of evidence, or because the evidence was not presented in a clear and easily digestible format.

18.2 Project Ideation

You should be seated with your partner for the rest of class.

User Ideation Extravaganza

There are A LOT of possible questions you could propose for this project. In the project document we prompt you to *choose an important question related to feature recognition, detection, or classification.* This should be rooted in a real context, and you will likely not be able to answer it entirely. Break off a small subquestion that you think you can answer in one week through an analytical approach that utilizes eigenfaces, another facial recognition algorithm, or linear regression. We recognize that this is a very open ended and somewhat ambiguous prompt, but we feel you are up to the challenge! We will be taking the remainder of class to generate lots of possible questions, refine ideas, and try to generate teams around shared interests.

Exercise 18.2

1. Do this part individually [10 minutes]: Write down as many different ideas for possible questions related to facial/feature recognition/detection/classification as you can on different sticky notes. Go wild!
2. Do this part with your table and the neighboring table [10 minutes]: Our goal now is to get all of the questions up on the back board and to group them (in other words, all of the questions around flagging and the potential bias in that process could go into one group). Draw on the board as needed. This may feel a bit chaotic but we will help you get there.
3. Do this part with your partner [5 minutes]: Walk around the room with your partner, reading the idea clusters from the other table groups. Get inspired!

Pair Project Ideation

This next stage is going to give you an opportunity to work with your partner to develop a complete project idea and to "pitch" it to the class.

Exercise 18.3

1. With a partner at your table, select a question (or group of questions) from the boards. Grab the appropriate sticky notes and bring them to your table. [5 minutes]
2. Collaboratively develop an idea that identifies a question, it's context, and how you could perform some analysis. Do some internet research to find out more about your question and its context. What are the ethical issues associated with your topic? What is known and what is still in question? Fill out the project pitch handout. You will need to define the question itself, the real world context, and details about the critical concepts from this module that will allow you to perform the desired analysis. [20 minutes]

18.3 Project Worktime

Get started on the details of your project. You should consider this an extension of the project ideation time. Play with ideas, and hopefully, by the end of class you'll feel like you've settled on an idea and have a direction to go in. Use this time to chat with the faculty.

Chapter 19
Night 9

Project work time. Remember that the outline of your report is due Monday. You should also get started on the analysis before Monday, but you don't have to turn in any results yet. The last few sections can simply be an outline of what you will do.

Chapter 20

Day 10

Project work time.

Module II

Fear the Ducky

Chapter 21

Day 1: Goodbye Faces, Hello Duckies

21.1 Schedule

- 0900-0945: Sharing your project
- 0945-1015: Module 1 "Cheat-sheet"
- 1015-1030: Coffee
- 1030-1130: Linear Algebra Grab Bag
- 1130-1200: Mind-map
- 1200-1230: Preview of Module 2 and Overnight

21.2 Sharing your project

You've all just completed – or at least tried to complete – a project on feature detection and classification. And you've worked in a variety of areas. We'd like you to take some time to share your work with others in the class. In particular, we'd like you to share *what you worked on*, *why you worked on it*, *how your approach worked (or didn't)*, and *what you learned from the process.*

To do this, we're going to have two twenty minute sessions. During the first session one partner from each group will visit a few other tables to have this conversation while the second partner stays "at home" presenting; during the second session you'll switch roles. We'll be circulating during this time as well to ask the same questions.

21.3 Fairness, Accountability, Transparency in Machine Learning: More Food for Thought (optional reading for outside of class)

Earlier in the module you engaged with some resources on ethics and bias in machine learning (e.g., Joy Buolamwini's testimony to congress or Google's guidelines for inclusive AI). Also, as part of your project you discussed the ethical context surrounding your chosen project topic (and perhaps you were able to connect this to the technical bits of your work). While reviewing the rough drafts of your papers last class, we were inspired by how much thought most groups had put into this aspect of their projects. That said, in a relatively short project we have only just begun to scratch the surface of this topic. If you are interested in issues of algorithms and fairness (and in particular as they relate to machine learning) and you'd like to learn more, please consider the following reading list.

- Gender Recognition or Gender Reductionism?: The Social Implications of Embedded Gender Recognition Systems
- Automated Experiments on Ad Privacy Settings
- Falsehoods Programmers believe about Gender
- Fast AI's lists of AI Ethics Resources

- Fairness and machine learning: Limitations and Opportunities
- Readings on the COMPAS NorthPointe Recidivism Prediction Algorithm (one of the most important case studies for the field of fairness in machine learning)
  - Report of The Sentencing Project to the United Nations Special Rapporteur on Contemporary Forms of Racism, Racial Discrimination, Xenophobia, and Related Intolerance
  - How We Analyzed the COMPAS Recidivism Algorithm
  - COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity
  - Technical Response to Northpointe
  - Injustice Ex Machina: Predictive Algorithms in Criminal Sentencing
- Sam Daitzman and Austin Veseliza's ML and Ethics Reading List
- If you have a specific interest in this area, please contact the teaching team and we'll see if we can suggest resources for following up.

As you engage with this topic further, here are some potential questions to reflect upon. Please note that there really are no "right" answers to these questions!!

- As you read this literature or think about the topic of AI and ethics, how are you personally relating to the issues raised? Do you feel defensive? Threatened? Angry? Detached? Confused? Powerless? Analyze your reactions to see if you can figure out *why* you might be relating to the issues in this manner.
- As we engage with these topics as a community (e.g., within this class, at Olin, or in society more broadly), how are you supporting others in this endeavor (e.g., taking into account the different backgrounds and experiences each of us bring to learning, taking into account the varied responses folks have when engaging with these topics, being an empathic listener, speaking up when you hear something troubling)?
- As you engage with these topics, are these issues shaping your personal and professional identity (e.g., opening up potential new areas of study or professional work, shifting your sense of moral / ethical responsibility with regards to technology)?

21.4 Module 1 Cheat sheet

If you go back over all the stuff we've done in QEA over the last weeks, you'll see it's actually quite a bit of material – from eigenvectors to linear regression to shear matrices to inverses to...

Whenever you complete something, it's good to take some time stepping back and reviewing what you've learned (and perhaps what you still have to learn!). To facilitate this, we'd like you to do a number of activities. The first is to spend thirty minutes making a "cheat sheet" for the material we have covered thus far in QEA. Imagine that you are going to be given an exam on the stuff we've done, and you are allowed to bring one 2-sided sheet of paper to the exam with you. You may write as small as you want. But no electronic scaling!

21.5 Bad coffee is better than no coffee

21.6 Linear Algebra Grab Bag

You might have thought that cheat sheet was a theoretical exercise (insert evil laugh here)...

Seriously, work with your table at the board on the following problems as a way to review some linear algebra ideas – they range from easy to more challenging, and should help you to remember some of the big ideas. There is no need to do them in order and no need to do them all.

Exercise 21.1

Recall that $\mathbf{v}$ is an eigenvector of A with associated eigenvalue λ if $A\mathbf{v} = \lambda\mathbf{v}$.

1. If $\mathbf{v}$ is an eigenvector of A, is $3\mathbf{v}$ also an eigenvector?
2. Let A be an invertible matrix and let λ be an eigenvalue of A with corresponding eigenvector $\mathbf{v}$. Show that λ^{-1} is an eigenvalue of A^{-1}. What's a corresponding eigenvector?
3. If a square matrix A satisfies $A^2 = A$, what are possible eigenvalues for A?

Exercise 21.2

Let A be a 2×3 matrix and B be a 3×2 matrix.

1. Find examples of A and B such that zero is not an eigenvalue of AB.
2. Show that zero is an eigenvalue of BA for all A and B.

Exercise 21.3

Let A be an invertible $n \times m$ matrix. Show that $A^T A$ is also invertible. (Hint: Use the fact that a matrix is invertible if and only if $A\mathbf{x} \neq 0$ except when $\mathbf{x}$ is the zero vector.)

Exercise 21.4

1. The points (1,11),(2,10), and (4,14) all belong to a parabola of the form $y = \beta_2 x^2 + \beta_1 x + \beta_0$. Find β_2, β_1 and β_0 by setting up and solving an appropriate system of linear equations.
2. The data points (1,12),(2,9),(4,19) and (5,22) were gathered in a lab experiment and were supposed to lie on a parabola for the form $y = \beta_2 x^2 + \beta_1 x + \beta_0$. Show why there can be no

such parabola.

3. Find the best fit parabola of the form $y = \beta_2 x^2 + \beta_1 x + \beta_0$ for the data points above. (This uses linear regression, i.e., smile detection, technology.)

Exercise 21.5

Let $A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \\ 1 & 1 \end{bmatrix}$. Find the singular value decomposition of A. (You can use the MATLAB function `eigs`, but do not use `svd`.)

Exercise 21.6

Let A be an $m \times n$ matrix and Q be an orthogonal $n \times n$ matrix. Show that A and AQ have the same singular values.

Exercise 21.7

Two matrices A and B are *similar* if there exists a matrix P such that $B = P^{-1}AP$.

1. Show that if A and B are similar matrices, then A and B have the same eigenvalues.
2. Show that if A and B are similar matrices, then A^2 and B^2 are similar matrices.

Exercise 21.8

Let $A = \begin{bmatrix} 4 & -1 \\ 3 & 0 \end{bmatrix}$.

1. Write down the eigenvalue decomposition $A = VDV^{-1}$ for A.
2. Use the eigenvalue decomposition to compute A^k for any positive integer k.

Exercise 21.9

Let A be an $n \times n$ matrix. Assume that the eigenvalues $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n$ are all positive and the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are orthogonal (i.e., they form an eigenbasis).

Let $\mathbf{x}$ be any n-dimensional vector. We can write $\mathbf{x}$ in terms of the eigenbasis

$$\mathbf{x} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n.$$

1. Compute $\mathbf{x}^T\mathbf{x}$ in terms of the c's and $\mathbf{v}$'s from the eigenbasis expansion above.
2. Compute $\mathbf{x}^T A\mathbf{x}$ in terms of the c's and $\mathbf{v}$'s from the eigenbasis expansion above.
3. Write out an expression for
$$R = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$$
using the results from the previous two questions. (This is known as the Rayleigh quotient.)
4. What values for c make R as large as possible?

21.7 Concept map

You've now created a cheat sheet, and you've done some problems. The last review thing we want you to do is to try to create a "big picture" of the ideas from this module. In particular, we'd like you to create a concept map on the board something like this (but about the material we've covered, as opposed to mechanics!):

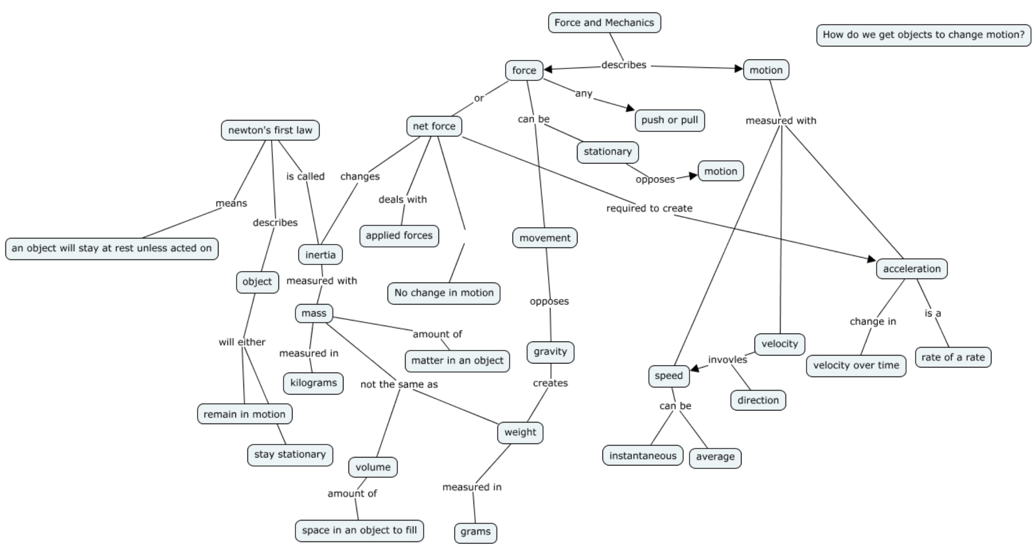


Figure 21.1: Mechanics Concept Map

Do this in groups of about 4. You can do this however you choose, and you're welcome to be creative about it! But please do be sure to make it pretty detailed: a concept map that only has "matrices", "matlab", and "eigenfaces" would miss the point.

21.8 Module 2 and Overnight Preview

Solution 21.1

Recall that $\mathbf{v}$ is an eigenvector of A with associated eigenvalue λ if $A\mathbf{v} = \lambda\mathbf{v}$.

1. If $\mathbf{v}$ is an eigenvector of A, is $3\mathbf{v}$ also an eigenvector? Yes, because $\mathbf{A}3\mathbf{v} = 3\mathbf{A}\mathbf{v} = 3\lambda\mathbf{v} = \lambda 3\mathbf{v}$

2. Let A be an invertible matrix and let λ be an eigenvalue of A with corresponding eigenvector $\mathbf{v}$. Show that λ^{-1} is an eigenvalue of A^{-1}. What's a corresponding eigenvector? Since $\mathbf{A}$ is invertible and $\lambda \neq 0$ we can multiple $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ by $\lambda^{-1}\mathbf{A}^{-1}$ to give $\mathbf{A}^{-1}\mathbf{v} = \lambda^{-1}\mathbf{v}$ so that λ^{-1} is an eigenvalue of $\mathbf{A}^{-1}$ with eigenvector $\mathbf{v}$.

3. If a square matrix A satisfies $A^2 = A$, what are possible eigenvalues for A?

Chapter 22

Night 1: Curves, Surfaces, Vectors, and Forces

Overview

During this mini-module you will be exploring some content in multi-variable calculus and mechanics. To start, we will be exploring the various mathematical representations of curves and surfaces. This material is important not only to functionally describing shapes for things like 3D design, but is the basis for mathematically describing the motion of bodies, which will be critical as we move into the robotics module.

Figure 22.1: This is “The Mule,” the test boat for the New York Yacht Club’s America’s Cup campaign. While “The Mule” relies on hydrofoil lift instead of buoyancy while flying, the curves of its hull are made beautifully clear. Credit: sailingworld.com

We need some building blocks about curves and surfaces in order to do this. There are very few books that are centered around curves and surfaces - these concepts typically show up in a dispersed fashion across a variety of courses, both within mathematics and in other disciplines. We will therefore be picking ideas from a variety of places to highlight the interdisciplinary nature of the material.

Learning Objectives

Concepts

- Distinguish between equations that represent explicit functions, implicit functions, and parametric functions.
- Identify exponential, polynomial and trigonometric relationships by the shape of their curves.
- Describe how changes in parameters affect the shape of curves or surfaces.
- Determine a mathematical approximation to the surface of real physical object.

MATLAB Skills

- Use MATLAB to define and visualize curves and surfaces defined by explicit functions, implicit functions, and parametric functions. These MATLAB functions are **plot, plot3, contour, surf, isosurface**.

Sources

There is no single book or website that captures all of this material. We have assembled several great textbooks on the rack in AC113, and the Library is an excellent source for materials as well. Some good sources include:

- College Algebra. Look for sections on basic functions and their visualization.
- Single-variable and Multi-variable calculus. Look for sections on functions, curves and surfaces, parametric curves, parametric surfaces.
- The Khan Academy's material on *Thinking about Multivariable functions* is probably useful.
- Computational geometry books, 3D game rendering books, CAD books - look for sections on curves and surfaces.
- The articles developed, written, and nurtured by Eric Weisstein at Wolfram Mathworld might be a great starting point: mathworld.wolfram.com.
- Wikipedia can be a useful source, but the quality of entries on technical material is highly variable.

22.1 Curves

Curves defined Explicitly

If you recall, single-variable calculus involved explicit functions of a single variable, e.g. $y = t^2$, $y = \sin(t)$, or more generally $y = f(t)$. You spent a lot of time visualizing these functions, solving equations with these functions, and computing related properties like derivatives and integrals.

Let's consider the function, $y = mt + b$, where m and b are parameters. We probably recognise this function, and that it's graph is a straight-line with slope m and intercept b.

In MATLAB we can visualize this function using the **plot** function which you are probably already familiar with.

```
>> m = 2
>> b = 1
>> t = linspace(-10,10,1000)
>> y = m*t+b
>> plot(t,y,'red')
```

First, we define a value of m and a value of b as an example. Second, we use the function **linspace** to generate 1000 equally-spaced points between -10 and +10. There is nothing special about this domain, except that the resulting graph captures the behavior of the function. Third, we evaluate the mathematical function at these points and store the result in y. Fourth, we call the **plot** function to generate the curve — we use red in this case because it looks great! Hopefully we recognize the classic straight-line which has the following features:

- y tends to $\pm\infty$ as $t \to \pm\infty$ if $m > 0$.
- y tends to $\mp\infty$ as $t \to \pm\infty$ if $m < 0$.
- The line passes through the point $(0, b)$.

One way to capture these different behaviors is to plot sample curves in each quadrant of a $b - m$ space.

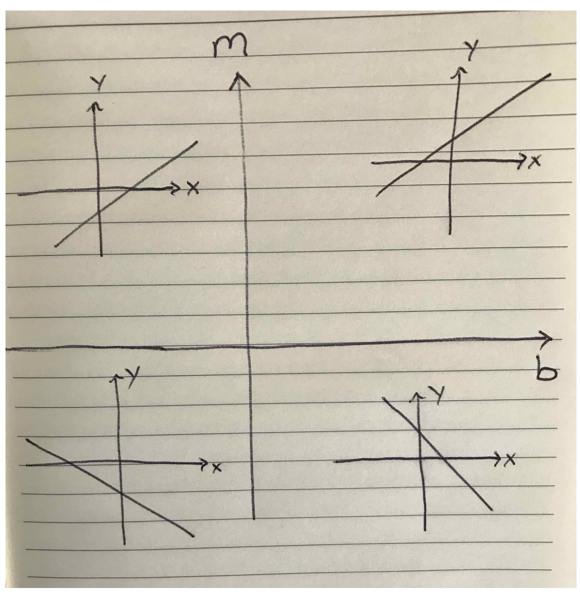


Figure 22.2: Straight-lines with different slopes and intercepts.

The meaning and interpretation of parameters and curves depends on their application. For example, if t is time and y is population, then this function could be used to model a population that is increasing or decreasing **linearly** in time. The parameter b would have units of population and would represent the initial population at $t = 0$. The parameter m would have units of population per time and would capture the linear growth or decay, e.g. $m = 0.1$ means we are adding 0.1 population units per unit time.

How long does it take a population to double if it is growing linearly in time? Since $y = b$ at $t = 0$ we can find the doubling time T at which $y = 2b$, i.e.

$$2b = mT + b \tag{22.1}$$

Solving for T gives

$$T = \frac{b}{m} \tag{22.2}$$

Notice that this doubling time is proportional to the initial population b and inversely proportional to the linear growth m. For example, if the initial population is $300M$ and we add $10M$ per year it would naturally take 30 years to reach $600M$.

Exercise 22.1

Consider the exponential function $y = Ae^{kt}$ where A and k are parameters.

1. What is the value of y when $t = 0$?
2. What happens to y as $t \to \pm\infty$? How does this limiting behavior depend on the sign of A and k?
3. Now sketch examples of the curves in the four quadrants of the $A - k$ space.
4. What is the effect of the parameters A and k on the curve?
5. Assuming that t is time and y is population, how would you interpret the parameters $A > 0$ and $k > 0$?
6. How long does it take an exponentially increasing population to double if $k = 0.1$?

Exercise 22.2

Consider the logistic function $y = A/(1 + e^{-kt})$ where A and k are parameters.

1. What is the value of y when $t = 0$?
2. What happens to y as $t \to \pm\infty$? How does this limiting behavior depend on the sign of k?
3. Now sketch examples of the curves in the four quadrants of the $A - k$ space.

4. What is the effect of the parameters A and k on the curve?
5. Assuming that t is time (years) and y is population, how would you interpret the parameters $A > 0$ and $k > 0$?
6. How long would it take an initial population of $A/2$ to reach 99% of the carrying capacity if $k = 0.1$?

Exercise 22.3

Consider the trigonometric function $y = A\sin(\omega t + \phi)$ with parameters A , ω, and ϕ.

1. Sketch some representative examples of these curves for different values of the parameters.
2. What features of the curve do A, ω, and ϕ control? Use the internet to deepen your understanding of these parameters.
3. Find the value(s) of t corresponding to $y = 2$ for $A = 3$, $\omega = 4$, and $\phi = 5$.

Exercise 22.4

Consider the quadratic polynomial in vertex form $y = g(x - h)^2 + k$, with parameters g,h,and k.

1. Sketch some representative curves for different parameter values.
2. What features of the curve do g, h, and k control?
3. What is the relationship between g,h, and k in the vertex form and a, b, and c in the standard form $y = c + bx + ax^2$? (This will require some algebra.)

The quadratic polynomial is probably very familiar to you. There are numerous ways to write this second-order polynomial, and we've used two forms here: the standard form and the vertex form. It is hard to tell the effect of each parameter in standard form. Using the vertex form, however, the effect of each parameter is much easier to interpret.

Curves defined Implicitly

Not every curve can be expressed in terms of an explicit function in which there is only one output for each value of the input. Curves can also be expressed implicitly through a relationship between 2 variables. A circle is a good example. For example, the equation for a circle of radius 1, centered at the origin, is

$$x^2 + y^2 - 1 = 0 \tag{22.15}$$

The left-hand side of this equation can be thought of as a function of two variables, $f(x, y) = x^2 + y^2 - 1$ and the set of points (x, y) where $f = 0$ defines a curve that we like to call the unit circle.

In order to visualize such curves in MATLAB we use the **contour** function. We begin by defining a grid of (x, y) points using the **meshgrid** function

```
>> [x,y]=meshgrid(linspace(-2,2,100),linspace(-3,3,200));
```

You will notice that both x and y are 200×100 matrices. There are 200 rows corresponding to the 200 y-values between -3 and 3. There are 100 columns corresponding to the 100 x-values between -2 and 2. There is nothing special about the limits of the domain or the number of points in each direction - we chose values here that would help explain the size of the resulting matrices.

Now that we have the grid defined, we compute the value of the function f at every point. Since x and y are already matrices we can use

```
>> f = x.^2 + y.^2 - 1
```

Notice that we use the . operator because we want every entry in the x matrix to be squared, and similarly for y. You will also notice in MATLAB that f is a 200×100 matrix. In theory, subtracting "1" (a scalar) from a matrix should not be permitted, but the good people at MATLAB have decided to interpret this for us automatically.

To plot the curve we now use the **contour** function

```
>> contour(x,y,f,[0 0])
>> axis equal
```

which should produce a circle of radius 1 centered at the origin. The last argument to the **contour** function tells it to draw the contour at $f = 0$. Don't ask why you have to put two zeros instead of just one because only MATLAB knows. Without the "axis equal" the curve would look like a ellipse due to the different scaling MATLAB will use in the x and y directions.

There is no end to the functions of two variables that you can define. There is, however, a set of functions that show up again and again, and these are the quadratic functions of two variables. The general form (containing all possible quadratic, linear and constant terms) is

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{22.16}$$

where a, b, c, d, e, f are arbitrary parameters, some of which may be zero. The curves defined by this equation are called **conic sections**, and represent the intersection of a double cone and a plane. The non-degenerate cases include circles, parabolas, ellipses, and hyperbolas. See the Wikipedia article on conic section for more information.

Exercise 22.5

Use the internet to find the implicit equation for a circle of radius R, centered at the point (a, b). Visualize the circle in MATLAB for different values of a, b, R.

This is a warm-up question. The implicit equation for a circle centered away from the origin should be easy to find, and you should use the visualization to check that changing the parameters moves the circle and changes its radius in the way you expect.

Exercise 22.6

Visualize an ellipse using the implicit definition

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0 \tag{22.18}$$

for different values of a and b. What features of the ellipse do a and b control? Use the internet to deepen your understanding of the parameters a and b—see for example the Wikipedia article on conic section.

This question requires a little modification to the visualization for the circle, and a little internet research to fully understand the parameters. Start with the article on conic section, and then spend a little time exploring after that - don't get lost in the world of the internet, and don't be surprised when you see lots of terminology that you don't understand.

Exercise 22.7

Visualize an hyperbola using the implicit definition

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} - 1 = 0 \tag{22.19}$$

for different values of $a > 0$ and $b > 0$. What features of the hyperbola do a and b control? Use the internet to deepen your understanding of the parameters a and b.

This question is similar to the one for the ellipse. In this case, however, interpreting the parameters without some additional reading is much harder because the precise impact of the parameters is not obvious from visualization. Again, start with the Wikipedia article on conic section and take it from there.

Exercise 22.8

Use the internet to find the conditions under which the solutions of

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{22.20}$$

define an ellipse, a parabola, a hyperbola, and a circle.

This question is meant to broaden your understanding of the possible solutions of this general quadratic polynomial in two variables. Start with the Wikipedia article on conic section.

Curves defined Parametrically

A more general representation of a curve involves expressing it's coordinates in terms of another independent variable or parameter as follows

$$x = f(u), y = g(u), u \in [a, b] \tag{22.21}$$

Each value of u defines a point with coordinates $(f(u), g(u))$. If we collect all the points defined by u in a specific interval, then we get a parametric curve. For example, the definition

$$x = cos(u), y = sin(u), u \in [0, 2\pi] \tag{22.22}$$

defines a unit circle centered at the origin which begins and ends at $(1, 0)$ and is traced out counterclockwise as u increases from 0 to 2π. The parameter u can therefore be thought of as the angle from the x-axis to the current point on the circle.

To visualize parametric curves in MATLAB we still use the **plot** function as follows

```
>> u = linspace(0,2*pi,100);
>> x = cos(u);
>> y = sin(u);
>> plot(x,y,'*')
>> axis equal
```

We first define a set of u points on the interval $[0, 2\pi]$. We then compute the x and y coordinates for every value of u. We finally plot the points, using an * for clarity and an "axis equal" so that we recognise the circle.

How do we "know" that these parametric equations trace out a circle, and not just a curve that looks like a circle? Let's check by substituting the definition of x and y into the equation for a circle of radius 1, centered at the origin,

$$x^2 + y^2 - 1 = cos^2(u) + sin^2(u) - 1 = 0 \tag{22.23}$$

which required the use of the trigonometric identity $\cos^2(u) + \sin^2(u) = 1$.

Exercise 22.9

Use the internet to find a set of parametric equations that define an ellipse, and use MATLAB to verify them visually. Show that the parametric equations satisfy the implicit equation for an ellipse.
This is a small change to the parametric equations for a circle, and finding parametric equations on the internet should be straight-forward - try searching on "parametric equations for ellipse" or start with the Wikipedia page on conic section or ellipse.

Exercise 22.10

A logarithmic spiral can be defined by the parametric equations

$$x = ae^{-bu}\cos(u), y = ae^{-bu}\sin(u), a > 0, b > 0, u \in [0, \infty) \tag{22.26}$$

Visualize the curve in MATLAB for different values of a and b—you won't be able to define an infinite domain but you can define a large one. How does a and b change the curve?
This question involves a curve that has been of interest to mathematicians and scientists for many centuries. Try searching on the internet for the term "logarithmic spiral".

Exercise 22.11

A helix in 3D can be defined by the parametric equations

$$x = a\cos(u), y = a\sin(u), z = bu, a > 0, b > 0, u > 0 \tag{22.27}$$

Visualize this curve for different values of a and b. How do a and b change the curve? (You will need to use **plot3** in MATLAB)
This question demonstrates that it is relatively simple to define a curve in 3D - just define the x, y, and z coordinates in terms of a single parameter. This curve is a good example, and has been widely studied in modern biology given its connection to the shape of DNA. Use the Wikipedia article on "helix" as a starting point.

Data-Driven Curves

We are often tasked with finding a curve that fits a set of data. You've probably seen informal approaches to this, particularly when finding the best-fit straight-line to a set of data points. Fortunately we have a robust, formal tool at our disposal now - orthogonal projection, often known as linear regression in this context.

Let's start with some data. Consider the 4 points $(0,1), (1,0), (3,2), (5,4)$. We can use MATLAB to plot these points

```
>> x = [0 1 3 5]';
>> y = [1 0 2 4]';
>> plot(x,y,'*')
```

Notice that we placed all of the x-coordinates in a column vector x and all of the y-coordinates in a column vector y. Let's now find the best-fit straight-line through these points, i.e. let's find the parameters m and b so that the straight-line defined by

$$y = mx + b \tag{22.28}$$

fits the points as well as possible.

The approach we take is motivated by our work in linear algebra. If we pack all of the x-coordinates into a vector $\mathbf{x}$ and all of the y-coordinates into a vector $\mathbf{y}$ then we would like to satisfy the vector equation

$$\mathbf{y} = m\mathbf{x} + b \tag{22.29}$$

as well as we can. Notice that there are 4 equations here (one for each point) and only two unknown parameters. An exact solution is impossible (unless the points happen to lie on a line) and so we use orthogonal projection to find the best solution. Let's define a matrix $\mathbf{A}$ and parameter vector $\mathbf{p}$ so that the vector equation for a straight-line becomes

$$\mathbf{A}\mathbf{p} = \mathbf{y} \tag{22.30}$$

where $\mathbf{A}$ and $\mathbf{p}$ are given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{x} & \mathbf{1} \end{bmatrix}, \mathbf{p} = \begin{bmatrix} m \\ b \end{bmatrix}$$

Notice that there is a coefficient of "1" in front of the "b" term so we had to create a column vector and fill it with 1's.

Recall that to find the best solution we multiply by $\mathbf{A}^T$,

$$\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y} \tag{22.31}$$

and solve this linear system for $\mathbf{p}$. In MATLAB we can simply use the backslash operator as follows (assuming we already defined x and y as earlier), since it will return the best fit solution when $\mathbf{A}$ has more rows than columns.

```
>> A = [x ones(4,1)]
>> p = A\y
```

For these data points we should find that $m = p(1) = 0.6949$ and $b = p(2) = 0.1864$. You should plot the straight-line defined by this slope and intercept to see how good the fit is.

Exercise 22.12

Find the best-fit parabola for these 4 data points. Recall that a parabola can be defined using the explicit function $y = ax^2 + bx + c$.

22.2 Surfaces

Surfaces defined Explicitly

If we assign the output of a function of two variables $f(x, y)$ to be a third variable, $z = f(x, y)$, then the set of points in 3D define a surface. For example, $z = x^2 + y^2$ defines a paraboloid. This surface can be visualized in MATLAB using the **surf** function.

```
>> [x,y]=meshgrid(linspace(-2,2,100),linspace(-2,2,100));
>> z = x.^2 + y.^2;
>> surf(x,y,z)
>> shading interp
```

First we lay down a grid of points in the xy-plane using **meshgrid**. Next we compute the value of the function at each of these points and assign the value to z. Finally we pass the x, y, z matrices to **surf** for rendering—we include a shading option to make the surface look nice and smooth.

It is often helpful to visualize a surface by drawing the contours defined by holding one of the variables constant. For example, if we define $z = 1$ in the equation for the paraboloid we obtain $x^2 + y^2 = 1$, which we know to be the equation of a circle of radius 1, centered at the origin. Choosing different values of z will define circles of radius $\sqrt{z}$. We already used the **contour** function in MATLAB earlier—here we will use it to draw the contours at different values of z

```
>> contour(x,y,z,'ShowText','On')
>> axis equal
```

In this case we are allowing MATLAB to pick the contour levels and we are including labels on the contours to show the corresponding value of z. We include the "axis equal" option in order to recognise that the contours are circles.

We can also "slice" the surface along the different coordinate directions. For example, if we wanted to plot the contours in the yz-plane where x is constant we would use

```
>> contour(y,z,x,'ShowText','On')
```

If we define $x = c$ and replace it into the definition of the function we see that

$$z = y^2 + c^2 \tag{22.32}$$

which is the equation of a parabola in the yz-plane and the value of c controls where it crosses the z-axis $(y = 0)$. The contour plot should support that analysis. We could also view the constant y contours in the xz-plane and we would find parabolas again—thus the reason we refer to the surface as a paraboloid.

Exercise 22.13

Visualize the elliptic paraboloid $z = x^2/a^2 + y^2/b^2$ for different values of $a > 0$ and $b > 0$.

1. Describe the contours in the yz-plane defined by $x = c$.
2. Describe the contours in the xz-plane defined by $y = c$.
3. Describe the contours in the xy-plane defined by $z = c$.

This question requires you to combine surface visualization with the curve visualization that we met earlier. To fully understand the parameters you should try to explain why the surface is called an elliptic paraboloid.

Surfaces defined Implicitly

A surface in three dimensions can also be implicitly defined by a function of three variables. For example, the equation for a unit sphere centered at the origin is

$$x^2 + y^2 + z^2 - 1 = 0 \tag{22.33}$$

The left hand side of this equation can be thought of as a function of three variables, $f(x, y, z)$, and the set of points where $f = 0$ defines the unit sphere. We can use the **isosurface** function in MATLAB to visualize:

```
>> [x,y,z] = meshgrid(linspace(-2,2,100),linspace(-2,2,100),linspace(-2,2,100));
>> f = x.^2 + y.^2 + z.^2 - 1;
>> isosurface(x,y,z,f,0)
>> axis equal
```

We first define a set of points in 3D space using the **meshgrid** function. Next we evaluate the function f at all of these points. We then use **isosurface** to render the surface defined by $f = 0$, and we use the "axis equal" option so that the resulting looks like a sphere.

There are lots of implicit surfaces, but a particularly important group is the quadratic (or quadric) surfaces, defined by the equation:

$$Ax^2 + By^2 + Cz^2 + Dyz + Ezx + Fxy + Gx + Hy + Iz + J = 0 \tag{22.34}$$

where A, B, C, D, E, F, G, H, I, and J are all arbitrary constants, some of which may be zero.

Exercise 22.14

Visualize the hyperboloid of one sheet defined by

$$x^2/a^2 + y^2/b^2 - z^2/c^2 - 1 = 0 \tag{22.35}$$

for different values of a, b, c. What features of the hyperboloid do a, b, c control?

Surfaces defined Parametrically

Finally, a more general representation of a surface involves expressing it's coordinates in terms of two independent variables as follows

$$x = f(u,v), y = g(u,v), z = h(u,v), u \in [a,b], v \in [c,d] \tag{22.36}$$

Each value of (u,v) defines a point in 3D with coordinates $(f(u,v), g(u,v), h(u,v))$. If we collect all the points defined by (u,v) in the specified domain, then we get a parametric surface. For example, the definition

$$x = sin(u)cos(v), y = sin(u)sin(v), z = cos(u), u \in [0,\pi], v \in [0,2\pi] \tag{22.37}$$

defines a unit sphere. In MATLAB we visualize a parametric surface using **surf**.

```
>> [u,v] = meshgrid(linspace(0,pi,100),linspace(0,2*pi,100));
>> x = sin(u).*cos(v);
>> y = sin(u).*sin(v);
>> z = cos(u);
>> surf(x,y,z), shading interp
>> axis equal
```

First we lay down a grid of points in the (u,v) space using **meshgrid**. We then compute x, y, z at each of these points, and we render the surface using **surf**.

Exercise 22.15

Lookup the parametric equations that define an ellipsoid, and use MATLAB to visualize.

Exercise 22.16

Visualize the following parametric surface

$$x = (a + r\cos(u))\cos(v), y = (a + r\cos(u))\sin(v), z = r\sin(u) \tag{22.38}$$

with $r < a$ and $u \in [0,2\pi], v \in [0,2\pi]$. Describe the surface and interpret the parameters a and r.

22.3 Designing Curves and Surfaces

Exercise 22.17

1. Pick a fruit or vegetable. Sketch it on paper from a variety of viewpoints. Now slice it in three ways, and sketch the sets of curves defined by each of these sets of slices.
2. Propose and evaluate a mathematical representation that is a good approximation to your

fruit or vegetable. You could represent the entire surface, or you could design a set of curves that are good approximations to the slices.

22.4 Vectors and Vector Operations: Cartesian

Many of the vectors we worked with in the first module were abstract, e.g. we treated day temperatures as a vector, or we treated an image as a vector. We are now going to turn out attention to **physical** vectors, that encode information like **position**, **velocity**, **acceleration**, **force**, **torque**, etc.

Exercise 22.18

Video Review: A few years ago Mark got tired of giving the same lecture over and over again, and he put together a bunch of videos. They may or may not be new ideas for you, but they contain valuable information and should be reviewed so everyone has a firm foundation moving forward in the course. Visit this link, and watch the Overview video as well as the videos in the first section on vectors and vector operations. This should take you about an hour if you watch it at regular speed and watch everything, or maybe 30 minutes if you watch at 2x speed. Which ever you choose to do, please complete the following table by listing the important ideas in the videos. For example, one important idea might be “Vectors have both direction and magnitude”. This table will be discussed at your table in debrief, so please make sure to identify key concepts.

Ideas in the videos I already knew	Ideas in the videos I had not seen before

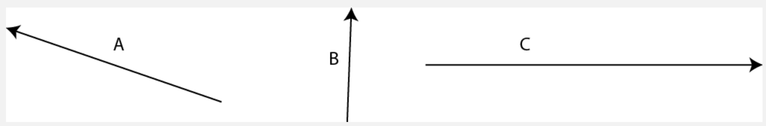
Conceptual Questions about Vectors

Exercise 22.19

Dot and Cross: What does the dot product of two vectors *tell you* about the two vectors? What about the cross product?

Exercise 22.20

Geometrical Vectors: The diagram shows three vectors, $\vec{A}$, $\vec{B}$, and $\vec{C}$. All three are in the plane of the page; their magnitudes are (respectively) 2, 1, and 3. For each operation below, either draw the results of the identified operations, or (if appropriate) give a best guess as to the value, or (if appropriate) identify the operation as nonsense.



1. $\vec{A} + \vec{B}$
2. $\vec{A} - \vec{C}$
3. $\vec{A} \cdot \vec{B}$
4. $(\vec{A} \times \vec{B}) \times \vec{C}$
5. $(\vec{A} \cdot \vec{B}) \times \vec{C}$

Exercise 22.21

Calculating Cartesian Vectors: Let $\vec{A} = 3\hat{\imath} + 4\hat{\jmath}$, $\vec{B} = \hat{\imath} - \hat{\jmath}$, and $\vec{C} = -5\hat{\jmath}$. Find the results of identified operations, or (if appropriate) identify the operation as nonsense.

1. $|\vec{A} + \vec{B}|$
2. $\vec{A} \times \vec{C}$
3. $\vec{A} \cdot \vec{B}$

Exercise 22.22

Constructing Useful Vectors: It is often the case in modeling that you want to construct a unit vector that points in a particular direction relative to some other vectors: “I want a unit vector that is perpendicular to the deck of the boat”, or “The Magnus force is perpendicular to the rotational axis and the velocity”. There are some nifty tricks for constructing this kind of vector; this problem asks you to think through them. $\vec{A}$ and $\vec{B}$ are two arbitrary, non-parallel vectors in three-dimensional space. Using them, construct the following vectors (i.e., find mathematical expressions for the specified vector in terms of the vectors $\vec{A}$ and $\vec{B}$):

1. The vector $\hat{A}$, which has a length of 1, and points in the direction of $\vec{A}$.
2. The vector $\hat{n}$, which has a length of 1, and is perpendicular to both $\vec{A}$ and $\vec{B}$.

22.5 Forces: Ideas and Models

In previous exposures to physics content you may have been introduced to various types of forces and force models. Two of the fundamental big-picture takeaways about forces are:

- Force is always a vector, and in order to fully specify the force, you need to give both magnitude and direction.
- When adding forces, they must be added vectorially.

Exercise 22.23

Video Review: Hooray, more videos! Again, they may or may not be new ideas for you, but these concepts will be important throughout the course. Visit this link, and watch the videos relating to Models for Forces. Again, this should take you about an hour if you watch it at regular speed and watch everything, or maybe 30 minutes if you watch at 2x speed. Which ever you choose to do, please complete the following table by listing the important ideas in the videos.

Ideas in the videos I already knew	Ideas in the videos I had not seen before

Exercise 22.24

Force Concepts

1. "I am pushing a coffee cup across my desk. Since it is in motion, I know that the magnitude of the frictional force acting on the cup is given by $\mu_k N$, where μ_k is the coefficient of kinetic friction and N is the normal force the table exerts on the cup." True or false? Why?

2. "I am pushing on a coffee cup that is sitting on my desk. Since it is not moving, I know that the magnitude of the frictional force acting on the cup is given by $\mu_s N$, where μ_s is the coefficient of static friction and N is the normal force the table exerts on the cup." True or false? Why? (Think carefully about this one!)

3. If an object of mass m is sitting alone and stationary on a level surface, what is the magnitude of the normal force that the surface exerts on the object?

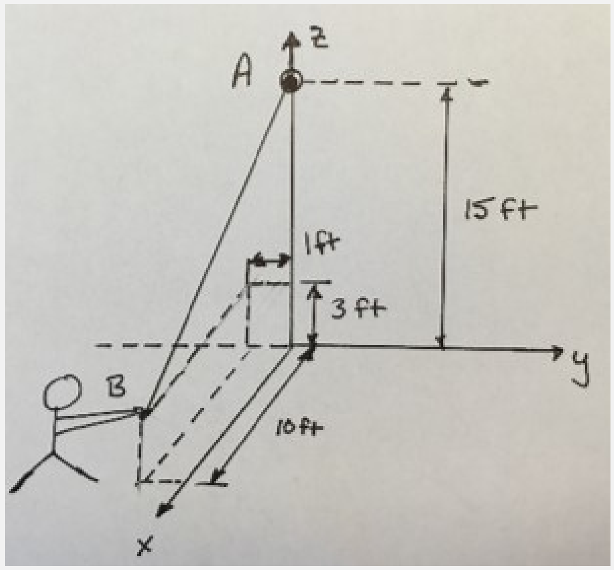
4. Now consider that we drop this object onto the surface. During the time that the object is decelerating to a stop when it comes into contact with the surface is the normal force exerted by the surface on the object greater than, less than, or equal to the force from part c?

5. Why does a boat actually float at a given level? Write an explanation invoking the concepts of hydrostatic pressure, gravity, and distributed forces. It would probably be helpful to make a few cross-sectional sketches of the boat: maybe one of the boat that is not at equilibrium, and one where the boat is at equilibrium. Your sketches should include a bunch of small arrows that indicate the distributed forces acting on the boat, as well as any equivalent discrete forces (e.g., gravity) acting at the appropriate point(s) on the boat.

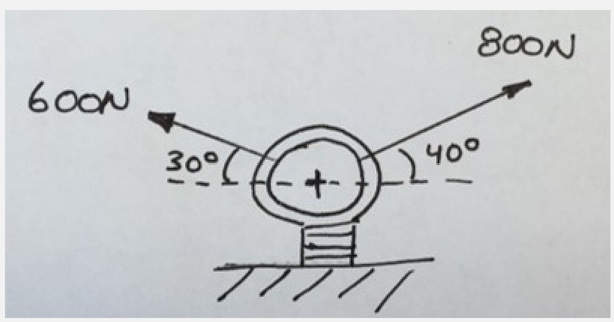
Exercise 22.25

Calculating Forces: The total force acting on an object is the sum of all the different force vectors acting on that object..

1. A man is pulling a rope running between points A and B with a force of 70 lbs. Describe the force of the rope on his hand as a vector in terms of components aligned with the x-y-z coordinate system shown.



2. Two forces are applied to an eye hook. They lie in the plane of the page with the orientation and magnitudes as shown. Determine the orientation and magnitude of the sum of these two forces acting on the hook.



Solution 22.1

1. The value of y at $t = 0$ is A.

2. For negative values of t the value of y tends to zero if $k > 0$ and it tends to $+\infty(A > 0)$ or $-\infty(A < 0)$ if $k < 0$. For positive values of t the value of y tends to zero if $k < 0$ and it tends to $+\infty(A > 0)$ or $-\infty(A < 0)$ if $k > 0$.

3.

4. The sign of the parameter A dictates whether the curve has positive or negative values of y – the curve also passes through the point $(0, A)$. The parameter k dictates whether the curve increases or decreases exponentially.

5. The parameter A could be the initial population at $t = 0$. The parameter k could be the exponential growth rate.

6. We seek the time T such that
$$2A = Ae^{kT} \tag{22.3}$$
A little algebra gives
$$e^{kT} = 2 \tag{22.4}$$
which has the solution
$$kT = ln(2) \tag{22.5}$$
With $k = 0.1$ we see that $T = 6.9$ years.

Solution 22.2

1. The value of y at $t = 0$ is $A/2$.

2. For negative values of t the value of y tends to zero if $k > 0$ and it tends to A if $k < 0$. For positive values of t the value of y tends to A if $k > 0$ and it tends to zero if $k < 0$.

3.

4. The parameter A changes the long-term behavior of the curve while the parameter k changes how quickly the curve tends to this value.

5. In terms of population growth, the parameter A is the maximum population that can be sustained, and is often known as the carrying capacity. The parameter k is the natural growth rate of the population.

6. We want to find the time T such that
$$0.99A = \frac{A}{1 + e^{-kT}} \tag{22.6}$$
A little algebra shows that
$$e^{-kT} = \frac{0.01}{0.99} \tag{22.7}$$

which has the solution

$$kT = 4.6 \tag{22.8}$$

With $k = 0.1$ we see that $T = 46$ years.

Solution 22.3

1.

2. Since a sin function returns values between 0 and 1, the parameter A controls the height of the function and is usually referred to as the amplitude. Since a sin function is periodic with a period of 2π , the period T of this function is determined by $\omega T = 2\pi$. Increasing ω decreases the period T , and ω is usually referred to as the angular frequency. Since a sin function is 0 when it's argument is 0, the parameter ϕ controls where it crosses the x-axis, and is usually referred to as the phase.

3. We seek the value(s) of T determined by

   $$2 = 3\sin(4T + 5) \tag{22.9}$$

   Re-arranging gives

   $$\sin(4T + 5) = \frac{2}{3} \tag{22.10}$$

   There are lots of solutions to this equation because the sin function sweeps periodically between -1 and +1. There are two solutions within $[0, 2\pi]$. The first one is obtained in MATLAB using

   ```
   >> asin(2/3)
   ```

   which returns 0.7297. There is another solution at $\pi - 0.7297 = 2.4119$. Due to the periodic nature of the sin function you can add or subtract integer multiples of 2π to these fundamental solutions. All solutions are therefore:

   $$4T + 5 = 0.7297 + 2\pi n \tag{22.11}$$
   $$4T + 5 = 2.4119 + 2\pi n \tag{22.12}$$

   Solving for T gives

   $$T = (0.7297 + 2\pi n - 5)/4 \tag{22.13}$$
   $$T = (2.4119 + 2\pi n - 5)/4 \tag{22.14}$$

Solution 22.4

1.

2. Graphing the vertex form reveals the effects of the parameters as follows. The vertex of the parabola is located at (h, k). The parabola opens upward if $g > 0$ and downward if $g < 0$. The parabola is narrow and steep for large positive values of g or large negative values of g. Changing h and k simply changes the location of the vertex.

3. Expanding the vertex form of the polynomial leads to $gx^2 - 2ghx + gh^2 + k$. Comparing to the standard form we see that $a = g$, $b = -2gh$, $c = gh^2 + k$. So although a has the same effect as g, the parameter b depends on g and h, and the parameter c depends on g, h, and k. This is why it is difficult to see the effect of the standard-form parameters on the curve.

Solution 22.5

The equation for a circle of radius R, centered at (a, b) is given by

$$(x-a)^2 + (y-b)^2 = R^2 \tag{22.17}$$

Solution 22.6

The parameters a and b determine the axes of the ellipse. The larger one is usually called the major axis and the smaller one is usually called the minor axis. This ellipse is oriented with its major and minor axes along the coordinate axes. Increasing a while holding b fixed results in a vertically-squished ellipse and vice versa.

Solution 22.7

There are two curves that define the hyperbola. Notice that the curves cross the x-axis at $x = -a$ and $x = a$ respectively. The rest of each curve is unbounded, but is asymptotic to the straight lines $y = (b/a)x$ and $y = -(b/a)x$.

Solution 22.8

The type of conic section is determined by the value of $b^2 - 4ac$ as follows:

- If $b^2 - 4ac < 0$ the equation represents an ellipse. In addition, if $a = c$ and $b = 0$ the equation represents a circle.
- If $b^2 - 4ac = 0$ the equation represents a parabola.
- If $b^2 - 4ac > 0$ the equation represents a hyperbola.

Solution 22.9

Although there are lots of parametric equations that trace out an ellipse, the most common are closely related to those for a circle and take the form

$$x = a\cos u, y = b\sin u, u \in [0, 2\pi] \tag{22.24}$$

where a and b represent the ellipses major and minor axes. The ellipse is traced out as u changes from $0 to 2\pi$, but note that u does not represent the angle between the x-axis and a point on the ellipse - see the Wikipedia page on "Ellipse" for an explanation of this. To confirm that these are valid parametric equations for an ellipse we substitute them into the implicit equation for an ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = \cos^2 u + \sin^2 u - 1 = 0 \tag{22.25}$$

where again we have used the trigonometric identity $\cos^2(u) + \sin^2(u) = 1$.

Solution 22.10

If $b = 0$ we see that the curve is a circle of radius a, and the parameter u corresponds to the angle of rotation. As you increase b, the circle changes into a spiral which tends to the origin as $u \to \infty$—the larger the value of b the quicker the curve spirals into the origin.

Solution 22.11

If $b = 0$ the curve is a circle of radius a in the x y - plane, and u is the angle of rotation. For $b > 0$ the curve continues to rotate as before when viewed from "above", but its height increases linearly - the resulting curve is a helix. The separation between each rotation of the curve is given by $2\pi b$, which is commonly known as the pitch of the helix.

Solution 22.12

Assuming we have already packed the x-coordinates of the data into $\mathbf{x}$ and the y-coordinates of the data into $\mathbf{y}$ we need to define the matrix $\mathbf{A}$ and solve for a vector $\mathbf{p}$ of unknown parameters. In MATLAB we would use

```
>> A = [x.^2 x ones(4,1)]
>> p = A\y
```

We should find that $a = p(1) = 0.1910$, $b = p(2) = -0.2663$ and $c = p(3) = 0.6784$. You should graph the parabola defined by these parameters and see how good the fit is.

Solution 22.13

Let's take slices through the surface along each of the coordinate axes.

1. If we choose $x = c$ then we obtain $z = c^2/a^2 + y^2/b^2$ which is a parabola in the (y, z)-plane that crosses the z-axis $(y = 0)$ at c^2/a^2.
2. If we choose $y = c$ then we obtain $z = x^2/a^2 + c^2/b^2$ which is a parabola in the (x, z)-plane that crosses the z-axis $(x = 0)$ at c^2/b^2.
3. If we choose $z = c$ then we obtain $c = x^2/a^2 + y^2/b^2$ which is an ellipse in the (x,y)-plane. If we divide both sides by c we get the standard form for an ellipse $1 = x^2/(a\sqrt{c})^2 + y^2/(b\sqrt{c})^2$ so that the major and minor axes are $a\sqrt{c}$ and $b\sqrt{c}$ - increasing the value of c increases the axes of the ellipse.

Solution 22.18

Please watch the videos and make your own table based on personal experience with the content.

Solution 22.19

Dot product tells you how parallel two vectors are. Cross product tells you how perpendicular two vectors are, and what the plane defined by these vectors are.

Solution 22.20

1. $\vec{A} + \vec{B}$
   Vector addition is accomplished geometrically by drawing the vectors head-to tail. In this

case, putting the tail of vector B at the head of vector A (or vice versa) will result in a vector longer than either A or B and pointed up and to the left.

2. $\vec{A} - \vec{C}$
   Vector subtraction is accomplished geometrically by drawing the vectors head-to head. In this case, putting the head of vector C at the head of vector A will result in a vector pointed strongly to the left and only a little up.

3. $\vec{A} \cdot \vec{B}$
   The dot product is a scalar equal to the product of the magnitudes of the vectors times the cosine of the angle between them. In this case we know the magnitudes (2 and 1) but don't know the angle, though we could estimate it as maybe 60 degrees.

4. $(\vec{A} \times \vec{B}) \times \vec{C}$
   $(\vec{A} \times \vec{B})$ will give you a vector pointing into the page with magnitude $2 * \sin\theta$. Crossing this with C will result in a vector back in the plane of the page and pointing down with magnitude $6 * \sin\theta$.

5. $(\vec{A} \cdot \vec{B}) \times \vec{C}$
   This one is nonsense because the dot product results as a scalar, as you can't cross a scalar with a vector. (Like the joke: what do you get when you cross a mosquito with a mountain climber?? Nothing! You can't cross a vector with a scalar!)

Solution 22.21

1. $|\vec{A} + \vec{B}|$
   5

2. $\vec{A} \times \vec{C}$
   $-15\hat{k}$

3. $\vec{A} \cdot \vec{B}$
   -1

Solution 22.22

1. The vector $\hat{A}$, which has a length of 1, and points in the direction of $\vec{A}$.
   $\vec{A}/|\vec{A}|$

2. The vector $\hat{n}$, which has a length of 1, and is perpendicular to both $\vec{A}$ and $\vec{B}$.
   $\pm\vec{A} \times \vec{B}/|\vec{A} \times \vec{B}|$

Solution 22.23

Please watch the videos and make your own table based on personal experience with the content.

Solution 22.24

Force Concepts

1. "I am pushing a coffee cup across my desk. Since it is in motion, I know that the magnitude of the frictional force acting on the cup is given by $\mu_k N$, where μ_k is the coefficient of kinetic friction and N is the normal force the table exerts on the cup." True or false? Why?
   True

2. "I am pushing on a coffee cup that is sitting on my desk. Since it is not moving, I know that the magnitude of the frictional force acting on the cup is given by $\mu_s N$, where μ_s is the coefficient of static friction and N is the normal force the table exerts on the cup." True or false? Why? (Think carefully about this one!)
   False. The quantity $\mu_s N$ is the LIMIT of the static friction force. Static friction is a constraint force which takes whatever value it needs to take in order keep the object stationary. Once that value exceeds the limit, the object will move

3. If an object of mass m is sitting alone and stationary on a level surface, what is the magnitude of the normal force that the surface exerts on the object?
   mg

4. Now consider that we drop this object onto the surface. During the time that the object is decelerating to a stop when it comes into contact with the surface is the normal force exerted by the surface on the object greater than, less than, or equal to the force from part c?
   Greater than: the surface has to not only support the weight of the object, but exert the force necessary to decelerate the object as well. Think about placing a rock on your hand vs. dropping a rock on your hand.

5. Why does a boat actually float at a given level? Write an explanation invoking the concepts of hydro-static pressure, gravity, and distributed forces. It would probably be helpful to make a few cross-sectional sketches of the boat: maybe one of the boat that is not at equilibrium, and one where the boat is at equilibrium. Your sketches should include a bunch of small arrows that indicate the distributed forces acting on the boat, as well as any equivalent discrete forces (e.g., gravity) acting at the appropriate point(s) on the boat.
   When the boat is floating in equilibrium, there are two equal and opposite forces: gravity and buoyancy, the latter of which is caused by hydro-static pressure where the boat contacts water. When the boat is too far out of the water, the total distributed force from hydro-static pressure (i.e., buoyancy) is not as great as the force of gravity, causing the boat to sink down until the buoyancy force matches the gravitational force. We expect a drawing with hydro-static force arrows pointing normal to and into the boat hull where it contacts the water and a downward gravitational force arrow acting on the middle of the boat. (If you were feeling fancy, you could have also included a distributed force from the air pressure above the boat–a force we often neglect.

Solution 22.25

1. The unit vector that describes the direction of the rope (pointing away from the man's hand) is $(-10, 1, 12)/\sqrt{245}$. The force is in the same direction with magnitude 70 lb. So the force vector is $70 * (-10, 1, 12)/\sqrt{245}$ lb or $(-44.7, 4.47, 53.7)$ lb.

2. The horizontal component of the resultant force would be $800\cos 40 - 600\cos 30$ and the vertical component would be $800\sin 40 + 600\sin 30$. Magnitude can be found from the square root of the sum of the squares of the two component. Angle can be found by taking the arc-tangent of the ratio, but it is also acceptable to just leave it in components.

Chapter 23

Day 2: Curves and Intro to Free Body Diagrams

23.1 Schedule

- 0900-0930: Debrief
- 0930-1030: Mathematical Representation of a Curved Surface
- 1030-1045: Coffee
- 1045-1115: Vector Operations and Force Models
- 1115-1215: Torque
- 1215-1230: Review and Preview

23.2 Debrief [30 minutes]

Exercise 23.1

1. **Representation of Curves:** In the overnight you dealt with different ways of representing curves. Make a cheat sheet that represents these ways. Give examples of each, and indicate how the ways differ.
2. **Mathematical Functions and Parameters:** In the overnight you dealt with a variety of different types of functions: sinusoids, exponentials, polynomials, etc. Make a "cheat sheet" table of these different types of functions, with a column that shows a sketch of the function and a column that identifies the impact of the different parameters.

23.3 Mathematical Representation of a Curved Surface [1hr]

In the overnight assignment we learned about describing, visualizing, and working with curves and surfaces in different ways. In this activity, we are going to apply what we learned and develop a mathematical representation of the hull of a boat. The boat we are going to model is called the *Spray*.

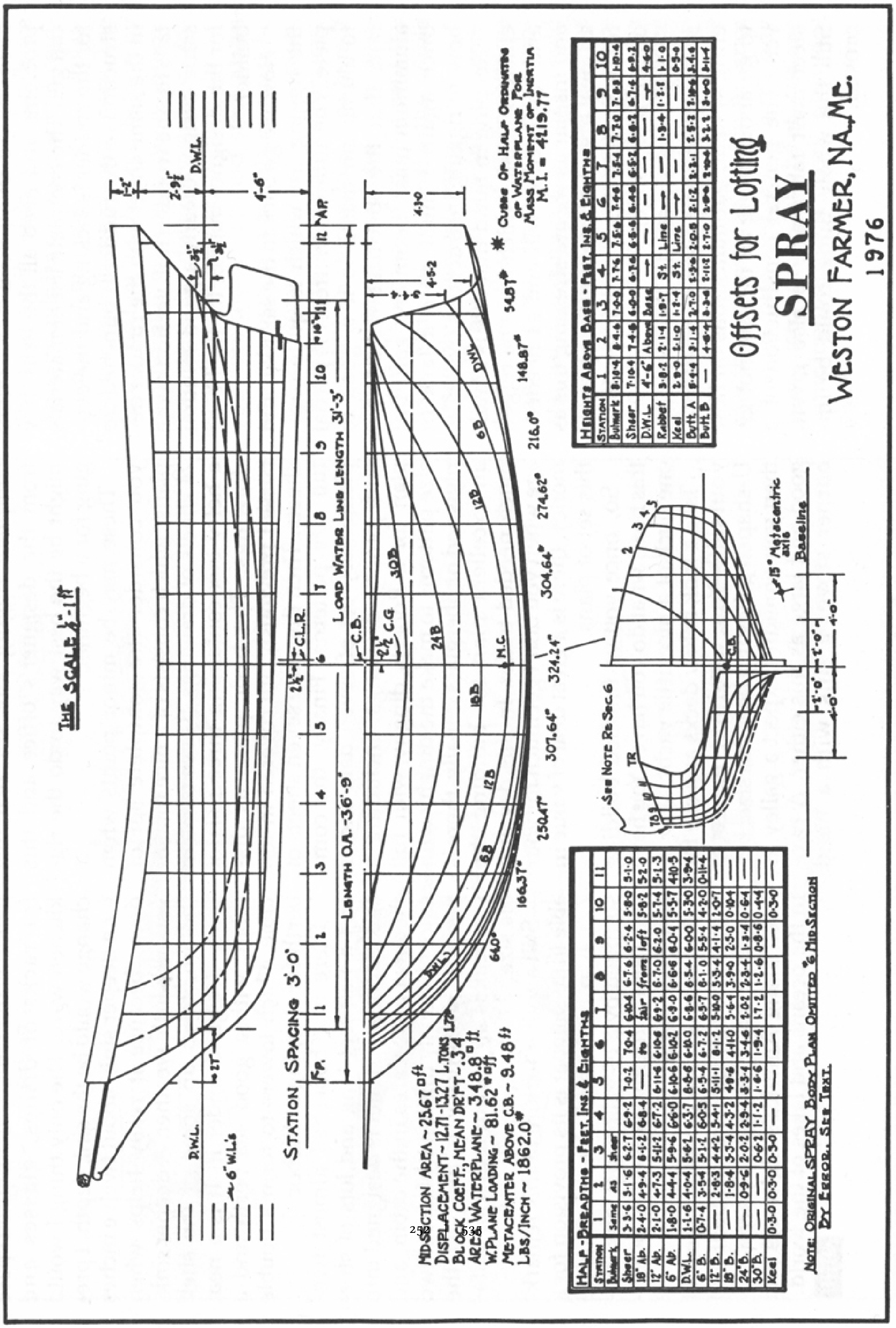
The *Spray* was used by Joshua Slocum in 1895 when he single-handedly sailed around the world. There has been much debate about the seaworthiness of the *Spray*. On the next page are the boat lines for the *Spray*. You'll have to take some time to understand these because there is a lot going on.

Once you think you've got it figured out, we'd like you to build up a representation of the hull by designing curves that are a good match for the *waterlines* and *sections*. You are going to do this by proposing particular functional forms and then finding the best-fit curve that captures the hull data.

Exercise 23.2

- Review the boat lines for the *Spray*. The *waterlines* are shown in the *plan view* in the central figure, and the *sections* are shown in the *section view* in the bottom figure. The *buttocks* are shown in the *profile view* in the top figure.
- Observe that data for the waterlines and the sections is presented in the table on the left: the waterlines are read from left to right, while the sections are read from bottom to top.

1. Propose a rectangular coordinate system (xyz) for the Spray, and discuss at least three options for where you might locate the origin.
2. Trace out the waterline called 18B on your lines plan and see how it falls in relationship to your axes and origin. Now propose a **quadratic** function to describe the waterline curve you have visualized. Can you **estimate** some of the function parameters that define your curve? Keep in mind that the data for the waterline and sections is in the table!
3. For your convenience, we have provided a MATLAB script that captures the data in the table on the *Spray* plans - it is called spray.mlx, and can be downloaded from CANVAS or this link.
4. Find the best-fit parameters for your quadratic curve using the data. Plot your best-fit curve and the original data points.
5. **If you have time** trace the section curve defined at station 2 in the section view and propose a **power** function of the form $y = x^a$ to describe it. Discuss how you might find a best-fit for this model.
6. **If you have time** discuss how you would find a best-fit surface to the entire hull.


THE SCALE ⅛"-1 ft
STATION SPACING 3'-0"
LENGTH O.A. -36'-9"
LOAD WATER LINE LENGTH 31'-3"
MIDSECTION AREA ~ 25.67 □ft
DISPLACEMENT ~ 12.71-13.27 L.TONS
BLOCK COEFF. MEAN DRFT. -.34
AREA WATERPLANE ~ 348.8 □ ft
W.PLANE LOADING ~ 81.62 #/□ft
METACENTER ABOVE C.B. ~ 9.48 ft
LBS/INCH ~ 1862.0#
CUBES OF HALF ORDINATES OF WATERPLANE FOR MASS MOMENT OF INERTIA M.I. = 4219.77
SEE NOTE RE SEC. 6
15° Metacentric axis
Baseline
HALF-BREADTHS - FEET, INS. & EIGHTHS
HEIGHTS ABOVE BASE - FEET, INS. & EIGHTHS
NOTE: ORIGINAL SPRAY BODY PLAN OMITTED ⅜ MID SECTION BY ERROR. SEE TEXT.
Offsets for Lofting
SPRAY
WESTON FARMER, NA, ME.
1976

23.4 Nectar of the Gods [15 minutes]

23.5 Vector Operations and Force Models [30 minutes]

During the overnight you watched several videos that covered topics in vector operations and types of forces. Some of these topics may have been new to you, and some may have been review, but they are all valuable for the upcoming material in the course.

Now that you are re-caffeinated, please work through the following exercises to debrief and synthesize the vector and forces material you encountered in the overnight.

Exercise 23.3

Vector Operations:

1. With your group, review the tables you made highlighting important topics in vector operations. Are there any concepts you are still confused about? If so, work with your table group to resolve these confusions.

2. In the overnight we asked you what the dot and cross product *tell* you about two vectors. At your table, discuss the answer to this question and draw a sketches to demonstrate the operations.

Exercise 23.4

May the Forces be With You

1. With your group, review the tables you made highlighting important topics modeling forces. Are there any concepts you are still confused about? If so, work with your table group to resolve these confusions.

2. The videos you watched identified several types of forces. With your table group, identify these types and draw an associated picture. On your picture, represent the force as a vector(s) and label it appropriately.

3. For the forces identified in the previous question, is there an equation that is used to model the force? If so, write it next to your drawing. What are the unknowns in your equations? Are there simple experiments you can do to characterize the forces?

4. Following the videos, we asked you a series of conceptual questions about forces. Please discuss those with your table group and resolve any questions.

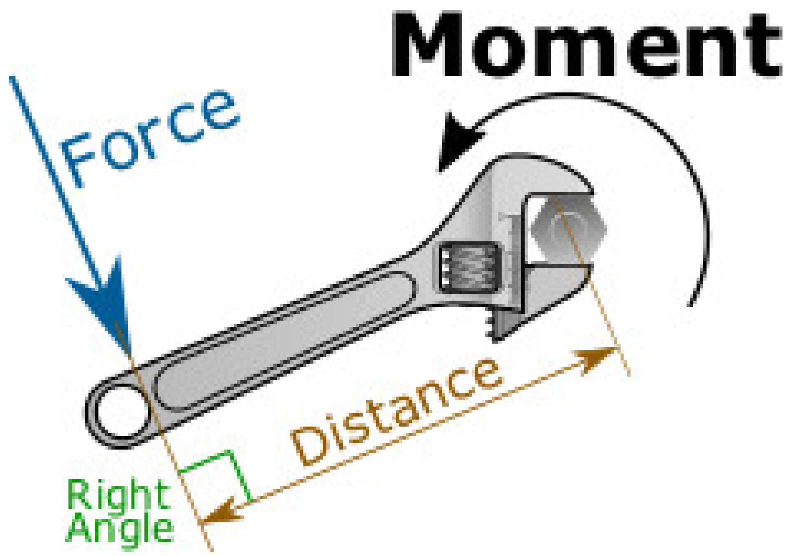


Figure 23.1: The perpendicular force acting on the wrench at some distance from the pivot point creates a torque. This "turning force" can also be represented as a moment at the pivot point.

23.6 Introduction to Torque [25 minutes]

During the previous overnight, and earlier in class today, we dove deep into the concept of forces and the various models of forces we use in engineering practice.

Closely related to forces is the concept of a *torque* (denoted by the vector $\boldsymbol{\tau}$ or $\vec{\tau}$). Just as a *force* changes *linear momentum*, a *torque* changes *angular momentum*. So, what does this mean in simple terms? We can think about a *torque* in two ways:

1. the thing that causes *angular momentum* to change ($\boldsymbol{\tau} = \frac{d\boldsymbol{L}}{dt}$), or more colloquially,
2. the thing that causes rotation.

For a nice review of *angular momentum*, feel free to take a few minutes and check out this summary video.

A note about terminology: The terms *torque* and *moment* are often used interchangeably in engineering. In free body diagrams, moments are denoted using a curved arrow, like the example in Figure 23.1.

Torque: The Picture The most common way to think about a torque is a "force applied at a distance from the point of rotation." We can illustrate this concept using the picture in Figure 23.2.

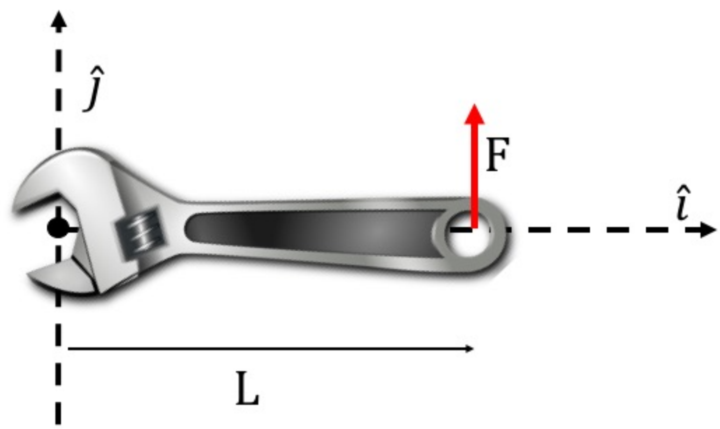


Figure 23.2: Force being applied to a wrench.

In Figure 23.2, a force $\vec{F}$ is being applied to the wrench at a distance $\vec{L}$ from the rotation point at the origin O (note that $\vec{F}$ and $\vec{L}$ are both vectors!). Because of the separation between the application point of the force and the origin, we expect this force would cause the wrench to rotate, increasing its *angular momentum*. While qualitatively this scenario is easy to understand, it is important to add mathematical definition to the scenario, then look at some additional cases.

Torque: Mathematical Definition The mathematical definition for a torque is given as:

$$\vec{\tau} = \vec{r} \times \vec{F}$$

which relies on the *cross product* that you encountered in the overnight. Looking at the equation for a torque, there are several important characteristics we notice immediately:

1. A torque, $\vec{\tau}$ is a vector, meaning that it has both a magnitude and a direction.
2. Both $\vec{r}$ and $\vec{F}$ are vectors, so both magnitude and direction matter.
3. Torque is dependent on the *cross product* between $\vec{r}$ and $\vec{F}$, so it is maximized when $\vec{r}$ and $\vec{F}$ are perpendicular.

Now that we have the definition of torque, let's work through a few examples to illustrate the characteristics above.

The ideas we will be exploring in these exercises are also in the videos about the definition of torque and choice of origin if you would like to review after class.

Exercise 23.5

Consider the wrench shown in Figure 23.3 that rotates about the origin O.

1. For each force vector in Figure 23.3, find the associated torque about the origin O. Make sure to specify both the direction and magnitude of the torque.
2. What change would need to be made to F_B in order for $\vec{\tau}_B = \vec{\tau}_A$ (where $\vec{\tau}_B$ is the torque due to F_B annd $\vec{\tau}_A$ is the troque due to F_A)?
3. If point of rotation for the wrench was changed from the origin to the point $(L, 0)$, how would that change the torque due to each force?

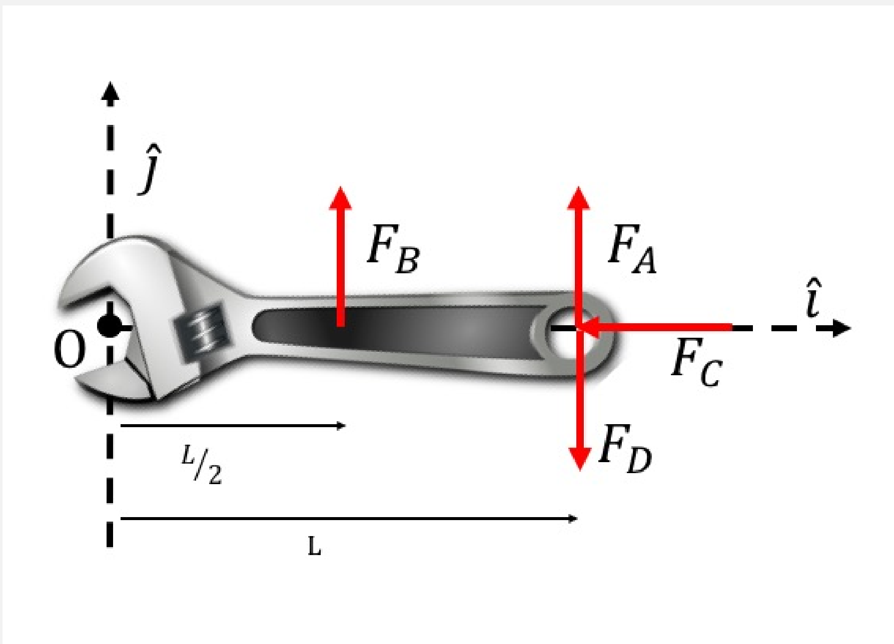


Figure 23.3: Wrench under different force conditions.

Exercise 23.6

Consider the wrench again, but with the force $F = 2\hat{\imath} + 1\hat{\jmath}$ as shown in Figure 23.4.

1. Find the torque about the origin O due to the force $\vec{F}$.

2. Find the torque about the point $(L, 0)$.

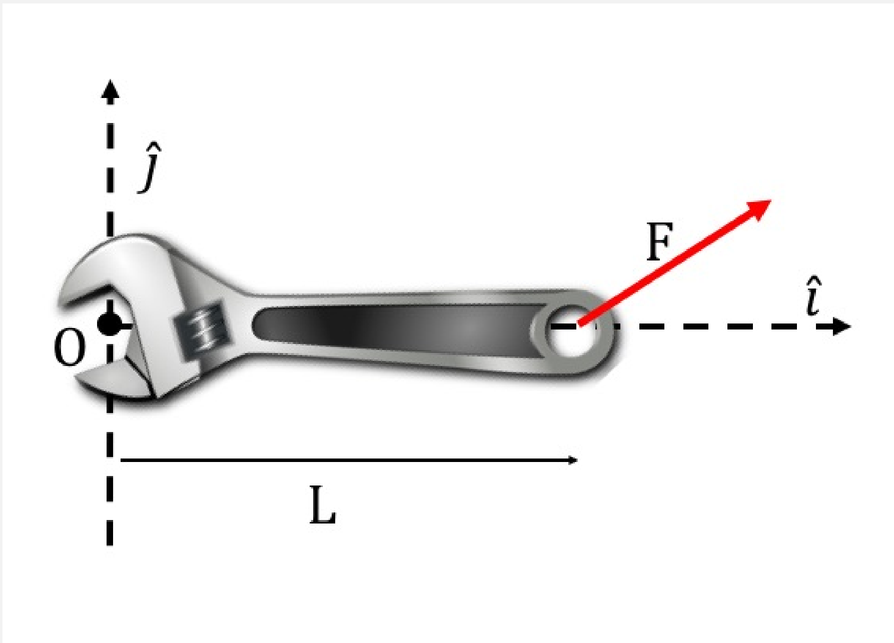


Figure 23.4: Wrench with a non-perpendicular applied force.

23.7 *Net Force and Net Torque [15 minutes]*

For these next exercises, we want to consider the concepts of *net force* and *net torque* acting on a body, which we define as:

$$\vec{F}_{net} = \sum F = m\vec{a}$$

$$\vec{\tau}_{net} = \sum \tau = I\vec{\alpha}$$

where $\vec{a}$ and $\vec{\alpha}$ are translational and rotational accelerations of the body, respectively, and m and I are mass and rotational moment of inertia of the body, respectively (note that $\vec{\tau}_{net} = \sum \tau = I\vec{\alpha}$ applies to the case of pure rotation or rotation about the center of mass, and there is a lot of interesting underlying angular momentum concepts. Talk to Mark with questions!) .

The ideas we will be exploring in these exercises are also in the video here if you would like to review after class.

Exercise 23.7

Consider the wrench in Figure 23.5. This figure is a snapshot of a moment in time (t=0), right when we apply the force, F.

1. What is the net force acting on the wrench?
2. What is the net torque acting on the wrench?
3. What motion would you expect to see in this scenario at t>0 seconds (assuming a friction-less surface)?

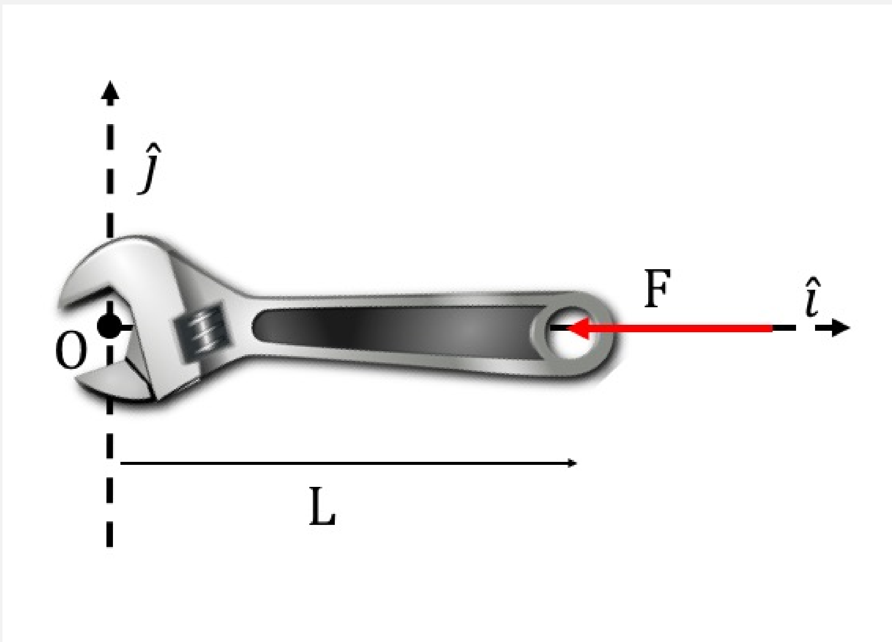


Figure 23.5: Yep, the wrench again.

Exercise 23.8

Consider the pencil in Figure 23.6. Let's say we are looking down on this pencil as it lies on a friction-less surface, at the initial moment (t=0) of applying the forces shown.

1. What is the net force acting on the pencil?

2. What is the net torque acting on the pencil?
3. What motion would you expect to see in this scenario (assuming a friction-less surface) at time >0 seconds?

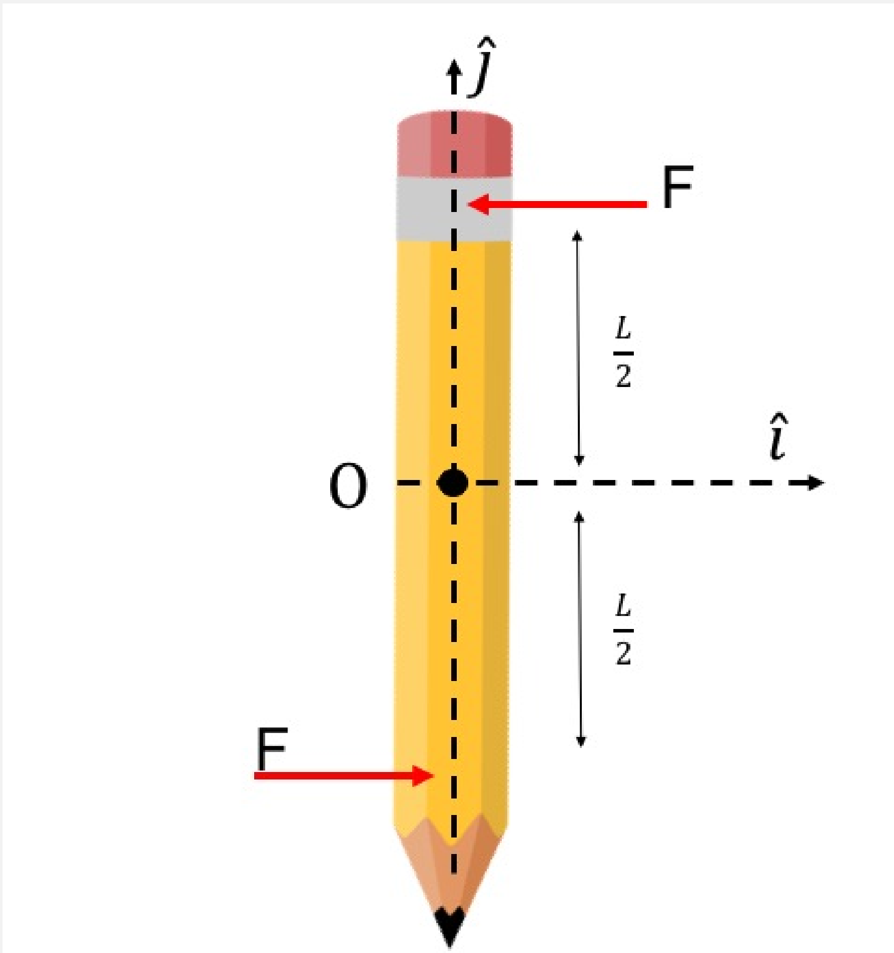
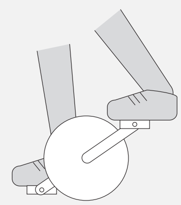


Figure 23.6: Oh hey, a pencil!

23.8 *Torque Practice [20 minutes]*

Exercise 23.9

Imagine that you are riding a bicycle (without cleats or toe clips so you are only able to push down on the pedals). Assuming you push down with constant force as you pedal, at what point during a pedaling cycle is the torque applied by your feet around the centerpoint of the crank greatest? When is it least? What is the direction of the torque?



Exercise 23.10

What might a wrench designed to help you exert a 100 N-m torque look like? About how long should the handle be and why?

Exercise 23.11

Let's say I'm trying to loosen a very stuck nut which is in a difficult to reach location. I am using a wrench which is 20 cm long. I am capable of exerting a force of 100 N, but because of the awkward position, the direction of the force I exert is at an angle of 60 degrees to the wrench (or 30 degrees off of the perpendicular to the wrench). If the nut will require a torque of 16 N-m to come loose, will I succeed?

Exercise 23.12

A force with vector representation $\vec{F} = 3\hat{\imath} + 2\hat{\jmath}$ is acting at a location $\vec{r} = 1\hat{\jmath} + 5\hat{k}$. What is the torque due to this force about the origin? What is the torque due to this force about the point $\vec{r}_0 = 7\hat{\imath} + 3\hat{k}$?

You might need to review the process of computing the cross-product of two vectors to complete this exercise. Consider the Rule of Sarrus for the procedure, or the physical interpretation.

Solution 23.5

1. For each force vector in Figure 23.3, find the associated torque about the origin O. Make sure to specify both the direction and magnitude of the torque.

$$\vec{\tau}_A = L\hat{i} \times F_A\hat{j} = LF_A\hat{k}$$
$$\vec{\tau}_B = \frac{L}{2}\hat{i} \times F_B\hat{j} = \frac{L}{2}F_B\hat{k}$$
$$\vec{\tau}_C = L\hat{i} \times -F_C\hat{i} = 0$$
$$\vec{\tau}_D = L\hat{i} \times -F_D\hat{j} = -LF_D\hat{k}$$

2. What change would need to be made to F_B in order for $\vec{\tau}_B = \vec{\tau}_A$? The magintude of F_B would need to double. Therefore, when you use a crescent wrench, you will need to apply less force if you grip it near the near the end of the handle. By the way, in real life, if you're exerting the forces F_A or F_B, you will want to flip the wrench 180 degrees about its horizontal axis so that the largest force is on the stationary jaw.

3. If point of rotation for the wrench was changed from the origin to the point $(L, 0)$, how would that change the torque due to each force?

$$\vec{\tau}_A = 0\hat{i} \times F_A\hat{j} = 0$$
$$\vec{\tau}_B = -\frac{L}{2}\hat{i} \times F_B\hat{j} = -\frac{L}{2}F_B\hat{k}$$
$$\vec{\tau}_C = 0\hat{i} \times -F_C\hat{i} = 0$$
$$\vec{\tau}_D = 0\hat{i} \times -F_D\hat{j} = 0$$

Solution 23.6

Consider the wrench again, but with the force $F = 2\hat{i} + 1\hat{j}$ as shown in Figure 23.4.

1. Find the torque about the origin O due to the force $\vec{F}$.

$$\vec{\tau} = L\hat{i} \times 2\hat{i} + 1\hat{j} = L\hat{k}$$

2. Find the torque about the point $(0, L)$. The torque about $(0, L)$ would be zero because $\vec{r}$=0

Solution 23.7

Consider the wrench in Figure 23.5.

1. What is the net force acting on the wrench? $\vec{F}_{net} = -F\hat{i}$

2. What is the net torque acting on the wrench? $\vec{\tau}_{net} = 0\hat{j} \times -F\hat{i} = 0$

3. What motion would you expect to see in this scenario (assuming a friction-less surface)? Translation in the $-\hat{i}$ direction. Specifically, we would expect a translational acceleration, since $\vec{F}_{net} = -F\hat{i} = m\vec{a}$.

Solution 23.8

Consider the pencil in Figure 23.6.

1. What is the net force acting on the pencil? $\vec{F}_{net} = \sum F_x = F\hat{i} - F\hat{i} = 0$. No forces are acting in the y-direction.

2. What is the net torque acting on the wrench?

$$\vec{\tau}_{net} = \frac{L}{2}\hat{j} \times -F\hat{i} + -\frac{L}{2}\hat{j} \times F\hat{i} = \frac{L}{2}F\hat{k} + \frac{L}{2}F\hat{k} = FL\hat{k}$$

3. What motion would you expect to see in this scenario (assuming a friction-less surface)? Counter-clockwise (positive) rotation about the center of the pencil. Specifically, we would expect an angular acceleration in the counter-clockwise direction, because $\vec{\tau}_{net} = I\vec{\alpha}$.

Solution 23.9

The maximum torque will be applied when the pedals are horizontal and the force applied by your foot is perpendicular to the crankset. The minimum torque will be applied when the pedals are aligned vertically, and the force from your feet is applied parallel to the crankset. At all points in the pedalling cycle, the torque will be oriented perpendicular to the pedaling plane (and toward your left as you sit on your bike).

Solution 23.10

What might a wrench designed to help you exert a 100 N-m torque look like? About how long should the handle be and why?
Most people can lift a 20 lb (10 kg-ish) dumbbell in one hand, so they can probably exert $F = mg \approx 10\text{ kg} \times 10\text{ m}^2/\text{s} = 100\text{ N}$ of force and they could do so perpendicular to the wrench. So, to make a wrench that could exert 100 N-m of torque, you probably want about a 1 m long handle.

Solution 23.11

Let's say I'm trying to loosen a very stuck nut which is in a difficult to reach location. I am using a wrench which is 20 cm long. I am capable of exerting a force of 100 N, but because of the awkward position, the direction of the force I exert is at an angle of 60 degrees to the wrench (or 30 degrees off of the perpendicular to the wrench). If the nut will require a torque of 16 N-m to come loose, will I succeed?
Yup! If you push at the very end of the handle (0.2 m), then your applied torque is $0.2\text{ m} \times 100\text{ N} \times \sin 60°$ = 17.3 N-m, which exceeds the required torque.

Solution 23.12

A force with vector representation $\vec{F} = 3\hat{i} + 2\hat{j}$ is acting at a location $\vec{r} = 1\hat{j} + 5\hat{k}$. What is the torque due to this force about the origin? What is the torque due to this force about the point $\vec{r}_0 = 7\hat{i} + 3\hat{k}$?
$\vec{r} \times \vec{F}$ gives you $-10\hat{i} + 15\hat{j} - 3\hat{k}$, $(\vec{r} - \vec{r}_0) \times \vec{F}$ gives you $-4\hat{i} + 6\hat{j} - 18\hat{k}$

Chapter 24

Night 2: Static Equilibrium and Free Body Diagrams

Overview

This assignment is focused on important concepts that enable the analysis of the forces and torques acting on stationary (static) and moving (dynamic) objects or groups of objects. You may have seen some of this material in your high school calculus and physics courses, but if not, that is ok! If anything here is really new to you, please be sure to ask for help and attend the Wednesday review session.

Learning Objectives

Concepts

- Use vector notation(s) to represent forces.
- Compute resulting vectors from given vectors using addition, subtraction, and cross-products.
- Draw a free-body diagram from a description of its physical situation to include: reaction forces, frictional forces and turning moments.
- Use a vector to account for frictional forces acting on a body.
- Compute net force and/or torque acting on a body from multiple known forces and/or torques.
- Resolve a force acting on an object into its components in a coordinate system rotated an angle of θ from the force.
- Use linear algebra to express forces in one coordinate system as their values in a coordinate system rotated an angle of θ relative to the original system.

24.1 Introduction to Free Body Diagrams

A foundational skill in engineering practice is the ability to distill complex scenarios into tractable problems that can be solved using the tools in your quantitative analysis toolbox. A primary component of doing this is drawing *free body diagrams (FBDs)*. A *free body diagram* is a representation of an isolated element from your system with indications of all forces and torques acting on that element. As with most modeling work, you need to make choices in representing the system: what forces will you include, and what forces will you ignore (usually because they are so small as to have little effect)? Will you think about distributed forces or equivalent forces? In the previous overnight and in class assignments you reviewed forces and torques, and their models, and now we will work to connect these concepts to physical systems.

Intro Video and Notes

Exercise 24.1

Video Review: Visit this link. Watch the video about Free Body Diagrams titled "concept and overview example." After watching the video, create a table of key concepts. Include simple sketches where appropriate to illustrate these concepts.

Static vs. Dynamic Systems

When drawing a FBD, one of the first questions that needs to be considered is "is this a static or dynamic system?" To answer this question, we need to understand what it means for a system to be in *static equilibrium*.

Exercise 24.2

Video Review: Visit this link. Watch the video about equilibrium titled "A Statics Example." After watching the video, create a table of key concepts. Include simple sketches where appropriate to illustrate these concepts.

Static Equilibrium A system is said to be in *static equilibrium* when the net force and net torque on every part of the system is zero. The net forces equaling zero is known as the *first condition for equilibrium*, and the net torque equaling zero is known as the *second condition for equilibrium*.

The two conditions for equilibrium lead to the following basic equations (where we use bold to represent a vector):

$$\sum \boldsymbol{F} = 0$$
$$\sum \boldsymbol{\tau} = 0$$

Exercise 24.3

Static Equilibrium Conceptual Questions

1. What do the two conditions for static equilibrium say about the linear and angular acceleration of the system?
2. For a system to be in static equilibrium, does it need to be at rest? Why or why not?
3. For a two-dimensional (x,y) system, how many equations are needed to prove static equilibrium? Please write them. How about for a three-dimensional (x,y,z) system?
4. For a two-dimensional (x,y) system in static equilibrium, what is the maximum number of unknown forces/torques that can be solved for? How do you know this?

Drawing FBDs

Having a consistent methodology when drawing free body diagrams can be really helpful when analyzing systems. Below are some questions to consider:

Questions to Consider Before the FBD

- How will the system behave? (i.e., Play the movie in your head. Is this a static or dynamic system?)
- What are the aspects of the system you care about?
- Should you divide your system into multiple components or subsystems? Keep in mind that sometimes you need to draw multiple free body diagrams in order to properly analyze a situation and identify interaction forces and torques, etc.
- What interactions do you need to capture in your analysis? (i.e., what are the forces and/or torques acting on the various parts of your system? Are these forces/torques captured in your free-body diagrams?)
- How can you capture those interactions on a diagram or a set of diagrams? (i.e., what does the free-body diagram look like for this system and its components?)
- What are the appropriate frames of references to analyze this system?

How you choose to visualize your system and its components can impact your ability to understand and ultimately complete your analysis. A clear and complete *free body diagram* (or **multiple** *free body diagrams*) can help you clarify your own understanding of the system, derive appropriate equations of motion (more on this later in the course), quickly evaluate the accuracy of your work, and get help from others (because they can see how you are understanding the system!).

Steps for Drawing a FBD As you begin to create free body diagrams, here are some suggested steps to consider following:

- Make a decision as to which system (a body or collection of bodies) you will be analyzing. You may identify the need to break your system into multiple subsystems.
- Choose a body or combination of bodies **isolated** from all surrounding bodies.
- Observe which forces and/or torques are exerted **on** the body or combination of bodies you have chosen.
- Label forces and torques, with appropriate directions. (NOTE: You may need to arbitrarily assign a direction at times, in which case you may subsequently find through analysis that the magnitude of the force in the direction you have chosen is negative, which is fine! Forces do not have to be positive, but it’s useful to draw them in the positive direction when possible.)
- Choose and label reference frames.

Exercise 24.4

Consider the *pinned beam* with constant cross section and mass distribution shown in Figure 24.1 below. The beam is supported at one end by a *pinned joint*. This particular type of joint can provide linear support for reaction forces in any direction, but cannot support a turning moment, since it rotates about the pin (If interested, you can learn more about types of load supports here. At the other end of the beam is a rope which can only support tension along its axis, attached to a fixed point. For this exercise, you can assume the beam is stationary ($\boldsymbol{a} = \boldsymbol{\alpha} = 0$, $\boldsymbol{v} = \dot{\boldsymbol{\theta}} = 0$).

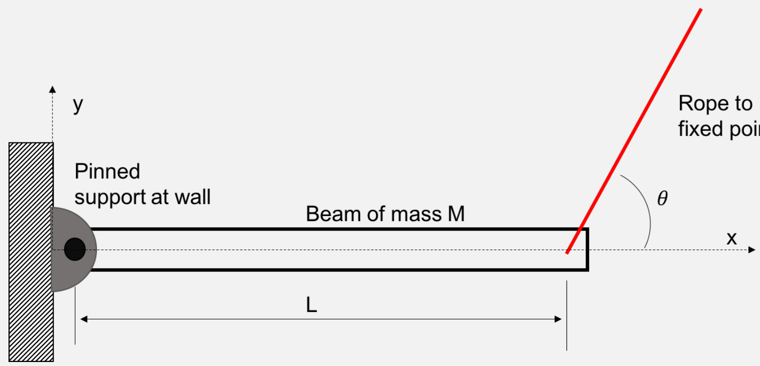


Figure 24.1: A pinned beam supported by a rope.

1. When drawing a free body diagram for the pinned beam in Figure 24.1, which body should you isolate?
2. Draw the FBD for the body you have isolated, paying special attention to the steps outlined above.
3. Is the beam in static equilibrium? How do you know?
4. Write the appropriate equations for find the unknown forces and torques acting on the beam. How many equations are there?
5. How many unknowns do you have in your system of equations?
6. Is the pinned beam system *statically determinant*? Why or why not?

Exercise 24.5

Consider the *cantilevered beam* with constant cross section and mass distribution shown in Figure 24.3 below. The beam has mass M, and is supported at one end by a *fixed joint*, which can provide linear support (or reaction) forces, and moments. For this exercise, you can assume the beam is stationary ($\boldsymbol{a} = \boldsymbol{\alpha} = 0$, $\boldsymbol{v} = \dot{\boldsymbol{\theta}} = 0$).

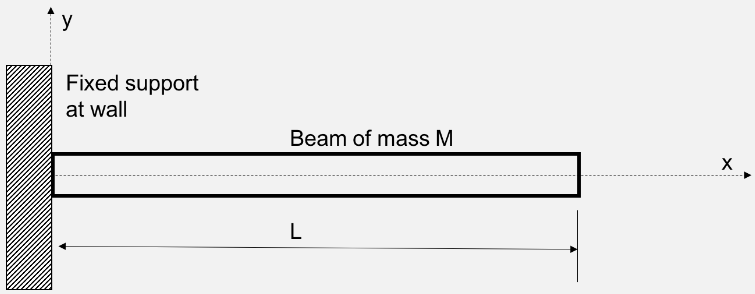


Figure 24.3: A cantilevered beam of mass M.

1. When drawing a free body diagram for the cantilevered beam in Figure 24.3, which body should you isolate?
2. Draw the FBD for the body you have isolated, paying special attention to the steps outlined above.
3. Is the beam in static equilibrium? How do you know?
4. Write the appropriate equations for find the unknown forces and torques acting on the beam. How many equations are there?
5. How many unknowns do you have in your system of equations?
6. Is cantilevered beam system *statically determinant*? Why or why not?

Exercise 24.6

You may have seen this coming, but let's consider the cantilevered beam, of constant cross section and mass distribution, with a rope, shown in Figure 24.5.

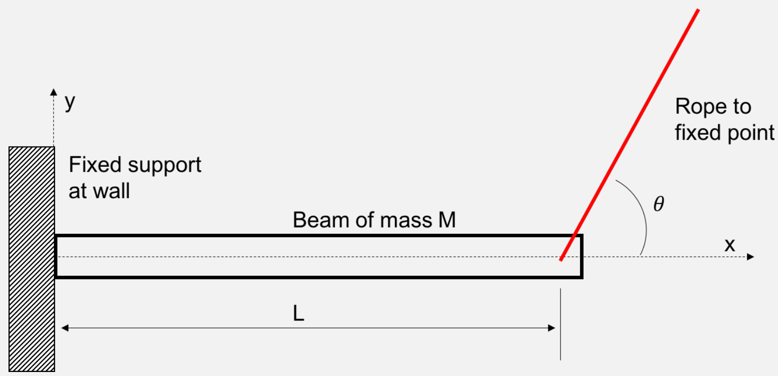


Figure 24.5: A cantilevered beam of mass M with a rope attached to a fixed point.

1. Draw the FBD for the system in Figure 24.5, paying special attention to the steps outlined above.
2. Write the appropriate equations for find the unknown forces and torques acting on the beam. How many equations are there?
3. How many unknowns do you have in your system of equations?
4. Is the system *statically determinant*? Why or why not?

24.2 *Additional FBD Practice*

Exercise 24.7

Qualitative FBDs

1. Consider a sprinter who is running (and accelerating) over a level surface. Draw a free body diagram for the person (assume one of the person's feet is in contact with the ground). Use appropriately directed arrows, in appropriate places, and of appropriate sizes to indicate the forces acting on the person. Make sure to label the arrows appropriately. What force is propelling the runner forward?
2. Consider a hot air balloon carrying a basket with two riders. It is floating at a given altitude. Draw free body diagrams for the following:

   a) Draw a FBD for the whole system (balloon + basket) using equivalent forces to represent all distributed (body and contact) forces (for example, consider the buoyancy of the balloon to be a single force acting on the center of buoyancy).

b) Draw an FBD for the basket using equivalent forces to represent all distributed forces.

c) Draw a FBD for the balloon (not including the basket) using distributed forces.

d) Comment on the utility of each of these FBDs: what questions would each help you to answer?

24.3 Free Body Diagrams With Multiple Bodies

In the previous exercises you drew simple FBDs for systems with a single body (a beam). Next, we will consider systems with multiple subsystems.

Exercise 24.8

Video Review: Just a bit to watch here. Visit this link. Watch the video about Free Body Diagrams titled "System and subsystem boundaries", and make a table of key concepts.

What is a Reference Frame?

A *reference frame* is an explicitly defined framework in which we can make observations and write physical laws. A reference frame is defined by a coordinate system, an observer, and some definition of time. For statics problems, the primary component we are interested in are relationships between different coordinate systems. For dynamic systems, however, time and the observer play a very important role. We will be diving deeper into the concept of reference frames in the Robotics module, and again in QEA 2.

Coordinate Systems: A coordinate system consists of an **origin** and a set of **basis vectors**. Recall that a set of vectors form a basis if they are linearly independent—if they are mutually orthogonal then we have an orthogonal coordinate system. The standard basis vectors for 2D are usually labelled $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$, but they are sometimes written as $\mathbf{e}_1$ and $\mathbf{e}_2$, or $\mathbf{e}_x$ and $\mathbf{e}_y$, or $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$.

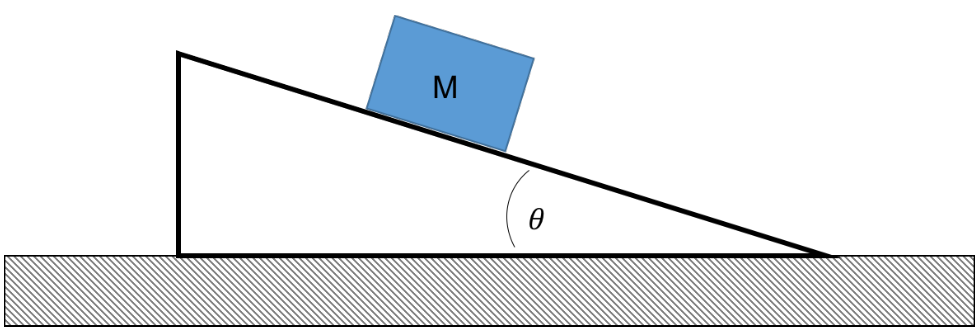


Figure 24.7: Box of mass M on an inclined ramp. The ramp is not fixed to the floor, and the box is not fixed to the ramp.

Choosing and Working With Multiple Frames

Exercise 24.9

For this exercise, refer to the system in Figure 24.7.

1. For the system in Figure 24.7, define two reference frames for the system using orthogonal unit vectors:
   a) One reference frame will be known as the "global frame" and defined by the unit vectors $[\hat{i}, \hat{j}]$. The $\hat{i}$ vector should be parallel to the floor, with the $\hat{j}$ vector perpendicular to the floor.
   b) The second reference frame will be known as the "ramp frame" and defined by the unit vectors $[\hat{x}, \hat{y}]$. The $\hat{x}$ vector should be parallel to the ramp, with the $\hat{y}$ vector perpendicular to the ramp.

Exercise 24.10

1. Draw a FBD for the box by isolating it from the ramp. Draw and label appropriate vectors for the forces acting on the box.
2. Draw a FBD for the ramp. Draw and label appropriate vectors for the forces.
3. Look carefully at the force vectors you have drawn for the box and ramp FBDs. What reference frame is each force defined in (e.g. for the box, gravity is likely in the global $\hat{j}$ direction, while the normal force is in the ramp $\hat{y}$ direction)? Make a table like the one below explicitly defining each force/frame.

Force	Acting On	Reference Frame	Direction (e.g. $+\hat{i}$)

Exercise 24.11

1. Write down an equation for the summation of all of the forces acting on the box. You should include the unit vector associated with each force (i.e. $\sum F = -F_g\hat{j} - F_{friction}\hat{x} + ...$).
2. Typically, for a 2D problem, we would split the force equation into two equations, $\sum F_{\hat{i}}$ and $\sum F_{\hat{j}}$ (or commonly $\sum F_x$ and $\sum F_y$). Can we do that here? What about the forces that are defined in the $[\hat{x}, \hat{y}]$ frame? Can forces in different frames be added?

Exercise 24.12

We need to represent all of our forces in a common reference frame in order to perform analysis. In the Faces Modules, you encountered the concept of *rotation matrices*, and we can apply that idea here to relate the global frame and the ramp frame.

1. Draw the unit vectors that define the global and ramp frame co-located at the same origin. Specify the angle θ that defines the rotation between the two coordinate systems.
2. Write equations for $\hat{x}$ and $\hat{y}$ in terms of $\hat{i}$ and $\hat{j}$.
3. Rewrite the equations above in the form of a *rotation matrix*.
4. Rewrite your force equations so all of the forces are expressed in the global reference frame $[\hat{i}, \hat{j}]$, then split each force equation into the $\hat{i}$ and $\hat{j}$ components (i.e. you should end up with four equations for $\sum F_{box\hat{i}}$, $\sum F_{box\hat{j}}$, $\sum F_{ramp\hat{i}}$, $\sum F_{ramp\hat{j}}$).

24.4 Revisiting Torque

Exercise 24.13

Video Review: Visit this link. Watch the video about Free Body Diagrams titled “Torque FBD Example Bike.” After watching the video, create a table of key concepts. Include simple sketches where appropriate to illustrate these concepts.

Exercise 24.14

The picture below shows a person trying to tighten a nut with a wrench. The point of view is that of standing, facing the end of the bolt as shown and turning clockwise. Draw a free body diagram for

1. The wrench only
2. The nut only

Then, using your FBDs, explain how the wrench is used to tighten the nut.

24.5 *Introduction to Center of Mass*

Exercise 24.15

We're going to deal with center of mass in our next class. Before you come to class, visit your friends at the Khan Academy to get an overview of the concept of center of mass, and identify important concepts. Once again, make a table of key concepts from this video.

Solution 24.3

Static Equilibrium Conceptual Questions

1. What do the two conditions for static equilibrium say about the linear and angular acceleration of the system?
   Applying Newton's second law, $\sum \boldsymbol{F} = m\boldsymbol{a}$ and $\sum \boldsymbol{\tau} = I\boldsymbol{\alpha}$ (for rotation of a rigid body about the center of mass) we know that the linear acceleration $\boldsymbol{a} = 0$ and the angular acceleration $\ddot{\boldsymbol{\theta}} = \boldsymbol{\alpha} = 0$.

2. For a system to be in static equilibrium, does it need to be at rest? Why or why not?
   No, for a system to be in static equilibrium, the net force and torques acting on the system must be zero, leading to linear acceleration $\boldsymbol{a} = 0$ and the angular acceleration $\ddot{\boldsymbol{\theta}} = \boldsymbol{\alpha} = 0$. This means that the system can be traveling at a constant angular or linear velocity while still satisfying the two conditions for static equilibrium.

3. For a two-dimensional (x,y) system, how many equations are needed to prove static equilibrium? Please write them. How about for a three-dimensional (x,y,z) system?
   For a two-dimensional (x,y) system, three equations are needed:

$$\sum F_x = 0$$

$$\sum F_y = 0$$

$$\sum \tau_z = 0$$

   For a three-dimensional (x,y,z) system, the number of equations increases to six:

$$\sum F_x = 0$$

$$\sum F_y = 0$$

$$\sum F_z = 0$$

$$\sum \tau_x = 0$$

$$\sum \tau_y = 0$$

$$\sum \tau_z = 0$$

4. For a two-dimensional (x,y) system in static equilibrium, what is the maximum number of unknown forces/torques that can be solved for? How do you know this? For a two-dimensional (x,y) system in static equilibrium, we know that there are three equations that must be satisfied as indicated above, all of which represent the net forces/torques in the respective directions,

$$\sum F_x = 0$$

$$\sum F_y = 0$$
$$\sum \tau_z = 0$$

If we have more unknowns than equations, we are not able to determine these unknowns. So, with only these three equations, we can solve a maximum of three unknowns. If the number of unknowns exceeds the number of equations, the system is know as “statically indeterminate,” and cannot be solved without additional information. If there are fewer unknowns than the number of equations, the system would be considered "overdetermined," i.e., not physically possible. When the number of equations equals the number of unknowns, our system is "determinate", which means "having exact and discernible limits."

Solution 24.4

1. When drawing a free body diagram for the pinned beam in Figure 24.1, which body should you isolate? For the pinned beam in Figure 24.1, you want to isolate just the beam portion of the drawing as shown below.

2. Draw the FBD for the body you have isolated, paying special attention to the steps outlined above. The FBD for the pinned beam is shown in 24.2. Notice that we've chosen to represent all of the weight as a point force, F_G, acting at the center of mass. This ends up being a simplifying shortcut that gives us an accurate result when the mass is equally distributed along the beam (i.e., the mass/length does not change). At the pin, we expect F_R as the reaction force, acting in some unknown direction. For our computational convenience, we define the x-component of F_R, acting along the x-axis, F_{Rx}, and F_{Ry} acting along the y-axis (these vectors are not necessarily drawn to scale in the figure). We are doing this in anticipation of summing the forces in the x and y directions. Incidentally, we are calling them ‘reaction’ forces as a nod to Newton's law that states, "For every action, there is an equal and opposite 're-' action." These forces are re-acting to the applied force labeled $\vec{T}$. (What happens to these re-acting forces when we remove $\vec{T}$)?

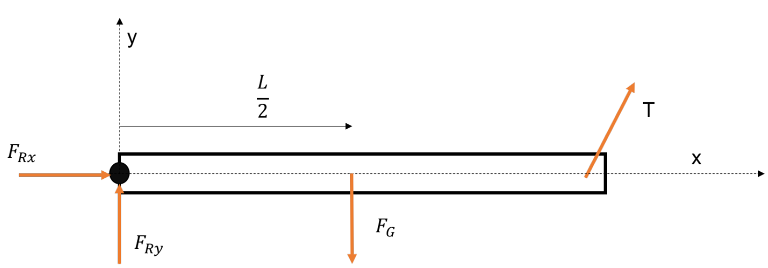


Figure 24.2: FBD for the pinned beam.

3. Is the beam in static equilibrium? How do you know? Yes, in the problem statement we were told for this exercise, you can assume the beam is stationary $\boldsymbol{a} = \boldsymbol{\alpha} = 0$, which satisfies the conditions for static equilibrium.

4. Write the appropriate equations for finding the unknown forces and torques acting on the beam. How many equations are there? There are three equations:

$$\sum F_x = 0 = F_{Rx} + T\cos\theta$$

$$\sum F_y = 0 = F_{Ry} - F_G + T\sin\theta$$

$$\sum \tau_z = 0 = -F_G\frac{L}{2} + T\sin\theta L$$

5. How many unknowns do you have in your system of equations? Three unknowns: T, F_{Rx}, and F_{Ry}.

6. Is the pinned beam system *statically determinant*? Why or why not? Yes, the system is statically determinant because the number of equations equals the number of unknowns.

Solution 24.5

1. When drawing a free body diagram for the pinned beam in Figure 24.1, which body should you isolate? For the cantilevered beam in Figure 24.3, you want to isolate just the beam portion of the drawing as shown below.

2. Draw the FBD for the body you have isolated, paying special attention to the steps outlined above. The FBD for the cantilevered beam is shown in 24.4. We've used the same treatment for the reaction force at the wall as we did for the pin in the above cantilever; its components, F_{Rx} and F_{Ry} are shown at the joint in the x and y directions, respectively. We have also drawn a circular arrow to represent a reactive turning moment, M_{Rz}, at the wall. We drew this on our FBD because we know by experience that the F_G produces a clockwise turning moment. Since the beam is not moving, this moment must be offset by an equal an opposite one at the wall connection.

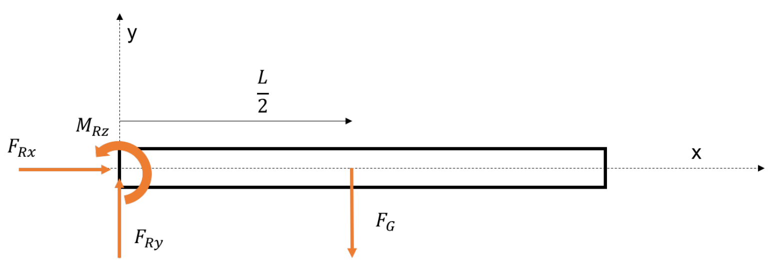


Figure 24.4: FBD for the cantilevered beam. Vectors are not necessarily drawn to scale.

3. Is the beam in static equilibrium? How do you know? Yes, in the problem statement we were told For this exercise, you can assume the beam is stationary $\boldsymbol{a} = \boldsymbol{\alpha} = 0$, which satisfies the conditions for static equilibrium.

4. Write the appropriate equations for find the unknown forces and torques acting on the beam. How many equations are there? There are three equations:

$$\sum F_x = 0 = F_{Rx}$$

$$\sum F_y = 0 = F_{Ry} - F_G$$

$$\sum \tau_z = 0 = -F_G \frac{L}{2} + M_{Rz}$$

5. How many unknowns do you have in your system of equations? Three unknowns: M_{Rz}, F_{Rx}, and F_{Ry}.

6. Is the cantilevered beam system *statically determinant*? Why or why not? Yes, the system is statically determinant because the number of equations equals the number of unknowns.

Solution 24.6

1. Draw the FBD for the system in Figure 24.5, paying special attention to the steps outlined above.

   The FBD for the cantilvered beam with a rope is shown in 24.6. In the diagram, F_{Rx} and F_{Ry} are the reaction forces from the fixed joint in the x and y directions respectively, and M_{Rz} is the reaction moment.

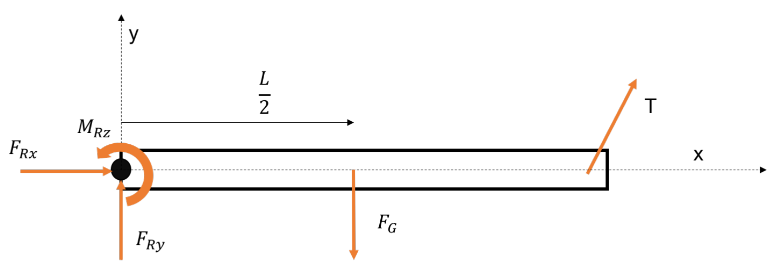


Figure 24.6: FBD for the cantilevered beam with a rope.

2. Write the appropriate equations for find the unknown forces and torques acting on the beam. How many equations are there? There are three equations:

$$\sum F_x = 0 = F_{Rx} + T\cos\theta$$

$$\sum F_y = 0 = F_{Ry} - F_G + T\sin\theta$$

$$\sum \tau_z = 0 = -F_G \frac{L}{2} + T\sin\theta L + M_{Rz}$$

3. How many unknowns do you have in your system of equations? Four unknowns: M_{Rz}, F_{Rx}, F_{Ry}, and T.

4. Is the system *statically determinant*? Why or why not? No, the system is statically indeterminant because the number of equations is less than the number of unknowns.

Solution 24.7

Please see diagrams at the end of the page.
2d) Comment on the utility of each of these FBDs: what questions would each help you to answer? The full system FBD (part a) could help you solve the necessary buoyant force to maintain a constant elevation (static equilibrium) or rise/fall. With this knowledge, the size of the balloon can be chosen. The basket diagram (part b) can help spec the cable attachments between the balloon and basket to prevent failure. The final diagram (part c) can help spec cables, or could help choose the material properties of the balloon itself to prevent failure of the canopy material.

Solution 24.9

The global and ramp reference frames are shown below. When focusing on forces, the position of the origin of each frame is not particularly important, but if we were concerned with the motion of the ramp and the box, we would want to think carefully about the position of each coordinate system (e.g. we may have three frames, with a global fixed coordinate system, and a coordinate systems with an origin fixed to the ramp and box respectively).

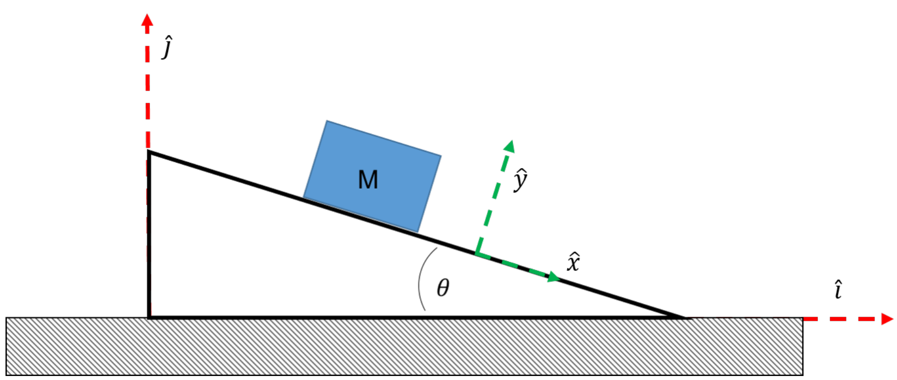


Figure 24.8: Global and ramp reference frames defined for the box-on-a-ramp.

Solution 24.10

1. Draw a FBD for the box by isolating it from the ramp. Draw and label appropriate vectors for the forces acting on the box.

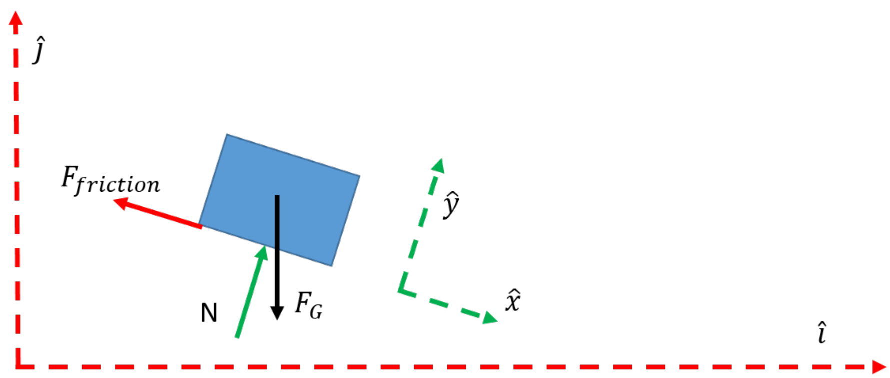


Figure 24.9: FBD for the isolated box.

2. Draw a FBD for the ramp (ramp mass=M_R). Draw and label appropriate vectors for the forces.

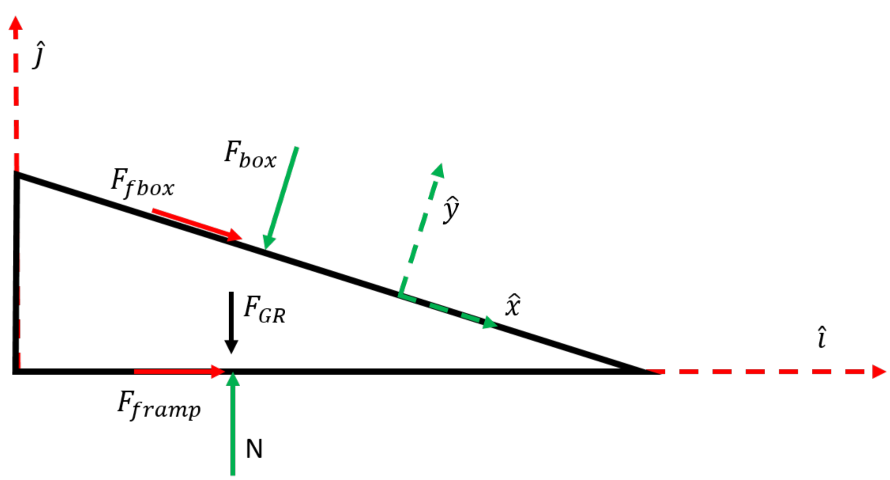


Figure 24.10: FBD for the isolated ramp.

3. Look carefully at the force vectors you have drawn for the box and ramp FBDs. What reference frame is each force defined in (e.g. for the box, gravity is likely in the global $\hat{j}$ direction, while the normal force is in the ramp $\hat{y}$ direction)? Make a table like the one below explicitly defining each force/frame.

Force	Acting On	Reference Frame	Direction
Gravitational force (weight of ramp)	Ramp	Global Frame	$-\hat{j}$
Floor normal force	Ramp	Global Frame	$+\hat{j}$
Floor frictional force	Ramp	Global Frame	$+\hat{i}$
Normal force from box to ramp	Ramp	Ramp Frame	$-\hat{y}$
Friction force from box to ramp	Ramp	Ramp Frame	$+\hat{x}$
Friction force from ramp to box	Box	Ramp Frame	$-\hat{x}$
Normal force on box from ramp	Box	Ramp Frame	$+\hat{y}$
Gravitational force (weight of box)	Box	Global Frame	$-\hat{j}$

Solution 24.11

1. Write down an equation each for the summation of all of the forces acting on the box and ramp. You should include the unit vector associated with each force (i.e. $\sum F_{box} = -F_g\hat{j} - F_{friction}\hat{x} + ...$).

$$\sum F_{box} = -F_g\hat{j} - F_{friction}\hat{x} + N\hat{y}$$

$$\sum F_{ramp} = -F_{GR}\hat{j} + N\hat{j} + F_{framp}\hat{i} + F_{fbox}\hat{x} - F_{box}\hat{y}$$

2. Typically, for a 2D problem, we would split the force equation into two equations, $\sum F_{\hat{i}}$ and $\sum F_{\hat{j}}$ (or commonly $\sum F_x$ and $\sum F_y$). Can we do that here? What about the forces that are defined in the $[\hat{x}, \hat{y}]$ frame? Can forces in different framed be added?

   Yes, we can split these equations into two separate equations, but not before we represent all of the forces in the same reference frame. Forces in different reference frames do not add, so we must choose one frame for the whole system.

Solution 24.12

We need to represent all of our forces in a common reference frame in order to perform analysis. In the Faces Modules, you encountered the concept of *rotation matrices*, and we can apply that idea here to relate the global frame and the ramp frame.

1. Draw the unit vectors that define the global and ramp frame co-located at the same origin. Specify the angle θ that defines the rotation between the two coordinate systems.

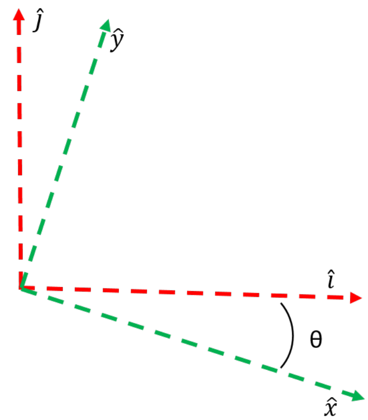


Figure 24.11: Unit vectors that define the two coordinate systems in the box-on-a-ramp problem.

2. Write equations for $\hat{x}$ and $\hat{y}$ in terms of $\hat{i}$ and $\hat{j}$.

$$\hat{x} = \cos(\theta)\hat{i} - \sin(\theta)\hat{j}$$
$$\hat{y} = \sin(\theta)\hat{i} + \cos(\theta)\hat{j}$$

3. Rewrite the equations above in the form of a *rotation matrix*.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix}$$

4. Rewrite your force equations so all of the forces are expressed in the global reference frame $[\hat{i}, \hat{j}]$, then split each force equation into the $\hat{i}$ and $\hat{j}$ components (i.e. you should end up with four equations for $\sum F_{box\hat{i}}$, $\sum F_{box\hat{j}}$, $\sum F_{ramp\hat{i}}$, $\sum F_{ramp\hat{j}}$).

Note: $F_{friction} = F_f$ for simplicity

$$\sum F_{box\hat{i}} = -F_f \cos\theta\hat{i} + N\sin\theta\hat{i} = 0$$

$$\sum F_{box\hat{j}} = -F_G\hat{j} + F_f\sin\theta\hat{j} + N\cos\theta\hat{j} = 0$$

$$\sum F_{ramp\hat{i}} = F_{framp}\hat{i} + F_{fbox}\cos\theta\hat{i} - F_{box}\sin\theta\hat{i} = 0$$

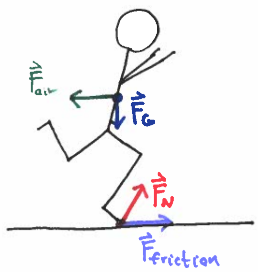
$$\sum F_{ramp\hat{j}} = -F_{GR}\hat{j} + N\hat{j} - F_{fbox}\sin\theta\hat{j} - F_{box}\cos\theta\hat{j} = 0$$

Solution 24.14

In the sprinter drawing below, our excited runner (note poor form with both arms forward!) has a toe planted on the ground. We've drawn a reactive normal force, F_N, to show the ground reacting to the toe of the sprinter. In the presence of a reactive force, particularly one that is normal to a surface, we should consider the presence of friction. We often model a frictional force as acting at the point of the normal force and being proportional to its magnitude, $F_{friction} = \mu F_N$, where μ is an appropriate coefficient of friction. If there were no frictional force at the sprinter's toe, what would happen? Notice that we've ignored the impact force that one feels in one's leg when running. We've ignored this because it is internal to the free body that we have defined like the tension in the cables of the Hot Air Balloon.

Qualitative FBDs

Sprinter:



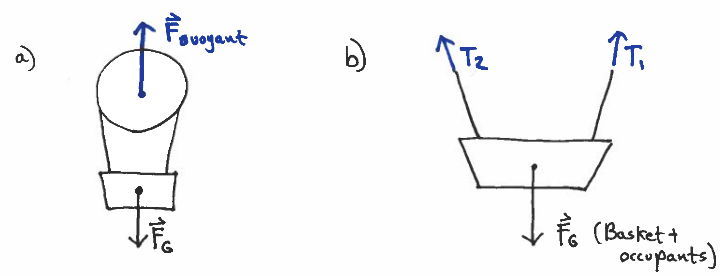
$\vec{F}_N$ = Normal force from ground

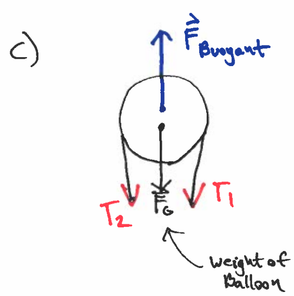
$\vec{F}_{friction}$ = Frictional force between foot and ground

$\vec{F}_G$ = Gravitational force

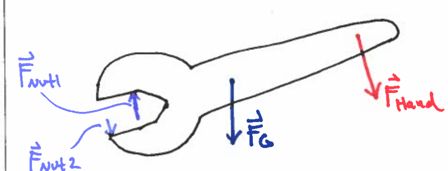
$\vec{F}_{air}$ = Air resistance

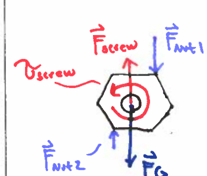
Hot Air Balloon





Wrench





Torque created by coupled forces from the wrench on the nut surfaces overcomes the resistive torque from the screw threads.

Chapter 25

Day 3: Center of Mass Workshop

25.1 Schedule

10:10-10:15	Welcome Back and Troubleshooting
10:15-11:00	1D and 2D Discrete COM
11:00-11:15	Coffee
11:15-12:00	Discretized Continuous COM
12:00-12:30	Conceptual COM

25.2 Welcome Back! [5 minutes]

Welcome back, at least virtually, to session II of QEA1!. We really hope that everything has been going well for each and every one of you, for your families, and for your travels away from Olin's campus. We recognize that the events of the past couple weeks have been incredibly challenging, scary, and disruptive. We feel that too, but we are going to do our best, as a community, to make the best of the situation and move forward with QEA1b. Today's in-class will be our first attempt at the new virtual reality, and we'll likely be tweaking the format and materials continuously as we move through the second half of the semester. It's going to be weird, but we're confident it will also be fun and you will learn a lot!

The exercises below reflect what we would have done in class on Thursday 3/12. The text in red are modifications to the exercises that we have made to adapt to the virtual course structure.

Figure 25.1: Don't be like T-Rex. Wash your hands.

25.3 Center of Mass

Gravity is a distributed force acting uniformly on all the mass within an object. For most purposes (i.e., when we don't need to consider deformation or breakage of an object) we can consider the entire gravitational force of the object to be acting upon the object at a single point: this point is the center of mass (COM, also referred to as Center of Gravity). The COM is the point about which the gravitational torque is zero: the balance point of an object. This is extremely important for ducks...or any other object that we wish to sit upright while being acted upon by gravity, since if there is no net torque around the COM, the object will not rotate.

COM of Systems of Discrete Objects [55 minutes]

COM in One Dimension Let's start by considering the COM for a collection of discrete, individual objects. For a set of objects each of mass m_i located at vector positions $\vec{r}_i$, the center of mass of the set is defined to be:

$$\vec{r}_{COM} = \frac{1}{\sum_i m_i} \sum_i m_i \vec{r}_i \tag{25.1}$$

Equation 25.1 looks like an expression for a weighted average because the COM is the *mass-weighted average position* of an object or group of objects.

Exercise 25.1

At the board, draw a picture and write a caption that uses an example to explain this mathematical definition.

Now let's start in one dimension. In one dimension this equation simplifies to:

$$x_{COM} = \frac{1}{\sum_i m_i} \sum_i m_i x_i \tag{25.2}$$

Exercise 25.2

Grab a ruler and a couple of the brass weights from the bin at the front of the room. Place the brass weights (start with two) at random positions on the ruler. Then, working at the board, calculate location of the center of mass from the masses and positions of the weights. Verify your calculation by finding (e.g., with your finger) the balance point for the ruler with the masses on it. Some things to consider:

Obviously we are not in the QEA rooms, but hopefully you each have a ruler, or a ruler-like object, to use for this exercise. In place of brass wights, you could use coins- that is what we will be using in the solution video. You will notice that in the equation for COM, the mass terms cancel, so the actual units of mass are not critical. Because of this, you can use each coin as a discrete mass unit (as long as you use all of the same type of coin), and can also represent the mass of the ruler in terms of "coin units." Unless you have a small scale at home, you will need to get creative in order to "weigh" the ruler. A simple balance might work well here. See the solution video for an example.

1. Considering that the weights have a significant size, what should you take for the position of the brass weights in the equation for COM?
2. Try using two masses which are the same mass. Then do the same exercise, same positions, with two masses which are very different. How does the position of the COM change with asymmetric masses?
3. Should you take into account the mass of the ruler? If so, what should you use for the position of the ruler?

COM in Two Dimensions In two dimensions, we are summing up vectors, which we can do component by component. Equation 25.1 can be expressed in two dimensions as:

$$x_{COM}\hat{i} + y_{COM}\hat{j} = \frac{1}{\sum_i m_i} \sum_i m_i(x_i\hat{i} + y_i\hat{j}) \tag{25.3}$$

By separating the above equation into x and y components, we can find the x and y components of the COM position individually:

$$x_{COM} = \frac{1}{\sum_i m_i} \sum_i m_i x_i \tag{25.4}$$

$$y_{COM} = \frac{1}{\sum_i m_i} \sum_i m_i y_i \tag{25.5}$$

Exercise 25.3

Grab an acrylic plate and a couple of different brass weights from the bin at the front of the room and weigh them.
In this problem we will be replacing the acrylic plate with a rectangular piece of cardboard. Make sure you choose a stiff piece of cardboard because you will be placing masses on it and looking for the COM point. Again, we can replace the brass weights with coins, either single coins or small stacks, in this exercise.

1. Where is the COM of your plate? Answer this question first based on what you know about COM, then find it experimentally and mark it using a dry erase marker.

2. Use your ruler to mark out a grid with dry erase marker on your acrylic plate. Grab two different brass weights. Using the above formulas and the plate COM as the origin, work out positions to place these two weights such that the COM of your plate stays at the origin. Try to do this with the weights moved off of center in both x and y directions (not just in x or y). Verify experimentally that your COM is the same as the unweighted plate.

3. Now move the weights to two arbitrary positions and compute the COM of the system of the plate and weights, again with the origin at the center of the plate. Test your prediction.

4. Now move your origin to the corner of the plate and re-compute the position of the center of mass for the system. Confirm your calculation experimentally. Does the choice of origin affect the position of the COM relative to the objects? How would you choose a convenient origin for a COM calculation?

Coffee Break!! [15 minutes]

Calculation of COM for Continuous Objects [45 minutes]

We can consider a continuous object to be made up of a sum of very small discrete objects. In this exercise, you will use this concept to compute the center of mass for a “two dimensional” hardboard object.

Exercise 25.4

1. Choose one of the hardboard shapes from stack at the front of the room. These will either already have a grid drawn in on the shape, or you will need to draw in evenly spaced grid lines. Determine the area of each square in the grid. We can refer to this incremental area of each little square as ΔA. The area density of the hardboard is 1.82 grams/square inch. What is the mass of each square in your grid? We can refer to this incremental mass of each little square as ΔM.

   A pdf version of the shapes was sent out to the class via email, and can be printed on standard 8.5" x 11" paper. The shapes are also available for download here. It is recommended to print two copies- one to work on for counting squares, and one to use as templates for cutting the shapes out of cardboard to verify your COM calculations. If you cut the shapes out of cardboard, you can assume the shape has uniform density of 1.

2. The area of the object is found by summing up the all the little areas $A_{total} = \sum \Delta A$, the total mass is found by summing up all the little masses $M_{total} = \sum \Delta M$. Calculate the total area and total mass of your object. You can verify your total mass by weighing your object on the scale in the front of AC113.
3. In order to find the center of mass of the object, we have to multiply each mass $m_i = \Delta M$ by the position vector of that mass square before summing. Choose an origin which you think is convenient, and find the COM by evaluating the COM equation for a continuous object:
4. Simplify the COM equation above for the case of a hardboard shape divided into N equal masses at positions r_i. Does the COM depend on the object's mass?
5. Once you have your final location for your center of mass of your object, mark it on your object and verify it experimentally by balancing the object with your finger under that point.

Conceptual COM for Continuous Objects [45 minutes]

Also at the front of the room are an assortment of real objects of various shapes. With a partner, consider a few different objects.

Exercise 25.5

1. Look at the distribution of mass on the object, including any regions made of differing materials. Can you predict where the center of mass should be?
2. Keeping in mind that the center of mass is a point in three dimensions, how can you *experimentally* locate the center of mass in all three dimensions? Look up the plumb bob method, and explain why it works.
3. Using a couple of objects, compare predictions and measurements of center of mass. How well do you do at guessing the center of mass for complex objects?
4. Some objects have symmetry: reflection symmetries, rotation symmetries. What do these symmetries tell you about the center of mass position?
5. Does the center of mass of an object have to be contained within the object? Can you find an example where the center of mass is not within the object?

6. Some objects can be considered to be made up of a system of separate objects, which can make it easier to find the composite center of mass. Can you find (or make) an example? How would you find the center of mass of the whole, if you can find the center of mass of each of the parts? (Write a mathematical expression for this).

7. The COM is a very important feature of most objects, impacting the functionality of many machines. As an example of this, our suite of objects includes three model airplanes. Where must the center of mass of an airplane be located for it to fly stably and safely? Where is the COM located on the models?

Solution 25.1

Lots of pictures could provide an excellent visual explanation of the mathematical definition for center of mass. The picture in Figure 25.2 is of two children of different masses on a seesaw. From intuition built on experience, we know that to balance the seesaw, the two children would have to sit different distances from the fixed pivot point. The mathematical definition of the COM is basically the inverse of this problem- for known positions and masses, where would the pivot point need to be placed in order to achieve the condition of perfect balance?

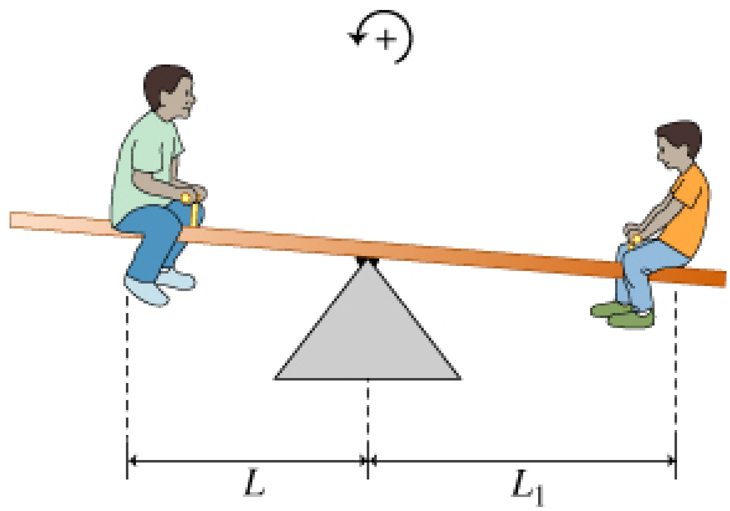


Figure 25.2: Two children of different mass balancing a seesaw by adjusting their distance from the fixed pivot point.

Solution 25.2

Grab a ruler and a couple of the brass weights from the bin at the front of the room. Place the brass weights (start with two) at random positions on the ruler. Then, working at the board, calculate location of the center of mass from the masses and positions of the weights. Verify your calculation by finding (e.g., with your finger) the balance point for the ruler with the masses on it. Some things to consider:

1. Considering that the weights have a significant size, what should you take for the position of the brass weights in the equation for COM? Take the position of their center of mass (the center for cylindrical weights) as their position.
2. Try using two masses which are the same mass. Then do the same exercise, same positions, with two masses which are very different. How does the position of the COM change with asymmetric masses? The center of mass of the system will shift towards the larger mass.

3. Should you take into account the mass of the ruler? If so, what should you use for the position of the ruler? Yes, and you should use the COM of the ruler! (See a pattern?)

Solution 25.3

Grab an acrylic plate and a couple of different brass weights from the bin at the front of the room and weigh them.

1. Where is the COM of your plate? Answer this question first based on what you know about COM, then find it experimentally and mark it using a dry erase marker. The plate's COM is in the middle of the rectangle.

2. Use your ruler to mark out a grid with dry erase marker on your acrylic plate. Grab two different brass weights. Using the above formulas and the plate COM as the origin, work out positions to place these two weights such that the COM of your plate stays at the origin. Try to do this with the weights moved off of center in both x and y directions (not just in x or y). Verify experimentally that your COM is the same as the unweighted plate.

3. Now move the weights to two arbitrary positions and compute the COM of the system of the plate and weights, again with the origin at the center of the plate. Test your prediction.

4. Now move your origin to the corner of the plate and re-compute the position of the center of mass for the system. Confirm your calculation experimentally. Does the choice of origin affect the position of the COM relative to the objects? How would you choose a convenient origin for a COM calculation? The choice of origin does not affect the COM relative to the objects. (Of course, it does affect the COM relative to the origin.) It's often convenient to choose an origin at the COM of an important, symmetrical object (like your plate) because it makes one term zero in your calculation and it may be easier to imagine the COM as an offset from the center of an important object in the system.

Solution 25.4

$$\vec{r}_{COM} = \frac{1}{N}\sum_i \vec{r}_i, \tag{25.6}$$

The COM does not depend on the object's mass, area, area density, or ΔM. Because the shapes are cut from a material with uniform area density, the COM only depends on the object's geometry.

Solution 25.5

1. Look at the distribution of mass on the object, including any regions made of differing materials. Can you predict where the center of mass should be?

2. Keeping in mind that the center of mass is a point in three dimensions, how can you *experimentally* locate the center of mass in all three dimensions? Look up the plumb bob method, and explain why it works. Gravity acts at the COM, and the object will rotate until the COM is directly below the hanging point. Therefore, each time you hang the object from a point, the vertical line through the object contains the COM. If you hang the object from two or more points and draw the vertical lines, the intersection of the lines will be the COM.

3. Using a couple of objects, compare predictions and measurements of center of mass. How well do you do at guessing the center of mass for complex objects?

4. Some objects have symmetry: reflection symmetries, rotation symmetries. What do these symmetries tell you about the center of mass position? The COM lies along the line of reflection for a 2-D object or on the plane of reflection for a 3-D object. For an object with rotational symmetry, the COM lies at the point of rotation for a 2-D object or along the line of rotation for a 3-D object.

5. Does the center of mass of an object have to be contained within the object? Can you find an example where the center of mass is not within the object? Nope! A coffee cup is one common object where the COM is not on the object.

6. Some objects can be considered to be made up of a system of separate objects, which can make it easier to find the composite center of mass. Can you find (or make) an example? How would you find the center of mass of the whole, if you can find the center of mass of each of the parts? (Write a mathematical expression for this).

7. The COM is a very important feature of most objects, impacting the functionality of many machines. As an example of this, our suite of objects includes three model airplanes. Where must the center of mass of an airplane be located for it to fly stably and safely? Where is the COM located on the models?

   The COM should be slightly in front of the center of lift, and on the center line of the aircraft. A nice explanation of trimming an aircraft can be found here.

Chapter 26

Day 4: Derivatives and Integrals

Learning Objectives

Concepts:

1. Define derivatives and integrals
2. Integrate and differentiate functions that involve: $x^n, \sin, \cos, \exp, \ln$.
3. Use integration to find the area between two curves.

Computing Skills:

1. Use WolframAlpha to compute derivatives and integrals

26.1 Schedule

- 1000-1005: Zoom and Technology (All of us)
- 1005-1020: Overview of the Day and Night (All of us)
- 1020-1115: Studio Time (Join your studio)
- 1115-1130: Stretch Break
- 1130-1230: Studio Time (Join your studio)

26.2 Overview and Orientation

In the "Duck on a Ramp" mini-project, we will be thinking about how to compute quantities like area, volume, and center of mass. These quantities involve the concept of integration. In this day assignment we will discuss some concepts from single-variable calculus, and in the night assignment we will discuss some concepts from multi-variable calculus. We will work exclusively in Cartesian coordinates, and use explicit or implicit function representations for curves and surfaces.

26.3 Resources

There is no shortage of resources available to help you with derivatives and integrals. Any single-variable or multi-variable calculus book will deal with these topics and you might have some useful resources from your high school calculus class. We'll focus on using two popular online resources:

- Khan Academy's videos
- Paul(not Ruvolo)'s online math notes

Both of these resources have practice problems, which we strongly recommend doing.

26.4 Single-variable calculus

Big idea: Single-variable calculus hinges on one fundamental idea: There is an intimate connection between the slope of the tangent line to a curve, and the area under the curve. The slope of the tangent line to the curve is the *derivative* and the area under the curve is the *integral*. They are connected by the *fundamental theorem of calculus*.

Derivative

Additional resources for this subsection:

- Khan Academy: Derivative as slope of curve
- Khan Academy: Formal definition of derivative as a limit
- Limit definition notes

Consider an explicit function $y = f(x)$. The slope of the tangent line at a point on the curve is the derivative of the function, defined as

$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

We often use the notation f' (especially if the independent variable represents a spatial coordinate) or $\dot{f}$ (often when the independent variable is time). Fortunately, we don't have to compute the derivative of functions using limits anymore, because humans have been doing this for over 300 years, and the derivative of lots of functions can be expressed in terms of elementary functions.

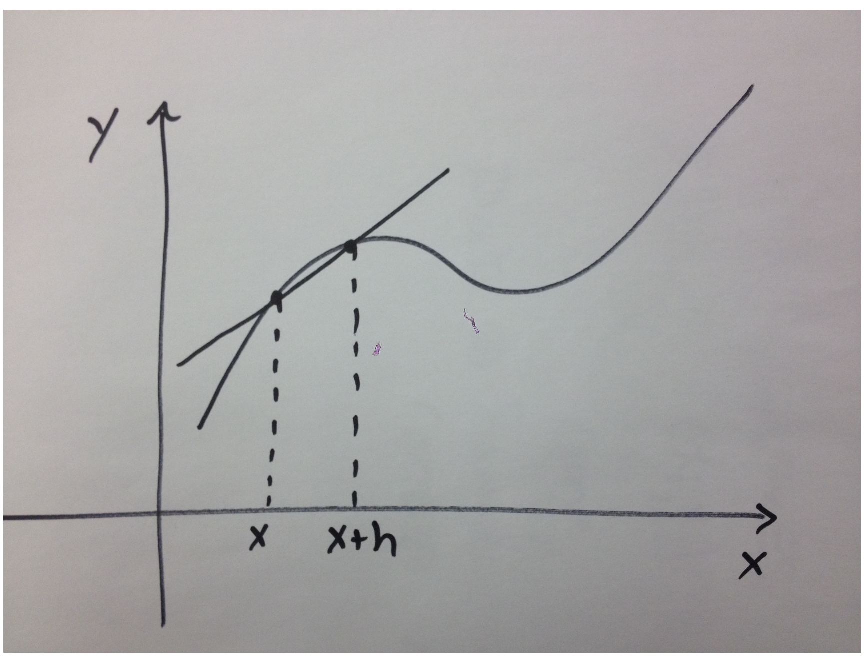


Figure 26.1: The derivative as a limit.

What if we give you a derivative, and ask you to figure out the function that it came from? Now you are finding the anti-derivative. However, this language is not always used, and many people refer to this as the indefinite integral or just the integral. This is unfortunate since it presupposes the fundamental theorem of calculus, which probably means you didn't even realize that this was a cool idea! Given that this terminology is widely used, we will just have to adopt it.

Definite Integrals

Additional resources for this subsection:

- Khan Academy: Integrals as Riemann sums

Again consider an explicit function $y = f(x)$. The area of the region below the curve defined by $y = f(x)$, above the line $y = 0$, and between the lines $x = a$ and $x = b$, is the definite integral of f from $x = a$ to $x = b$,

$$\int_a^b f(x)\,dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\Delta x_i.$$

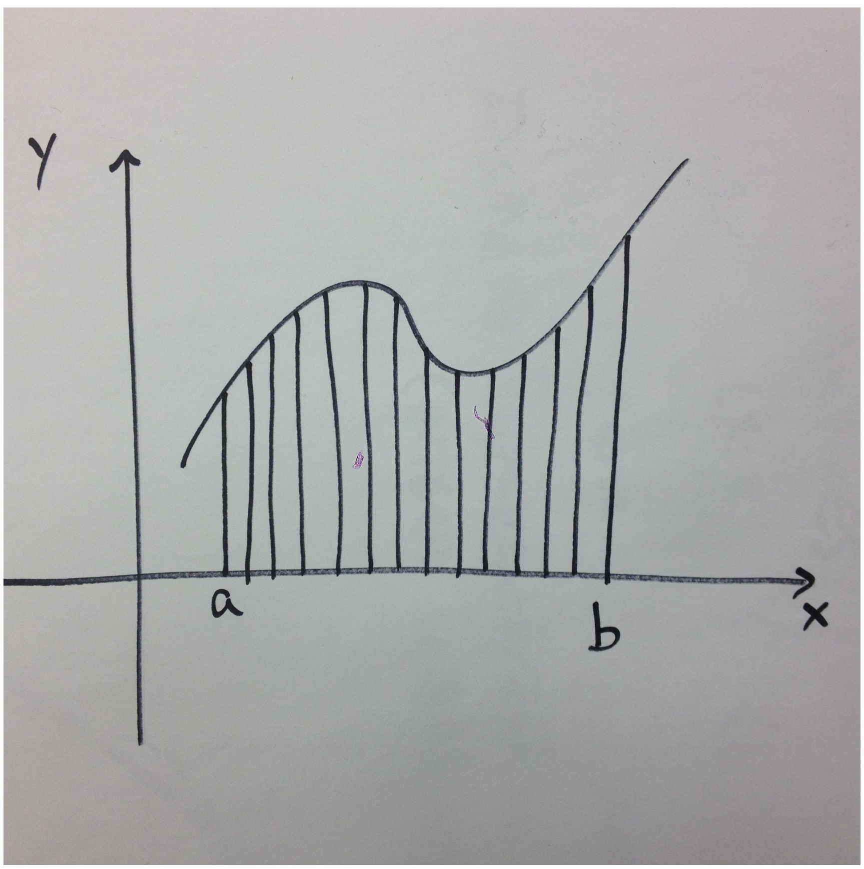


Figure 26.2: The integral as a limit.

The fundamental theorem of calculus and anti-derivatives

Additional resources for this section:

- Khan Academy: Fundamental theorem of calculus
- Khan Academy: Fundamental theorm of calculus and indefinite integrals
- Indefinitely integral notes

The fundamental theorem of calculus (one of its forms anyway) states that

$$\int_a^b f(x)\ dx = F(b) - F(a)$$

where F is the anti-derivative (or indefinite integral if you insist) of f, or $F' = f$. In other words, integrating the slope of a function between two points gives the change in the function between the end-points. For example,

$$\int_0^{\pi/2} \cos(x)\ dx = \sin(x)|_0^{\pi/2} = \sin(\pi/2) - \sin(0) = 1$$

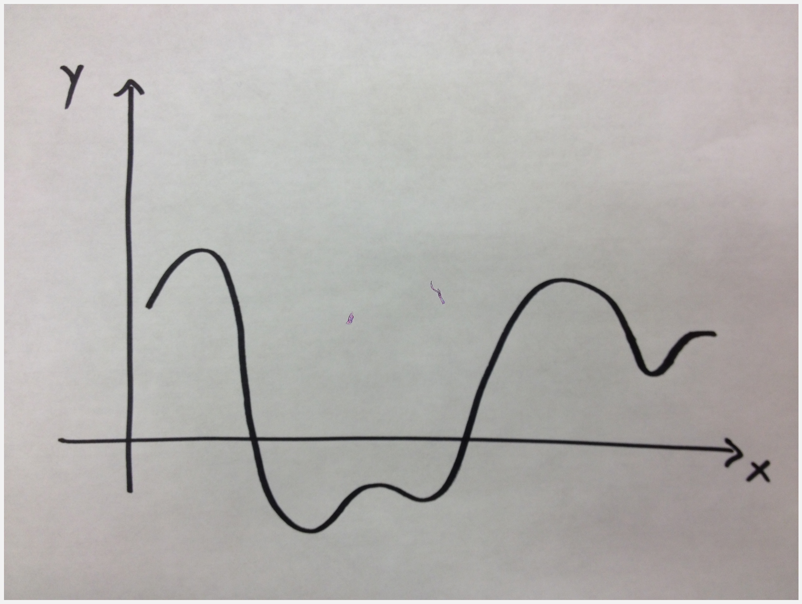
Exercise 26.1

1. Create a table of the five fundamental functions x^n, $\sin(x)$, $\cos(x)$, $\exp(x)$, and $\ln(x)$. List both their derivatives and their anti-derivatives (integrals). Include in your table at least one other example. Use any resource to find the derivatives and integrals. For example, you could type in **WolframAlpha**

   ```
   derivative of x^n
   ```

   ```
   integral of x^n
   ```

2. Consider the sketch of the function below. Now try to sketch the derivative and an anti-derivative.



26.5 Properties and Rules of Derivatives and Integrals

There are some key properties and rules of derivatives and integrals that we use over and over again. We include them here for completeness and ask one or two simple questions about them. These are summarized below:

Linearity of the Derivative and Integral (f and g are functions, c is a constant)

$$\begin{aligned}(f+g)' &= f'+g' \\ (cf)' &= cf' \\ \int_a^b (f+g)\ dx &= \int_a^b f\ dx + \int_a^b g\ dx \\ \int_a^b cf\ dx &= c\int_a^b f\ dx\end{aligned}$$

Exercise 26.2

Use your table of fundamental functions and these properties to evaluate the derivative and integral of $4x^3 + 3x^2 - 5x + 4$. Verify your answer using **WolframAlpha**.

Chain Rule

$$\frac{d}{dx} f(u(x)) = f'(u(x))\, u'(x)$$

Exercise 26.3

Use your table of fundamental functions and the chain rule to determine the derivative of $(x^3 - 1)^{100}$. Verify your answer using **WolframAlpha**.

Substitution Rule

$$\int_a^b f(u(x))u'(x)\, dx = \int_{u(a)}^{u(b)} f(u)\, du$$

Exercise 26.4

Use your table of fundamental functions and the substitution rule to evaluate $\int_0^4 2\sqrt{2x+1}\, dx$. Verify your answer using the Mathematica function **Integrate**.

Product Rule

$$\frac{d}{dx} f(x)g(x) = f\frac{dg}{dx} + g\frac{df}{dx}$$

Exercise 26.5

Use your table of fundamental functions and the product rule to determine the derivative of $x^2 \sin(x)$. Verify your answer using **WolframAlpha**

Integration by Parts

$$\int_a^b f(x)g'(x)\ dx \quad = \quad f(x)g(x)|_a^b - \int_a^b g(x)f'(x)dx$$

Exercise 26.6

Use your table of fundamental functions and integration by parts to determine $\int_1^2 x\exp(-x)\ dx$. Verify your answer using **WolframAlpha**.

General Practice

Exercise 26.7

Find the derivative and integral of the following functions, and verify your answers using **WolframAlpha**. Assume the following are constant values: $A, k, m, a, b, n, \omega, \phi, g, h$.

1. $$f(t) = Ae^{kt}$$
2. $$f(x) = mx + b$$
3. $$f(x) = ax^n + b$$
4. $$f(t) = A\sin(\omega t + \phi)$$
5. $$f(x) = g(x-h)^2 + k$$

26.6 Areas enclosed by curves

Additional resources for this section:

- Khan Academy videos and practice problems

Single-variable calculus gives us the tools to compute the area of regions bounded by curves. Consider the region bounded on top by $y = f(x)$, on the bottom by $y = g(x)$, and on the sides by $x = a$ and $x = b$. Appealing to the properties of integrals, the area of this region is

$$\int_a^b (f(x) - g(x))\, dx$$

We should note that integration will return a signed area. For example, the integral will be negative if the value of the function g is greater than that of f.

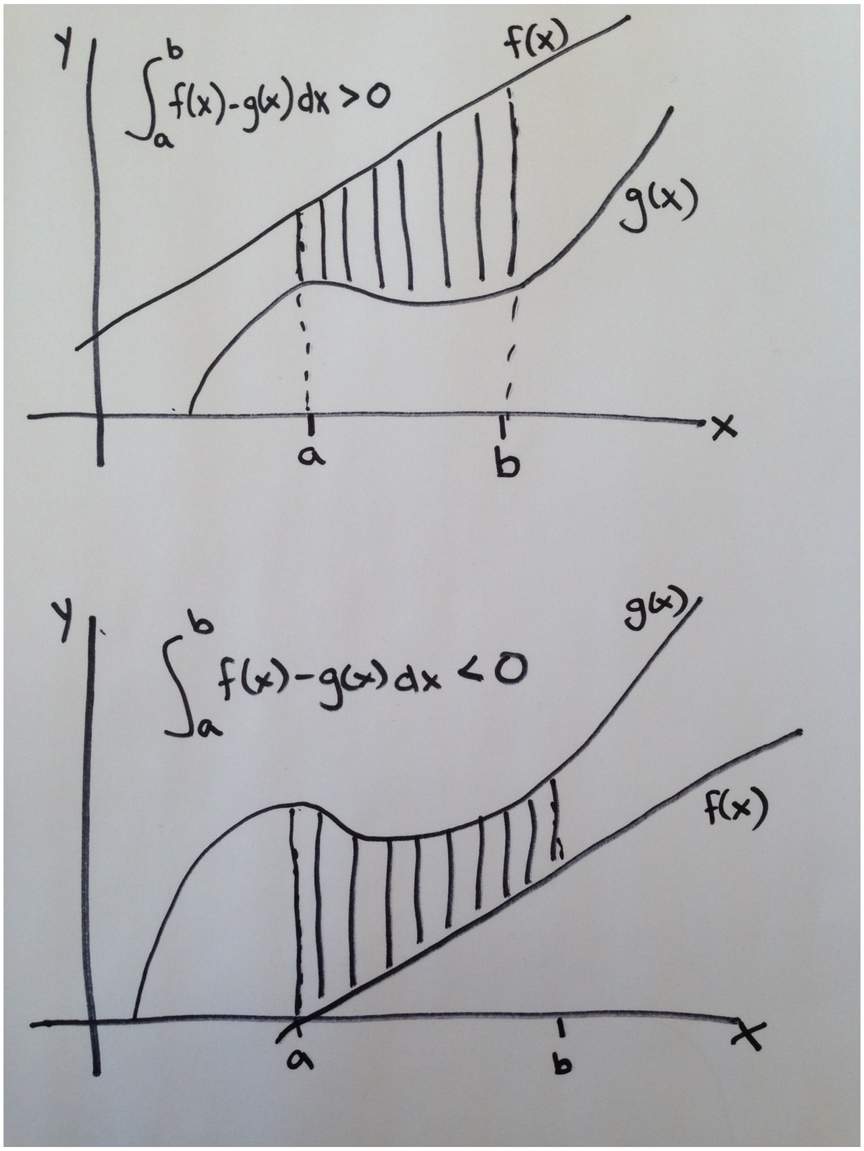


Figure 26.3: The area defined by integration can be positive or negative.

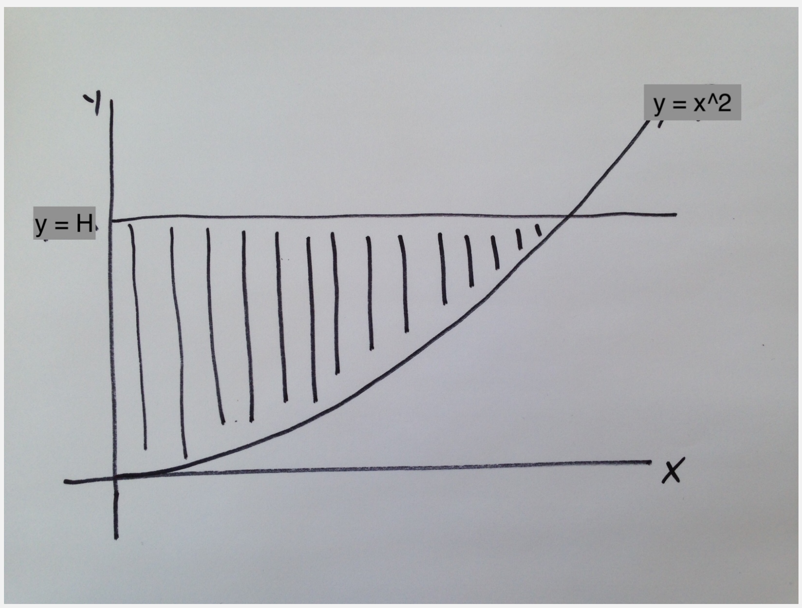
Exercise 26.8

Consider the first four fundamental functions x^n, $\sin(x)$, $\cos(x)$, and $\exp(x)$. For each function, sketch the region which is bounded above by the function, below by the x-axis and between $x = 0$ and $x = 1$. Use an integral to find the area of the region, and use **WolframAlpha** to verify your calculations. To visualize the regions you could type the following in **WolframAlpha**

```
plot 0 < y < x^2 and 0 < x < 1
```

Exercise 26.9

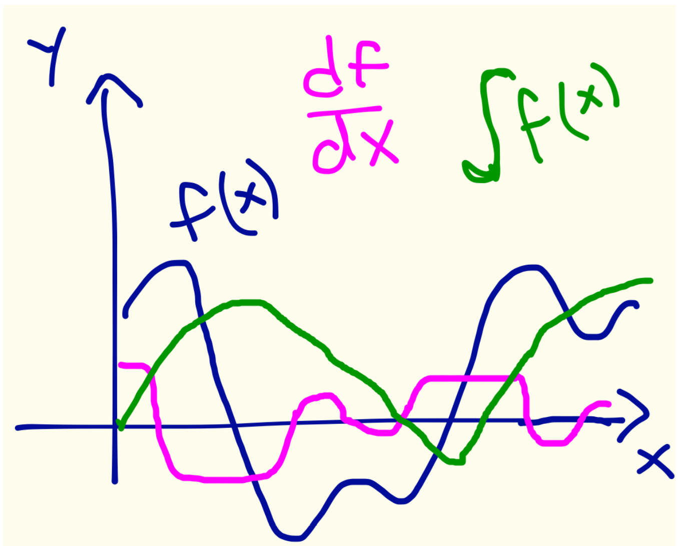
Consider the parabola defined by $y = x^2$. Propose an integral that would determine the area enclosed on the top by $y = H, H > 0$, on the bottom by $y = x^2$, on the left by $x = 0$, and on the right by the intersection of the top and bottom functions. Evaluate it by hand and verify the result using **WolframAlpha**.



Solution 26.1

1.

$f(x)$	$\frac{df}{dx}$	$\int f(x)\,dx$
x^n	nx^{n-1}	$\frac{x^{n+1}}{n+1}$ $(n \neq -1)$
$\sin x$	$\cos x$	$-\cos x$
$\cos x$	$-\sin x$	$\sin x$
$\exp x$	$\exp x$	$\exp x$
$\ln x$	$\frac{1}{x}$	$x\ln x - x$

2. 

Solution 26.2

$$\begin{aligned}
(4x^3 + 3x^2 - 5x + 4)' &= (4x^3)' + (3x^2)' + (-5x)' + 4' \\
&= 4(x^3)' + 3(x^2)' - 5(x)' + 4(x^0)' \\
&= 4(3x^2) + 3(2x) - 5(1) + 4(0) \\
&= 12x^2 + 6x - 5
\end{aligned}$$

$$\begin{aligned}
\int (4x^3 + 3x^2 - 5x + 4)\,dx &= \int 4x^3\,dx + \int 3x^2\,dx + \int -5x\,dx + \int 4\,dx \\
&= 4\int x^3\,dx + 3\int x^2\,dx - 5\int x\,dx + 4\int x^0\,dx \\
&= 4(\frac{x^4}{4}) + 3(\frac{x^3}{3}) - 5(\frac{x^2}{2}) + 4(\frac{x^1}{1}) \\
&= x^4 + x^3 - \frac{5}{2}x^2 + 4x
\end{aligned}$$

Solution 26.3

To use the formula above for the chain rule, we let $f(x) = x^{100}$ and $u(x) = x^3 - 1$. Then

$$\frac{d}{dx}(x^3-1)^{1000} = \frac{d}{dx}f(u(x)) = f'(u(x))u'(x) = 1000(x^3-1)^{999}(3x^2) = 3000x^2(x^3-1)^{999}.$$

Solution 26.4

To use the formula above for the substitution rule, we let $u(x) = 2x+1$ and $f(x) = \sqrt{x}$. Then

$$\int_0^4 2\sqrt{2x+1}\ dx = \int_0^4 f(u(x))u'(x)\ dx = \int_1^9 \sqrt{u}\ du = \frac{u^{3/2}}{3/2}|_1^9 = \frac{52}{3}.$$

Solution 26.5

To use the formula above for the product rule, let $f(x) = x^2$ and $g(x) = \sin(x)$. Then

$$\frac{d}{dx}x^2\sin(x) = \frac{d}{dx}f(x)g(x) = f\frac{dg}{dx} + g\frac{df}{dx} = x^2\cos(x) + \sin(x)(2x).$$

Solution 26.6

To use the formula above for integration by parts, let $f(x) = x$ and $g(x) = -\exp(-x)$. Then

$$\begin{aligned}\int_1^2 x\exp(-x)\ dx = \int_1^2 f(x)g'(x)\ dx \\ &= f(x)g(x)|_1^2 - \int_1^2 g(x)f'(x)\ dx \\ &= -x\exp(-x)|_1^2 - \int_1^2 -\exp(-x)\ dx \\ &= -x\exp(-x)|_1^2 - \exp(-x)|_1^2 \\ &= -2\exp(-2) + \exp(-1) - \exp(2) + \exp(-1) \\ &= -3\exp(-2) + 2\exp(-1)\end{aligned}$$

Solution 26.7

f	f'	$\int f$
Ae^{kt}	Ake^{kt}	$\frac{A}{k}e^{kt}$
$mx+b$	m	$\frac{1}{2}mx^2+bx$
ax^n+b	anx^{n-1}	$\frac{a}{n+1}x^{n+1}+bx$
$A\sin(\omega t+\phi)$	$A\omega\cos(\omega t+\phi)$	$-\frac{A}{\omega}\cos(\omega t+\phi)$
$g(x-h)^2+k$	$2g(x-h)$	$\frac{1}{3}g(x-h)^3+kx$

Solution 26.8

$$\int_0^1 x^n \, dx = \frac{x^{n+1}}{n+1}\Big|_0^1 = \frac{1}{n+1}$$

$$\int_0^1 \sin x \, dx = -\cos x|_0^1 = 1 - \cos 1$$

$$\int_0^1 \cos x \, dx = \sin x|_0^1 = \sin 1$$

$$\int_0^1 \exp x \, dx = \exp x|_0^1 = \exp(1) - 1$$

Solution 26.9

The top function is $f(x) = H$ and the bottom function is $g(x) = x^2$. The left limit is $x = 0$ and the right limit is $x = \sqrt{H}$, since this is where the top and bottom functions meet. To find the area of the region described in the problem we need to solve the integral

$$\int_0^{\sqrt{H}} (H - x^2) \, dx.$$

This gives

$$\begin{aligned}
\int_0^{\sqrt{H}} (H - x^2) \, dx &= (Hx - \frac{1}{3}x^3)|_0^{\sqrt{H}} \\
&= H^{\frac{3}{2}} - \frac{H^{\frac{3}{2}}}{3} \\
&= \frac{2}{3} H^{\frac{3}{2}}
\end{aligned}$$

Chapter 27

Night 4: Multiple Integrals and COM for Continuous Shapes

Learning Objectives

Concepts:

1. Compute a double integral.
2. Draw the region associated with a double integral. Vice versa, create a double integral to compute the area of a given region.
3. Compute a volume using a double integral.
4. Use a double integral to find the COM of an object.

In the day assignment, you learned that integral computes the "area under a curve." In this night assignment, we'll expand on that idea and introduce the concept of multiple integrals.

27.1 Areas as double integrals

Additional resources for this section:

- Khan Academy: Vertical area between curves (This series of videos covers the case where you are computing $\int\int\ dy\, dx$.)
- Khan Academy: Horizontal area between curves (This video covers the case where you are computing $\int\int\ dx\, dy$.)

First, we're going to compute the area of a rectangle in the silliest way possible.

Consider a rectangle enclosed by $x = a$, $x = b$, $y = c$, and $y = d$. The area of the rectangle according to calculus is

$$\int_a^b (d - c)\ dx$$

which (fortunately) evaluates to $(b - a)(d - c)$. Consider the integrand, $d - c$. According to the fundamental theorem of calculus, this difference could be expressed as an integral

$$d - c = \int_c^d dy$$

which is at the very least an interesting thing to do. Replacing this expression into the earlier one means the area of the rectangle could be expressed as a *double integral*

$$\int_a^b \int_c^d dydx$$

A word on notation: the inner integral is with respect to y, with limits defined by $y = c$ and $y = d$. The outer integral is with respect to x, with limits defined by $x = a$ and $x = b$. Since we could have expressed the area as

$$\int_c^d (b-a)\ dy$$

the area of the rectangle can also be expressed as the *double integral*

$$\int_c^d \int_a^b dxdy$$

Notice how the order of integration and corresponding limits have changed, but (presumably) the result hasn't.

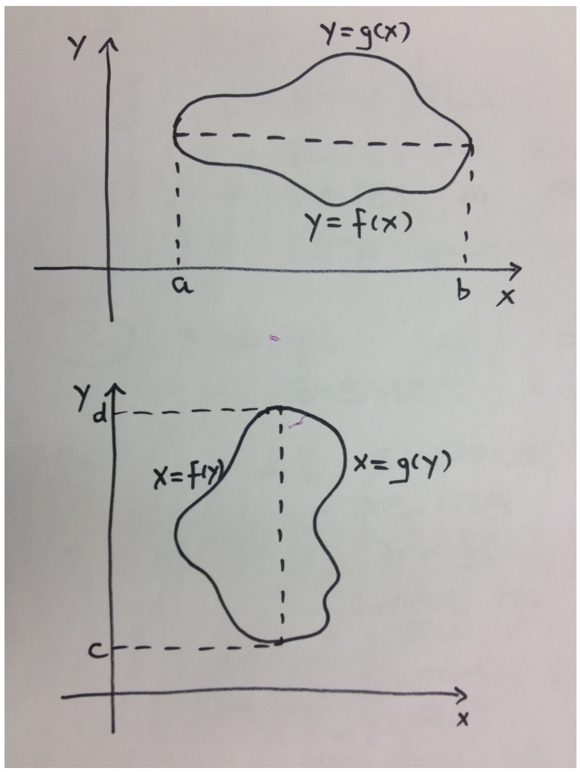


Figure 27.1: Simple regions in the plane that enclose areas.

Now consider a region enclosed by two functions $y = f(x)$ and $y = g(x)$ and two lines $x = a$ and $x = b$. Using the same reasoning, the area of this enclosed region can be expressed as a double integral

$$\int_a^b \int_{f(x)}^{g(x)} dydx$$

Consider the following double integral

$$\int_0^1 \int_0^x dydx$$

The region of integration is defined between $y = 0$ and $y = x$, and from $x = 0$ to $x = 1$. If we sketch this region we see that it is a triangle, and it should have an area of $1/2$. In **WolframAlpha** we can issue the request

```
integral of 1 from y = 0 to y = x and x = 0 to x = 1
```

and we will happily find that it returns $1/2$ as the result.

Exercise 27.1

Sketch the regions of integration and compute the area of the enclosed region by evaluating the double integral. Verify the result using **WolframAlpha**.

1. $\int_0^1 \int_0^x dydx$
2. $\int_1^2 \int_0^{\ln x} dydx$

What if the region must be described by two functions $x = f(y)$ and $x = g(y)$, and two lines $y = c$ and $y = d$? In this case we would integrate with respect to x first, and then with respect to y,

$$\int_c^d \int_{f(y)}^{g(y)} dxdy$$

Exercise 27.2

Sketch the regions of integration and compute the area of the enclosed region by evaluating the double integral. Verify your answer using **WolframAlpha**.

1. $\int_0^1 \int_y^{2-y} dxdy$
2. $\int_0^1 \int_y^{\exp y} dxdy$

For a given region in the plane, we are faced with a choice of the **order of integration**. Do we integrate with respect to x first and then y or vice versa? Since we are computing an area, the result should be the same, but often times one order of integration is much easier than another—sometimes one order of integration can't even be evaluated exactly!

Exercise 27.3

Repeat the previous two questions, but change the order of integration. The hard part is redefining the limits of integration.

27.2 Volumes enclosed by surfaces defined explicitly

Additional resources for this section:

- Khan Academy: Volume with cross sections – video series

Consider the volume enclosed by the surfaces defined by $z = f(x, y)$, $z = 0$, $x = a$, $x = b$, $y = c$, and $y = d$. How would we compute the volume of this region? One option would be to slice the surface up in sections parallel to one of the coordinate planes. For example, if we make a slice at $x = x_1$, then each planar region is bounded by $z = f(x_1, y)$, $y = c$, and $y = d$. The area of the enclosed region is therefore

$$\int_c^d f(x_1, y)\, dy$$

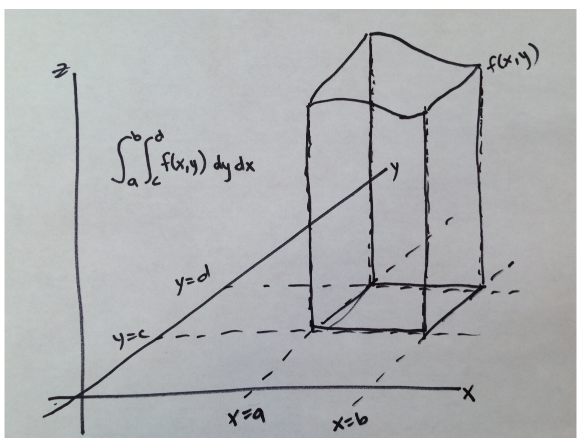


Figure 27.2: Volume defined as a double integral over a rectangle in the plane.

If we repeat this for different values of x, then we could compute the area of each cross-section as a function of x

$$area(x) = \int_c^d f(x, y)\, dy$$

What happens if we now integrate the area function from $x = a$ to $x = b$? We should get a volume, and it should be the volume of the original enclosed region,

$$Volume = \int_a^b area(x)\, dx = \int_a^b \int_c^d f(x, y)\, dydx$$

As we saw earlier, it shouldn't matter whether we change the order of integration.

Notice that the region of integration is the rectangle in the plane defined by $(x, y) \in [a, b] \times [c, d]$. There is no reason that the integral can't be computed over more general regions. In general we will use the coordinate-free notation

$$\int_D f \, dA$$

to define the integral of a function over a general region D in the plane.

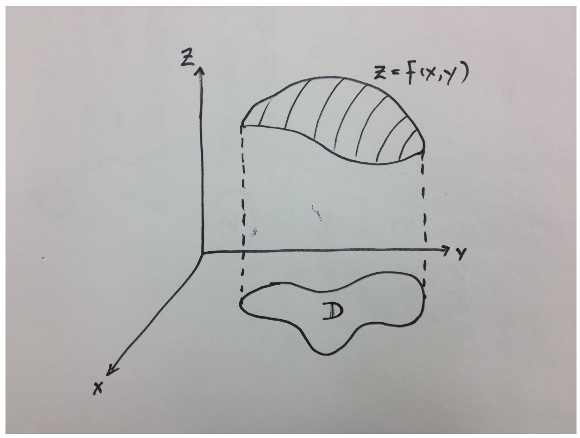


Figure 27.3: Volume defined as a double integral over a general region in the plane.

For example, consider the following integral

$$\int_D \frac{4y}{x^3+2} \, dA, \ D = \{(x, y) | 1 \leq x \leq 2, 0 \leq y \leq 2x\}$$

To sketch this region we draw a line at $y = 0$ and a line defined by $y = 2x$. We add lines at $x = 1$ and $x = 2$. The region of integration is enclosed by these definitions. We can evaluate this integral using **WolframAlpha** by issuing the following request

```
integral of 4y/(x^3+2) from y = 0 to y = 2x and x = 1 to x = 2
```

and we find that the result is approximately 3.21059.

Exercise 27.4

Sketch the following region of integration in the plane, and evaluate the integral using **WolframAlpha**

$$\int_D x \cos y \, dA,$$

where D is bounded by $y = 0, y = x^2, x = 0, x = 1$.

27.3 Applications of Double Integrals

So far we have been thinking exclusively in terms of geometry. In the same way that single integrals are used widely to compute quantities that are not areas per se, we can use double integrals to compute physically-relevant quantities like mass, center of mass, etc.

As an example, consider a thin plate (thickness Hcm) in 2D with variable mass density $\rho gm/cm^3$, i.e. the plate could be made of different material with a mass density that varies from location to location. The total mass M of the plate is represented by a double integral of the mass density over the plate. In coordinate-free notation, we can write

$$M = H \iint_D \rho \, dA$$

where D is the region in the plane occupied by the plate, and we evaluate the double integral depending on how we describe the plate. Likewise, the center of mass can be expressed as a double integral, and the relevant expressions are

$$x_{com} = \frac{H}{M} \iint_D x\rho \, dA$$

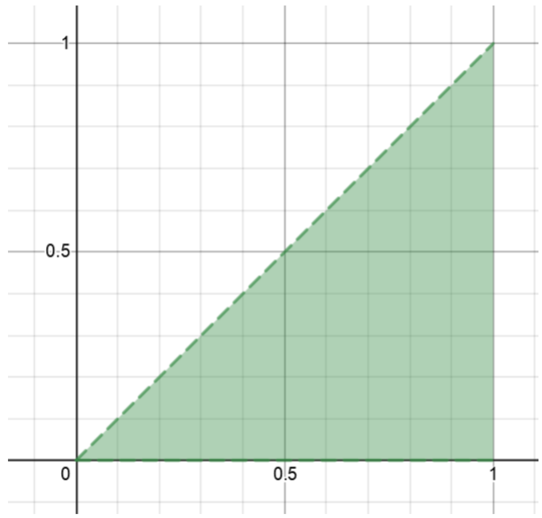
$$y_{com} = \frac{H}{M} \iint_D y\rho \, dA$$

Exercise 27.5

Find the total mass and center of mass of the 1 cm thin aluminum plate bounded by the parabola $y = x^2$, $y = 10$, and $x = 0$. Assume x and y are measured in centimeters. Use **WolframAlpha** to confirm your answer.

Solution 27.1

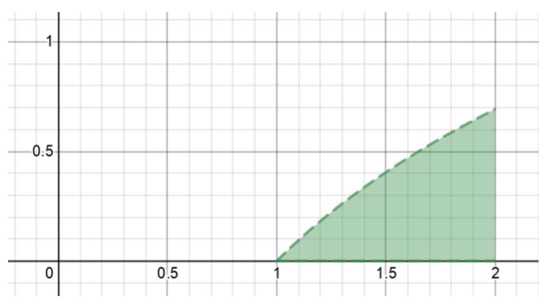
1. The region is



and the integral evaluates to

$$\int_0^1 \int_0^x dy\, dx = \int_0^1 x\, dx = 1/2.$$

2. The region is

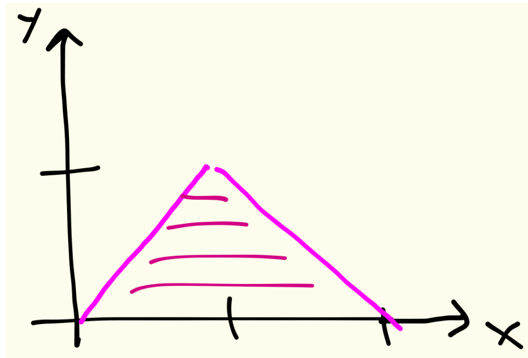


and the integral evaluates to

$$\int_1^2 \int_0^{\ln x} dy\, dx = \int_1^2 \ln(x)\, dx = \ln(2).$$

Solution 27.2

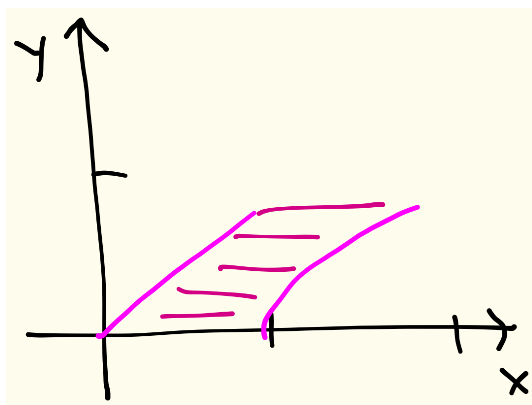
1. The region is



and the integral is

$$\int_0^1 \int_y^{2-y} dx\, dy = \int_0^1 (2-2y)\, dy = 2y - y^2|_0^1 = 1.$$

2. The region is



and the integral is

$$\int_0^1 \int_y^{\exp y} dx\, dy = \int_0^1 (\exp y - y)\, dy = \exp y - \frac{1}{2}y^2|_0^1 = \exp(1) - \frac{3}{2}.$$

Solution 27.3

1. To exchange the order of integration we have to integrate in two parts

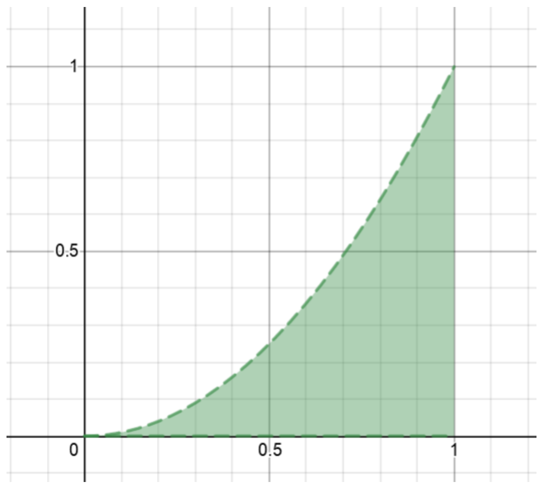
$$\int_0^1 \int_0^x dy\, dx + \int_1^2 \int_0^{2-x} dy\, dx = \int_0^1 x\, dx + \int_1^2 (2-x)\, dx = \frac{1}{2} + \frac{1}{2} = 1.$$

2. To exchange the order of integration we have to integrate in two parts

$$\int_0^1 \int_0^x dy\,dx + \int_1^e \int_{\ln(x)}^1 dy\,dx = \int_0^1 x\,dx + \int_1^e (1-\ln(x)\,)dx = \frac{1}{2} + \exp(1) - 2.$$

Solution 27.4

The region of integration is



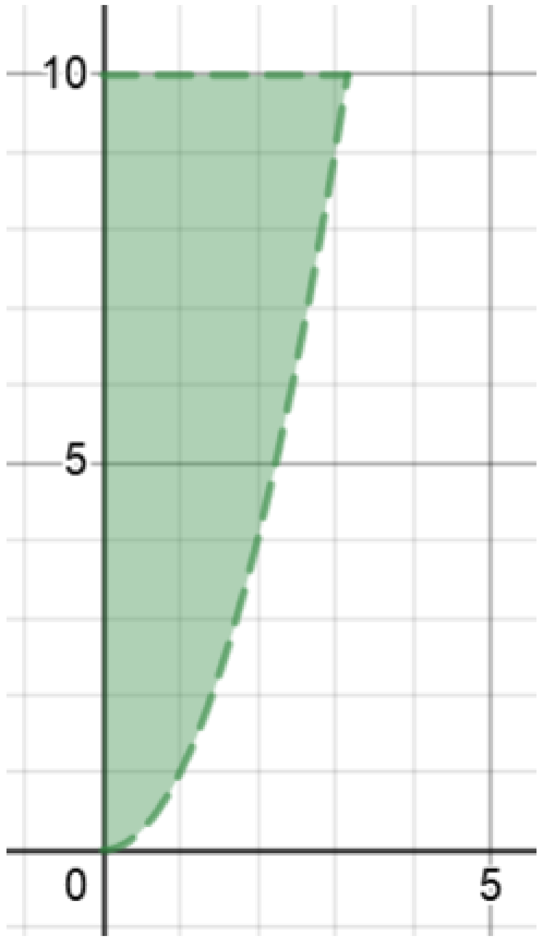
and so the integral will be

$$\int_0^1 \int_0^{x^2} x\cos y\,dy\,dx$$

which we evaluate

$$\begin{aligned}
\int_0^1 \int_0^{x^2} x\cos y\,dy\,dx &= \int_0^1 x\sin y|_0^{x^2}\,dx \\
&= \int_0^1 x\sin(x^2)\,dx \\
&= \frac{1}{2}\int_0^1 \sin(u)\,du \\
&= \frac{1}{2}(-\cos u)|_0^1 \\
&= \frac{1}{2}(1-\cos(1))
\end{aligned}$$

Solution 27.5



To find the total mass we use the expression from above with $H = 1cm$

$$\begin{aligned} M &= \int_0^{10} \int_0^{\sqrt{y}} \rho \, dx \, dy \\ &= \rho \int_0^{10} \sqrt{y} \, dy \\ &= \rho \frac{y^{3/2}}{3/2} |_0^{10} \\ &= \rho \frac{2}{3} 10^{3/2} gm \end{aligned}$$

where ρ is the density of Aluminum in gm/cm^3 (about 2.7).

To find the x center of mass we compute

$$\begin{aligned}
x_{com} &= \frac{3}{\rho 2(10^{3/2})}\int_0^{10}\int_0^{\sqrt{y}} \rho x \, dx \, dy \\
&= \frac{3}{\rho 2(10^{3/2})}\rho\int_0^{10} \frac{x^2}{2}|_0^{\sqrt{y}} \, dy \\
&= \frac{3}{2(10^{3/2})}\int_0^{10} \frac{y}{2} \, dy \\
&= \frac{3}{2(10^{3/2})}\left(\frac{y^2}{4}|_0^{10}\right) \\
&= \frac{3}{2(10^{3/2})}\left(\frac{10^2}{4}\right) \\
&= \frac{3}{8}10^{1/2} cm \\
&\approx 1.186 cm
\end{aligned}$$

Notice that ρ cancels out, assuming it is uniform. In other words, the center of mass for an object of uniform density only depends on its geometry!
To find the y center of mass we compute

$$\begin{aligned}
y_{com} &= \frac{3}{\rho 2(10^{3/2})}\int_0^{10}\int_0^{\sqrt{y}} \rho y \, dx \, dy \\
&= \frac{3}{\rho 2(10^{3/2})}\rho\int_0^{10} xy|_{x=0}^{x=\sqrt{y}} \, dy \\
&= \frac{3}{2(10^{3/2})}\int_0^{10} y^{3/2} \, dy \\
&= \frac{3}{2(10^{3/2})}\left(\frac{y^{5/2}}{5/2}|_0^{10}\right) \\
&= \frac{3}{2(10^{3/2})}\left(\frac{10^{5/2}}{5/2}\right) \\
&= 6 cm
\end{aligned}$$

Chapter 28

Day 5: Multiple-Integral Synthesis and Ramp Intro

28.1 Schedule

- 1000-1015: Zoom and Tech (All of us)
- 1015-1030: Live Overview (All of us)
- 1030-1115: Debrief and Synthesis
- 1115-1130: Stretch Break
- 1130-1230: Ducky on a Ramp

28.2 Zoom and Tech [15 minutes]

Settle in, say hi to your peers and the teaching team, and show everyone your pets. Or at least Jeff, he wants to meet your dogs.

28.3 Live Overview [15 minutes]

Introduction to the day and night assignments, especially the long-rumoured "Ducky on a Ramp."

28.4 Debrief and Synthesis [45 minutes]

Debrief

- Check-in with the folks in your break-out room about the Night 4 assignment. Resolve any issues, asking for help from your studio instructor if necessary!

Triple Integrals

In the same way that we could view area as a single or double integral (with the appropriate integrand), we can think of volume as a double or triple integral. If the region in the plane is a rectangle, and we are computing the volume under the surface define by $z = f(x, y)$,

$$\int_c^d \int_a^b f(x,y)\ dxdy$$

then we can use the fundamental theorem of calculus to express the integrand as an integral along the z-direction

$$f(x,y) - 0 = \int_0^{f(x,y)} dz$$

In this case then, the volume is defined as the triple integral

$$\int_c^d \int_a^b \int_0^{f(x,y)} dzdxdy$$

and more general regions could be dealt with in a similar manner. Hopefully, the order of integration does not matter and there are now 6 different ways that a triple integral could be written.

Exercise 28.1

1. Visualize the solid defined by the limits of integration, and evaluate its volume. Verify your results using Wolfram Alpha.

$$\int_0^1 \int_0^z \int_0^{x+z} dydxdz$$

   To visualize the solid, start by visualizing the surfaces in xyz-space defined by the inner most integral bounds, i.e. y = 0 and y = x+z. Next visualize the domain of definition in the xz-plane using the outer integral bounds.

2. Choose one other order of integration, determine the new limits, and evaluate.

In Wolfram Alpha you will need to spell out the order of integration by using the following syntax:

```
integrate 1 dy dx dz from y=0 to y=x+z and x=0 to x=z and z=0 to z=1
```

Now that we have come this far, there is no reason that we can't consider the triple integral of a function of three variables over a solid in 3D

$$\iiint_D f(x,y,z)\ dxdydz$$

We can, if we like, think of this as the hyper-volume of an object in 4D enclosed by the solids $w = 0$, $w = f(x,y,z)$, and the coordinate hyper-planes. This could very well make your head hurt.

Exercise 28.2

1. Visualize the solid defined by the limits of integration, and evaluate the triple integral. Verify your result using Wolfram Alpha.

$$\int_0^1 \int_x^{2x} \int_0^y 2xyz\ dzdydx$$

28.5 Introduction to the Ducky on a Ramp [60 minutes]

In the overnight assignment you will be solving a timeless problem that has plagued generations of the world's brightest engineers- what angle does a ducky assume when sitting on an inclined surface?

Figure 28.1: It is important not to confuse a duckling on a ramp (left) with a ducky on a ramp (right). Ducklings have fancy balancing systems comprised of brains and muscles and things. Duckies do not.

You may think this problem is ridiculous, and you are absolutely right, especially since our "ducky" is a simple circular segment. Even though the framing of this problem is intentionally silly, the learning objectives are very real and crucially important to engineering practice. The ability to analyze a problem, draw accurate free body diagrams, and interpret and define forces is foundational to the advanced skills you will learn later in QEA, at Olin, and in your professional careers.

In the overnight you will calculate the equilibrium angle for the "ducky" sitting on an inclined ramp. You will do this through analysis, not through experiments. You will also predict the maximum ramp angle will cause the "ducky" to either flip over *or* slide down the ramp.

If we were still at Olin, You would then test your predictions live, in class, in real time. You would not be allowed to perform experiments to develop your angle predictions, so you would need to trust your calculations. Unfortunately, in our current remote working environment, we will not be able to do these experiments as a group. We would still like you to perform these calculations and explain your process, and additional details will be in the night assignment.

28.6 Problem Framing

For this exercise, the "ducky" is defined as a circular segment with radius r and segment height h as shown in Figure 28.2. The circular segment is defined in the reference frame of the ramp with unit vectors $[\hat{x}, \hat{y}]$. The circular segment has unit thickness, and is constructed from a uniform density material. The center of mass of the circular segment is located at $[\bar{x}, \bar{y}]$.

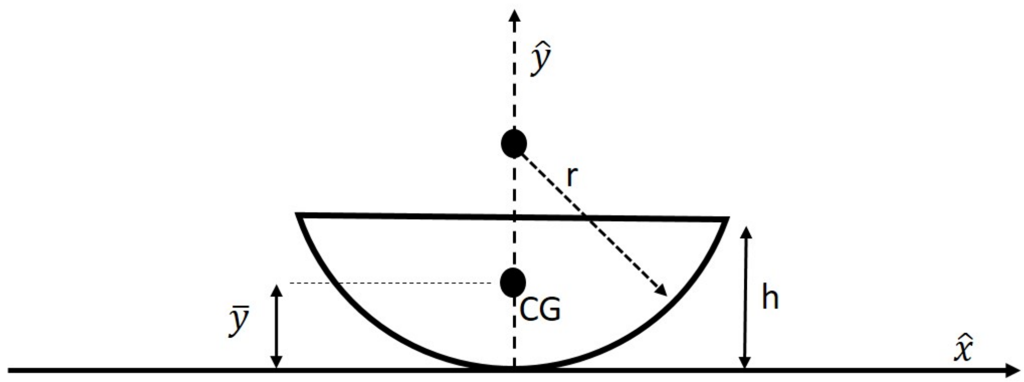


Figure 28.2: The circular segment "ducky."

Exercise 28.3

1. Write the general equation for a circle with center position at point $[m, n]$.
2. Apply the general form of the circle equation to the circular segment in Figure 28.2.
3. Solve the above equation for x.

28.7 *Visualizing the Problem*

In previous classes and overnight assignments we introduced the concept of free body diagrams (FBDs) as a way to visualize the forces acting on a body. In order to understand the problem of a "ducky" on a ramp, we need to draw multiple free body diagrams of relevant scenarios. The goal of these FBDs is to understand the forces and torques acting on the body, and identify if an equilibrium condition is possible.

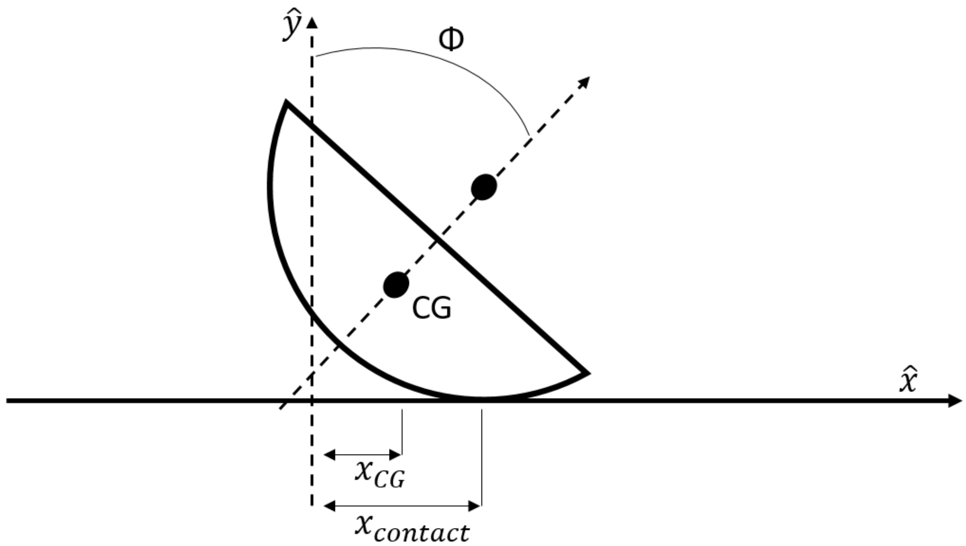


Figure 28.3: The circular segment "ducky" rolled to an angle ϕ relative to the $\hat{y}$ axis.

Exercise 28.4

1. Draw a FBD of the circular segment sitting on flat ground. Make sure of label your forces appropriately.
2. Draw a FBD of the circular segment sitting on flat ground, but rolled to an angle ϕ relative to the $\hat{y}$ axis, as shown in Figure 28.3.
3. Is the circular segment in equilibrium in both pictures? If not, why not?
4. Is the circular segment stable or unstable? How do you know?

Now, we need to consider the case where the circular segment "ducky" is sitting on a ramp tilted at angle θ. There are two interesting scenarios to consider, the initial condition of the segment before it rolls ($\phi = 0°$), and after the segment has reached an equilibrium condition. A diagram of the circular segment on the ramp is shown in Figure 29.4.

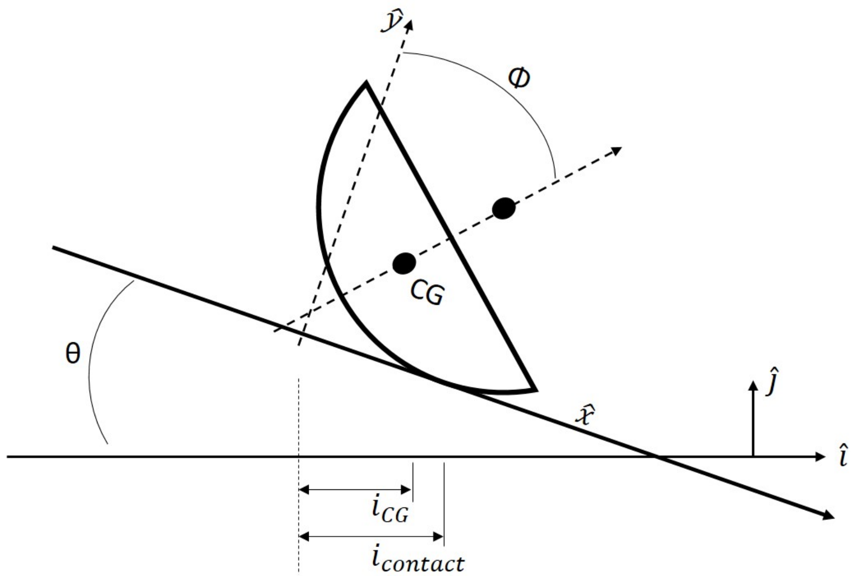


Figure 28.8: The circular segment “ducky” rolled to an angle ϕ on an inclined ramp with angle θ.

Exercise 28.5

1. Draw a free body diagram of the circular segment on a ramp titled at angle θ and with $\phi = 0°$.
2. Draw a free body diagram of the circular segment on a ramp titled at angle θ and roll angle ϕ.
3. For the case of $\phi = 0°$, is the circular segment in static equilibrium? Explain.
4. What conditions would need to exist for the circular segment to be in static equilibrium?

Solution 28.3

1. Write the general equation for a circle with center position at point $[m, n]$.

   a) For a circle with center position at point $[m, n]$, the general equation is $r^2 = (x-m)^2 + (y-n)^2$.

2. Apply the general form of the circle equation to the circular segment in Figure 28.2 (the equation will be for a full circle at this point).

   a) For the circular segment in Figure 28.2, the general equation becomes $r^2 = (x)^2 + (y - r)^2$.

3. Solve the above equation for x.

   a) Solving for x, we arrive at $x = \pm\sqrt{r^2 - (y-r)^2}$

Solution 28.4

1. Draw a FBD of the circular segment sitting on flat ground. Make sure of label your forces appropriately.

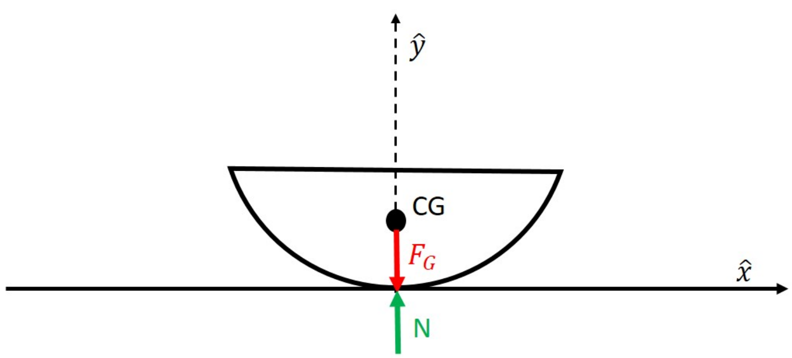


Figure 28.4: Free body diagram for the circular segment at $\phi = 0°$.

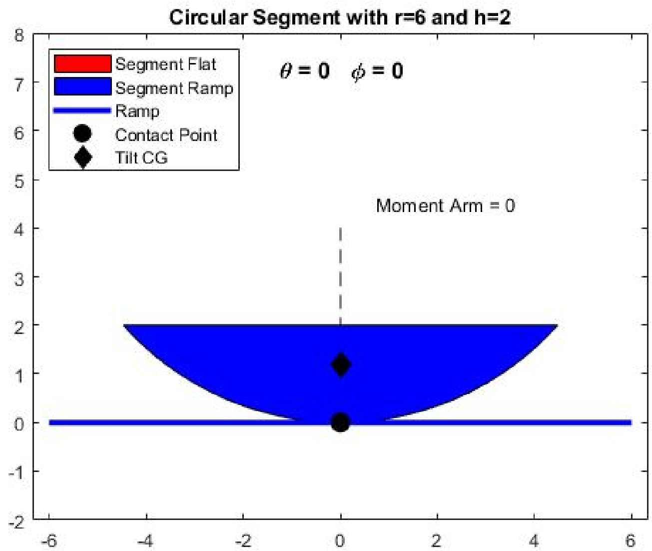


Figure 28.5: Simulation result for a circular segment on a flat surface.

2. Draw a FBD of the circular segment sitting on flat ground, but rolled to an angle ϕ relative to the $\hat{y}$ axis, as shown in Figure 28.3.

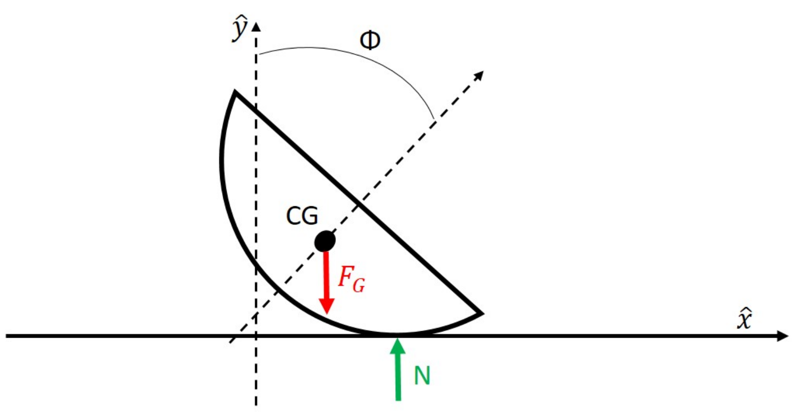


Figure 28.6: Free body diagram for the circular segment at an angle of roll ϕ.

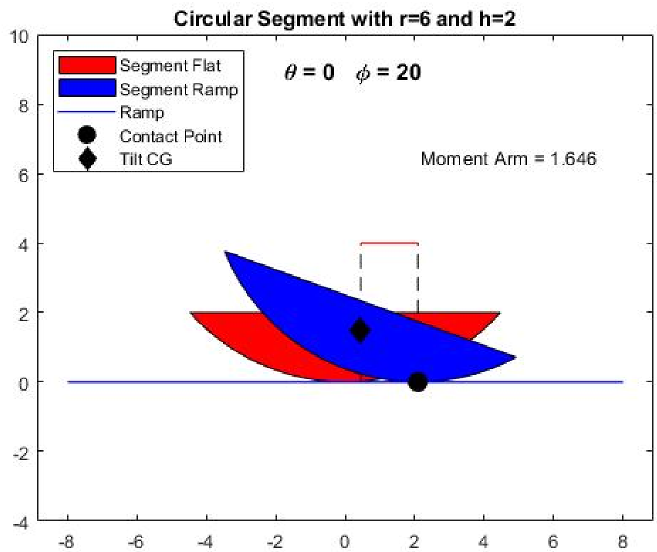


Figure 28.7: Simulation result for the circular segment at an angle of roll $\phi = 20°$.

3. Is the circular segment in equilibrium in both pictures? If not, why not?

   a) In Figure 28.4, the circular segment is in static equilibrium because $\sum F_{\hat{y}} = 0 = -F_G + N$, $\sum \tau = 0$, and $\sum F_{\hat{x}} = 0$. In Figure 28.6 the segment is in equilibrium in the $\hat{x}$ and $\hat{y}$ directions, but the horizontal separation between the center of gravity and the contact point create a moment about the contact point.

4. Is the circular segment stable or unstable? How do you know?

   a) The circular segment is stable. When $\phi > 0$, a restoring moment is created by the horizontal separation of N and F_G. For small angles, this restoring moment will act to restore ϕ to $\phi = 0°$.

Solution 28.5

1. Draw a free body diagram of the circular segment on a ramp titled at angle θ and with $\phi = 0°$.

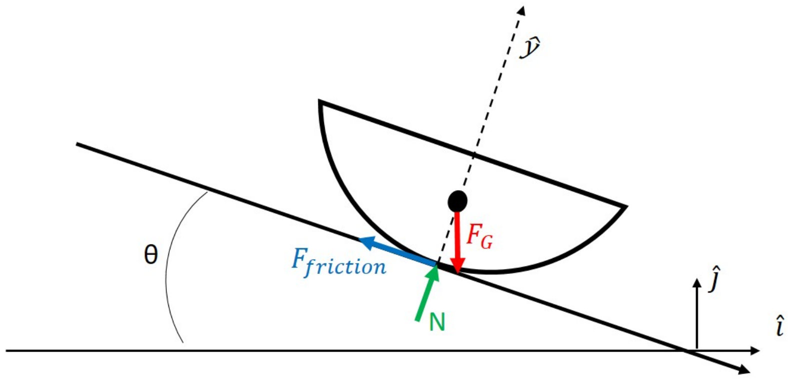


Figure 28.9: Free body diagram for the circular segment on a ramp at angle θ and with $\phi = 0°$.

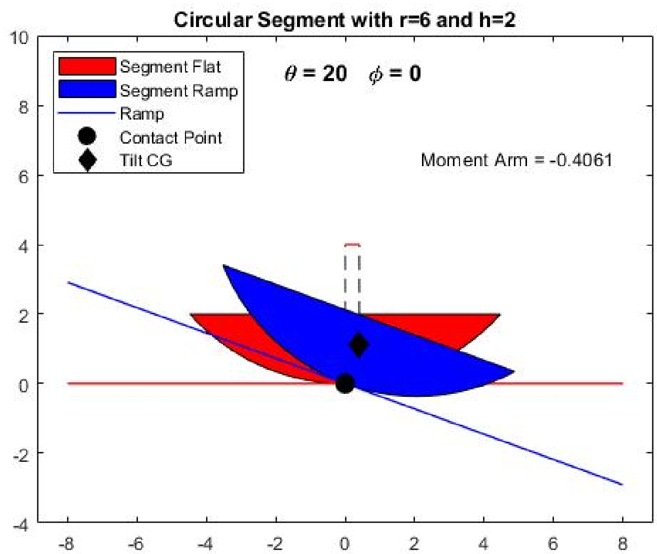


Figure 28.10: Simulation result for the circular segment on a ramp at angle $\theta = 20°$ and with $\phi = 0°$. A negative moment arm means a clockwise torque that will roll the segment towards $\phi > 0°$.

2. Draw a free body diagram of the circular segment on a ramp titled at angle θ and roll angle ϕ.

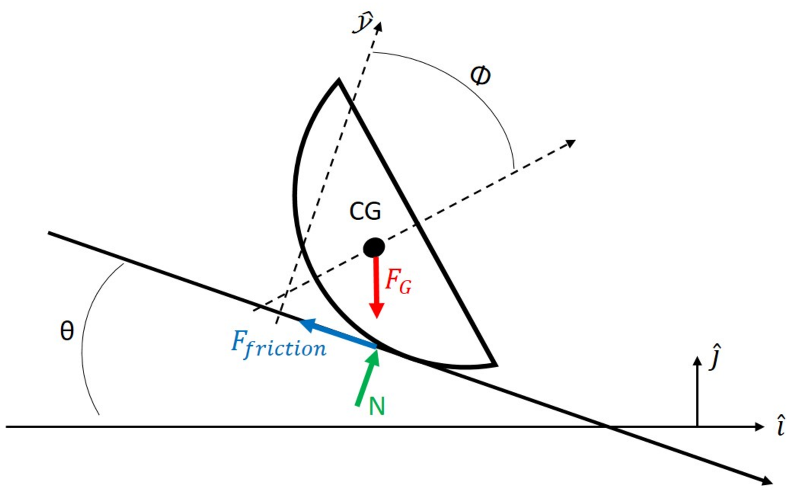


Figure 28.11: Free body diagram for the circular segment on a ramp at angle θ and with roll angle ϕ.

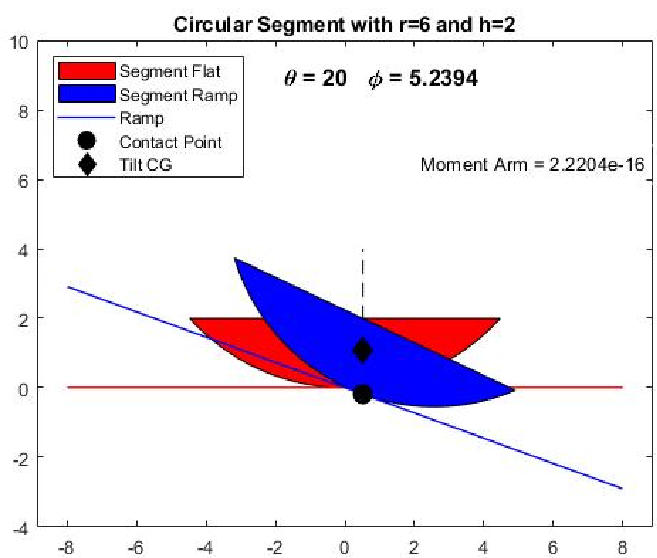


Figure 28.12: Simulation result for the circular segment on a ramp at angle $\theta = 20°$ and with roll angle $\phi = \phi_{equilibrium} = 5.24°$.

3. For the case of $\phi = 0°$, is the circular segment in static equilibrium? Explain.

   a) No, when the circular segment is at $\phi = 0°$ and the ramp is at an incline θ, the center of gravity, CG, is no longer directly above the contact point between the "ducky" and the ramp. Because the gravitational force, F_G acts at CG, and the normal force N acts at the contact point, a clockwise moment is present. The segment might also be out of equilibrium if the frictional force between the ramp and the segment is not sufficient and the segment slips down the ramp.

4. What conditions would need to exist for the circular segment to be in static equilibrium?

    a) From looking at the FBD, we can identify two important criteria. The frictional force most be sufficient to prevent slipping, and the gravitational force must act through the contact point, e.g. CG and the contact point are aligned vertically in the global reference frame, so that there is no moment.

Chapter 29

Night 5: The Ramp

Learning Objectives

Concepts

- Draw free body diagrams.
- Identify and model the forces acting on a body.
- Recognize and evaluate an equilibrium condition.
- Solve for the centroid of a three dimensional shape.
- Write multiple integrals for areas/volumes including the bounds of integration.
- Identify and work with multiple reference frames.

Matlab Skills

- Use symbolic math in Matlab to solve multiple integrals.
- Use the *solve* function.

Night Assignment Introduction

During the in-class assignment you worked with your breakout room to draw FBDs of several scenarios for the circular segment on a ramp. These included the segment sitting on flat ground, the segment on an inclined surface before an equilibrium condition was reached, and while in equilibrium on the inclined surface. Please look back at the day assignment solutions to review these FBDs and make sure you understand forces acting on the circular segment.

In this overnight assignment, we will move from the qualitative FBD analysis of the ramp problem, to a quantitative analysis where we will calculate the center of mass of the circular segment, and find the tilt angle $\phi_{equilibrium} = \phi_e$ which satisfies the static equilibrium condition. This analysis will require you to combine concepts from mechanics and multi-variable calculus that you saw in previous in-class and night assignments. We will also be using symbolic math in Matlab to help solve some "messy" integrals.

29.1 The Center of Mass/Gravity

In the problem framing, we identified that the center of gravity, CG, is located at some position $[\bar{x}, \bar{y}]$ on our circular segment (note: we are assuming a uniform gravitational field here, so center of mass and center of gravity are identical). In general, the position $\mathbf{R}$ of the center of mass within a solid Q with variable density $\rho(\mathbf{r})$ is given by:

$$\mathbf{R} = \frac{\iiint_Q \rho(\mathbf{r})\mathbf{r}dV}{\iiint_Q \rho(\mathbf{r})dV}$$

Because the expression $\iiint_Q \rho(\mathbf{r})dV$ represents the total mass of the solid, M, this expression for the center of mass can be written as:

$$\mathbf{R} = \frac{1}{M}\iiint_Q \rho(\mathbf{r})\mathbf{r}dV$$

For our "ducky," the density of the material is uniform (i.e., ρ=constant), and the circular segment is of unit depth. Therefore, the position of the center of mass is given by the geometric centroid in the $\hat{x}$ and $\hat{y}$ directions.

$$\bar{x} = \frac{\iint xdA}{\iint dA}$$

$$\bar{y} = \frac{\iint ydA}{\iint dA}$$

Exercise 29.1

1. When solving for the centroid of the circular segment, do you need to setup and solve the multiple integrals for both $\bar{x}$ and $\bar{y}$? Explain.
2. It is often useful to write a double integral of the form $A = \iint dA$ as $A = \int xdy$ or $A = \int ydx$. For the y-centroid, $\bar{y}$, write the double integral for area as a single integral, including an appropriate representation of x. Hint: take advantage of symmetry.
3. What are the bounds of the integrals in the equation for $\bar{y}$.
4. Write, but do not solve, the full equation for $\bar{y}$.

Symbolic Matlab Implementation The equation for the y-centroid developed in Exercise 29.1 is non-trivial to do by hand. If you would like to flex your integration muscles, feel free to give it a try (this video has a nice walk through of a similar problem).

Exercise 29.2

1. Use the code snippet linked here and shown below to utilize symbolic math in Matlab to find $\bar{y}$ and the total area of the segment:

   Note: If you copy/paste the code below, carefully check the comments are correct on your local machine.

```
%define circular segment parameters
r=6;
```

```
h=2;

%specify symbolic values using the matlab 'syms' function
syms r_sym h_sym y

%first define x using the definition of a circle with center at [0,R], ...
    x^2+(y-R)^2=R^2
x=sqrt(r_sym^2-(y-r_sym)^2);

%Setup the integral, symbolically, for the area of the circular segment. ...
    This %integral is A=2*int(x). We will do this using the 'int' ...
    function in Matlab which %takes inputs as int(symbolic_equation, ...
    symbolic_variable_of_integration, [lower %bound, upper bound])
A=2*int(x,y,[0 h_sym]);

%Do the integral for the centroid symbolically
ybar=int(y*x,y,[0 h_sym])/int(x,y,[0 h_sym]);

%substitute numerical values for r and h and convert the output to a ...
    double. The 'subs' function takes inputs in the form subs(equation, ...
    old, new)
A=double(subs(A,[r_sym, h_sym],[r, h]));
ybar=double(subs(ybar,[r_sym, h_sym],[r, h]));
```

2. For the case of $r = 6$ and $h = 2$, what are A and $\bar{y}$?

29.2 *Position of CG and the Contact Point*

To find the angle ϕ at which the circular segment is in static equilibrium for a given ramp angle, θ, we must find the position of the center of mass and the contact point between the body and the ramp.

To understand the motion of the center of mass, we can visualize the motion of the point CG on a circle of radius r at distance (r-$\bar{y}$) from the center, where $\bar{y} < r$. This case is shown in Figure 29.1, where $r = a$, (r-$\bar{y}$)=b and point P represents the point CG on our segment. The trajectory of point P is called a *curtate cycloid*. In our case, we are only using a segment of the circle, so we cannot physically roll through a full revolution of range 0 to 2π, but the case of a point on a circle helps visualize the motion.

The position of the center of mass in the frame of reference of the ramp, as a function of the roll angle, ϕ, can be described as a *curtate cycloid* with equations of the form:

$$x_{cg} = (r\phi - (r - \bar{y})sin(\phi))\hat{x}$$

$$y_{cg} = (r - (r - \bar{y})cos(\phi))\hat{y}$$

Exercise 29.3

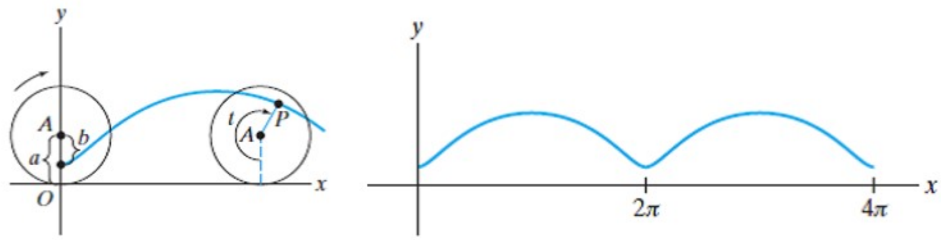


Figure 29.1: Left: When $b < a$, point P traces a curtate cycloid. Right: The trajectory of point P over two full revolutions of the circle. Images from Chegg.

1. In Matlab, plot the position of the center of mass for the range $0 \leq \phi \leq 4pi$. (Imagine you are plotting the position of a point on a disk as it completes two full rotations. You should find a trajectory that looks like Figure 29.1.)
2. What are the equations for the position of the contact point, as a function of ϕ in the reference frame of the ramp?
3. Plot the position of the CG and the contact point on the same plot for the range $0 \leq \phi \leq \frac{pi}{4}$. Keep the number of points small so it is easy to interpret the relative position of the CG and the contact point. What do you notice?

29.3 Reference Frames

For the case of the circular segment "ducky" rolling through small angles on a flat surface, we observe that the horizontal position of the CG always lags the horizontal position of the contact point. This is supported by both the FBD and the calculated positions of the CG and contact point, respectively. Because a horizontal separation exists between the contact point and the CG, and restoring moment is generated for small angles ϕ, and we consider the segment to be in stable static equilibrium at $\phi = 0$. When placed on an inclined surface, however, the horizontal separation between the CG and the contact point can be equal to zero at $\phi \neq 0$.

To find the angle ϕ at which the circular segment is in a stable static equilibrium (assuming no slip) on an inclined surface ($\theta \neq 0$), we must consider the positions of the CG and the contact point in the *global reference frame* which is defined by the unit vectors $[\hat{i}, \hat{j}]$, as seen in Figure 29.4.

Exercise 29.4

1. Write expressions for the unit vectors $[\hat{x}, \hat{y}]$ that define the *ramp frame* in terms of the vectors

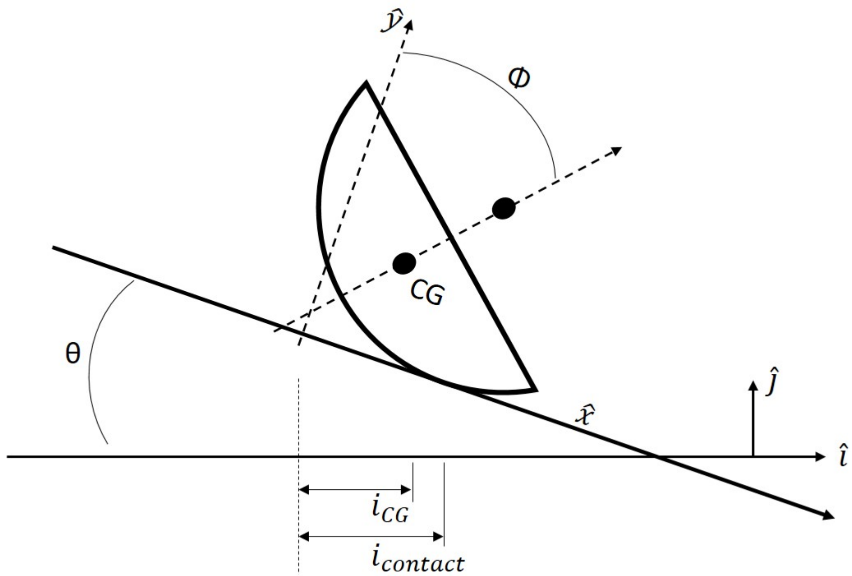


Figure 29.4: The circular segment "ducky" rolled to an angle ϕ on a inclined ramp with angle θ.

$[\hat{\imath}, \hat{\jmath}]$ which define the *global frame*.

2. To represent the positions of the CG and the contact point in the *global frame*, substitute the expressions for $[\hat{x}, \hat{y}]$ in terms of $[\hat{\imath}, \hat{\jmath}]$ you found above into the equations for the position of the CG and contact point in the *ramp frame*.
3. Following the substitution above, collect terms to find the vertical and horizontal positions of the CG and contact point in the *global frame*.
4. Based on your FBD of the circular segment on an inclined ramp, what condition must the horizontal positions of the CG and contact point satisfy for the "ducky" to be in static equilibrium (assuming no slip)?

29.4 *Solving for the Equilibrium Angle*

There are several ways to find the equilibrium angle in Matlab. The code snippet linked here and shown below creates symbolic equations for the horizontal positions of the CG and contact point, and then uses the Matlab `solve` function to find the angle ϕ where they are equal.

```
%define circular segment parameters
r=6;
h=2;

%ramp angle in degrees converted to radians
theta=deg2rad(30);

%specify symbolic values
syms r_sym h_sym y

%we need to setup our integrals. For the centroid, ybar=intydA/intdA
%We use dA=xdy

%first define x using the definition of a circle with center at [0,R], x^2+(y-R)^2=R^2
x=sqrt(r_sym^2-(y-r_sym)^2);

%Setup the integral, symbolicaly for the area of the circular segment. This integral ...
    is A=2*int(x)dy
A=2*int(x,y,[0 h_sym]);

%do the integral for the centroid symbolically and convert to a double
ybar=int(y*x,y,[0 h_sym])/int(x,y,[0 h_sym]);

%substitute numerical values for r and h and convert the output to a double
A=double(subs(A,[r_sym, h_sym],[r, h]))
ybar=double(subs(ybar,[r_sym, h_sym],[r, h]))

% create the symbolic variable phi
syms phi

%position of the CG in global frame
i_cg=(r*phi-(r-ybar)*sin(phi))*cos(theta)+(r-(r-ybar)*cos(phi))*sin(theta);
j_cg=-(r*phi-(r-ybar)*sin(phi))*sin(theta)+(r-(r-ybar)*cos(phi))*cos(theta);

%position of contact point in global frame
i_contact=r*phi*cos(theta);
j_contact=r*phi*-sin(theta);

% find the equilibrium angle using the solve function
eqn = i_cg == i_contact; %create equality
phi_eq=rad2deg(double(solve(eqn,phi,'Real',true))) %use rad2deg to convert to degrees
```

Exercise 29.5

1. Find the static equilibrium angle for a circular segment with r=1.75 and h=1.5 for θ=5, 15, 25, 35 degrees. Is equilibrium possible for all of these ramp angles?
2. Use this script to visualize the circular segment and ramp angles given above. For the 35 degree case, you can set the ϕ angle manually to get a qualitative sense of what is happening. Why do you think a solution is not solvable?
3. The `solve` function returns two ϕ angles for a given ramp angle θ. Are both results physically valid? Why or why not?

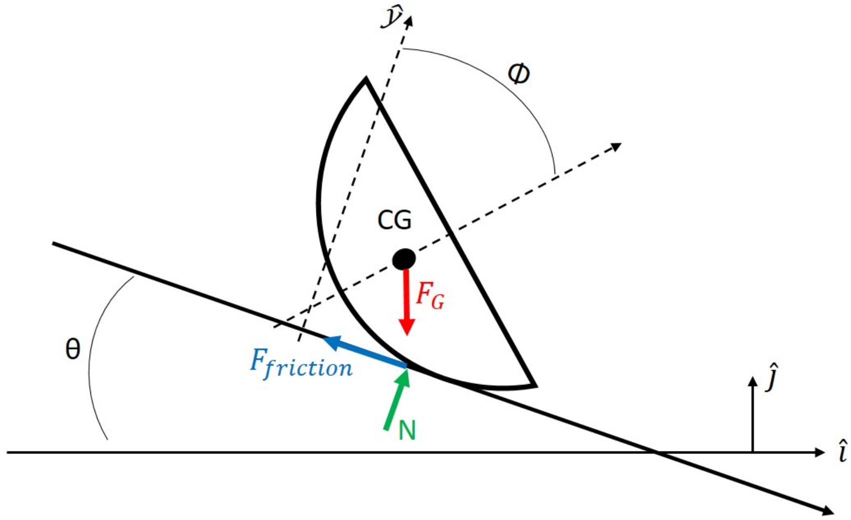


Figure 29.11: Free body diagram for the circular segment on a ramp at angle θ and with roll angle ϕ.

29.5 Will it Slip?

Up to this point, we have assumed a no-slip condition for the "ducky," but in reality, the "ducky" may slip down the ramp before flipping over. Considering the FBD in Figure 29.11, we recognize that a frictional force exists between the "ducky" and the ramp that opposes motion in the $\hat{x}$ direction. In the case of static equilibrium, we consider this frictional force to be static friction, given as $F_{fs} = \mu_s N$ where μ_s is the coefficient of static friction which is dependent on the material and geometry of the object, and N is the normal force.

Exercise 29.6

1. Write the force equations in the ramp $[\hat{x}, \hat{y}]$ frame for the "ducky" in static equilibrium.
2. When will the "ducky" slip?
3. What can be done to increase the slip-angle for the "ducky"? What are the trade-offs for the modifications?

29.6 The Ramp Challenge

We will now combine the skills developed during this assignment with those from the “Center of Mass Workshop” earlier in the semester. In Figure 30.1 is a three-dimensional shape with constant density. The shape is a composite- the bottom portion is a circular segment with r=4.75cm and h=3.8cm. The middle section is a rectangle 7cm x 2.3cm. The upper section is a circular segment with with r=4.75cm and h=2.5cm. Each segment is 5cm thick. The density is uniform across the composite ducky shape.

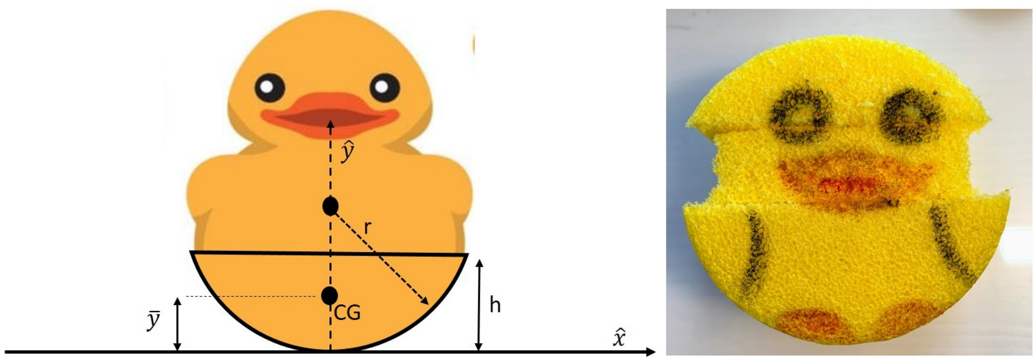


Figure 29.12: Pinterest vs. reality, am I right?

Exercise 29.7

For the final ramp challenge, you will perform the following calculations and put your answers in the “Ducky Night 5” quiz on Canvas. This exercise replaces the typical conceptual quiz that you usually have on night assignments. Submissions are due before the start of class of Thursday, April 2nd.

For the composite ducky shape:

1. Find the position of the center of mass for the composite made-from-Jeff’s-foam-roller ducky.
2. Predict the equilibrium angle $\phi_{eq}(\theta)$ for $\theta = [2°, 4°]$.
3. (text entry on Canvas) Explain your process for finding the CG and ϕ_{eq}.

Solution 29.1

1. When solving for the centroid of the circular segment, do you need to setup and solve the multiple integrals for both $\bar{x}$ and $\bar{y}$? Explain.

   No, because of symmetry you know that $\bar{x}$ is located at zero in the $\hat{x}$ direction.

2. It is often useful to write a double integral of the form $A = \int\int dA$ as $A = \int x dy$ or $A = \int y dx$. For the y-centroid, $\bar{y}$, write the double integral for area as a single integral, including an appropriate representation of x. Hint: take advantage of symmetry.

   Because of symmetry we will focus on writing the area integral for the half of the circular segment to the right of the $\hat{y}$ axis. The integral would then be $A = 2\int_a^b \sqrt{r^2 - (y-r)^2} dy$, where the bounds a, b will be answered in the next question.

3. What are the bounds of the integrals in the equation for $\bar{y}$.

   The integral bounds are 0 and h, or $A = 2\int_0^h \sqrt{r^2 - (y-r)^2} dy$.

4. Write, but do not solve, the full equation for $\bar{y}$.

   The full equation for the y-centroid is: $\bar{y} = \frac{2\int_0^h y\sqrt{r^2-(y-r)^2}dy}{2\int_0^h \sqrt{r^2-(y-r)^2}dy}$.

Solution 29.2

1. Just run the code as given.

2. For the case of $r = 6$ and $h = 2$, what are A and $\bar{y}$?

   We find $A = 12.39$ and $\bar{y} = 1.19\hat{y}$. Incidentally, if r and h are in cm, A is in cm^2 and $\bar{y}$ is in cm.

Solution 29.3

1. In Matlab, plot the position of the center of mass for the range $0 \le \phi \le 2 * pi$ (imagine you are plotting the position of a point on a disk as it completes two full rotations).

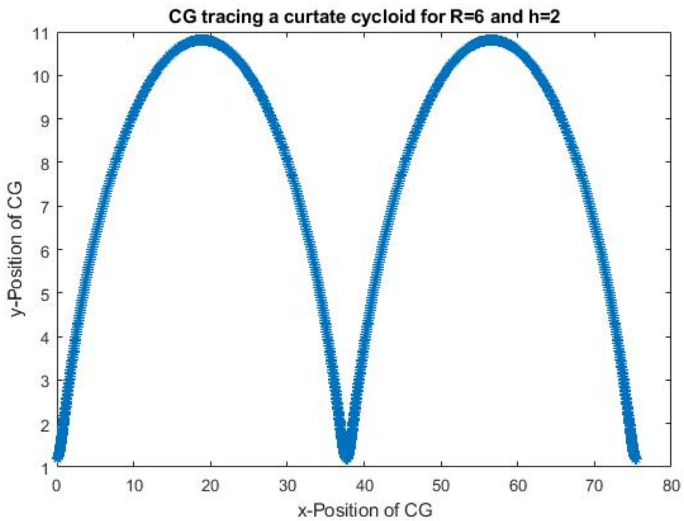


Figure 29.2: The position of the CG as the circular segment rolls is described by a curtate cycloid.

2. What are the equations for the position of the contact point, as a function of ϕ in the reference frame of the ramp?

   The equations for the contact point are:

$$x_{contact} = (r\phi)\hat{x}$$

$$y_{contact} = 0$$

3. Plot the position of the CG and the contact point on the same plot for the range $0 \leq \phi \leq \frac{pi}{4}$. Keep the number of points small so it is easy to interpret the relative position of the CG and the contact point. What do you notice?

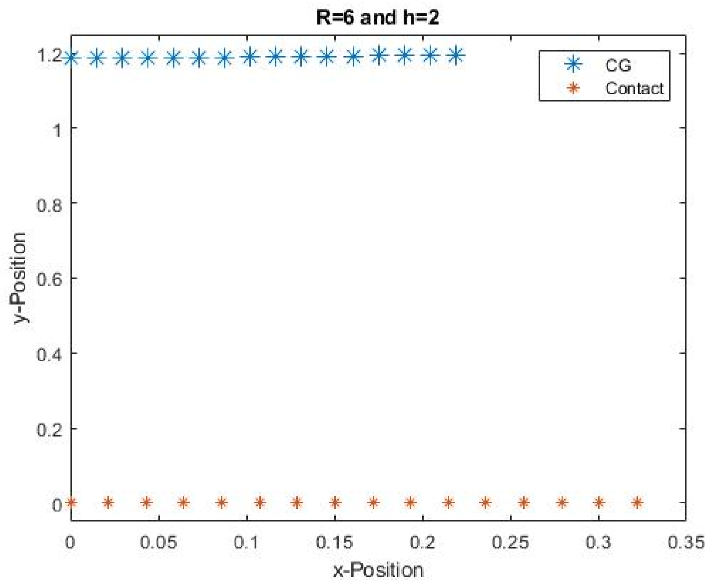


Figure 29.3: Position of the CG and contact point as the circular segment rolls.

As the circular segment rolls, the horizontal translation of the contact point leads the horizontal position of the CG. This creates a restoring moment.

Solution 29.4

1. Write expressions for the unit vectors $[\hat{x}, \hat{y}]$ that define the *ramp frame* in terms of the vectors $[\hat{i}, \hat{j}]$ which define the *global frame*.

$$\hat{x} = \cos(\theta)\hat{i} - \sin(\theta)\hat{j}$$
$$\hat{y} = \sin(\theta)\hat{i} + \cos(\theta)\hat{j}$$

2. To represent the positions of the CG and the contact point in the *global frame*, substitute the expressions for $[\hat{x}, \hat{y}]$ in terms of $[\hat{i}, \hat{j}]$ you found above into the equations for the position of the CG and contact point in the *ramp frame*.

$$x_{cg} = (r\phi - (r - \bar{y})sin(\phi))(\cos(\theta)\hat{i} - \sin(\theta)\hat{j})$$
$$y_{cg} = (r - (r - \bar{y})cos(\phi))(\sin(\theta)\hat{i} + \cos(\theta)\hat{j})$$
$$x_{contact} = (r\phi)(\cos(\theta)\hat{i} - \sin(\theta)\hat{j})$$
$$y_{contact} = 0$$

3. Following the substitution above, collect terms to find the vertical and horizontal positions of the CG and contact point in the *global frame*.

$$i_{cg} = ((r\phi - (r - \bar{y})\sin(\phi))\cos(\theta) + (r - (r - \bar{y})\cos(\phi))\sin(\theta))\hat{i}$$
$$j_{cg} = -((r\phi - (r - \bar{y})\sin(\phi))sin(\theta) + (r - (r - \bar{y})\cos(\phi))\cos(\theta))\hat{j}$$
$$i_{contact} = r\phi\cos(\theta)\hat{i}$$
$$j_{contact} = -r\phi\sin(\theta)\hat{j}$$

4. Based on your FBD of the circular segment on an inclined ramp, what condition must the horizontal positions of the CG and contact point must satisfy for the "ducky" to be in static equilibrium (assuming no slip)?

   To be in static equilibrium, the horizontal position of the CG (i_{CG}) and the contact point ($i_{contact}$ must be equal in the *global frame* ($\hat{i}$ direction). This is because the gravitational force, F_G acts in the $\hat{j}$ direction, and to ensure zero moment, there needs to be no moment arm. The two figures below illustrate these conditions. In Figure 29.5, the CG and contact point are not aligned in the $\hat{i}$ direction, creating a clockwise torque that will roll the segment. In Figure 29.6, the segment has reached an equilibrium condition where CG and contact point are aligned in the $\hat{i}$ direction and the moment arm is zero.

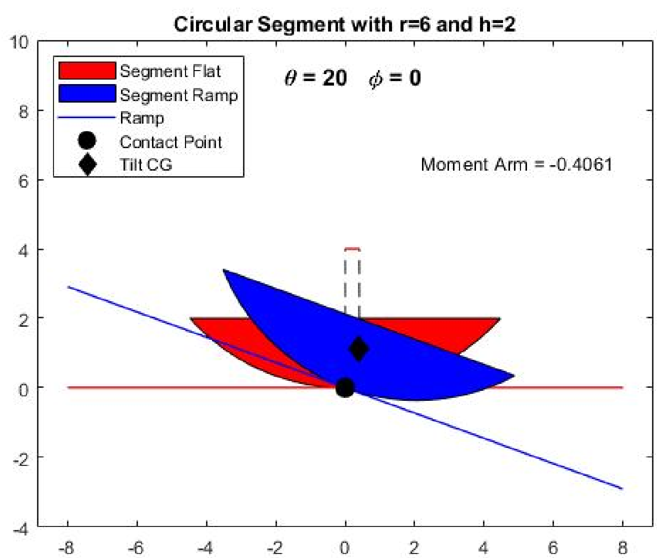


Figure 29.5: Simulation result for the circular segment on a ramp at angle $\theta = 20°$ and with $\phi = 0°$. A negative moment arm means a clockwise torque that will roll the segment towards $\phi > 0°$.

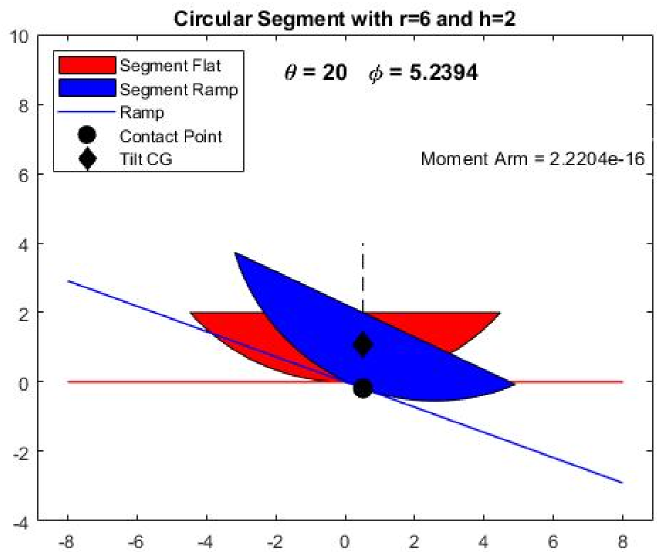


Figure 29.6: Simulation result for the circular segment on a ramp at angle $\theta = 20°$ and with roll angle $\phi = \phi_{equilibrium} = 5.24°$.

Solution 29.5

1. Find the static equilibrium angle for a circular segment with r=1.75 and h=1.5 for ϕ=5, 15, 25, 35 degrees. Is equilibrium possible for all of these ramp angles?

   $\phi_{equilibrium} = 5°, 16°, 32°$. An equilibrium angle is not solvable for $\theta = 35°$.

2. Use this script to visualize the circular segment and ramp angles given above. For the 35 degree case, you can set the ϕ angle manually to get a qualitative sense of what is happening. Why do you think a solution is not solvable?

   The solution for $\theta = 25°$ is shown in Figure 29.7. The same script can be used to generate the other angles. To better understand the $\theta = 35°$ case, we have manually set $\phi = 60$, which is just before the segment motion switches from a rolling motion to a hinge motion around the edge of the circular segment (the problem gets really crazy at that point, talk to Mark or Jeff if interested in digging deeper). The visualization shows that even at this extreme tilt angle, the righting arm is negative, creating a clockwise torque, and causing the segment to keep rolling. This can also be observed in Figure 29.9.

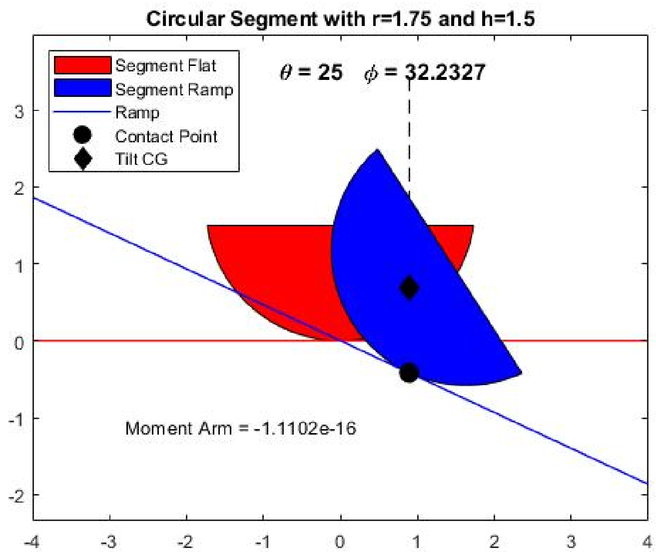


Figure 29.7: Simulation result for the circular segment on a ramp at angle $\theta = 25°$ and with roll angle $\phi = \phi_e = 32.23°$.

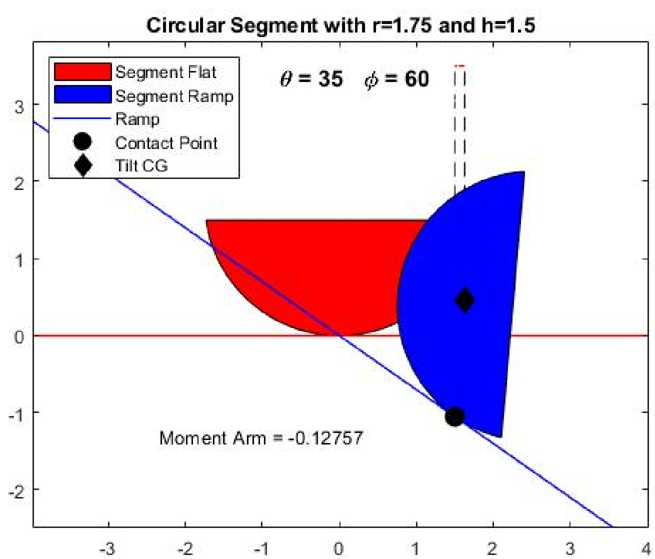


Figure 29.8: Simulation result for the circular segment on a ramp at angle $\theta = 35°$ and with roll angle $\phi = 60°$.

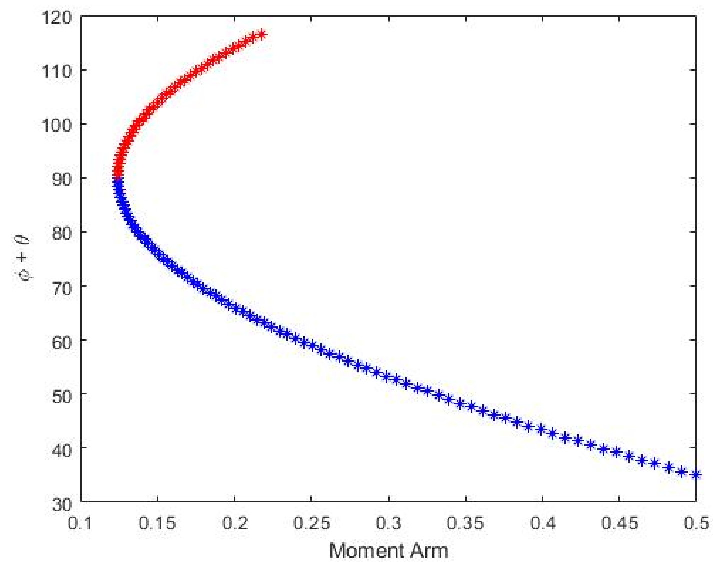


Figure 29.9: Moment arm as a function of $\phi + \theta$ for $\theta = 35°$. In this case, the moment arm is never reduced to zero and equilibrium is never reached. NOTE: this is an older figure where **positive** moment arm produces a clockwise torque instead of negative.

3. The `solve` function returns two ϕ angles for a given ramp angle θ. Are both results physically valid? Why or why not?

   No, only one of the solutions is valid. The equation for the positions of the CG and contact point assume a full circle, not a circular segment. For the case of a full circle with a CG not at the middle because of a non-uniform mass distribution, a second equilibrium position is possible, as shown in Figure 29.10 using the segment visualizer. This second equilibrium position is unstable. In our case, a circular segment can only roll so far before it flips over.

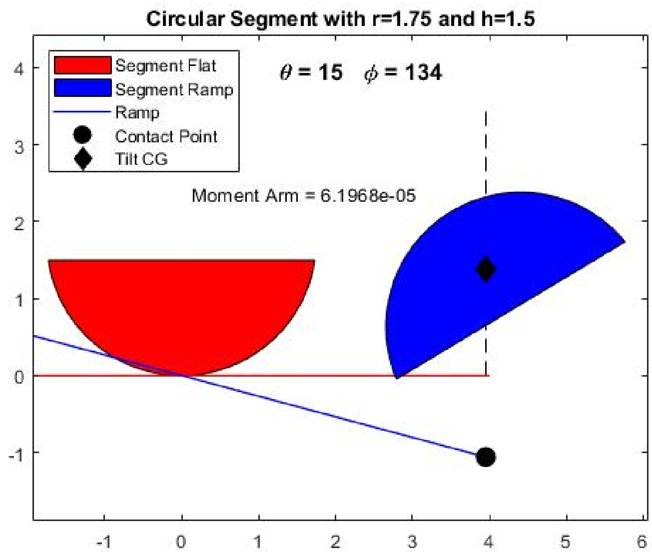


Figure 29.10: Simulation result for the circular segment on a ramp at angle $\theta = 15°$ and with roll angle $\phi = 134°$. This would represent the second, unstable, equilibrium condition if the segment were a full circle (with CG not at center).

Solution 29.6

1. Write the force equations in the ramp $[\hat{x}, \hat{y}]$ frame for the "ducky" in static equilibrium.

$$\sum F_{\hat{x}} = -F_{fs} + F_G \sin\theta = 0$$
$$\sum F_{\hat{y}} = N - F_G \cos\theta = 0$$

2. When will the "ducky" slip?

   From the definition of static friction, the maximum value of the static frictional force is $F_{fs} = \mu_s N$. If equilibrium requires a frictional force greater than this, we get slipping instead of equilibrium. Manipulating the equations above, we see that $N = F_G \cos\theta$ at equilibrium. Making the substitution into the equation for the $\hat{x}$ direction, at equilibrium $F_{fs} \cos\theta = F_G \sin\theta$, so the 'duck" will slip if $\mu_s F_G \cos\theta < F_G \sin\theta$ or, for a given angle θ, $\mu_s < \tan\theta$.

3. What can be done to increase the slip-angle for the "ducky"? Increasing the coefficient of friction μ_s will increase the maximum angle before slipping. Counter to intuition, increasing the mass of the "ducky" will not increase the maximum slipping angle.

Chapter 30

Night 5: Composite Duck Solution

30.1 The Ramp Challenge

We will now combine the skills developed during this assignment with those from the "Center of Mass Workshop" earlier in the semester. In Figure 30.1 is a three-dimensional shape with constant density. The shape is a composite- the bottom portion is a circular segment with r=4.75cm and h=3.8cm. The middle section is a rectangle 7cm x 2.3cm. The upper section is a circular segment with with r=4.75cm and h=2.5cm. Each segment is 5cm thick. The density is uniform across the composite ducky shape.

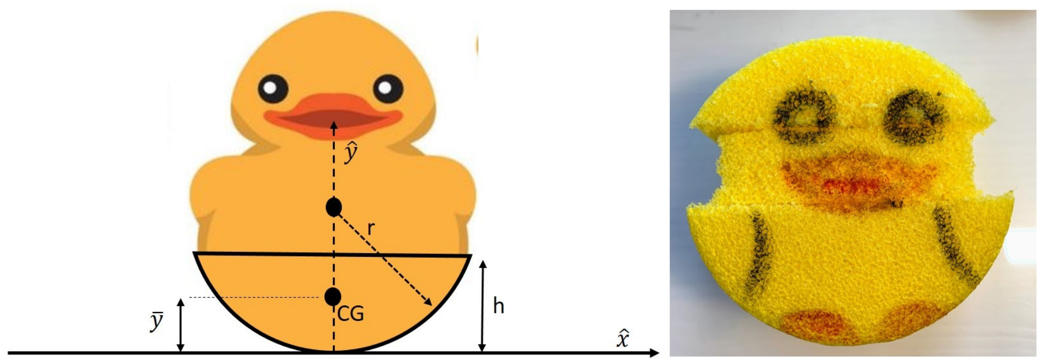


Figure 30.1: Pinterest vs. reality, am I right?

Exercise 30.1

For the final ramp challenge, you will perform the following calculations and put your answers in the "Ducky Night 5" quiz on Canvas. This exercise replaces the typical conceptual quiz that you usually have on night assignments. Submissions are due before the start of class of Thursday, April 2nd.
For the composite ducky shape:

1. Find the position of the center of mass for the composite made-from-Jeff's-foam-roller ducky.
2. Predict the equilibrium angle $\phi_{eq}(\theta)$ for $\theta = [2°, 4°]$.
3. (text entry on Canvas) Explain your process for finding the CG and ϕ_{eq}.

Module III

Robo: Curves, Outliers, and Optimization

Chapter 31

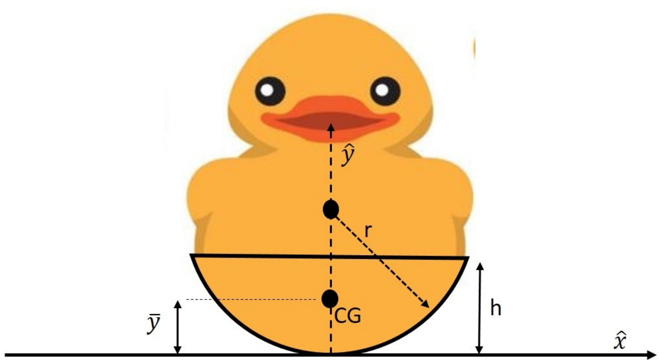
Day 1: Curves and Intro to Mobile Robotics

31.1 Schedule

- 1000-1005: Hello and Zoom Tech Time (everyone)
- 1005-1020: Debrief on Ducky Challenge Problem (breakout groups)
- 1020-1040: Ducky Concept Mapping (breakout groups)
- 1040-1115: Braitenberg Vehicles Start (breakout groups)
- 1115-1130: Coffee
- 1130-1145: Braitenberg Vehicles End (breakout groups)
- 1145-1225: Motion of Rigid Bodies (breakout groups)

31.2 Ducky D-Day

We'll start out by debriefing (in groups) on the problem where you were to predict the equilibrium angle of Jeff's fabulous foam ducky. We'll have our predictions (and measurements) posted for you to look at. Remember, you had to predict the equilibrium angle for 2 and 4 degrees of angle.



31.3 Ducky (and Module) Concept Mapping

Create a concept map of the various concepts you used to solve the ducky on a ramp problem (e.g., center of mass, static equilibrium, etc.). Expand your concept map to include any additional concepts from this mini-module that were not utilized in the ducky problem.

Note: while we've done this before, it's probably worth a quick reminder that one method of doing concept mapping is to write major concepts as bubbles and use lines to connect related concepts to each other. If you'd like, you can annotate particular connections to further define how they are related.

31.4 Robo Ninja Warrior

Welcome to Module 3 entitled "Robo Ninja Warrior." In this module you'll be learning some of the fundamental ideas, concepts, and algorithms that lie at the heart of robotics. Along the way we'll be revisiting some mathematical and analytical concepts we touched upon earlier in the semester. Not only will we be applying these concepts in new contexts and to new purposes, but also extending them in important ways. The module is structured around a series of challenges in which you will be programming your robot to perform various tasks autonomously. As you and your robot face tougher and tougher challenges, you will need to carefully integrate a wider range of techniques in order to successfully complete the task at hand.

When we typically do this module, we use the Neato BotVac as our robot platform (you'll learn more about the platform soon, but briefly it is a really nice robot platform that has some very advanced sensors). Of course we are now separated from our beloved Neatos, but never fear! The QEA team has got you covered. In this module we'll be using a Neato simulator, and we are excited about some of the cool things we'll be able to do when working in simulation. You'll learn more about the Neato in the night assignment section called "Meeto Your Neato."

31.5 What is a Robot?

Before diving into the challenges, let's take a step back and look at some definitions of the word "robot". Merriam-Webster provides three definitions of the word.

- a machine that looks like a human being and performs various complex acts (such as walking or talking) of a human being
- a device that automatically performs complicated often repetitive tasks
- a mechanism guided by automatic controls

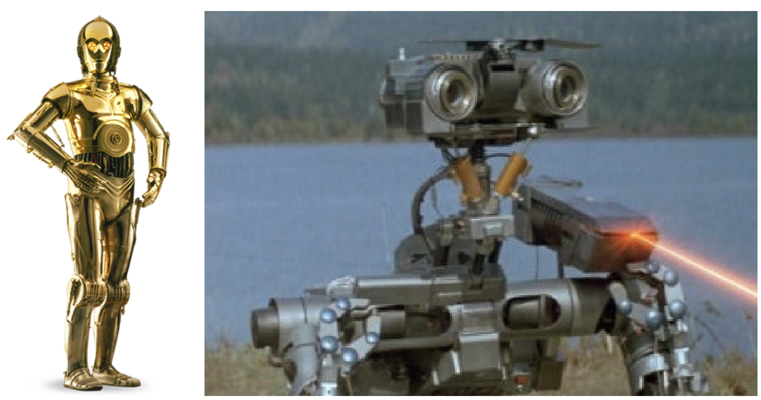


Figure 31.1: **Left:** C-3PO from the Star Wars franchise. **Right:** Johnny Five from the Short Circuit movies.

Exercise 31.1

Jot down a list of devices. For each device, determine which, if any, of the three definitions of the word "robot" apply.

These disparate definitions highlight the fact that depending on who you ask, the answer to the question "what is a robot?" will likely be very different. In a sense these three definitions proceed along a continuum of more restrictive to looser definitions, with the definition "a mechanism guided by automatic controls" being the loosest. Under this definition many things that you probably wouldn't intuitively call "robots" are just that. Take for example a thermostat. A thermostat is a mechanism that automatically regulates the heat in a building by comparing the measured temperature with a "desired" temperature. By the third definition, a thermostat is certainly a robot. At this point you may be thinking that if something as simple as a thermostat is a robot, then definition 3 must be completely bogus. After all, robots are supposed to be complicated and hard! However, today we will see that robots can in fact be quite simple. Further, simple robots can do some pretty complicated things.

31.6 Sensory-Motor Loops

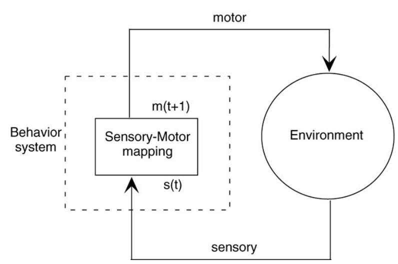


Figure 31.2: A schematic of a robot controlled by a sensory-motor loop.

We ended our last section on a somewhat cryptic note. If we seek to design simple robots, how on earth can they do complex things? The answer to this question lies in the fact that robots are not isolated machines, but instead interact with a complex and ever-changing world. A simple model that captures this idea is the sensory-motor loop (see Figure 31.2). The model situates the robot within an environment that it interacts with through two pathways.

The first is a **motor pathway** by which the robot executes actions which affect its environment. In our thermostat example these actions would be turning the heating system on or off. In a more conventional example of a robotic arm in a manufacturing plant, this could be operating the motors that control the joints of the arm.

The second is a **sensory pathway** by which the robot perceives its environment. In our thermostat example this could be a temperature sensor such as a thermocouple. In the case of a robotic arm in a manufacturing plant, this could be a potentiometer that measures the angle of each of the arm's joints or pressure sensors that measure contacts between the robot arm and other objects.

The "brain" of the robot, if you will, is defined by the box labeled "behavior system". In the general case you could imagine that the robot's brain might integrate multiple pieces of sensory information over time to form representations of the world around it. Take for example a robot mapping a building. The robot could build a progressively more detailed map by moving around in the building and collecting sonar readings (which provide an estimate of distance to objects in the world) over time. Putting aside this more complex form, let's restrict ourselves to robots with fairly simple behavior systems. **What about a robot that has no memory at all?** Such a robot would have to make all of its decisions based on its current sensory information.

Exercise 31.2

Design robots using the model in Figure 31.2. Restrict yourselves to robots that have no memory (i.e. ones that act at any moment in time directly based on their sensory input). To help get your creative juices flowing, it may help to make lists of sensors and actuators. You can then create interesting ideas by seeing what would happen if you paired a particular sensor with a particular actuator in a particular context. Don't worry too much about trying to design useful robots (whimsical is good), the goal here is to be creative and to think through the mental simulations necessary to understand how your robot would behave. Here are some suggestions for sensors and actuators.
Sensors: vibration sensor, microphone, camera, thermal camera, wheel rotation sensor, pressure sensor, light intensity sensor, laser range sensor, bump detector, temperature sensor, breathalyzer, etc. (Wikipedia has a good list).
Actuators (which is just a more general term for something that causes an action, e.g., a motor): DC motors, combustion engines, stepper motors, solenoids, speakers, lasers, LEDs, etc.

31.7 Grey Walter's Tortoises

Two very early examples of electric robots that worked using the principle of sensory-motor mappings were Grey Walter's robotic "Tortoises" Elmer and Elsie (see left panel of Figure 31.3). This YouTube video probably tells the story better than we possibly could.

As Grey Walter said himself, the robots behave as if they had a very simple two-cell nervous system that specifies the sensory-motor mapping (or behavior system). Despite this striking simplicity, the robots are capable of complex behavior such as obstacle avoidance and phototaxis (navigating towards the light that marks the charging kennel). This is an example of what we've been alluding to several times in this document: simple sensory-motor mappings can lead to complex behavior when put into a complex environment.

31.8 Braitenberg Vehicles

The pioneering work of Grey Walter was extended by a number of others. One particularly interesting line of research was conducted by Valentino Braitenberg. Valentino Braitenberg was interested in how vehicles controlled by very simple sensory-motor loops could execute behaviors that when viewed by humans would cause them to ascribe emotion and feelings of intelligence and intentionality to these vehicles. The name typically used to refer to these hypothetical robots is "Braitenberg Vehicles". While Braitenberg never actually built these vehicles (he was more interested in how these simple vehicles might inform various philosophical issues, particularly in the area of philosophy of mind), others have followed up and actually built these vehicles. Here is a video from a group at MIT that built several of Braitenberg's vehicles.

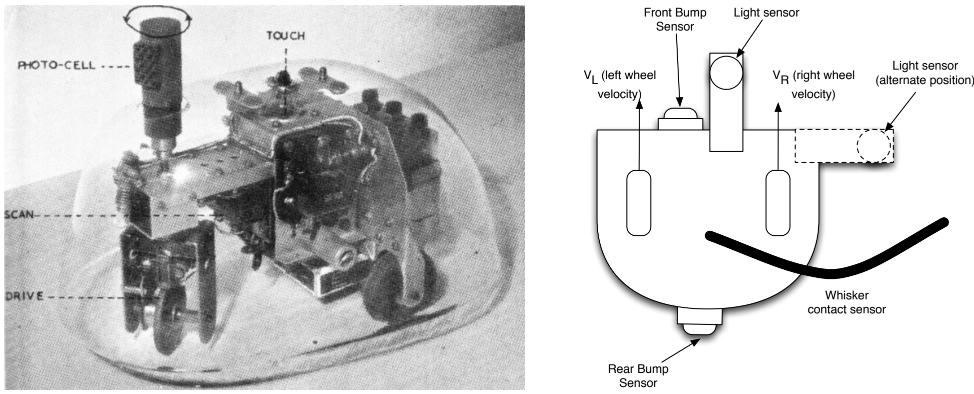


Figure 31.3: **Left:** Gray Walter's Tortoise Elsie. **Right:** A schematic of the vehicle from the real-life Braitenberg vehicles video. The robot is a differential drive vehicle with two wheels. We use V_L and V_R to refer to the velocities of the left and right wheels (positive is forward by convention). Also labeled are the other sensors on the robot, including the light sensor that reads higher values when exposed to more light, a whisker sensor that reads 0 when it is not in contact with something and 1 if it is, and two bump sensors that read 1 when they hit something and 0 otherwise.

A schematic of the vehicle (or robot) in the video is shown in the right panel of Figure 31.3. The robot has two motors that each control one of the robot's wheels. We can use the symbols V_L and V_R to refer to the velocities of each of the wheels (positive indicating a forward velocity). Using the framework from the previous section, these are our actuators. The robot also has a number of sensors. A light sensor outputs a continuous value which reads out larger values when in the presence of bright light and smaller values in the presence of low light (this is basically a one-pixel camera!). Additionally the robot has two bump sensors that have binary outputs. That is, they output 1 when they strike an object and 0 otherwise. Finally, the robot has a whisker sensor that is also binary and outputs a 1 when it contacts something and 0 otherwise.

Before getting on to the task of figuring out how one might program this robot, we need to understand a bit about how the drive systems of these robots work. The configuration shown in Figure 31.3 is known as *differential drive*. We will be thinking much more systematically about differential drive in the first robot challenge, but for now let's work to understand it from a qualitative perspective.

Exercise 31.3

To build a qualitative understanding of differential drive it helps to understand a few limiting cases. Good ones to start with are ones that involve the wheels moving at equal speeds in either the forward (positive) or reverse (negative) direction. In these cases the robot will either move forward in a straight line or backwards in a straight line. In these cases the speed of the robot is directly proportional to the speed of its wheels. Now, let's consider cases where the velocities of the two wheels are unequal. To help you with your intuition it might help to imagine the right wheel pulling either forwards or backwards on the right side of the robot and the left wheel pulling either forwards or backwards on the left side of the robot. Here is a potential list of limiting cases to consider. Make predictions about what would happen in these cases. It may help to sketch a couple of key frames (poses of the robot) over time.

1. What if V_L is positive and $V_R = -V_L$?
2. What if V_R is positive and $V_L = -V_R$?
3. What if $V_L = 0$ and V_R is positive?
4. What if $V_R = 0$ and V_L is positive?
5. What if $V_L = 0$ and V_R is negative?
6. What if $V_R = 0$ and V_L is negative?
7. What if V_R is positive and $V_L = \frac{1}{2}V_R$?
8. What if V_L is positive and $V_R = \frac{1}{2}V_L$?

31.9 Programming a Robot on a (Zoom) Whiteboard

Next, you will be programming this vehicle to perform the behaviors you saw in the video of the real life Braitenberg vehicles (the one with wary, obsessive, etc.). However, instead of programming the robot using a computer, you will be programming it on everyone's favorite piece of software: the Zoom! How can you program using a whiteboard (we imagine you conveniently might ask)?!? Remember, we are thinking of our robot's brain as a sensory-motor mapping. Translated into the language of mathematics this simply means that our robot program is specified by a function from sensors to motors!

There are many ways to represent a function. One way is as an equation. For instance, if we use the symbol ℓ to represent the reading of the robot's light sensor, then a robot that moves faster and faster as it see more and more light might have $V_L(\ell) = \ell^2$ and $V_R(\ell) = \ell^2$. Another way to represent a function is to define it graphically. You could draw a function that has ℓ on the x-axis and V_L on the y-axis. You could then sketch the relationship between those two quantities. Doing this graphically you could either have a quantitatively accurate sketch or a sketch that simply characterizes the function's qualitative behavior. For sensors that have binary values, like the bump sensors, you can exhaustively enumerate all conditions. For instance, here is the program of a robot that drives forward at a $0.5\frac{m}{s}$ until it rams into something

$$V_L(bump_F) = \begin{cases} 0.5\frac{m}{s}, & bump_F = 0 \\ 0\frac{m}{s}, & bump_F = 1 \end{cases} \tag{31.1}$$

$$V_R(bump_F) = \begin{cases} 0.5\frac{m}{s}, & bump_F = 0 \\ 0\frac{m}{s}, & bump_F = 1 \end{cases} \tag{31.2}$$

where $bump_F$ is the value of the forward bump sensor (1 when in contact with something, 0 otherwise).

Exercise 31.4

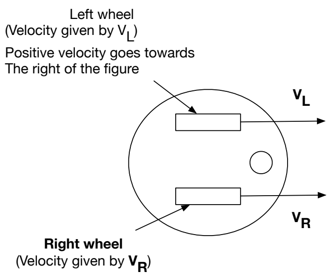
Work through generating robot programs to realize the behaviors in the video of the real life Braitenberg vehicles. Before jumping in, read the questions below and make sure to look at the template on the next page.

1. In order to validate your proposed program, it helps to do a quick whiteboard simulation. Sketch out a few key instants in time, what the robot's sensors would read, and what the wheel velocities would be. After the first couple you may be able to easily see whether or not your "program" is correct without actually sketching the key frames. **If this seems a little vague, check out the template on the next page, which we have made to help scaffold this activity. If it is still vague, check out the sample solution for the "wary" behavior**.

2. At least one of the behaviors cannot be reproduced without some primitive form of memory (although perhaps if you are very creative it can work). Which behaviors are these? How can you tell?

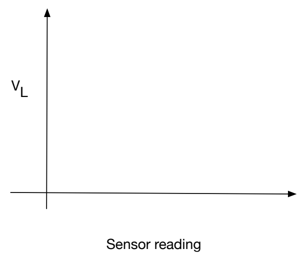
This template is designed for annotating in Zoom

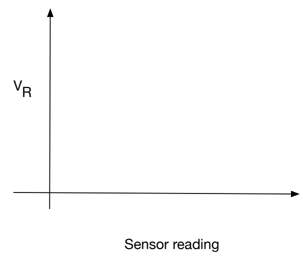
Step 1: Label your motors and sensors



Step 2: Define your Sensory-Motor Mapping

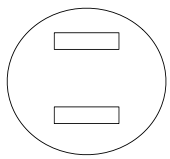
You could do this graphically using the provided axes, or as a function (as shown in the document)



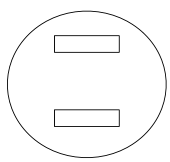


Step 3: Sketch some key frames by showing where the robot is in its environment, its sensor readings, and its resultant velocity

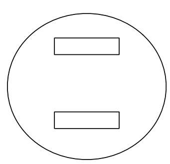
Time 1



Time 2



Time 3



Exercise 31.5

Next, Imagine your robot can remember a small amount of information. Specifically, your robot has access to a single flag that starts out with the value 0. Its value can be toggled from 0 to 1 or from 1 to 0 when a particular event occurs. For instance, if the light sensor reads a certain value, the robot might toggle its flag. The value of the flag can then inform the behavior of the robot. Given this new capability, implement any behaviors in the video that you couldn't before.

If you get done early and you're feeling excited about this, consider adding an additional light sensor to your robot. Now that you have two light sensors, you can get the robot to do a richer set of behaviors. Sketch the configuration of sensors on your new Braitenberg vehicle (equipped with two light sensors). Try to reproduce behaviors such as light seeking and light avoiding. For more ideas see the Wikipedia page on Braitenberg vehicles.

31.10 The Motion of Rigid Bodies

Our first challenge in Robo Ninja Warrior is going to involve the motion of rigid bodies (a rigid body just means an object that doesn't change shape as it moves around). You're probably all familiar with the idea of the velocity of a body from your previous coursework in physics, but you probably haven't thought too much about the concept of the angular velocity of a rigid body. The angular velocity of rigid bodies will turn out to be important for programming our robot to move along precisely defined paths. In this section, you'll be starting to get a feel for the concept.

We are going to explore the concept of angular velocity using an "app" for your phone. Please download and install "SensorKinetics" to your phone, and open the "Gyroscope sensor" (note: for newer iPhones, the app does not seem to display properly. If you run into trouble, download Sensors Toolbox - Multitool by Andrew Neal).

Exercise 31.6

Qualitative: For each of the questions below, you should "plot" the data on the board, and interpret the data qualitatively in terms of the coordinate system of the phone.

1. Place your phone on the table and spin it, in place, counterclockwise. Note that you should be spinning it about an axis that is orthogonal to the face of the phone. Which axis is this on the data graph? What happens if you spin it clockwise?

2. Now spin the phone about the two other axes. Which axis is which? Which direction is clockwise and which is counterclockwise? Clearly draw the coordinate system the phone is using - use the unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$.

3. What happens if you move the phone along a straight line (on the table) without turning the phone?

4. What happens if you move the phone uniformly in a circle **without turning the phone**? i.e. the orientation of the phone does not change.

5. What happens if you move the phone uniformly in a circle **while turning the phone at the same time**? i.e. imagine the phone is a car and you are driving in a circle.

Now that we've explored angular velocity using the phone, let's make it quantitative. We will use the following notation for the angular velocity

$$\boldsymbol{\omega} = \omega_x \hat{\mathbf{x}} + \omega_y \hat{\mathbf{y}} + \omega_z \hat{\mathbf{z}}$$

so that ω_x is the component of angular velocity corresponding to rotation about the x-axis and so on. The units of angular velocity are in radians per second, e.g., rotating in a complete circle in one second would be a 2π radian/second rotation.

Exercise 31.7

Quantitative: For each of the questions below, you should "plot" the data on the board (please interpret these quotes as meaning "sketch" what you think the data would look like, don't start writing MATLAB code), and interpret the data quantitatively in terms of the coordinate system of the phone.

1. Predict the angular velocity if you place the phone down on the table and then spin the phone in place so that it spins once in 5 seconds. Confirm your prediction using the data.

2. Predict the angular velocity if you uniformly move the phone in a circle in 5 seconds, and confirm your prediction using the data. Does it matter how large the circle is?

3. What type of motion would give rise to a constant ω_z of 2 radians per second for 5 seconds, with both $\omega_x = 0$ and $\omega_y = 0$? Confirm your prediction with the phone.

4. What type of motion would give rise to a sinusoidal ω_z with amplitude of 2 radians per second, and a period of 10 seconds, with both $\omega_x = 0$ and $\omega_y = 0$.

5. What type of motion would give rise to a constant ω_z and ω_x of 2 radians per second for 5 seconds, with $\omega_y = 0$. Confirm your prediction with your phone.

6. Sketch a graph of your own choosing of ω_x, ω_y, and ω_z and challenge yourself to produce it using the phone!

Solution 30.1

For the final ramp challenge, you will perform the following calculations and put your answers in the "Ducky Night 5" quiz on Canvas. This exercise replaces the typical conceptual quiz that you usually have on night assignments. Submissions are due before the start of class of Thursday, April 2nd.

For the composite ducky shape:

1. Find the position of the center of mass for the composite made-from-Jeff's-foam-roller ducky.

   The composite ducky in Figure 30.1 is made up of three distinct sections, as seen in Figure 30.2. Each of these sections has symmetry across the $\hat{y}$ axis, so we know that $\bar{x_1} = \bar{x_2} = \bar{x_3} = 0\hat{x}$. Because the composite shape has uniform density and uniform thickness, we can simplify the equation for the center of mass to:

   $$\bar{y}_{composite} = \frac{1}{A_{total}} \sum(\bar{y_1} A_1 + \bar{y_2} A_2 + \bar{y_3} A_3)$$

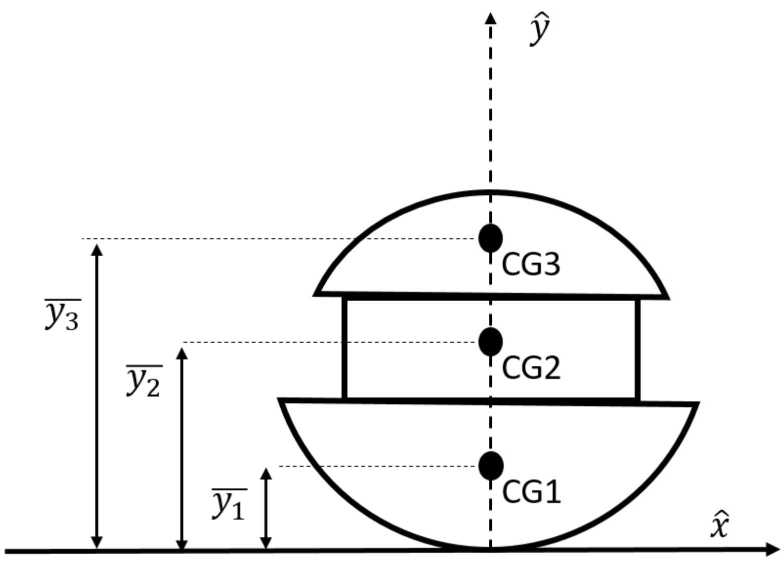
   

   Figure 30.2: Center of mass locations for the composite ducky.

   For segment 1, we can use the existing code for a circular segment with r=4.75cm and h=3.8cm to find $\bar{y_1}$=2.21cm. For segment 2, we know the CG would be located at half of the height of the rectangle because of symmetry. Looking at Figure 30.2, we know that $\bar{y_2}$ needs to be measured from the $\hat{x}$ axis, so $\bar{y_2} = h_1 + \frac{1}{2}h_2$ =3.8cm+1.15cm=4.95cm. For segment three, we again use the existing code for the circular segment with r=4.75cm and h=2.5cm, which gives a value of $\bar{y}$=1.47cm. However, we must recognize that this solutions the segment is upright, not inverted as in the composite ducky. Again looking at Figure 30.2, we find that $\bar{y_3} = h_1 + h_2 + (h_3 - 1.47cm)$=7.13cm.

Segment	$\bar{y}$ in cm	Area in cm^2
1	2.21	26.48
2	4.95	16.10
3	7.13	14.90

Table 30.1: Parameters for the three segments that make up the composite duck.

The segment parameters in Table 30.1 can be used in the equation

$$\bar{y}_{composite} = \frac{1}{A_{total}} \sum (\bar{y_1} A_1 + \bar{y_2} A_2 + \bar{y_3} A_3) = 4.25cm$$

2. Predict the equilibrium angle $\phi_{eq}(\theta)$ for $\theta = [2^\circ, 4^\circ]$.

   Adding segments two and three to create the composite ducky shape raises the center of mass compared to circular segment one alone. The translation of the contact point, however is not changed. The position of the center of mass still moves in a curtate cycloid as the roll angle, ϕ, of the ducky changes.

   To predict the equilibrium angle of the composite duck, the Duck_simplesolve.m script works well with r=4.75cm and h=3.8cm. The only modification needed is to manually set the variable *ybar* to be equal to 4.25cm. You can also try the script Composite_Duck.m that solves both parts of the composite problem together. It requires the functions DuckyParam.m and DuckySolve.m. The equilibrium angles are given in Table 30.2.

Ramp Angle θ	Equilibrium Angle ϕ_{eq}
2°	17.47°
4°	37.77°

Table 30.2: Equilibrium angles for the composite duck.

3. (text entry on Canvas) Explain your process for finding the CG and ϕ_{eq}.

   Please see extended answers to each part above.

Please see below for experimental validation of equilibrium angle (acknowledging limitations in measurement accuracy and ducky fabrication skill).

Figure 30.3: Ramp at $\theta = 2°$.

Figure 30.4: Composite duck in equilibrium at $\theta = 2°$.

Figure 30.5: Composite duck in equilibrium at $\theta = 2^\circ$ and $\phi_{eq} \approx 17^\circ$.

Figure 30.6: Ramp at $\theta = 4°$.

Figure 30.7: Composite duck in equilibrium at $\theta = 4°$.

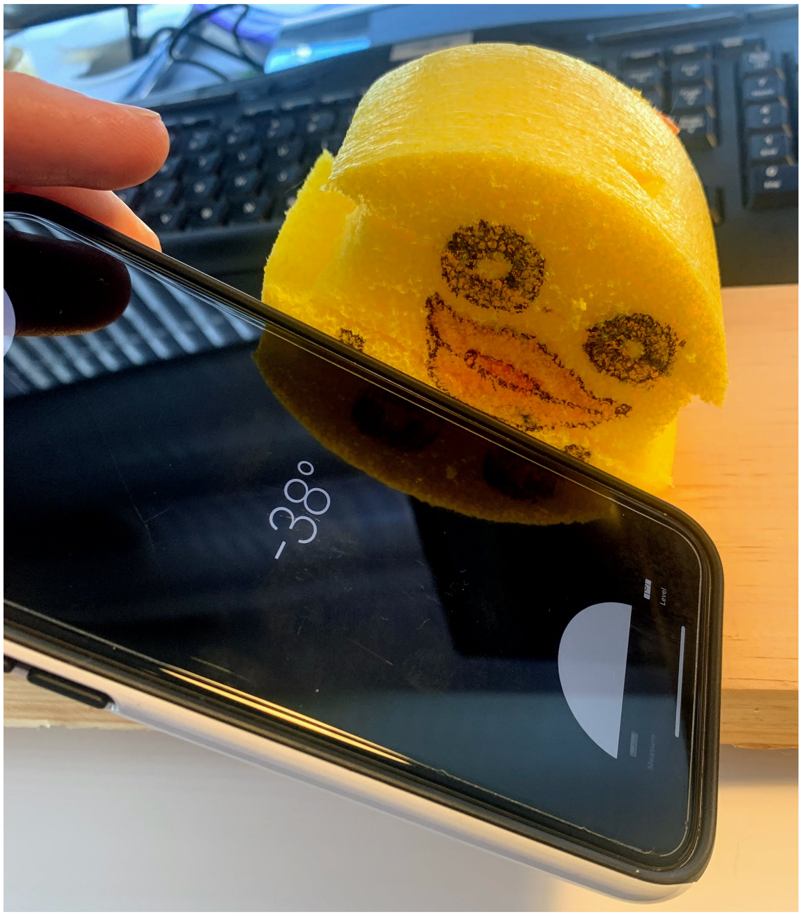


Figure 30.8: Composite duck in equilibrium at $\theta = 4°$ and $\phi_{eq} \approx 38°$.

Solution 31.1

Here are a few examples to get you started.

- A thermostat (2) and (3)
- R2D2 (3) (maybe also (2)?)
- A toaster oven (2) (most toasters don't really have a true control system)
- C3PO (1) and (3)
- Spot from Boston Dynamics (3) (maybe (2)?)

Solution 31.2

Sorry, no solution here! You all will come up with cooler things than we ever could.

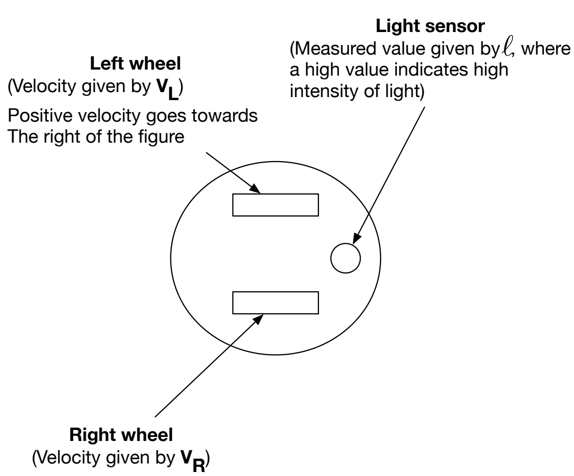
Solution 31.3

1. The robot will rotate in place in the clockwise direction if you were looking down at the robot from above (i.e., spin to the right).
2. The robot will rotate in place in the counterclockwise direction (i.e., spin to the left).
3. The robot will rotate in the counterclockwise direction with the center of rotation at its left wheel.
4. The robot will rotate in the clockwise direction with the center of rotation at its right wheel.
5. The robot will rotate in the clockwise direction with the center of rotation at its left wheel.
6. The robot will rotate in the counterclockwise direction with the center of rotation at its right wheel.
7. The robot will rotate in the counterclockwise direction while also translating forward.
8. The robot will rotate in the clockwise direction while also translating forward.

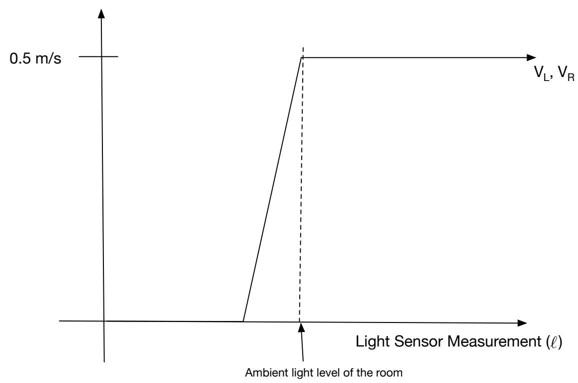
Solution 31.4

1. Here is a potential solution for the *wary* behavior. You could also do this one similarly to the example given above by writing out the functions as in Equations 31.1-31.2, but we wanted to show you an example of writing the "program" graphically.

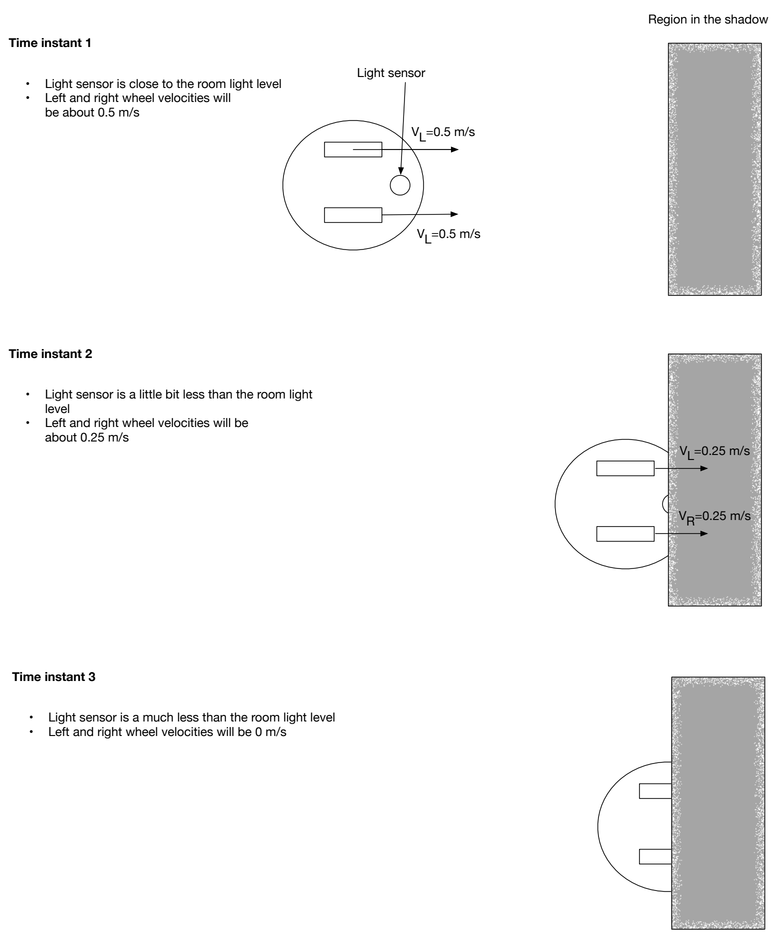
   First, sketch the robot and label the sensors (in this case the light sensor) and its actuators (the wheels). Assign notation to relevant quantities.



Write down V_L and V_R as functions of the light sensor reading ℓ.



Sketch some keyframes to build confidence in your solution.



2. If you are strict about defining the velocity just based on the sensor readings (and not on the current velocities of the wheels), then the obsessive behavior would be impossible to implement without some sort of memory. The reason would be that when you are halfway between the two shoes the robot wouldn't know whether to go left or right (since the sensor of "no bump detected" would be the same in either case. If you imagine the robot's motor system working a bit differently (e.g., maintaining the current velocity unless otherwise commanded), then you could argue that no memory is needed.

Solution 31.5

For the obsessive behavior you could do something like.
Start with $flag = 0$.

$$V_L(bump_F, bump_R, flag) = \begin{cases} -0.5\frac{m}{s}, & bump_F = 1 \\ 0.5\frac{m}{s}, & \text{else if } bump_R = 1 \\ 0.5\frac{m}{s}, & \text{else if } flag = 0 \\ -0.5\frac{m}{s}, & \text{else if } flag = 1 \end{cases} \tag{31.3}$$

$$V_R \text{ same as } V_L \tag{31.4}$$

We would also need to update the flag whenever a bump occurs by changing it from a 1 to a 0 or vice-versa.

Solution 31.6

1. The graph shows a positive angular velocity int the z-direction. When you spin it clockwise, the sign of the angular velocity becomes negative.

2. On my iPhone the positive y-axis runs along the long axis of the phone away from the bottom of the phone (where the home button would be on older models). I can tell this since a counterclockwise rotation about this axis yields a positive angular velocity in the y-direction. Using similar logic, I can tell that my phone's positive x-axis points to the right, across the short edge of the phone screen. You can see this in the iOS documentation using this link: https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events.

3. The gyroscope reads values close to 0 for all axes.

4. The gyroscope reads values close to 0 for all axes.

5. The gyroscope reads a relatively constant value in the z-direction. The magnitude of the value corresponds to how long it takes to go around the circle (radius does not matter) and the sign tells us which direction we are traveling around the circle.

Solution 31.7

1. The angular velocity should be approximately $\frac{2\pi}{5}$radians/s $\hat{\mathbf{z}}$ (assuming we spin it counter clockwise).

2. The angular velocity should be approximately $\frac{2\pi}{5}$ radians/s $\hat{\mathbf{z}}$ (assuming we move counter-clockwise around the circle). The radius of the circle doesn't matter.

3. Any motion that involves rotating the phone purely about $\hat{\mathbf{z}}$ (the screen is aligned to $\hat{\mathbf{z}}$) at the specified rate. An example would be moving the phone uniformly around a circle a total of $\frac{5s \times 2\ radians/s}{2\pi radians/revolution} \approx 1.6\ revolutions$ in 5 seconds (other motions will work as well, e.g., the linear motion can be anything you want).

4. While many motions could achieve this, one possibility is to spin the phone about its $\hat{\mathbf{z}}$ axis, starting with slow rotation, gradually ramping up in speed, then down in speed, reversing direction, getting faster, slowing down, and ultimately reversing direction. This complete cycle should take 10 seconds.

5. Spinning the phone around an axis pointing out of the phone at an angle of 45 degrees to the right (good luck actually doing this!). The phone should rotate once in about 2.25 seconds.

6. Let's see what you can come up with!

Chapter 32

Night 1: Parametric Curves and Motion

Learning Objectives

Concepts

- Determine the radii and origin for a given circle or ellipse in 2D space.
- Design a 2D parametric vector function whose trace for the parameter, from [0 - 2pi], will produce an ellipses of desired radii.
- Discuss the relationship between a curve on interval [a,b] and its following vectors: Tangent, Normal and Binormal
- State the physical significance of a non-zero curvature or torsion of a vector function.

MATLAB skills

- Alter MATLAB starter code for a unit circle for the general case of a circle or ellipse of given dimensions, centered at [h,k].
- Compute the curvature and torsion of a curve using the symbolic math toolbox.
- Compute the length of a curve using the symbolic math toolbox.

Overview and Orientation

In Module 1 we introduced the concept of parametric curves. We are now going to return to this subject, but in a more general framework using vectors.

This assignment draws from material in multi-variable and vector calculus, and any textbook in these subjects will have related material. Keywords include **parametric curves**, **curve length**, and **line integral**. Good sources include **Paul's Online Math Notes**—the section on Calculus III. The relevant **Kahn** videos are also useful.

Visualizing parametric curves and associated quantities like tangent vectors etc is a critical part of developing an understanding of the connection between parametric curves and the motion of moving bodies. Linda has created a video about using **CalcPlot3D** to visualize parametric curves—it is available here.

Although you can evaluate the derivatives and integrals in this assignment by hand, we would also like you to use the **Symbolic Toolbox in MATLAB** for this purpose—we will be using this toolbox throughout this robotics module. You will find a starter MATLAB script here that uses symbolic Matlab to work through the circle example below.

32.1 Parametric Curves

In Module I we considered curves in the plane, represented by either an explicit function, $y = f(x)$ or $x = f(y)$, an implicit function $f(x, y) = 0$, or a set of parametric equations

$$x = f(u), y = g(u)$$

where we treat u as a parameter. Each value of u defines a point $(f(u), g(u))$ which we can plot. If we collect all the points defined by $u \in [a, b]$, then we get a parametric curve. In Module 1, we did not limit ourselves to curves in the plane. For example, in 3D we defined

$$x = f(u), y = g(u), z = h(u)$$

and the collection of points so defined trace out a curve in 3D.

An alternative to these coordinate definitions involves representing each point with a position vector, $\mathbf{r}(u)$. Since the position vector depends on a single parameter u, the end of the position vector traces out a curve in space. If we limit ourselves to 3D, we will usually use the following notation

$$\mathbf{r}(u) = x(u)\hat{\mathbf{i}} + y(u)\hat{\mathbf{j}} + z(u)\hat{\mathbf{k}},\ u \in [a, b]$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are the standard Cartesian unit vectors. In a sense the vector function $\mathbf{r}(u)$ lifts the interval $[a, b]$ and transforms it in order to produce a curve in 3D space.

One major advantage of this notation is that we can take derivatives of this vector function with respect to the parameter u

$$\begin{aligned} \mathbf{r}'(u) &= \frac{d}{du}\left(x(u)\hat{\mathbf{i}} + y(u)\hat{\mathbf{j}} + z(u)\hat{\mathbf{k}}\right), \\ &= x'(u)\hat{\mathbf{i}} + y'(u)\hat{\mathbf{j}} + z'(u)\hat{\mathbf{k}}, \end{aligned}$$

since the Cartesian unit vectors are constant. We can interpret the derivative as follows: for any given value of u this vector is tangent to the parametric curve. At times we might be more interested in a unit tangent vector $\hat{\mathbf{T}}$, which we can obtain by normalizing the derivative

$$\hat{\mathbf{T}} = \frac{\mathbf{r}'}{|\mathbf{r}'|}$$

The trace of the tangent vector tip as u varies, $\hat{\mathbf{T}}(u)$, also produces a curve; taking its derivative ($\frac{d}{du}$), produces a vector, tangent to $\hat{\mathbf{T}}(u)$, which we will define as the normal vector to the original curve, $\mathbf{r}(u)$. The unit normal vector $\hat{\mathbf{N}}$ is therefore

$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|}$$

Finally, we can use both the unit tangent vector and the unit normal vector to define a unit binormal vector $\hat{\mathbf{B}}$ as follows

$$\hat{\mathbf{B}} = \hat{\mathbf{T}} \times \hat{\mathbf{N}}$$

Taken together, these three unit vectors are mutually orthogonal and therefore form an orthonormal basis of 3D space. This is known as the Frenet-Serret frame, and some applications can be found on the Wikipedia page concerning `Frenet-Serret Formulas`.

In addition to these unit vectors, parametric curves in 2D and 3D are often described in terms of their curvature κ and torsion τ. The curvature is the normalized rate of change of the unit tangent vector

$$\kappa = \frac{|\hat{\mathbf{T}}'|}{|\mathbf{r}'|}$$

and measures how quickly a curve is changing direction - a large value of the curvature means the curve is changing direction rapidly. The curvature is always non-negative. A straight-line would have zero curvature.

The torsion is the normalized rate of change of the unit binormal vector in the direction opposite to the unit normal

$$\tau = -\hat{\mathbf{N}} \cdot \frac{\hat{\mathbf{B}}'}{|\mathbf{r}'|}$$

and measures the rate at which a curve is twisting out of the plane - a large value of the torsion means the curve is rapidly twisting out of the plane. A curve in the plane has zero torsion. The torsion can be positive or negative, and convention dictates that a right-handed curve has positive torsion. As Wikipedia says "Intuitively, curvature measures the failure of a curve to be a straight line, while torsion measures the failure of a curve to be planar."

Now that we know how to define a general parametric curve, we are ready to compute with it. For example, we could compute the length of the curve. In order to do so, let's lay down a set of points in the u-domain separated by Δu. Each point is mapped to the space curve, so the approximate length of each section of the curve is,

$$\Delta L = |\mathbf{r}'(u)|\Delta u$$

Refining this for smaller Δu and then summing up the pieces results in the integral

$$L = \int_a^b |\mathbf{r}'(u)| \, du$$

which defines the length of the curve.

The Parametric Circle

Let's take a few minutes to digest some of the theory by looking at a very common example—the parametric representation for a circle of radius R, centered at the origin in the xy-plane. If we define the position vector

$$\mathbf{r}(u) = R\cos u \,\hat{\mathbf{i}} + R\sin u \,\hat{\mathbf{j}}, \ u \in [0, 2\pi]$$

then the circle is traced out once in the counterclockwise direction starting at $(R, 0)$. In this way, we can identify the parameter u as being the angle from the x-axis to a point on the circle.

Let's compute (by hand) the various unit tangent vectors, the curvature, the torsion, and the length of the curve. The first derivative of the position vector is

$$\mathbf{r}'(u) = -R\sin u \,\hat{\mathbf{i}} + R\cos u \,\hat{\mathbf{j}}$$

The unit tangent vector is

$$\begin{aligned} \hat{\mathbf{T}} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \\ &= -\sin u \,\hat{\mathbf{i}} + \cos u \,\hat{\mathbf{j}} \end{aligned}$$

and is tangent to the circle (you could check that $\mathbf{r} \cdot \hat{\mathbf{T}} = 0$). The unit normal vector is

$$\begin{aligned} \hat{\mathbf{N}} &= \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|} \\ &= -\cos u \, \hat{\mathbf{i}} - \sin u \, \hat{\mathbf{j}} \end{aligned}$$

and is normal to the circle, pointing inward (you could check that $\hat{\mathbf{T}} \cdot \hat{\mathbf{N}}$). The unit binormal vector is

$$\begin{aligned} \hat{\mathbf{B}} &= \hat{\mathbf{T}} \times \hat{\mathbf{N}} \\ &= (-\sin u \, \hat{\mathbf{i}} + \cos u \, \hat{\mathbf{j}}) \times (-\cos u \, \hat{\mathbf{i}} - \sin u \, \hat{\mathbf{j}}) \\ &= (\sin^2 u + \cos^2 u) \, \hat{\mathbf{k}} \\ &= \hat{\mathbf{k}} \end{aligned}$$

and is out of the plane of the circle. The curvature of the circle is

$$\begin{aligned} \kappa &= \frac{|\hat{\mathbf{T}}'|}{|\mathbf{r}'|} \\ &= \frac{1}{R} \end{aligned}$$

and is inversely proportional to the radius of the circle. The torsion of the circle is

$$\begin{aligned} \tau &= -\hat{\mathbf{N}} \cdot \frac{\hat{\mathbf{B}}'}{|\mathbf{r}'|} \\ &= 0 \end{aligned}$$

is zero because the circle is a plane curve and the binormal vector is a constant. For the length of the curve, we have

$$|\mathbf{r}'(u)| = +\sqrt{(-R)^2 \sin^2 u + (R)^2 \cos^2 u}$$

and

$$|\mathbf{r}'(u)| = R$$

and the integral becomes

$$L = \int_0^{2\pi} R du = 2\pi R$$

which is the circumference of a circle of radius R as expected!

Note: For the following problems, computing the various vectors by hand can get very messy as the complexity of the function increases. It is recommended that you first visualize the parametric curves using CalcPlot3D, and then learn how to you use the Symbolic Toolbox in MATLAB to find the expressions for each vector symbolically

Exercise 32.1

Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is a circle centered at (x_0, y_0) with radius R.

1. Use **CalcPlot3D** to visualize the circle for different centers and radii, along with the unit tangent and normal vectors.

   We already examined the vector function for a circle centered at the origin; how would that vector function change so that it's tail starts at (x_0, y_0) rather than $(0, 0)$?

2. Determine (by hand) the unit tangent vector.
3. Determine (by hand) the unit normal vector.
4. Determine (by hand) the unit binormal vector.
5. Determine (by hand) the curvature and torsion.

   Does the curvature or torsion depend on the location of the circle center?

6. Set up the integral to compute the perimeter of the circle, and evaluate it by hand.

   Given that it's a circle, what do you expect the length of the curve to be for u=0 to 2π?

Exercise 32.2

A helix in 3D can be defined by the vector function

$$\mathbf{r}(u) = a\cos u\,\hat{\mathbf{i}} + a\sin u\,\hat{\mathbf{j}} + bu\,\hat{\mathbf{k}}, a > 0, b > 0, u \geq 0$$

1. Use **CalcPlot3D** to visualize the helix for different centers and radii, along with the unit tangent, normal, and binormal vectors.
2. Determine (by hand) the unit tangent vector.
3. Determine (by hand) the unit normal vector.
4. Determine (by hand) the unit binormal vector.
5. Determine (by hand) the curvature and torsion.
6. Set up the integral to compute the length of the helix corresponding to 5 complete turns, and evaluate it (by hand).

The helix is a great example of a curve in 3D, and shows up in all sorts of places. Visualizing the unit vectors associated with the curve is more challenging because they live in 3D, and we are asking you to define the domain so that the helix completes 5 turns. You should also find that the curvature and torsion of the helix are constant, and indeed any space curve with constant curvature and torsion is an helix. We often define a helix in terms of its radius a, and its pitch $2\pi b$, which is the height of the helix after one complete turn.

Exercise 32.3

Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is an ellipse centered at (x_0, y_0) with semi-major axis a, and semi-minor axis b, $b < a$.

1. Use **CalcPlot3D** to visualize the ellipse for different centers and semi-major axes, along with the unit tangent, normal, and binormal vectors.
2. Determine the unit tangent vector using the **Symbolic Toolbox** in MATLAB.
3. Determine the unit normal vector using the **Symbolic Toolbox** in MATLAB.
4. Determine the unit binormal vector using the **Symbolic Toolbox** in MATLAB..
5. Determine the curvature and torsion using the **Symbolic Toolbox** in MATLAB.
6. Set up the integral to compute the perimeter of the ellipse using the **Symbolic Toolbox** in MATLAB, and numerically evaluate it for a specific case of $(x_0, y_0), a, b$.

You will find that the integral for the perimeter of the ellipse involves elliptic integrals which cannot be evaluated using elementary functions. Check out this page at the American Mathematical Society for an interesting review.

32.2 Motion of Bodies

So far we've been talking about the intrinsic geometry of curves. However, there is an intimate connection between the geometry of curves and the motion of bodies. Assume a body is moving in space and is described by a position vector $\mathbf{r}(t)$ defined in terms of a fixed frame. In component form we write

$$\mathbf{r}(t) = x(t)\hat{\mathbf{i}} + y(t)\hat{\mathbf{j}} + z(t)\hat{\mathbf{k}}$$

where t is time. The units of $\mathbf{r}(t)$ are length.

The derivative with respect to time of this position vector defines the velocity of the body

$$\mathbf{v}(t) = \mathbf{r}'(t) = x'(t)\hat{\mathbf{i}} + y'(t)\hat{\mathbf{j}} + z'(t)\hat{\mathbf{k}}$$

and the second derivative with respect to time of this position vector defines the acceleration of this body

$$\mathbf{a}(t) = \mathbf{r}''(t) = x''(t)\hat{\mathbf{i}} + y''(t)\hat{\mathbf{j}} + z''(t)\hat{\mathbf{k}}$$

It will be instructive now to ask how the velocity vector and acceleration vector are oriented with respect to the path of the body as it moves through space. If we treat t as a parameter, and view the motion of the body as a parametric curve, then we can use the machinery of parametric curves to answer this question.

Let's start with the velocity vector. We know from our earlier work that the derivative $\mathbf{r}'(t)$ is tangent to the curve, which implies that the velocity must be tangent to the curve. Recall that earlier we defined the unit tangent to be

$$\hat{\mathbf{T}}(t) = \frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}$$

Re-arranging this definition we can express the velocity in terms of the unit tangent vector $\hat{\mathbf{T}}$

$$\mathbf{v}(t) = v(t)\hat{\mathbf{T}}(t)$$

where $v(t) = |\mathbf{r}'(t)|$ is the linear speed of the body in the tangent direction, and we explicitly note that $\hat{\mathbf{T}}(t)$ is a function of t, i.e. the unit tangent direction changes as we move along the curve.

What about the acceleration $\mathbf{a}(t)$? If we take the derivative of the velocity we see that

$$\begin{aligned} \mathbf{a}(t) &= \frac{d}{dt}\left(v(t)\hat{\mathbf{T}}(t)\right) \\ &= \frac{dv}{dt}\hat{\mathbf{T}} + v\frac{d\hat{\mathbf{T}}}{dt} \end{aligned}$$

because both the magnitude and direction of velocity can change in time, and we need to use the product rule for derivatives. Recall from earlier that the rate of change of the unit tangent vector is related to the unit normal vector

$$\frac{d\hat{\mathbf{T}}}{dt} = |\hat{\mathbf{T}}'|\hat{\mathbf{N}}$$

and the magnitude of the rate of change of the unit tangent vector is related to the curvature

$$|\hat{\mathbf{T}}'| = \kappa|\mathbf{r}'|$$

Since $|\mathbf{r}'| = v$ we have

$$\frac{d\hat{\mathbf{T}}}{dt} = \kappa v \hat{\mathbf{N}}$$

so that the acceleration becomes

$$\mathbf{a}(t) = \frac{dv}{dt}\hat{\mathbf{T}} + \kappa v^2 \hat{\mathbf{N}}$$

Let's pause for a moment to consider this. We have expressed the acceleration of a moving body in terms of the unit tangent vector and the unit normal vector, so we will often talk about the tangential acceleration and normal acceleration of a body. You might recall that because velocity is a vector, it can accelerate by changing in magnitude ("speed") or changing in direction. The magnitude of the tangential acceleration is just the rate of change of the linear speed. On the other hand, the magnitude of the normal acceleration is caused by the constant change in the direction of the velocity vector as it moves around the circle. This normal acceleration is proportional to the square of the linear speed and the curvature of the path along which the body is moving and is directed normal to the curve.

The Circle Example

Consider the example of a body moving in a circle of radius R at some constant linear speed v. What is the position vector for such a body? Well, we know that it should look like a parametric circle, so let's define

$$\begin{aligned} x &= R\cos(\omega t) \\ y &= R\sin(\omega t) \end{aligned}$$

where ω is a variable that we need to define. In vector notation we have

$$\mathbf{r}(t) = R\cos(\omega t)\hat{\mathbf{i}} + R\sin(\omega t)\hat{\mathbf{j}}$$

Using the approach from the previous section we can compute the unit tangent vector and unit normal vector to this curve,

$$\hat{\mathbf{T}} = -\sin(\omega t)\hat{\mathbf{i}} + \cos(\omega t)\hat{\mathbf{j}}$$

$$\hat{\mathbf{N}} = -\cos(\omega t)\hat{\mathbf{i}} - \sin(\omega t)\hat{\mathbf{j}}$$

The velocity vector is

$$\mathbf{v}(t) = -R\omega\sin(\omega t)\hat{\mathbf{i}} + R\omega\cos(\omega t)\hat{\mathbf{j}}$$

which we expect to be tangential to the curve. One way to check is to find the tangential and normal components of velocity by evaluating $\mathbf{v}\cdot\hat{\mathbf{T}}$ and $\mathbf{v}\cdot\hat{\mathbf{N}}$. We obtain

$$\mathbf{v}\cdot\hat{\mathbf{T}} = R\omega$$

$$\mathbf{v}\cdot\hat{\mathbf{N}} = 0$$

which means that the velocity can be expressed as

$$\mathbf{v}(t) = R\omega\hat{\mathbf{T}}$$

We should now recognize ω as the angular velocity of the body as it moves in uniform circular motion. The circular motion is uniform because it moves a constant number of radians per second. The acceleration vector is

$$\mathbf{a}(t) = -R\omega^2\cos(\omega t)\hat{\mathbf{i}} - R\omega^2\sin(\omega t)\hat{\mathbf{j}}$$

Since the linear speed of motion is constant ($v = R\omega$) we would expect the acceleration to be in the normal direction only. Again we can compute the tangential and normal components of acceleration by evaluating $\mathbf{a} \cdot \hat{\mathbf{T}}$ and $\mathbf{a} \cdot \hat{\mathbf{N}}$. We obtain

$$\mathbf{a} \cdot \hat{\mathbf{T}} = 0$$
$$\mathbf{a} \cdot \hat{\mathbf{N}} = R\omega^2$$

which means that the acceleration can be expressed as

$$\mathbf{a}(t) = R\omega^2 \hat{\mathbf{N}}$$

A body in uniform circular motion has no tangential component of acceleration - it is purely normal. Using the earlier expression for the angular velocity we could just as easily write the normal component of the acceleration as

$$a_N = \frac{v^2}{R}$$

which hopefully agrees with some results you saw a long time ago in school. It also connects to our earlier expression since the normal component of acceleration should be κv^2, and $\kappa = \frac{1}{R}$ for a circle.

Exercise 32.4

Consider a body undergoing non-uniform circular motion with radius R so that it's angular velocity is linearly increasing with time, $\omega = \alpha t$, i.e. the body starts off at rest and speeds up. Assume position is measured in meters and time is measured in seconds.

1. Define a position vector for this moving body. How long should it take the body to make its first loop? How about its second loop?
2. Visualize the motion in CalcPlot3D.
3. Determine (by hand) the velocity of this moving body. How does the linear speed depend on R and α?
4. Determine (by hand) the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions. Explain the result.

You've probably seen and thought about uniform circular motion before. Often it is easier to understand a concept by looking at one just a little more complicated, so here we ask you to consider non-uniform circular motion. Although the path is the same curve, the motion is different.

Exercise 32.5

Consider a body moving in 3D with position vector

$$\mathbf{r}(t) = a\cos(ct)\hat{\mathbf{i}} + a\sin(ct)\hat{\mathbf{j}} + bct\hat{\mathbf{k}}$$

Assume that the position is measured in meters and time is measured in seconds.

1. Describe the path that the body should take. Interpret the variables a, b, and c.
2. Visualize the motion in CalcPlot3D.
3. Determine (by hand) the velocity of this moving body. How does the linear speed depend on a, b, and c?
4. Determine (by hand) the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.

Here we have a body moving in 3D on a curve that we have seen before. Pay attention to the direction of the unit tangent and unit normal as the body moves along this curve.

Exercise 32.6

Consider a body moving with the following position vector

$$\mathbf{r}(t) = a\cos(ct)\hat{\mathbf{i}} + b\sin(ct)\hat{\mathbf{j}}$$

where $a > 0$, $b > 0$, and $c > 0$. Assume that position is measured in meters and time in seconds.

1. Describe the path that the body takes. How long does it take to return to its starting position?
2. Visualize the motion in CalcPlot3D.
3. Use the Symbolic Toolbox in MATLAB to determine the velocity of this moving body. How does the linear speed depend on a, b, and c?
4. Use the Symbolic Toolbox in MATLAB to determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.

Here we have a body moving on a path that corresponds to a curve that we studied earlier, and so in many ways this is nothing new. However, the motion of a body consists of both the path and "how" it moves along it. Interpreting the velocity and acceleration in terms of the variables a, b, and c will help you build your understanding of these concepts.

Solutions for Module 3, Night 1 : Parametric Curves and Motion

- Exercise 1: Find a vector function r(u) in the plane whose trace is a circle
  centered at (x0 , y0) with radius R.

```
In[33]:= r[u_] = {xo + R * Cos[u], yo + R * Sin[u], 0}
Out[33]= {xo + R Cos[u], yo + R Sin[u], 0}
```

- b) What is the unit tangent vector

```
In[34]:= That[u_] = Simplify[r'[u] / Sqrt[r'[u].r'[u]], {u ≥ 0, R > 0}]
Out[34]= {-Sin[u], Cos[u], 0}
```

- c) What is the unit normal vector?

```
In[35]:= Nhat[u_] = Simplify[That'[u] / Sqrt[That'[u].That'[u]], {u ≥ 0, R > 0}]
Out[35]= {-Cos[u], -Sin[u], 0}
```

- d) What is the unit binormal vector

```
In[36]:= Bhat[u_] = Simplify[That[u] × Nhat[u], {u ≥ 0, R > 0}]
Out[36]= {0, 0, 1}
```

- e) What is the curvature and torsion?

```
In[37]:= κ[u_] = Simplify[Sqrt[That'[u].That'[u]] / Sqrt[r'[u].r'[u]], {u ≥ 0, R > 0}]
```

Out[37]= $\frac{1}{R}$

```
In[38]:= τ[u_] = Simplify[-Nhat[u] × (Bhat'[u] / Sqrt[r'[u].r'[u]]), {u ≥ 0, R > 0}]
Out[38]= {0, 0, 0}
```

f) Setup and evaluate the integral to find the perimeter of the circle

In[39]:=
```
Integrate[Simplify[Sqrt[r'[u].r'[u]], {u ≥ 0, R > 0}], {u, 0, 2 * Pi}]
```
Out[39]= $2\pi R$

Exercise 2 : A helix in 3 D can be defined by the vector function r (u) = acos[u] $\hat{i}$ + asin[u] $\hat{j}$ + bu $\hat{k}$, a > 0, b > 0, u >= 0

In[40]:=
```
r[u_] = {a * Cos[u], a * Sin[u], b * u}
```
Out[40]= {a Cos[u], a Sin[u], b u}

b) What is the unit tangent vector?

In[41]:=
```
That[u_] = Simplify[r'[u] / Sqrt[r'[u].r'[u]], {u ∈ Reals, a > 0, b > 0}]
```
Out[41]= $$\left\{-\frac{a\,\mathrm{Sin}[u]}{\sqrt{a^2+b^2}}, \frac{a\,\mathrm{Cos}[u]}{\sqrt{a^2+b^2}}, \frac{b}{\sqrt{a^2+b^2}}\right\}$$

c) What is the unit normal vector?

In[42]:=
```
Nhat[u_] = Simplify[That'[u] / Sqrt[That'[u].That'[u]], {u ∈ Reals, a > 0, b > 0}]
```
Out[42]= {-Cos[u], -Sin[u], 0}

d) What is the unit binormal vector?

In[43]:=
```
Bhat[u_] = Simplify[That[u] × Nhat[u], {u ∈ Reals, a > 0, b > 0}]
```
Out[43]= $$\left\{\frac{b\,\mathrm{Sin}[u]}{\sqrt{a^2+b^2}}, -\frac{b\,\mathrm{Cos}[u]}{\sqrt{a^2+b^2}}, \frac{a}{\sqrt{a^2+b^2}}\right\}$$

e) What is the curvature and torsion?

In[44]:=
```
κ[u_] =
  Simplify[Sqrt[That'[u].That'[u]] / Sqrt[r'[u].r'[u]], {u ∈ Reals, a > 0, b > 0}]
```

Out[44]= $\frac{a}{a^2 + b^2}$

In[45]:=
```
τ[u_] = Simplify[-Nhat[u] × (Bhat'[u] / Sqrt[r'[u].r'[u]]), {u ∈ Reals, a > 0, b > 0}]
```

Out[45]= $\left\{\frac{b\,\mathrm{Cos}[u]^2}{a^2 + b^2}, \frac{b\,\mathrm{Sin}[u]^2}{a^2 + b^2}, 0\right\}$

f) Find the length of the curve for 5 turns of the helix

In[46]:=
```
Integrate[Simplify[Sqrt[r'[u].r'[u]], {u ∈ Reals, a > 0, b > 0}], {u, 0, 10 * pi}]
```

Out[46]= $10\sqrt{a^2 + b^2}\;\mathrm{pi}$

Exercise 3 : Find a vector function r (u) in the plane whose trace is an ellipse centered at (x0, y0) with semi - major axis a, and semi - minor axis b, b < a.

a) Visualize it for different centers and semi - major and semi - minor axes

In[48]:=
```
r[u_] = {xo + a * Cos[u], yo + b * Sin[u], 0}
```

Out[48]= $\{xo + a\,\mathrm{Cos}[u], yo + b\,\mathrm{Sin}[u], 0\}$

b) What is the unit tangent vector

In[49]:=
```
That[u_] = Simplify[r'[u] / Sqrt[r'[u].r'[u]], {u ∈ Reals, a > 0, b > 0}]
```

Out[49]= $\left\{-\frac{a\,\mathrm{Sin}[u]}{\sqrt{b^2\,\mathrm{Cos}[u]^2 + a^2\,\mathrm{Sin}[u]^2}}, \frac{b\,\mathrm{Cos}[u]}{\sqrt{b^2\,\mathrm{Cos}[u]^2 + a^2\,\mathrm{Sin}[u]^2}}, 0\right\}$

c) What is the unit normal vector?

In[50]:=
```
Nhat[u_] = Simplify[That'[u] / Sqrt[That'[u].That'[u]], {u ∈ Reals, a > 0, b > 0}]
```

Out[50]= $\left\{-\frac{b\,\mathrm{Cos}[u]}{\sqrt{b^2\,\mathrm{Cos}[u]^2 + a^2\,\mathrm{Sin}[u]^2}},\ -\frac{a\,\mathrm{Sin}[u]}{\sqrt{b^2\,\mathrm{Cos}[u]^2 + a^2\,\mathrm{Sin}[u]^2}},\ 0\right\}$

d) What is the unit binormal vector?

In[51]:=
```
Bhat[u_] = Simplify[That[u] × Nhat[u], {u ∈ Reals, a > 0, b > 0}]
```

Out[51]= {0, 0, 1}

e) What is the curvature and torsion?

In[52]:=
```
κ[u_] = Simplify[Sqrt[That'[u].That'[u]] / Sqrt[r'[u].r'[u]], {u ∈ Reals, a > 0, b > 0}]
```

Out[52]= $\frac{a\,b}{\left(b^2\,\mathrm{Cos}[u]^2 + a^2\,\mathrm{Sin}[u]^2\right)^{3/2}}$

In[53]:=
```
τ[u_] = Simplify[-Nhat[u] × (Bhat'[u] / Sqrt[r'[u].r'[u]]), {u ∈ Reals, a > 0, b > 0}]
```

Out[53]= {0, 0, 0}

f) Setup and numerically evaluate the integral to find the perimeter of the ellipse

In[54]:=
```
NIntegrate[Simplify[Sqrt[r'[u].r'[u]], u ∈ Reals] /. {a → 1, b → 2}, {u, 0, 2 * Pi}]
```

Out[54]= 9.68845

Exercise 4: Consider a body undergoing non-uniform circular motion with radius R so that it’s angular velocity is linearly increasing with time, $\omega = \alpha\, t$, i.e. the body starts off at rest and speeds up. Assume that position is measured in

meters and time in seconds.

- (a) Define a position vector for this moving body. How long should it take the body to make its first loop? How about its second loop?

```
In[95]:= r[t_] = {R * Cos[α * t * t], R * Sin[α * t * t], 0}
         r[0]
         r[Sqrt[2 * Pi / α]]
         r[Sqrt[4 * Pi / α]]
```

Out[95]= $\{R \operatorname{Cos}[t^2 \alpha], R \operatorname{Sin}[t^2 \alpha], 0\}$

Out[96]= $\{R, 0, 0\}$

Out[97]= $\{R, 0, 0\}$

Out[98]= $\{R, 0, 0\}$

- (c) Determine the velocity of this moving body. How does the linear speed depend on R and α?

```
In[80]:= v[t_] = r'[t]
```

Out[80]= $\{-2 R t \alpha \operatorname{Sin}[t^2 \alpha], 2 R t \alpha \operatorname{Cos}[t^2 \alpha], 0\}$

```
In[82]:= speed[t_] = Simplify[Sqrt[r'[t].r'[t]], Assumptions → R > 0 && α >  0 && t >= 0]
```

Out[82]= $2 R t \alpha$

- (d) Determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.Explain the result.

```
In[85]:= a[t_] = Simplify[v'[t], Assumptions → R > 0 && α > 0 && t >= 0]
```

Out[85]= $\{-2 R \alpha (2 t^2 \alpha \operatorname{Cos}[t^2 \alpha] + \operatorname{Sin}[t^2 \alpha]), 2 R \alpha (\operatorname{Cos}[t^2 \alpha] - 2 t^2 \alpha \operatorname{Sin}[t^2 \alpha]), 0\}$

```
In[86]:= That[t_] = Simplify[r'[t] / Norm[r'[t]], Assumptions → R > 0 && α > 0 && t >= 0]
```

Out[86]= $\{-\operatorname{Sin}[t^2 \alpha], \operatorname{Cos}[t^2 \alpha], 0\}$

In[87]:=
```
Nhat[t_] = Simplify[That'[t] / Norm[That'[t]], Assumptions → R > 0  && α > 0 && t  >= 0]
```
Out[87]= $\{-\mathrm{Cos}[t^2\,\alpha],\,-\mathrm{Sin}[t^2\,\alpha],\,0\}$

In[88]:=
```
aT[t_] = Simplify[a[t].Nhat[t]]
```
Out[88]= $4\,R\,t^2\,\alpha^2$

In[89]:=
```
aN[t_] = Simplify[a[t].That[t]]
```
Out[89]= $2\,R\,\alpha$

Exercise 5: Consider a body moving in 3D with position vector $r(t) = \;\mathrm{acos(ct)}\,\hat{i} + \mathrm{asin(ct)}\,\hat{j} + \mathrm{bct\ khat}$ Assume that position is measured in meters and time in seconds.

- (a) Describe the path that the body takes. Interpret the variables a, b, and c.

We saw this path earlier - it is an helix. The parameter a is the radius if the helix. The parameter c is related to how long it takes to make a single loop. The parameter b tells us how quickly the helix climbs as it turns.

In[100]:=
```
r[t_] = {a * Cos[c * t], a * Sin[c * t], b * c * t}
```
Out[100]= $\{a\,\mathrm{Cos}[c\,t],\,a\,\mathrm{Sin}[c\,t],\,b\,c\,t\}$

- (c) Determine the velocity of this moving body. How does the linear speed depend on a, b, and c?

In[101]:=
```
v[t_] = r'[t]
```
Out[101]= $\{-a\,c\,\mathrm{Sin}[c\,t],\,a\,c\,\mathrm{Cos}[c\,t],\,b\,c\}$

In[102]:=
```
speed[t_] = Simplify[Sqrt[r'[t].r'[t]], Assumptions → c > 0 && b > 0  && a >  0 && t >= 0]
```
Out[102]= $\sqrt{a^2 + b^2}\;c$

(c) Determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.

In[103]:=
```
a[t_] = Simplify[v'[t], Assumptions → b > 0 && a > 0 && c > 0 && t >= 0]
```
Out[103]= $\{-a\, c^2 \,\text{Cos}[c\, t],\ -a\, c^2\, \text{Sin}[c\, t],\ 0\}$

In[104]:=
```
That[t_] = Simplify[r'[t] / Norm[r'[t]], Assumptions → c > 0 && b > 0 && a > 0 && t >= 0]
```
Out[104]= $\left\{-\frac{a\,\text{Sin}[c\,t]}{\sqrt{a^2+b^2}},\ \frac{a\,\text{Cos}[c\,t]}{\sqrt{a^2+b^2}},\ \frac{b}{\sqrt{a^2+b^2}}\right\}$

In[105]:=
```
Nhat[t_] =
  Simplify[That'[t] / Norm[That'[t]], Assumptions → c > 0 && b > 0 && a > 0 && t >= 0]
```
Out[105]= $\{-\text{Cos}[c\,t],\ -\text{Sin}[c\,t],\ 0\}$

In[106]:=
```
aT[t_] = Simplify[a[t].Nhat[t]]
```
Out[106]= $a\,c^2$

In[107]:=
```
aN[t_] = Simplify[a[t].That[t]]
```
Out[107]= 0

Exercise 6: Consider a body moving with the following position vector

$$r(t) = a\cos(ct)\,\hat{i} + b\sin(ct)\,\hat{j}$$

where a > 0, b > 0, and c > 0. Assume that position is measured in meters and time in seconds.

- (a) Describe the path that the body takes. How long does it take to return to its starting position?

We know that this is an elliptical path with semi-major axis a and semi-minor axis b. We know that a complete orbit of the ellipse occurs when the parameter ct=2π, so the time it takes to return to the starting position is t=(2π)/c.

```
In[57]:= r[t_] := {a * Cos[c * t], b * Sin[c * t], 0}
         r[0]
         r[2 * Pi / c]
Out[58]= {a, 0, 0}
Out[59]= {a, 0, 0}
```

(c) Determine the velocity of this moving body. How does the linear speed depend on a, b, and c?

```
In[60]:= v[t_] = r'[t]
Out[60]= {-a c Sin[c t], b c Cos[c t], 0}
```

```
In[61]:= speed[t_] = Simplify[Sqrt[r'[t].r'[t]], Assumptions → c > 0 && b > 0  && a > 0 && t >= 0]
```

Out[61]= $c\sqrt{b^2\,\mathrm{Cos}[c\,t]^2 + a^2\,\mathrm{Sin}[c\,t]^2}$

(c) Determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.Explain the result.

```
In[62]:= a[t_] = Simplify[v'[t], Assumptions → b > 0  && a > 0 && c > 0 && t >= 0]
```

Out[62]= $\{-a\,c^2\,\mathrm{Cos}[c\,t],\ -b\,c^2\,\mathrm{Sin}[c\,t],\ 0\}$

```
In[63]:= That[t_] = Simplify[r'[t] / Norm[r'[t]], Assumptions → c > 0 && b > 0  && a > 0 && t >= 0]
```

Out[63]= $\left\{-\frac{a\,\mathrm{Sin}[c\,t]}{\sqrt{b^2\,\mathrm{Cos}[c\,t]^2 + a^2\,\mathrm{Sin}[c\,t]^2}},\ \frac{b\,\mathrm{Cos}[c\,t]}{\sqrt{b^2\,\mathrm{Cos}[c\,t]^2 + a^2\,\mathrm{Sin}[c\,t]^2}},\ 0\right\}$

```
In[64]:= Nhat[t_] =
          Simplify[That'[t] / Norm[That'[t]], Assumptions → c > 0 && b > 0  && a > 0 && t >= 0]
```

Out[64]= $\left\{-\frac{b\,\mathrm{Cos}[c\,t]}{\sqrt{b^2\,\mathrm{Cos}[c\,t]^2 + a^2\,\mathrm{Sin}[c\,t]^2}},\ -\frac{a\,\mathrm{Sin}[c\,t]}{\sqrt{b^2\,\mathrm{Cos}[c\,t]^2 + a^2\,\mathrm{Sin}[c\,t]^2}},\ 0\right\}$

In[65]:=
```
aT[t_] = Simplify[a[t].Nhat[t]]
```

Out[65]=
$$\frac{\sqrt{2}\ a\ b\ c^2}{\sqrt{a^2 + b^2 + \left(-a^2 + b^2\right) \operatorname{Cos}[2\ c\ t]}}$$

In[66]:=
```
aN[t_] = Simplify[a[t].That[t]]
```

Out[66]=
$$\frac{\left(a^2 - b^2\right) c^2 \operatorname{Sin}[2\ c\ t]}{\sqrt{2}\ \sqrt{a^2 + b^2 + \left(-a^2 + b^2\right) \operatorname{Cos}[2\ c\ t]}}$$

Chapter 33

Day 2: Curves, Motion, and Neato

33.1 Schedule

- 1000-1005: Tech time and awkward conversation (main room)
- 1005-1025: Debrief (breakout rooms)
- 1025-1100: Parametric Avenue (breakout rooms)
- 1100-1115: Coffee
- 1115-1225: Differential Drive in Action (breakout rooms) (note: optional walkthrough of problem 8, see below)
- 1210-1225: Walkthrough of Simple Wheel Velocity Experiment (exercise 8) (main room, optional)
- 1225-1230: Fill out day survey (individually)

Learning Objectives

Concepts

- From $\mathbf{r}(t)$ data for a rigid body,
  - estimate average speed and distance traveled.
  - approximate $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$, and $\hat{\mathbf{B}}$ at various points on the space curve.
- From an $\mathbf{r}(t)$ parametric vector function for a rigid body,
  - approximate its space curve (i.e., path of travel)
  - compute the vector functions for $\mathbf{r}'(t)$ and $\mathbf{r}"(t)$

Matlab Skills

- Compute the angular velocity vector and its amplitude using the symbolic toolbox
- Direct the NEATO to drive forward
- Use NEATO position information to verify a motion model.

33.2 Debrief [20 minutes]

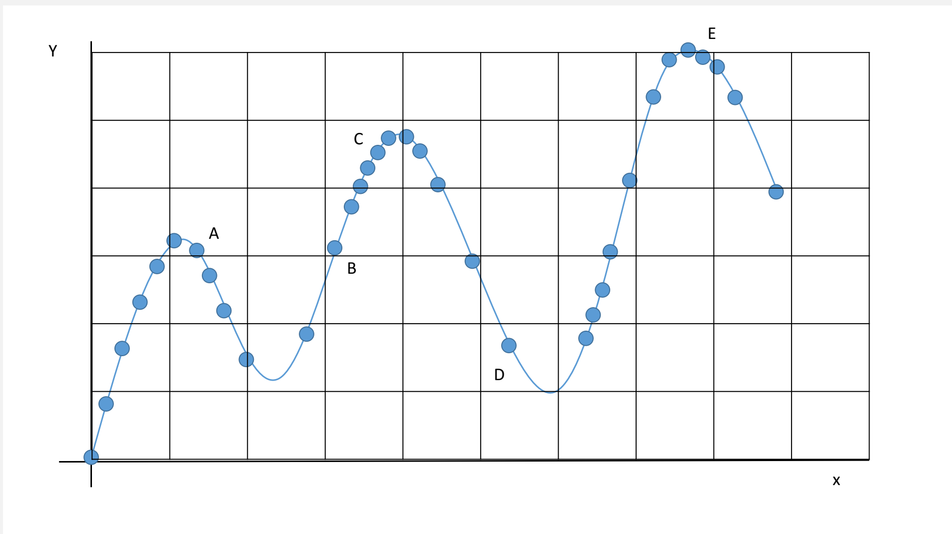
Discuss the overnight with your table-mates, and make a list of concepts you feel solid on, and concepts you feel shaky on. Make a list of plus and deltas for this assignment. This debrief is short, but you will be applying the concepts from the overnight during the next exercise.

33.3 The NEATO Goes for a Drive on Parametric Avenue [35 minutes]

Here we will work to connect the work you did over the weekend to the upcoming challenge.

A hypothetical NEATO has just gone for a drive, and we have recorded its position vector $\mathbf{r}(t)$. Its path is shown below, and the dots indicate its position at equally-spaced points in time. (You can download a printable version of this image here). It starts in the bottom left corner and moves along the path until it reaches the last point. Let's assume that this is a relatively long drive - the grid spacing is 1m and the time between samples is 1s.

Exercise 33.1



1. Roughly how long does the NEATO travel for? Roughly how far does it travel? Roughly what is its average speed?
2. Thinking about the curve only, draw the unit tangent vector $\hat{\mathbf{T}}$ and unit normal vector $\hat{\mathbf{N}}$ at each of the points indicated: A, B, C, D and E.
3. Thinking about the motion of the NEATO now, choose one of the points (A,B,C,D or E) and compute the magnitude of the velocity vector. Draw the velocity vector to scale at that point. Do the same for the acceleration vector, decomposing it into tangential and normal components. How do these relate to the unit tangent and unit normal vectors? (Do more points if you have time!)
4. Draw a picture of the NEATO at A, B, C, D, and E, i.e. you are looking down on the NEATO as it drives along the curve.
5. The NEATO has its own internally defined coordinate system which is fixed to the robot (this is called a 'body-fixed' coordinate system). This coordinate system is used, among other

things, for the readouts from the LIDAR scanner. The NEATO's coordinate system has the x-axis pointing forward, the y-axis pointing left, and the z-axis pointing up. For each of your NEATO pictures on the curve, indicate the orientation of the NEATO's own coordinate system. How does this relate to the $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ vectors?

6. In order for the NEATO to follow the curve, what must be true about the orientation of the NEATO's x-axis as it traverses the curve? What about its y-axis?

7. The NEATO is an example of a 'rigid body': an object which has a size (unlike a point particle, which you often consider in introductory physics) but for which the different parts of the body do not move with respect to one another (no stretching, bending, etc). When we study the motion of a rigid body, we can think about decomposing the motion into two components: the motion OF the center of the object and the motion ABOUT the center of the object. In other words, when we talk about the motion of the NEATO, in order to give a complete description, we need to specify the velocity of the center of the robot in the lab x direction, the velocity of the center of the robot in the lab y direction, and the rotational motion of the NEATO around its center relative to the x axis (θ=0 when the NEATO is going forward). This system has three degrees of freedom: x,y, and θ and we have to give the velocity for each!

   a) On your picture, indicate the orientation θ of the NEATO at each of the indicated points. Define +θ as the angle of the forward direction of the NEATO (NEATO's x axis) measured counter-clockwise from the lab x direction.

   b) The angular velocity $\boldsymbol{\omega}$ is a vector quantity,

   $$|\boldsymbol{\omega}| = \frac{d\theta}{dt}$$

   and its direction is the axis of rotation. For an object constrained to move in the xy plane, we can express the angular velocity as

   $$\boldsymbol{\omega} = \omega\hat{\mathbf{k}}$$

   where $\omega = d\theta/dt$ is the rate of change of angle of orientation. What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the right? What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the left? (Recall: the angular velocity follows the right hand rule.) What characteristic vector of your parametric curve is also along this direction?

   c) If the object is constrained to always be oriented along its path (a NEATO that isn't slipping for example), then a consistent mathematical definition of angular velocity $\boldsymbol{\omega}$ is

   $$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

   Note: you'll see a derivation of this relationship in the night assignment, but for now let's just use it as a given. With this definition in mind, give a rough indication of the angular velocity vector of the NEATO at points A,B,C,D and E.

Exercise 33.2

This is an optional practice problem that you might consider doing either if you have time at the end of today or later on your own. Our expectation is that you won't have time to do this during our Zoom call today. We recommend that you skip it and only return to it if you have time. If you never do this problem, you will also be fine. You will be doing problems that get at similar concepts in the night assignment.

Consider a NEATO that moves according to the following position vector

$$\mathbf{r}(t) = 0.05t\hat{\mathbf{i}} + 0.05t^2\hat{\mathbf{j}}$$

where $\mathbf{r}(t)$ has units of meters and t has units of seconds.

1. On the board, sketch the path that the NEATO moves along in 10 seconds, roughly indicating its location every second.
2. Approximate and sketch $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ at 1, 2, 5 and 10 seconds on your curve.
3. Compute by hand a vector function for $\hat{\mathbf{T}}$ as a function of time.
4. The direction of $\hat{\mathbf{N}}$ is defined by $\frac{d\hat{\mathbf{T}}}{dt}$. Compute $\frac{d\hat{\mathbf{T}}}{dt}$ (you can leave it as an un-simplified equation).
5. Use CalcPlot3D to visualize unit vectors ($\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$) for the 10 second period. How does each change during the period?
6. Derive the vector functions that represent the linear velocity and acceleration as the NEATO moves along the path.
7. Pull up the MATLAB symbolic starter code. Discuss how you would alter it to determine the angular velocity and visualize $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$ as the NEATO travels the path.

33.4 *Coffee [15 minutes]*

33.5 *Differential Drive in Action [70 minutes]*

In the first overnight you learned how to determine the velocity and acceleration of a particle moving along a parametric curve. In the previous activity, you connected these quantities to the motion of the Neato moving along the curve. Next, you'll extend this by considering not just the overall linear and rotational motion of the Neato as it moves along the curve, but how the motion of the Neato's wheels must be set in order to achieve this overall motion. To accomplish this you'll be working through the basic mechanics of differential drive vehicles. Specifically, you'll need to understand how the movement of each of the Neato's wheels translates into movement of the robot itself. In this section you will be solving two important, and closely related, problems related to robot motion:

a. Given a desired forward and angular velocity, determine the appropriate velocities of each of the robot's wheels.

b. Given the robot's current position, heading, and the velocities of its wheels, determine the robot's new position and heading. *Note: solving this problem can be quite useful when you have a sensor that estimates the actual wheel velocities of your robot (as the Neato does). In this way you can correct for discrepancies between the intended motion and what your robot actually did.*

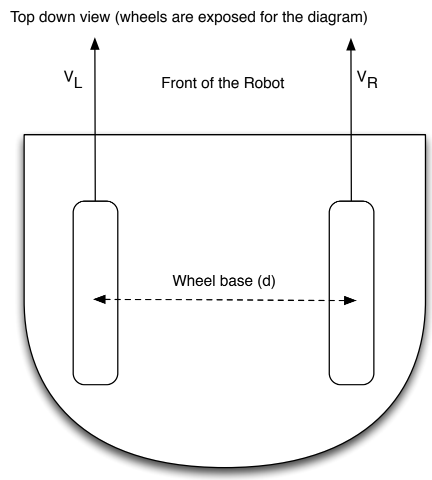


Figure 33.1: A diagram of the Neato's differential drive system.

Formalizing the Problem The Neato has two wheels equally spaced about its centerline (see Figure 33.1). As we saw during day 1, driving the wheels at different velocities (labeled V_L and V_R in the diagram) will achieve different linear and angular velocities.

Exercise 33.3

In day 1 of this module you built up an intuition for how movement of each of the wheels for a differential drive vehicle would translate into motions of the robot. Before we dive into a quantitative treatment of the subject, let's remind ourselves of some limiting cases. Determine qualitatively how the Neato would move (in terms of both linear and rotational motion) in these cases.

- What if both wheels move forward (positive velocity) with equal speed?
- What if both wheels move backward (negative velocity) with equal speed?
- What if one wheel drives forward and the other moves backward with equal speed?

- What if one wheel drives forward while the other remains stationary?

Now that you have refreshed your intuition, let's solve the problem quantitatively. The key insight is that the robot cannot move laterally, but instead must have a linear velocity parallel to the direction of its wheels. As the robot moves along a curve, the robot rotates about its center in order to keep itself aligned with the forward motion. Let's assume that the center of rotation of the robot is located midway between the wheels.

Exercise 33.4

Assuming no wheel slippage, the linear speed in the Neato's direction of motion V and angular velocity ω can be expressed in terms of the left and right wheel velocities V_L and V_R, and the robot's wheel base d (the distance between the two wheels).

$$V = \frac{V_L + V_R}{2} \tag{33.1}$$

$$\omega = \frac{V_R - V_L}{d} \tag{33.2}$$

Does this make sense? Can you confirm these expressions? Can you think of some test cases to validate these expressions? (hint: you just thought about some in the previous exercise!)
Note: we are giving you these equations rather than deriving them. If you want to explore how to derive theses equation for yourself, you can consult section 1 of this document.

Exercise 33.5

Solve equations (1) and (2) in order to express the left and right wheel velocities in terms of the linear and angular velocities and wheel base and show that

$$V_L = V - \omega\frac{d}{2} \tag{33.3}$$

$$V_R = V + \omega\frac{d}{2} \tag{33.4}$$

Does this make sense? Can you think of some test cases to validate these expressions?

Validating your Model

The Equations 33.1 and 33.2 define a motion model for your robot in the sense that they allow us to figure out what the resultant linear and angular motion of the robot would be for a given left and right wheel velocity. Similarly, we could use Equations 35.1 and 35.2 to figure out what the left and right wheel velocities of our robot should be in order to achieve a particular linear and angular motion. This computation can be extremely useful because we often have a goal of making our robot move in a particular fashion and the left and right wheel velocities are just a means to achieving that goal.

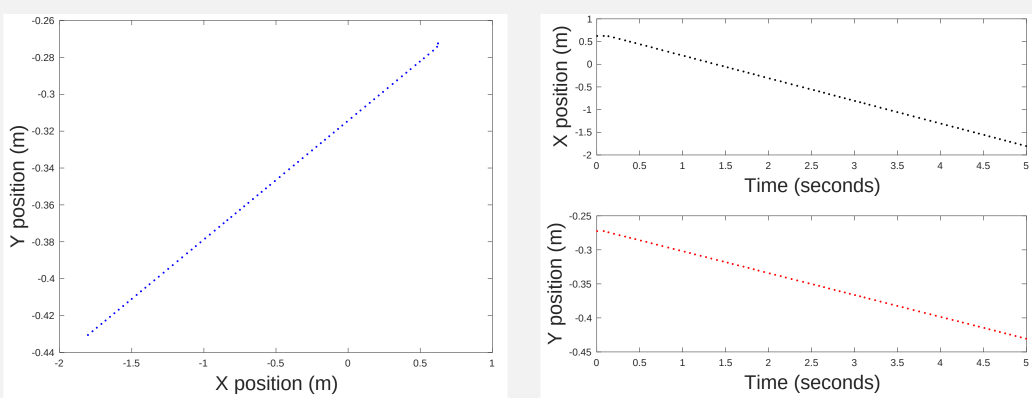
Let's further suppose that someone has measured the wheel base parameter d and obtained an estimate of $0.2m$. If we look at Equations 33.1 and 33.2 we see that now that we have an estimate of d we can plug in the wheel velocities V_L and V_R and compute the linear and angular motion of the robot This all sounds great, but of course as with any model (this is after all a motion model of the robot), you shouldn't expect it to be perfect. What we'd like to do is run an experiment to see if our model's predictions actually match what happens in real life (or simulated life in our current mode of operation).

Exercise 33.6

Design an experiment to determine how well the motion model in Equations 33.1 and 33.2 matches the robot's actual behavior. To further clarify what we mean, here are some things to keep in mind.

- Assume that you can set the left and right wheel velocities to any values you'd like between -3m/s and 3m/s (each wheel could be set to a different velocity).
- If you'd like you can vary the left and right wheel velocities over time.
- Assume that you can measure the robot's position as a function of time.

For example, you might decide that for your experiment you are going to set V_L and V_R to 0.5m/s for 5 seconds and then stop your robot. The results of this experiment might look like this.



Note that in this case the robot started in the upper right corner and moved to the lower left (as you can see from the right plots). You should notice that if you take the distance between the two end points in the left figure you get approximately 2.5m (which makes sense given you went at 0.5m/s for 5 seconds). In this case we are able to easily plot the position of the robot since we are using a simulator. For a real robot, it would be a more challenging (although doable with some additional instrumentation of your robot or its environment).

1. Design an experiment you would carry out to validate the motion model in equations 33.1 and 33.2. Describe in some detail what your experiment would entail.
2. Sketch some potential results of the experiment if the motion model proves to be relatively accurate.
3. What if it turns out that the motor on the left wheel is underpowered and moves at only 80% of the velocity you command it to. Qualitatively, how might your results look in this case?
4. Suppose you were to run your experiment and obtain some results, how might you use this experimental data to quantify the accuracy of the motion model in equations 33.1 and 33.2? Focus on high-level strategy rather than necessarily coming up with an equation.
5. Do the results of your proposed experiment tell you everything you need to know about your motion model? If not, what other experiments might you carry out and what information would you hope to gain from running them?

Exercise 33.7

In this problem we're going to run through a very basic experiment to test the validity of the motion model. In this experiment we're going to set $V_L = 0.2m/s$ and $V_R = 0.1m/s$ and let the robot go for 20 seconds. For those 20 seconds we'll be measuring the x and y position of our robot. Once 20 seconds have elapsed, we're going to stop the robot.

1. Assuming the motion model is relatively accurate, sketch some potential results for the experiment. If you get stuck, check the solution given for the previous problem.
2. Suppose you run the experiment and collect this experimental data (graph 1 is the y position versus the x position of the center of the Neato, graph 2 has x-position versus time and y-position versus time) (note: we made it a link so you won't see the results and spoil part 1).

   Based on this data is the motion model accurate? What are some potential sources of mismatch between the model and the experimental results? If you were going to revise your motion model, what might you change?

Exercise 33.8

For this problem you have one of two choices.

- *Option 1:* If you or one of the people in your group has the robot simulator up and running on your computer, you can run the experiment described in the previous problem on one of your computers. (detailed instructions on how to run the experiment are given below).
- *Option 2:* You can report to the main room at 1215 and we'll do a run through of the experiment for you (Paul will do a screen share to walk folks through this). Note that if you attend our walkthrough you can always do it yourself at a later time.

Instructions to run the experiment with the robot simulator.

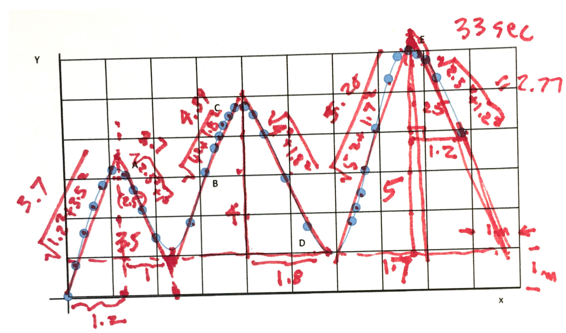
- Step 1: Go to the Using the Neato Simulator section of the Meeto your Neato page.
- Step 2: Copy paste the docker command into PowerShell with the following critical modification. Where you see *gauntlet_no_spawn* replace it with *empty_no_spawn*.
- Step 3: Once the robot simulator is up and running, to see the robot, as described on the Meeto Your Neato page, go to `http://localhost:8080`
- Step 4: Open up MATLAB and connect to the simulated robot by running the `rosinit` command from the Meeto your Neato page (check under Programming the Robot in MATLAB).
- Step 5: Download runBasicWheelVelocityExperiment.m script and run it to execute the experiment and generate plots. Note: we are not expecting you to be able to read through the code in that script at this time. This is not something we have taught you yet and you should not feel like it is something you should know how to do (yet).

Other (fun) stuff to try.

- Try different wheel velocities by specifying inputs to the function (e.g., `runBasicWheelVelocityExperiment(1.0, 0.8)` would drive the left wheel with velocity 1.0m/s and the right wheel with velocity 0.8m/s).
- Load the ice rink world by executing step 2 above but replacing *gauntlet_no_spawn* with *ice_rink*. In this new world, see what happens if you drive the wheels to fast!

Solution 33.1

1. Roughly how long does the NEATO travel for? Roughly how far does it travel? Roughly what is its average speed? By counting up the points and using 1 second between points, it traveled for 33 seconds. To determine the distance traveled, there are lots of ways to estimate these values. I turned the path into a series of triangles and computed the approximate length.

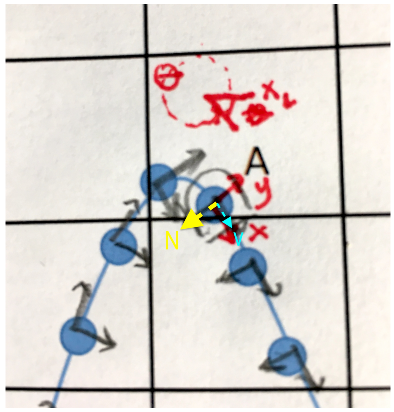


From this, I got 23.2 m. The average speed from these values is $0.7\bar{0}m/s$. The overline indicates that the "0" is significant. I used 2 significant figures since I only had 2 in the estimation of the triangles.

2. Thinking about the curve only, draw the unit tangent vector $\hat{\mathbf{T}}$ and unit normal vector $\hat{\mathbf{N}}$ at each of the points indicated: A, B, C, D and E. See the image below. $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ are in pencil.

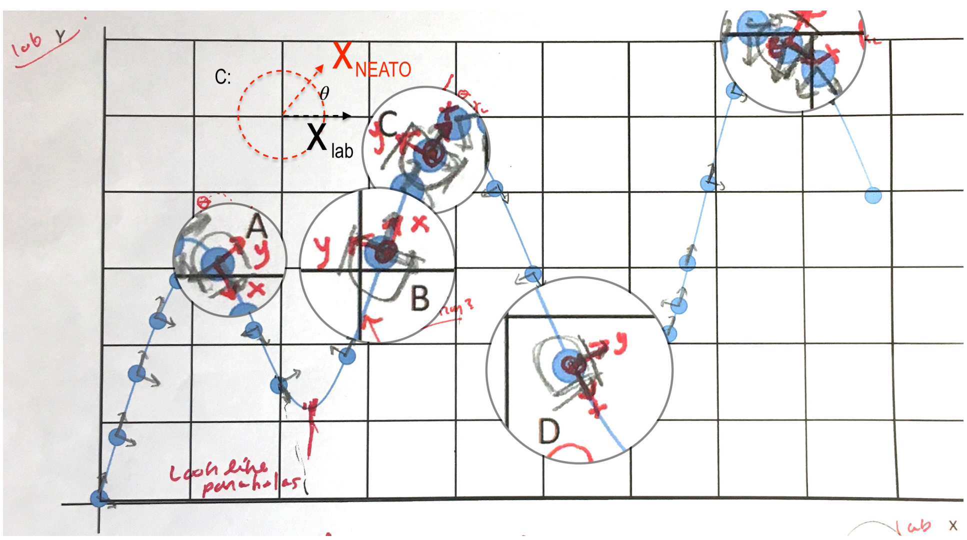
3. Thinking about the motion of the NEATO now, choose one of the points (A,B,C,D or E) and compute the magnitude of the velocity vector. Draw the velocity vector to scale at that point. Do the same for the acceleration vector, decomposing it into tangential and normal components. How do these relate to the unit tangent and unit normal vectors? (Do more points if you have time!) Point A: I'm going to use the points on the curve to estimate the ΔL over the 2 second interval in the point before and after A. Assuming that the grid is "to scale," I estimate the speed at A to be,

$$v = \frac{\Delta L}{\Delta t} \approx \frac{0.77}{2s} = 0.39m/s$$



4. Draw a picture of the NEATO at A, B, C, D, and E, i.e. you are looking down on the NEATO as it drives along the curve.

5. The NEATO has its own internally defined coordinate system which is fixed to the robot (this is called a 'body-fixed' coordinate system). This coordinate system is used, among other things, for the readouts from the LIDAR scanner. The NEATO's coordinate system has the x-axis pointing forward, the y-axis pointing left, and the z-axis pointing up. For each of your NEATO pictures on the curve, indicate the orientation of the NEATO's own coordinate system. How does this relate to the $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ vectors? $\hat{\mathbf{T}}$ is co-incident with x-axis of the NEATO. $\hat{\mathbf{N}}$ is parallel to y-axis of the NEATO.



6. In order for the NEATO to follow the curve, what must be true about the orientation of the NEATO's x-axis as it traverses the curve? What about its y-axis? The NEATO x-axis must be co-incident with $\hat{\mathbf{T}}$. The NEATO y-axis has to be parallel or anti-parallel to $\hat{\mathbf{N}}$.

7. The NEATO is an example of a 'rigid body': an object which has a size (unlike a point particle, which you often consider in introductory physics) but for which the different parts of the body do not move with respect to one another (no stretching, bending, etc). When we study the motion of a rigid body, we can think about decomposing the motion into two components: the motion OF the center of the object and the motion ABOUT the center of the object. In other words, when we talk about the motion of the NEATO, in order to give a complete description, we need to specify the velocity of the center of the robot in the lab x direction, the velocity of the center of the robot in the lab y direction, and the rotational motion of the NEATO around its center relative to the x axis (θ=0 when the NEATO is going forward). This system has three degrees of freedom: x,y, and θ and we have to give the velocity for each!

    a) On your picture, indicate the orientation θ of the NEATO at each of the indicated points. Define +θ as the angle of the forward direction of the NEATO (NEATO's x axis) measured counter-clockwise from the lab x direction. See image above. θ is shown for the point C, measured relative to the x-axis of the lab.

    b) The angular velocity $\boldsymbol{\omega}$ is a vector quantity,

    $$|\boldsymbol{\omega}| = \frac{d\theta}{dt}$$

    and its direction is the axis of rotation. For an object constrained to move in the xy plane, we can express the angular velocity as

    $$\boldsymbol{\omega} = \omega\hat{\mathbf{k}}$$

    where $\omega = d\theta/dt$ is the rate of change of angle of orientation. What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the right? What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the left? (Recall: the angular velocity follows the right hand rule.) What characteristic vector of your parametric curve is also along this direction? When curving to the left, $\frac{\Delta\theta}{dt} > 0$ and $\boldsymbol{\omega}$ points in +$\hat{\mathbf{k}}$. When curving to the right, $\frac{\Delta\theta}{dt} < 0$ and $\boldsymbol{\omega}$ points in -$\hat{\mathbf{k}}$. Our $\hat{\mathbf{B}}$ is along this same direction.

    c) If the object is constrained to always be oriented along its path (a NEATO that isn't slipping for example), then a consistent mathematical definition of angular velocity $\boldsymbol{\omega}$ is

    $$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

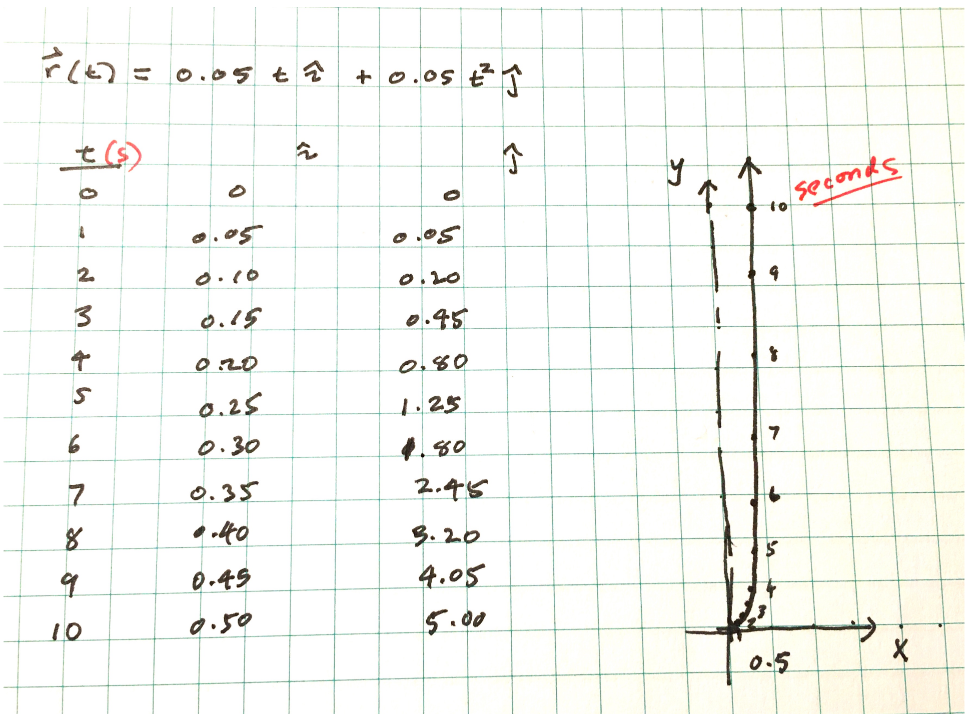
    Note that $\boldsymbol{\omega}$ is a vector obeys the right-hand rule. With this definition in mind, which general direction does $\boldsymbol{\omega}$ point for the NEATO at points A,B,C,D and E? (This definition will be explored more in the overnight). For all points, $\hat{\mathbf{T}}$ is co-incident with the +x-axis of the NEATO. To figure out the direction of $\boldsymbol{\omega}$, we need to consider $\frac{d\hat{\mathbf{T}}}{dt}$ for each point. Conceptually, the vector $\frac{d\hat{\mathbf{T}}}{dt}$ represents the change in the direction of $\hat{\mathbf{T}}$ with time (the magnitude of $\hat{\mathbf{T}}$ is a constant of 1).We could attempt to assess $\frac{d\hat{\mathbf{T}}}{dt}$ at each of the points, but we can also remember that $\frac{d\hat{\mathbf{T}}}{dt}$ will be in the same direction as $\hat{\mathbf{N}}$, as

    $$\hat{\mathbf{N}} = \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|}$$

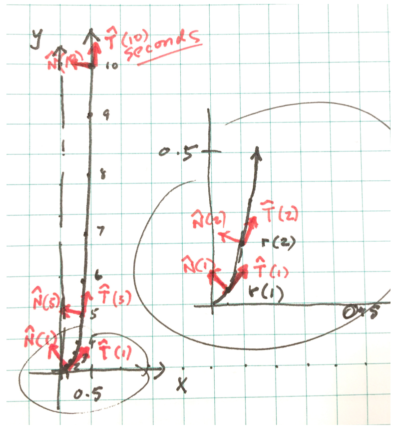
So we can use our drawings of $\hat{\mathbf{N}}$ as a proxy for $\frac{d\hat{\mathbf{T}}}{dt}$ at points A, B, C, D and E. From the image above, we will get the following result, using the right-hand rule for $\boldsymbol{\omega}$ at these points: A: $-\hat{\mathbf{k}}$; B: Hard to tell, just past the inflection point, so guessing $-\hat{\mathbf{k}}$; C: $-\hat{\mathbf{k}}$; D: $+\hat{\mathbf{k}}$; E: $-\hat{\mathbf{k}}$. Notice these results are consistent with the results above–turning left gives $+\hat{\mathbf{k}}$, turning right gives $-\hat{\mathbf{k}}$.

Solution 33.2

1. On the board, sketch the path that the NEATO moves along in 10 seconds, roughly indicating its location every second. For one second increments, a sketch might look like this:



2. Approximate and sketch $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ at 1, 2, 5 and 10 seconds on your curve.



3. Compute by hand a function that represents $\hat{\mathbf{T}}$ as a function of time. $\hat{\mathbf{T}}=\frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}$, so

$$\mathbf{r}'(t) = \frac{d}{dt}(0.05t\hat{\mathbf{i}} + 0.05t^2\hat{\mathbf{j}})$$

$$= 0.05\hat{\mathbf{i}} + 0.10t\hat{\mathbf{j}}$$

and

$$|\mathbf{r}'(t)| = \sqrt{0.05^2 + (0.10t)^2}$$
$$= 0.05\sqrt{1 + 4t^2}$$

so

$$\hat{\mathbf{T}} = \frac{1}{\sqrt{1+4t^2}}(\hat{\mathbf{i}} + 2t\hat{\mathbf{j}})$$

4. The direction of $\hat{\mathbf{N}}$ is defined by $\frac{d\hat{\mathbf{T}}}{dt}$. Compute $\frac{d\hat{\mathbf{T}}}{dt}$ (you can leave it as an un-simplified equation).

$$\frac{d\hat{\mathbf{T}}}{dt} = |\hat{\mathbf{T}}'|\hat{\mathbf{N}}$$

$$\frac{d\hat{\mathbf{T}}}{dt} = \frac{d}{dt}\frac{1}{\sqrt{1+4t^2}}(\hat{\mathbf{i}} + 2t\hat{\mathbf{j}})$$

We would need to use the chain rule:

$$= \frac{1}{\sqrt{1+4t^2}}\frac{d}{dt}(\hat{\mathbf{i}} + 2t\hat{\mathbf{j}}) + (\hat{\mathbf{i}} + 2t\hat{\mathbf{j}})\frac{d}{dt}\frac{1}{\sqrt{1+4t^2}}$$

$$= \frac{1}{\sqrt{1+4t^2}}(2\hat{\mathbf{j}}) + (\hat{\mathbf{i}} + 2t\hat{\mathbf{j}})(\frac{-1}{2})(1+4t^2)^{-\frac{3}{2}}(8t)$$

To evaluate $|\hat{\mathbf{T}}'|$, we would need to compute the magnitude of the vector that results from the messy chain-rule derivative. (MATLAB to the rescue!) Computing $\hat{\mathbf{N}}$ involves computing $\hat{\mathbf{T}}'/|\hat{\mathbf{T}}'|$.

5. Use CalcPlot3D to visualize unit vectors ($\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$) for the 10 second period. How does each change during the period? We can see in this short video that $\hat{\mathbf{T}}$ begins aligned with the x-axis and turns toward the y-axis as t increases; $\hat{\mathbf{N}}$ begins aligned with the y-axis at t=0 and turns toward -x as t increases; $\hat{\mathbf{B}}$ does not change–it remains in the $\hat{\mathbf{k}}$ direction throughout the time of travel.

6. Derive the vector functions that represent the linear velocity and acceleration as the NEATO moves along the path. From above,

$$\mathbf{r}'(t) = \frac{d}{dt}(0.05t\hat{\mathbf{i}} + 0.05t^2\hat{\mathbf{j}})$$

$$= 0.05\hat{\mathbf{i}} + 0.10t\hat{\mathbf{j}}$$

and

$$\mathbf{r}"(t) = \frac{d}{dt}(0.05\hat{\mathbf{i}} + 0.10t\hat{\mathbf{j}})$$
$$= 0.10\hat{\mathbf{j}}$$

7. Pull up the MATLAB symbolic starter code. Discuss how you would alter it to determine the angular velocity, $\boldsymbol{\omega}$, and visualize $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$ as the NEATO travels the path? From the starter code, we would need to change the parametric equations, change the t range ("u" in the code), expand the plotting range, and add the equations to compute $\boldsymbol{\omega}$, using the equation,

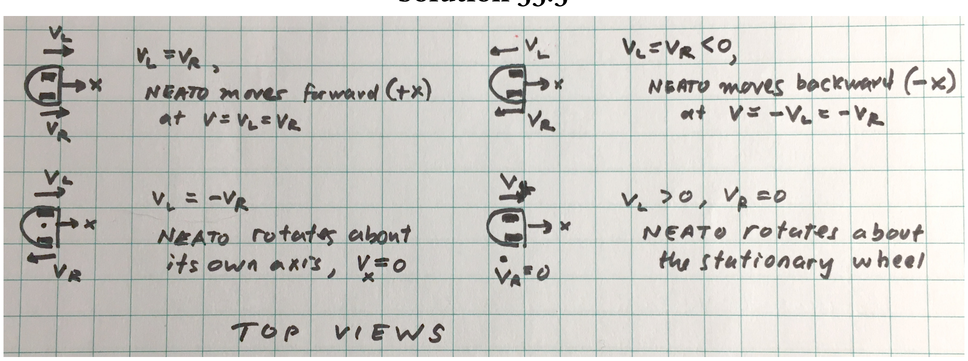
$$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

An example of a MATLAB line of code that would compute this is

```
%Compute the angular velocity vector
omega=simplify(cross(T_hat,dT_hat))
```

Here is an example of altered MATLAB script that would work.

Solution 33.3



Solution 33.4

Let's take the case where $V_L = V_R > 0$. We know intuitively in this case that the NEATO will be moving forward at $V = V_L = V_R$. Using the equation for V above, we get,

$$V = \frac{V_L + V_R}{2} = \frac{2V_L}{2} = V_L$$

Let's take another case where $V_L = -V_R$ and $V_L < 0$. In this case, we know intuitively that the forward motion of the NEATO stops and it is simply spinning to the left about its center (when viewed from the top–counter clockwise). The velocity is then,

$$V = \frac{V_L + -V_R}{2} = 0$$

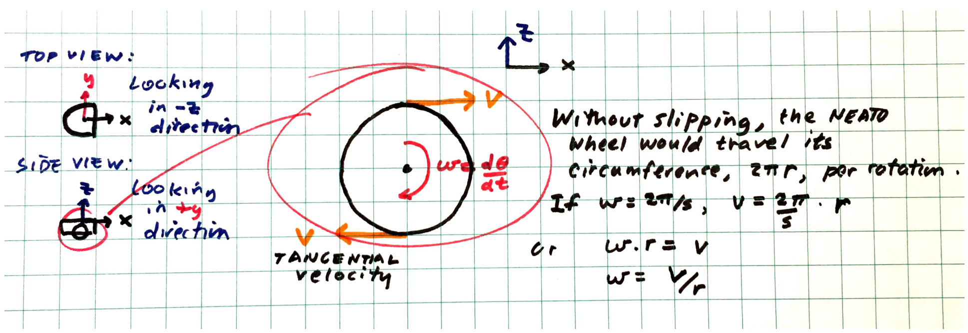
Let's use the same scenarios for computing ω. We find that for $V_L = V_R$,

$$\omega = \frac{V_R - V_L}{d} = 0$$

In other words, the NEATO is traveling in a straight line. That checks out. What about $V_L = -V_R$ and $V_L < 0$? We know intuitive that this result in a counter clockwise rotation about the NEATO center and therefore $\boldsymbol{\omega} > 0$. We get the magnitude ω from,

$$\omega = \frac{V_R - V_L}{d} = \frac{-V_L - V_L}{d} = \frac{-2V_L}{d}$$

Because $V_L < 0$, we can picture this motion, if viewing the NEATO along its y-axis, as



The wheel turns at a rate of $\omega = \frac{d\theta}{dt}$. For uniform circular motion, where ω is a constant value, $\omega = \frac{\Delta\theta}{\Delta t}$. The linear velocity would be the distance covered in the Δt, which would be $v = \frac{2\pi r}{\Delta t}$, giving,

$$v = \frac{2\pi r\omega}{\Delta\theta} = \frac{2\pi r\omega}{2\pi} = r\omega$$

Solution 33.5

Our goal will be to eliminate V_L from Equations 33.1 and 33.2. We can do this by solving each equation for V_L and then equating the two results.

$$\begin{aligned}
V &= \frac{V_L + V_R}{2} && \text{Starting from Equation 33.1} \\
2V &= V_L + V_R \\
2V - V_R &= V_L \\
\omega &= \frac{V_R - V_L}{d} && \text{Starting from Equation 33.2} \\
d\omega &= V_R - V_L \\
V_R - d\omega &= V_L \\
V_R - d\omega &= 2V - V_R && \text{Equating our two expressions for } V_L \\
2V_R &= 2V + d \\
V_R &= V + \frac{d}{2}
\end{aligned}$$

To get an expression for V_L we can follow the same strategy, but solve each equation for V_R.

$$\begin{aligned}
V &= \frac{V_L + V_R}{2} && \text{Starting from Equation 33.1} \\
2V &= V_L + V_R \\
2V - V_L &= V_R \\
\omega &= \frac{V_R - V_L}{d} && \text{Starting from Equation 33.2} \\
d\omega &= V_R - V_L \\
V_L + d\omega &= V_R \\
V_L + d\omega &= 2V - V_L && \text{Equating our two expressions for } V_L \\
2V_L &= 2V - d \\
V_L &= V - \frac{d}{2}
\end{aligned}$$

To sanity check our solution, we could set $\omega = 0$ and observe that this would result in both wheels being commanded to move at the same velocity. For a positive ω we have that the right wheel goes faster, which makes sense since it is on the outside of a counterclockwise turn.

Solution 33.6

1. A reasonable experiment would be to set $V_L = 0.1m/s$ and $V_R = 0.2m/s$. If we let the robot travel with these velocities for some amount of time, e.g., 30 seconds, we would expect it to trace out a circular path in the counterclockwise direction.

2. We would expect the resultant path to be a counterclockwise circle. Given $d = 0.2m$ we expect $\omega = \frac{0.2m/s - 0.1m/s}{0.2m} = 0.5rad/s$. It would take the Neato $\frac{2\pi}{0.5rad/s} = 4\pi$ seconds to make a full trip around the circle. Further, we expect the linear speed to be $0.15m/s$. Given

this we would expect the circumference of the circle the Neato traces to be 0.6π and the radius to be 0.3 m.

3. In this case we would expect the Neato to travel in a tighter circle since ω will be larger.

4. We could compute the radius of the circle the Neato actually drives and compare it to the predicted radius. We could also compare the time it takes to complete a traversal of the full circle to the predicted value.

5. This experiment will tell us a fair amount. We wouldn't be able to distinguish between the case where one wheel is underpowered (e.g., the hypothetical case posed above) or the wheel base being measured incorrectly. To distinguish between these two cases, we could add an additional experiment of having the robot drive straight for some time to see if it fits predictions.

Solution 33.7

1. Using the same logic as in the solution to the previous problem, we'd expect the Neato to travel around a circle of radius 0.3m with a period of 4π seconds. The only difference is the Neato will now be moving in clockwise direction.

2. The experimental data doesn't match the prediction as well as we might look since the measured radius of the circle the Neato moves around is about 0.35m. Perhaps the wheel base is incorrect?

Solution 33.8

There's no solution to this problem, but we'll post a recording of the walkthrough that we do at 1210.

Chapter 34
Night 2: Angular Velocity, NEATOs, and Partial Derivatives

Learning Objectives

Concepts

- Predict the direction of $\boldsymbol{\omega}$ an object's space curve.
- Compute the partial derivatives of functions of more than one variable.

Matlab Skills

- Compute an object's motion properties from its parametric vector function.
- Use basic ROS commands to send instructions and receive sensor data from a simulated NEATO.

34.1 Angular Velocity Revisited

Suppose we have a 2D parametric curve $\mathbf{r}(t) = f(t)\hat{\mathbf{i}} + g(t)\hat{\mathbf{j}}$. We saw in the Night 1 assignment that the linear velocity vector is given by $\mathbf{r}'(t) = f'(t)\hat{\mathbf{i}} + g'(t)\hat{\mathbf{j}}$.

Determining the expression for the angular velocity $\boldsymbol{\omega}(t)$ of our robot is more involved. Before we derive the correct expression, we will revisit the notion of angular velocity vectors. Recall, that in day 1 of this module you encountered this idea when we you experimented with moving your phone and viewing the components of the angular velocity vector in a mobile app. In this document we'll expand on this idea, so if you still had some confusions from the previous activity with the mobile phones, that is to be expected. While expressing angular velocity as a vector may seem overly complex (since in this case we know that the robot will rotate about the z-axis, and it seems like we should just be able to compute the scalar magnitude along with whether to turn clockwise or counterclockwise), thinking about the angular velocity as a vector will enable our derivation to be done in a much more straightforward and generalizable manner.

Angular velocity vectors point in the direction about which the body rotates (which for our robot will be along either the positive or negative z-axis). For right-handed coordinate systems (such as the one we are using here), by convention, a positive rotation happens counterclockwise about the direction of the rotation axis. Further, the magnitude of the angular velocity vector indicates the angular speed.

In the problem at the beginning of class on Monday we discussed the coordinate system attached to the robot: the body fixed frame. Because the heading of the robot is locked to the tangent vector $\hat{\mathbf{T}}$ of the curve, we can think of the vector $\hat{\mathbf{T}}$ as being a constant in the body fixed frame of the robot. The body fixed frame is rotating with some unknown angular velocity vector $\boldsymbol{\omega}$ with respect to the room coordinate system. If we wish to know the time derivative of the tangent vector $\hat{\mathbf{T}}$ in the room coordinate system, we can use the generalized relationship between the time derivatives of vectors in two coordinate systems which are rotating with an angular velocity vector $\boldsymbol{\omega}$ with respect to each other. This expression is

$$\frac{d\hat{\mathbf{T}}}{dt}|_{room} = \frac{d\hat{\mathbf{T}}}{dt}|_{body} + \boldsymbol{\omega} \times \hat{\mathbf{T}} \tag{34.1}$$

(Full mathematical derivation here; nice heuristic explanation here.) In the body frame of the robot, $\hat{\mathbf{T}}$ is unchanging, since it is always aligned with the forward direction, so the term $\frac{d\hat{\mathbf{T}}}{dt}|_{body}$ is zero leaving us with

$$\frac{d\hat{\mathbf{T}}}{dt}|_{room} = \boldsymbol{\omega} \times \hat{\mathbf{T}} \tag{34.2}$$

Then we make use of the scalar triple product, which states that $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$. Using this we can derive our angular velocity vector as follows

$$\begin{aligned}
\frac{d\hat{\mathbf{T}}}{dt}|_{room} &= \boldsymbol{\omega} \times \hat{\mathbf{T}} \\
\hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}|_{room} &= \hat{\mathbf{T}} \times \boldsymbol{\omega} \times \hat{\mathbf{T}} \\
&= \boldsymbol{\omega}(\hat{\mathbf{T}} \cdot \hat{\mathbf{T}}) - \hat{\mathbf{T}}(\hat{\mathbf{T}} \cdot \boldsymbol{\omega}) \\
&= \boldsymbol{\omega}(1) - \hat{\mathbf{T}}(0) \\
&= \boldsymbol{\omega} \\
\Rightarrow \boldsymbol{\omega} &= \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}|_{room}
\end{aligned} \tag{34.3}$$

The x and y components of the angular velocity vector will always be zero because $\hat{\mathbf{T}}$ and $\frac{d\hat{\mathbf{T}}}{dt}|_{room}$ are in the x-y plane and orthogonal. The magnitude of the z-component is the angular speed. If the z-component is positive, we turn counterclockwise at that speed. When it is negative, we turn clockwise at that speed.

Exercise 34.1

In the Night 1 assignment you found the unit tangent and normal vectors for various parameterized curves. We will use that information to find linear and angular velocities, then translate those to left and right wheel velocities for the NEATO.

The vector for a circle centered at the origin in the x-y plane is given by:

$$\mathbf{r}(t) = R\cos\alpha t\ \hat{\mathbf{i}} + R\sin\alpha t\ \hat{\mathbf{j}} + 0\hat{\mathbf{k}}$$

We will make the following assumptions about our parameters:

- t starts at 0 seconds and increases with time ($\mathbf{r}(t)$ gives us the Neato's position at time t).
- $R > 0$ and provides the radius of the circle.
- α can be positive or negative (we'll ask you to interpret its meaning later in the problem).

If you decide to work in MATLAB (either to check your solutions or to do the problems), we'd advise you to build off the your code from the night 1 assignment (or use the starter code provided in that assignment). If you're finding your answers are not in a form that you expect, considering the following hints.

- Specify assumptions (e.g., t, α, R are all real and $R > 0, t \geq 0$). Note as stated above: α does not have to be positive.

- Use the MATlAB function `simplify` on your answer.

1. How does the sign of α affect the path that the Neato takes around the circle?
2. What are the linear velocity vector and linear speed?
3. What is the unit tangent vector for the circle? Make sure your answer makes sense for both positive and negative values of α.
4. What is the unit normal vector?
5. What is the angular velocity vector?
6. For the uniform circular motion we have been investigating so far, what does the parameter we have labeled α represent? How is it related to the time it takes to complete one traverse of the circular trajectory?
7. How would you modify the initial equation for the position vector when its trace is circle of radius 1 m?
8. What value would you choose for α if you want your robot to complete a counterclockwise path around the circle in 30 seconds?
9. What are the equations for the left and right wheel velocities for the uniform circle? (d=wheel displacement). You can leave your answer in terms of d or substitute a reasonable value for the Neato of $d = 0.235m$.
10. What are the left and right wheel velocities needed for a 1 m radius counterclockwise circle to be completed in 30 seconds?

Exercise 34.2

The vector for a counterclockwise path around an ellipse is given by:

$$\mathbf{r}(t) = a\cos\alpha t\;\hat{\mathbf{i}} + b\sin\alpha t\;\hat{\mathbf{j}},\;\; \alpha t \in [0, 2\pi]$$

In this problem we can assume that α, a, b are all positive and that $t \geq 0$.

1. What is the tangent vector for the ellipse?
2. What is the unit tangent vector for the ellipse?
3. What is the linear velocity vector? How does it differ from the example of the circle?
4. What is the unit normal vector?

5. What is the angular velocity vector? How does it differ from the circle?
6. What are the left and right wheel velocities?
7. Plot the linear velocity vector as a function of time for various combinations of the parameters a, b, and α.
8. Plot the angular velocity vector as a function of time for various combinations of the parameters a, b, and α.
9. Plot the left and right wheel velocities as a function of time for various combinations of the parameters a, b, and α.

34.2 Fun with (Simulated) NEATOs

In the previous section we found the left and right wheel velocities needed to drive a particular trajectory. In this section of the assignment, we will be thinking about how to translate the velocity vectors to a Matlab program that will control your NEATO.

The control of your NEATO is built on top of the Robotic Operating System (ROS), so you will be using ROS commands to control the velocity values for your robot. We will start by playing with some very basic commands.

Consider the program drivefoward.m.

This code snippet defines the function "driveforward" which will cause your NEATO to.... you guessed it, drive forward. You will notice that this function does not have a meaningful output, its sole purpose is to move your robot forward.

The structure of a Simple Robot Program

While the program has lots of comments, on the page linked above we elaborate on specific lines of the code to help you understand its structure. Please read through the notes on that page.

Exercise 34.3

Download the above program and open the m-file in Matlab. It is a function, so it can be called from the command window using the form:

```
driveforward(distance, speed)
```

Start the simulator with an empty world by modifying the command under the *Using the Simulator* section of the Meeto Your Neato page and modifying the command by replacing *gauntlet_no_spawn* with *empty_no_spawn* in the long command that starts with *docker*. Once the simulator is up and running, connect to the visualizer as described on the Meeto Your Neato page.

Using this new function, try driving the NEATO for several combinations of distances and speeds. Do the final distance and time match your expectations? Note that squares of the grid on the simulator visualization are 1 meter by 1 meter.

Receiving Sensor Data

In the previous example program we published to a ROS topic to set the NEATO wheel velocities. In ROS you can also subscribe to a topic to do things like receive sensor data. Download and open the program driveUntilBump in Matlab.

Exercise 34.4

1. In line 2 the 'rossubscriber' command is introduced. From the code, what sensor output are we monitoring? Note: it's pretty hard to figure this out with the simulator since there is no way to actually physically touch the robot, just look at the solution for this one!
2. The variable 'bumpmessage' is a structure. What is the size of 'bumpmessage.Data'? What do the values contained in that variable mean? Note: it's pretty hard to figure this out with the simulator since there is no way to actually physically touch the robot, just look at the solution for this one!
3. What is the 'driveUntilBump' code commanding the robot to do?
4. Test the 'driveUntilBump' code on a NEATO and verify that your interpretation is correct.
5. Modify the 'driveUntilBump' code to make it a function where the robot velocity is an input.
6. Using what you have learned from the examples above, write a program that meets the following requirements:
   - The program commands the robot to drive a designated distance at a chosen speed, and stops when that distance is reached.
   - If the bump sensor is triggered, the robot reverses direction and backs up for 5 seconds then stops.

   Try developing this code on your own first, then if you get stuck, take a look at the program driveUntilBumpThenRunAway for inspiration.

34.3 *Partial Derivatives*

The following notes about Partial Derivatives might be helpful. You can check your calculations using WolframAlpha or the Symbolic Toolbox in Matlab.

Consider a function of two variables $f(x, y)$. If we identify $z = f(x, y)$, then we can visualize this function as a surface in 3D. At any point on the surface, (a, b, c), we can ask about the slope of the tangent line in the x-direction and in the y-direction. In the first case, we intersect the surface with the plane $y = b$ and consider the rate of change of f in the x-direction only. In the second case, we use the plane $x = a$ and consider the rate of change of f in the y-direction only. There are therefore two fundamental derivatives,

$\frac{\partial f}{\partial x}$ is the partial derivative of f with respect to x

$\frac{\partial f}{\partial y}$ is the partial derivative of f with respect to y

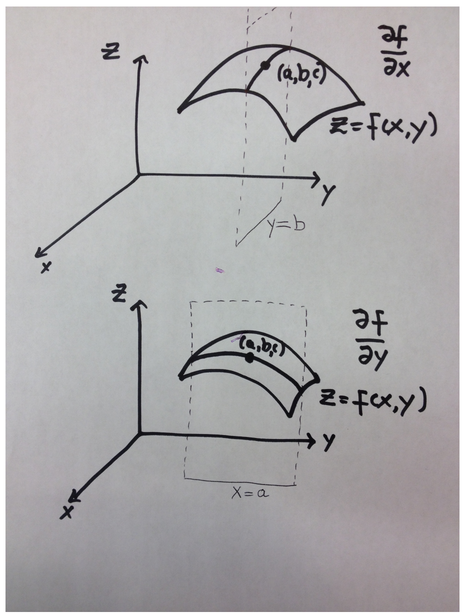


Figure 34.1: Partial derivatives of a function of two variables.

In each case we compute the derivative with respect to one variable by holding the other one fixed, i.e. treating it as a constant.

Exercise 34.5

Evaluate $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ for each of the following functions.

1. $f(x, y) = x^2 \sin(xy^2)$
2. $f(x, y) = 4 + x^3 + y^3 - 3xy$

We can also evaluate higher-order derivatives, but now there are several possibilities. We could take two derivatives with respect to x,

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x} = \frac{\partial^2 f}{\partial x^2}.$$

We could take two derivatives with respect to y,

$$\frac{\partial}{\partial y}\frac{\partial f}{\partial y} = \frac{\partial^2 f}{\partial y^2}.$$

Or we could take a derivative with respect to x and then with respect to y, and vice versa,

$$\frac{\partial}{\partial y}\frac{\partial f}{\partial x} = \frac{\partial^2 f}{\partial y \partial x}$$

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial y} = \frac{\partial^2 f}{\partial x \partial y}.$$

In all of the functions that we will be dealing with, the mixed partials are equal

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}.$$

Exercise 34.6

Evaluate all four second-order derivatives of the following functions. What do you notice about the mixed partial derivatives?

1. $f(x, y) = x^2 \sin(xy^2)$
2. $f(x, y) = 4 + x^3 + y^3 - 3xy$

Under very gentle conditions it is generally true that the mixed partial-derivatives are always equal, i.e.

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}$$

Exercise 34.7

Using the power of the internet, under what conditions are the mixed partial derivatives of f equal?

The Gradient and the Hessian

We've seen that a function of two variables, $f(x,y)$, has two partial derivatives, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$. Rather than thinking of these separately, we can package them into a vector known as the gradient vector. In terms of notation we write

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Exercise 34.8

Find the gradient vector of the function $f(x,y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$ by hand and check your answer using WolframAlpha or the Symbolic Toolbox in Matlab. Evaluate it at the point $(1,2)$.

In the same vein, we can package the four second-order partial derivatives into a matrix called the Hessian matrix. In terms of notation we write

$$Hf = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Exercise 34.9

Find the Hessian matrix of the function $f(x,y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$ by hand and check your answer using WolframAlpha or the Symbolic Toolbox in MATLAB. Evaluate it at the point $(1,2)$.

Solution 34.1

1. A positive value of α corresponds to a counterclockwise path around the circle, whereas a negative value of α corresponds to a clockwise path.

2. The linear velocity vector is given by $\mathbf{v}(t) = \mathbf{r}'(t) = x'(t)\hat{\mathbf{i}} + y'(t)\hat{\mathbf{j}} + z'(t)\hat{\mathbf{k}}$ or $\mathbf{v}(t) = v(t)\hat{\mathbf{T}}(t)$. For the case of the circle, we know:

$$\mathbf{r}'(t) = -R\alpha \sin \alpha t\ \hat{\mathbf{i}} + R\alpha \cos \alpha t\ \hat{\mathbf{j}}$$

   So, the linear velocity vector is $\mathbf{v}(t) = \alpha R(-\sin \alpha t\ \hat{\mathbf{i}} + \cos \alpha t\ \hat{\mathbf{j}})$ and the magnitude (linear speed) is $v(t) = |\alpha|R$ in unts of $\frac{\mathrm{m}}{\mathrm{s}}$.

3. The unit tangent vector is:

$$\begin{aligned} \hat{\mathbf{T}} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \\ &= -\frac{\alpha}{|\alpha|} \sin \alpha t\ \hat{\mathbf{i}} + \frac{\alpha}{|\alpha|} \cos \alpha t\ \hat{\mathbf{j}} \end{aligned}$$

   Note that the quantity $\frac{\alpha}{|\alpha|}$ gives us the sign of α. This means that if we negate α that will cause $\hat{\mathbf{T}}$ to negate as well (corresponding to, as we would expect, moving about the circle in the opposite direction).

4. The unit normal vector is

$$\begin{aligned} \hat{\mathbf{N}} &= \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|} \\ &= -\cos \alpha t\ \hat{\mathbf{i}} - \sin \alpha t\ \hat{\mathbf{j}} \end{aligned}$$

5. The angular velocity is constant: $\boldsymbol{\omega} = \alpha\hat{\mathbf{k}}$.

6. The parameter α is the angular frequency of motion, often denoted by the scalar ω (we'll use α here since we are already using $\boldsymbol{\omega}$ for the angular velocity vector). For a uniform circular motion, the frequency and angular velocity are equal. The time to complete one traverse of the circle is given by the period $T = \frac{2\pi}{|\alpha|}$.

7. Set R=1 m

8. For a positive value of α (corresponding to a counterclockwise path) we know that the product αt must go from 0 to 2π for the robot to complete one cycle of the parameterized curve. So, after one complete trip around the circle, $\alpha T = 2\pi$, so $\alpha = \frac{2\pi}{T} = 0.21$ with units $\frac{1}{s}$.

9. We can use the equations from Day 2 of the module that relate left and right wheel velocities to linear speed and angular velocity. Plugging the expressions for linear speed and angular velocity that we found earlier in this exercise into those equations we arrive at $V_L = R|\alpha| - \frac{d\alpha}{2}$ and $V_R = R|\alpha| + \frac{d\alpha}{2}$

   Note: that for the simulated Neato (and the real Neato), the wheel base is approximately $d = 0.235m$.

10.

$$
\begin{aligned}
V_L &= R\alpha - \frac{d\alpha}{2} \\
&= (1m)\left(0.21\frac{1}{s}\right) - \frac{(0.235m)(0.21\frac{1}{s})}{2} \\
&= 0.185\frac{m}{s} \\
V_R &= R\alpha + \frac{d\alpha}{2} \\
&= (1m)\left(0.21\frac{1}{s}\right) + \frac{(0.235m)(0.21\frac{1}{s})}{2} \\
&= 0.235\frac{m}{s}
\end{aligned}
\tag{34.4}
$$

Solution 34.2

1. What is the unit tangent vector for the ellipse? I altered the symbolic MATLAB starter code. This code computes almost all the needed properties ($\hat{\mathbf{T}}$, etc.), but I have to change the parameters to fit the ellipse. In my code, I used 'w' for α. Here is a MATLAB solution. When you run the MATLAB code, you will see that if you substituted α for 'w' in the code and used the trigonometric identity, $cos^2(\theta) = 1 - sin^2(\theta)$, the unit tangent vector reduces to,

$$\hat{\mathbf{T}} = \frac{-asin(\alpha t)}{\sqrt{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}}\hat{\mathbf{i}} + \frac{bcos(\alpha t)}{\sqrt{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}}\hat{\mathbf{j}}$$

2. What is the unit normal vector? See $\hat{\mathbf{N}}$ in the output of the MATLAB code. Using the trig identity above, you will get a 'simplified' $\hat{\mathbf{N}}$,

$$\hat{\mathbf{N}} = \frac{-bcos(\alpha t)}{\sqrt{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}}\hat{\mathbf{i}} + \frac{-asin(\alpha t)}{\sqrt{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}}\hat{\mathbf{j}}$$

3. What is the angular velocity vector? How does it differ from the circle? See ω in the output below. Because the NEATO motion is confined to the xy-plane, the angular velocity vector is coincident with the $+\hat{\mathbf{k}}$. However, unlike the circle, which has a constant $\frac{d\theta}{dt}$, the $|\boldsymbol{\omega}| = f(t)$:

$$\boldsymbol{\omega} = \frac{ab\alpha}{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}\hat{\mathbf{k}}$$

For both the circle and the ellipse, $\boldsymbol{\omega} > 0$, since the motion in counter clockwise

4. What are the left and right wheel velocities? Using the equations, from Day 2 we have,

$$V_L = V - \omega\frac{d}{2} \tag{34.5}$$

$$V_R = V + \omega\frac{d}{2} \tag{34.6}$$

In order to map what we've been doing in this document to these equations, we can keep in mind the following.

- ω (notice that ω is not bolded and refers to a scalar) represents the component of the angular velocity in the $\hat{\mathbf{k}}$ direction
- V is the linear speed, which is given as $|r'(t)|$.

$$V_L = V - \omega\frac{d}{2} \tag{34.7}$$

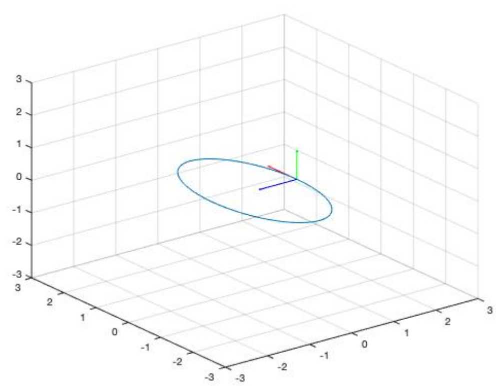
$$= \alpha\sqrt{a^2sin(\alpha t)^2 + b^2cos(\alpha t)^2} - \frac{ab\alpha}{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}\frac{d}{2} \tag{34.8}$$

$$V_R = V + \omega\frac{d}{2} \tag{34.9}$$

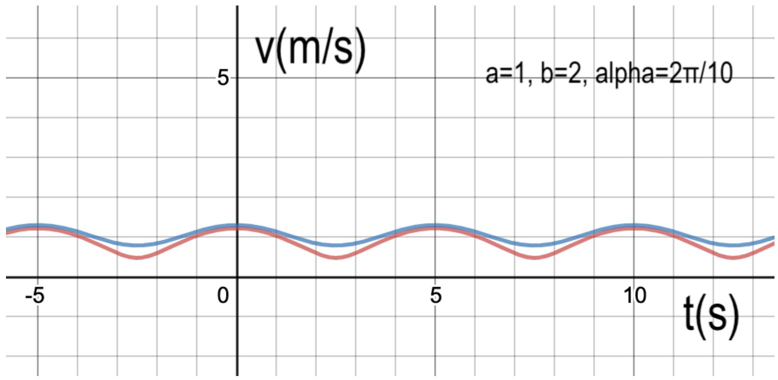
$$= \alpha\sqrt{a^2sin(\alpha t)^2 + b^2cos(\alpha t)^2} + \frac{ab\alpha}{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}\frac{d}{2} \tag{34.10}$$

5. Plot the linear velocity vector as a function of time for various combinations of the parameters a, b, and α. When you do this, you will see that the trace of the linear velocity function vector follows the same path as the space curve, $r(t)$, but is shifted in phase by $\frac{\pi}{2}$. We say that $r'(t)$ leads $r(t)$ because at $\alpha t = 0$, $r'(t)$ is positioned at $\frac{\pi}{2}$ radians ahead of $r(t)$.

6. Plot the angular velocity vector as a function of time for various combinations of the parameters a, b, and α. Because $r(t)$ is constrained to counterclockwise motion in the xy-plane, $\boldsymbol{\omega} > 0$ and in the direction of $\hat{\mathbf{k}}$. Its magnitude varies with the choice of parameters. For $a = 1, b = 2, \alpha = \frac{2\pi}{10}$, a complete rotation of $r(t)$ around the ellipse takes takes 10 seconds. The trace of $r'(t)$ will follow the same ellipse, starting at $r'(0) = (0, \frac{2\pi}{2})$ and ending at $r'(10) = (0, \frac{2\pi}{2})$



7. Plot the left and right wheel velocities as a function of time for various combinations of the parameters a, b, and α. Choosing $a = 1$ and $b = 2$ and $\alpha = \frac{2\pi}{10}$, gives



Regardless of the combinations of the parameters, as long as the NEATO is moving in the counterclockwise direction, $V_L < V_R$.

Solution 34.3

The behavior of the actual should match closely with your expectations. Although, if you set the distance to high for the specified time, the robot might exceed its maximum velocity.

Solution 34.4

- the bump sensor (which is located at the front of the robot and triggers when the front of the robot contacts something)
- 'bumpmessage.Data' contains four numbers one for each of the Neato's bump sensors. Note that in the simulated Neato the bump sensors are either all on or all off.
- It tells the robot to drive forward with velocity of 0.1 m/s until it runs into something and then stops.
- Open the robot simulator using the instructions on Meeto your Neato (you can use the default world of gauntlet_no_spawn). Note that you can actually pickup and move the robot around if you like (check out the gzweb user guide for more information).
- See driveUntilBumpWithVelInput
- See driveUntilBumpThenRunAwayForATime

Solution 34.5

1.

$$\begin{aligned}\frac{\partial f}{\partial x} &= 2x\sin(xy^2) + x^2y^2\cos(xy^2)\\ \frac{\partial f}{\partial y} &= 2x^3y\cos(xy^2)\end{aligned}$$

2.

$$\begin{aligned}\frac{\partial f}{\partial x} &= 3x^2 - 3y \\ \frac{\partial f}{\partial y} &= 3y^2 - 3x\end{aligned}$$

Solution 34.6

1.

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= (2 - x^2y^4)\sin(xy^2) + 4xy^2cos(xy^2) \\ \frac{\partial^2 f}{\partial y^2} &= 2x^3\cos(xy^2) - 4x^4y^2\sin(xy^2) \\ \frac{\partial^2 f}{\partial y \partial x} &= 6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2) \\ \frac{\partial^2 f}{\partial x \partial y} &= 6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2)\end{aligned}$$

As we would expect, the mixed partial derivatives are equal.

2.

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= 6x \\ \frac{\partial^2 f}{\partial y^2} &= 6y \\ \frac{\partial^2 f}{\partial y \partial x} &= -3 \\ \frac{\partial^2 f}{\partial x \partial y} &= -3\end{aligned}$$

As we would expect, the mixed partial derivatives are equal.

Solution 34.7

If the derivatives exist and are continuous then they are equal. See Paul's Online Math Notes about this.

Solution 34.8

Let's first evaluate the first-derivatives:

$$\begin{aligned}\frac{\partial f}{\partial x} &= 6xy - 6x \\ \frac{\partial f}{\partial y} &= 3x^2 + 3y^2 - 6y\end{aligned}$$

and then we simply package them into a vector

$$\nabla f = \begin{bmatrix} 6xy - 6x \\ 3x^2 + 3y^2 - 6y \end{bmatrix}$$

If we evaluate the gradient vector at $(1, 2)$ we see that

$$\nabla f(1, 2) = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

Solution 34.9

Let's first evaluate the second-derivatives:

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= 6y - 6 \\ \frac{\partial^2 f}{\partial y^2} &= 6y - 6 \\ \frac{\partial^2 f}{\partial y \partial x} &= 6x \\ \frac{\partial^2 f}{\partial x \partial y} &= 6x \end{aligned}$$

and then we simply package them into a matrix

$$Hf = \begin{bmatrix} 6y - 6 & 6x \\ 6x & 6y - 6 \end{bmatrix}.$$

Note that we expect the Hessian to be symmetric because the mixed partials are equal. If we evaluate the Hessian matrix at $(1, 2)$ we see that

$$Hf(1, 2) = \begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix}$$

Chapter 35

Day 3: Encoders, Mapping, and Intro to the Bridge of Doom

35.1 Virtual Class Schedule

- 1000-1005: Tech time
- 1005-1015: Introduction to the Day
- 1015-1055: Debrief and Synthesis
- 1055-1105: Coffee
- 1105-1145: Measured Paths
- 1145-1220: Encoders in Action
- 1220-1230: Intro to BOD

35.2 Introduction [10 minutes]

35.3 Debrief and Synthesis [40 minutes]

Exercise 35.1

Briefly discuss the overnight with your table-mates, and make a list of concepts you feel solid on, and concepts you feel shaky on. Make a list of important definitions that you encountered in the overnight.

Exercise 35.2

During class and in the overnight exercises we have been building capacity towards having the NEATO robot drive along a curve. Of course there are lots of different ways to parameterize a curve, each of which would correspond to different motion of the NEATO. Let's take a few minutes to review those concepts here. With your table, please answer the following questions on the board given a circle of radius R,

$$\mathbf{r}(u) = R\cos u\hat{\mathbf{i}} + R\sin u\hat{\mathbf{j}},\ u \in [0, 2\pi]$$

and the following parameterizations:

$$\begin{aligned} u(t) &= \beta t \\ u(t) &= \beta t^2 \\ u(t) &= \beta(2+\sin(t))t \end{aligned}$$

where $\beta > 0$ is a parameter that we can tune.

1. Qualitatively describe the motion of the NEATO in each case.
2. How long does it take to traverse the curve once? (Your answer should depend on β)
3. How would you find the linear velocity of a robot traveling along the curve? What is the direction of this velocity? (It would be really great if you employed the chain rule here and kept your work as general as possible.)
4. How would you find the angular velocity? What is the direction of the angular velocity? (It would be really great if you employed the chain rule here and kept your work as general as possible.)
5. Having found the linear and angular velocity, how would you find the left and right wheel speeds for the differential drive?
6. The robot has a maximum wheel speed of 0.3 m/s. How would you choose β to ensure your robot never exceeds this speed limit?

35.4 *Coffee Break [10 minutes]*

35.5 *Measured Paths [40 minutes]*

One potential source of error that you may have identified in the in class and overnight exercises is that your robot is not able to instantaneously achieve a desired V_L and V_R when you send it a particular motor command. Given the pesky laws of physics, instead, the robot needs to accelerate to the desired velocity. In order to get a more accurate picture of what the robot actually did, we can use *measurements* of the wheel velocities to give us a more accurate estimate of the robot's actual path in the world. Our Neato is outfitted with sensors called *wheel encoders*, which provide accurate estimates of the linear travel of each wheel over time. Knowing the linear travel and the time between measurements, the velocity of each wheel can be calculated. Next, you'll be determining formulas to update the robot's position and heading given measured values of V_L and V_R.

Exercise 35.3

Suppose that at time t your robot is at position $\mathbf{r}(t)$, with a heading of $\theta(t)$. Let's further assume that at $t = 0$ the robot is stationary and pointing along $\theta = 0$, which corresponds to the robot facing along the positive x-axis of the room.

1. Draw a picture on the board to make sure that you are clear as to the definition of the coordinate system.

2. Sketch a (fairly smooth) potential trajectory that your robot will be driving. Choose several points along the curve separated by Δt. Sketch in the unit tangent and unit normal vector at each point.
3. Sketch an estimated plot of your robot's linear and angular speeds as a function of time as it traverses your curve.
4. Sketch an estimated plot of your robot's tangential and normal components of acceleration as a function of time as it traverses your curve.
5. On the board, sketch a qualitative version of your estimate of right and left wheel velocity as a function of time.

Exercise 35.4

For a discretized path (expressed in terms of short time increments rather than continuously) you can, assuming that the time-step Δt is small, approximate a path by a series of movements in $\mathbf{r}$ and movements about the center of the robot in θ. The velocity of the robot is

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= v\hat{\mathbf{T}} \\ \frac{d\theta}{dt} &= \omega \end{aligned}$$

where v is the linear speed, ω is the angular velocity in the $\hat{\mathbf{k}}$ direction, and since the robot is always oriented along the path we can define $\hat{\mathbf{T}}$ in relation to the global (classroom fixed) coordinate system as

$$\hat{\mathbf{T}} = \cos\theta\hat{\mathbf{i}} + \sin\theta\hat{\mathbf{j}}$$

Given measured values for V_L and V_R determine the values of $\mathbf{r}(t + \Delta t)$, and $\theta(t + \Delta t)$ which represent the position and heading of your robot at time $t + \Delta t$.

35.6 Encoders in Action [35 minutes]

Next, you will complete a series of simple experiments with the NEATO, similar to those conducted at the end of Monday's class, while simultaneously collecting motion data. In order to do this, we have provided you a nice little script written by Paul Ruvolo (edited by Jeff Dusek) which you can use to collect the wheel position encoder data while you are running your experiment (which you can easily convert to velocities by taking the difference between two adjacent positions and dividing it by the timestep). The Matlab function *diff* will likely be useful in this process. Note that the robot's initial position is arbitrary.

The script for collecting encoder data is called collectDataset_sim.m and is linked here and to the Canvas assignment. To collect encoder data, run the function *collectDataset_sim('filename.mat')* from your command window. This will bring up a new figure window with the title "Dataset Collection Window". To start data collection , hit the space bar while focusing on the figure window. You will see the message "Starting Dataset Collection" in your command window if everything is working as intended. You can then run your personal script to control the robot, and encoder data will be collected in the background. When your robot motion has concluded, re-focus on the "Dataset Collection Window", and hit the space bar to stop data collection. You will see the message "Stopping Dataset Collection" in your command window.

After you stop the data collection, you will have a file *filename.mat* in your current directory. If you load this file, you will find a matrix "dataset" that contains the encoder and accelerometer data recorded from the robot. For this exercise and the upcoming challenge you only care about the encoder data in columns 2 and 3, and the time stamps in column 1 (recall this data is linear travel of the wheel). The form of the data is:

$$dataset = [time, Pos_{left}, Pos_{right}, AccelX, AccelY, AccelZ] \tag{35.5}$$

Important: If you include a loop in your personal robot control script, make sure to include a pause of the form *pause(0.1)* within the loop. Otherwise, Matlab will try to execute that loop as fast as possible and will prevent the data collection script from recording encoder data.

Complete the following simple experiments. For each experiment, record the encoder data for analysis using the *collectDataset* function.

Exercise 35.5

Using the provided DriveCircles.m function, have the robot spin counterclockwise for ten seconds, then spin clockwise for ten seconds around around the robot's center (i.e V_R=-V_L or the opposite). The function takes inputs of left and right wheel velocity as `DriveCircles(vL,vR)`, and the times are already set. Collect and plot the left and right wheel encoder data.

1. Does the plot of the wheel linear travel look as you would expect?
2. Find the left and right wheel velocity at each time step and plot them. Do they match your expectations? If you know d, you can also find and plot the linear and angular velocity.

Exercise 35.6

This exercise is optional during the virtual class. On a real robot, the actual distance traveled will differ from the commanded distance due to factors including inertia, friction, slight errors in the motor model, etc. The simulator is too "perfect" in the sense that a lot of these real-world considerations will not be present on the simulated robot. Using the driveforward.m function introduced in the overnight, conduct three experiments where the robot drives a specified distance at increasing speeds. Collect and plot the encoder data.

1. Does the linear distance traveled collected by the encoders match the distance input to the function?
2. If the values do not match, why not?
3. Plot the linear and angular velocity as a function of time. Do they match your expectations?

Exercise 35.7

If you have time in class: Calculate the left and right wheel velocities needed to drive a circle of radius 0.5m in 20s. Use the code runBasicWheelVelocityExperiment.m from Monday's class to drive the circle while simultaneously using thecollectDataset_sim.m to collect the wheel encoder data.

1. Plot the anticipated trajectory of your robot in Matlab. You may want to use the Matlab command *axis equal* to make sure your circle looks nice and circular.
2. In the same figure, plot the actual trajectory of your robot. How closely do they match?
3. Quantifying error is an important part of any experiment. For the circular path, how would you calculate error between the anticipated trajectory and the actual path of the robot? Are you interested in total accumulated error? Average error and each time step? Distance away from the "ideal" curve at each time stop (e.g. are you stying within a lane)? Does something else make sense? On the whiteboard draw sketches of what each of these types of error would represent visually, and come up with mathematical expressions.

Optional Extension: Encoders in Real-Time

You can also read the robot's wheel positions real-time from the */encoders* ROS topic and use this to plot the robot's motion real-time as it moves through the exercise...or even to correct for motion errors to bring it back closer to the desired path!

Exercise 35.8

Modify the driveforward.m function to use encoder feedback instead of time to determine when to stop the robot.

Exercise 35.9

Try writing a function that commands the NEATO to drive a square with the side length as an input variable. Think about using the encoder feedback to determine when to turn, and whether the robot has turned 90 degrees. The *'/cmd_vel'* Ros topic might be a good option here because it takes the linear and angular velocity as inputs. If the square is too easy, how about driving a star?

Exercise 35.10

Feedback from the encoders can be used to correct the robot's position if it strays from the planned trajectory. Think about how you could calculate error between the anticipated and actual path, and what actions would need to be taken to reduce that error.

35.7 Preview of the Overnight [10 minutes]

Introduction to the Bridge of Doom challenge.

Solution 35.2

1. Qualitatively describe the motion of the NEATO in each case.

   a) For the parameterization $u(t) = \beta t$, the NEATO drives around the circle described by $\mathbf{r}(u)$ at a constant velocity. The velocity of the NEATO will be impacted by the choice of β.

   b) For the parameterization $u(t) = \beta t^2$, the NEATO drives around the circle described by $\mathbf{r}(u)$ with an increasing velocity due to the t^2 term in the parameterization.

   c) For the parameterization $u(t) = \beta(2 + \sin(t))t$, the NEATO will travel around the circle described by $\mathbf{r}(u)$, but the velocity will oscillate between positive and negative, causing the robot to actually backup along the path at times.

2. How long does it take to traverse the curve once? (Your answer should depend on β) To traverse the circle once, the parameter $u \in [0, 2\pi]$. To determine the time to complete a traverse, we must solve for t in each of the parameterizations, and plug in $u = 2\pi$.

   a) For the parameterization $u(t) = \beta t$, $t = \frac{u(T)}{\beta}$, so $t_{final} = \frac{2\pi}{\beta}$.

   b) For the parameterization $u(t) = \beta t^2$, $t = \sqrt{\frac{u(T)}{\beta}}$, so $t_{final} = \sqrt{\frac{2\pi}{\beta}}$. Note that this time is less than the previous parameterization because the robot is speeding up throughout the traverse of the curve.

   c) For the parameterization $u(t) = \beta(2 + \sin(t))t$, $\frac{u(t)}{\beta t} = 2 + \sin(t)$, so $\frac{u(t_{final})}{\beta t} = 2 + \sin(t_{final})$. This can be solved using Matlab or any other numerical solver.

3. How would you find the linear velocity of a robot traveling along the curve? What is the direction of this velocity? (It would be really great if you employed the chain rule here and kept your work as general as possible.) The linear velocity would be found by

$$\mathbf{v}(t) = \frac{d}{dt}[\mathbf{r}(u(t))] = \mathbf{r}'(u(t))u'(t)$$

after applying the chain rule. For the function

$$\mathbf{r}(u(t)) = R\cos(u(t))\hat{\mathbf{i}} + R\sin(u(t))\hat{\mathbf{j}},$$

we can write

$$\frac{d}{dt}\mathbf{r}(u(t)) = -R\sin(u(t))u'(t)\hat{\mathbf{i}} + R\cos(u(t))u'(t)\hat{\mathbf{j}}$$

where

$$u'(t) = \frac{d}{dt}u(t).$$

The velocity is in the direction of the unit tangent vector

$$\hat{\mathbf{T}} = \frac{\mathbf{r}'}{|\mathbf{r}'|} = \frac{-R\sin(u(t))u'(t)\hat{\mathbf{i}} + R\cos(u(t))u'(t)\hat{\mathbf{j}}}{Ru'(t)} = -\sin(u(t))\hat{i} + \cos(u(t))\hat{j}$$

.

4. How would you find the angular velocity? What is the direction of the angular velocity? (It would be really great if you employed the chain rule here and kept your work as general as possible.) The angular velocity can be found as $\omega = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}|_{room} = u'(t)\hat{k}$. The angular velocity will always be in the $\pm\hat{k}$ direction. Positive $\hat{k}$ indicates a counterclockwise rotation, while negative $\hat{k}$ indicates a clockwise rotation.

5. Having found the linear and angular velocity, how would you find the left and right wheel speeds for the differential drive?

$$V_L = V - \omega\frac{d}{2} \tag{35.1}$$

$$V_R = V + \omega\frac{d}{2} \tag{35.2}$$

6. The robot has a maximum wheel speed of 0.3 m/s. How would you choose β to ensure your robot never exceeds this speed limit?

   We know from above that $\frac{d}{dt}\mathbf{r}(u(t)) = -R\sin(u(t))u'(t)\mathbf{\hat{i}} + R\cos(u(t))u'(t)\mathbf{\hat{j}}$, and the magnitude of the linear velocity is $|r'(u(t))| = R|u'(t)|$. We also know that $|\omega| = |u'(t)|$. Looking at the equations for left and right wheel velocity above, we would choose β so:

$$0.3 \geq Ru'(t) - |u'(t)|\frac{d}{2} \tag{35.3}$$

$$0.3 \geq Ru'(t) + |u'(t)|\frac{d}{2} \tag{35.4}$$

Solution 35.3

Solutions to this exercise will vary by breakout room based on your robot's trajectory.

Solution 35.4

Given measured values for V_L and V_R determine the values of $\mathbf{r}(t+\Delta t)$, and $\theta(t+\Delta t)$ which represent the position and heading of your robot at time $t+\Delta t$.

Conceptually, we can think about the robot making discrete moves in space over the period Δt. Each of these moves would consist of a translation and a rotation. From V_L and V_R, we can find the linear and angular velocities as:

$$V = \frac{V_L + V_R}{2}$$

$$\omega = \frac{V_R - V_L}{d}$$

With these velocities, for a robot at position $\mathbf{r}(t) = x(t)\hat{i} + y(t)\hat{j}$ in the global frame, we can then say:

$$\mathbf{r}(t+\Delta t) = (x(t) + V_x\Delta t)\hat{i} + (y(t) + V_y\Delta t)\hat{j}$$

$$\mathbf{r}(t+\Delta t) = (x(t) + V\cos(\theta(t))\Delta t)\hat{i} + (y(t) + V\sin(\theta(t))\Delta t)\hat{j}$$

To fully define the position of the robot, we must know its orientation $\theta(t)$ with respect to the global reference frame, in addition to its position $\mathbf{r}(t)$

$$\begin{aligned} \mathbf{r}(t+\Delta t) &= \mathbf{r}(t) + v\hat{\mathbf{T}}\Delta t \\ \theta(t+\Delta t) &= \theta(t) + \omega\Delta t \end{aligned}$$

Solution 35.5

1. Does the plot of the wheel linear travel look as you would expect? Example plot below for the case of $V_L = -0.3$ and $V_R = -0.3$ initially.

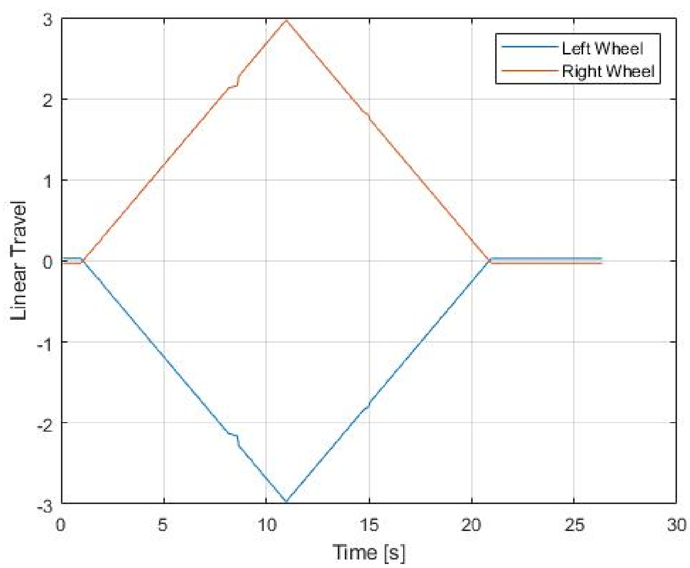


Figure 35.1: Linear travel of left and right wheel while spinning in circles around center.

2. Find the left and right wheel velocity at each time step and plot them. Do they match your expectations? Example plot below for the case of $V_L = -0.3$ and $V_R = -0.3$ initially.

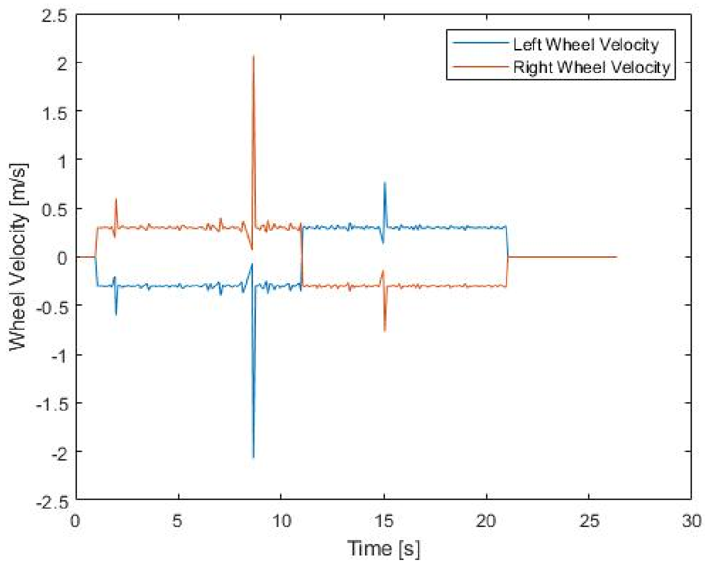


Figure 35.2: Velocity of left and right wheel while spinning in circles around center.

Solution 35.6

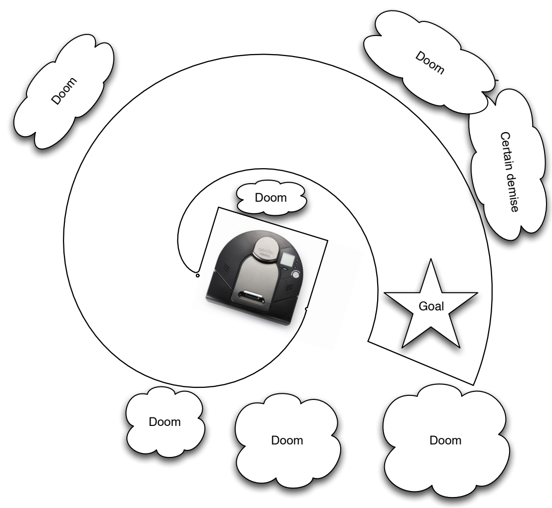
On a real robot, the actual distance traveled will differ from the commanded distance due to factors including inertia, friction, slight errors in the motor model, etc.

Chapter 36

Crossing the Bridge of Doom

36.1 Overview

Welcome to Robo Ninja Warrior. Your first challenge, should you choose to accept it (and you should!), will be crossing the *Bridge of Doom*. This challenge will push you and your robot literally to the brink, requiring you to be at the height of your analytical powers. Along the way you'll be building your knowledge of parametric curves, deriving robot motion models, and learning powerful validation and debugging techniques.



The Bridge of Doom.

Learning Objectives

By the end of this challenge, you should be able to:

1. Compute tangent vectors and normal vectors to parametric curves, and connect these vectors to motion.
2. Derive a motion model of a robot.
3. Validate a motion model of a robot empirically.
4. Control a robot using an open-loop control strategy.
5. Map the path of a robot based on encoder values.

36.2 The Challenge

You will write a program to autonomously pilot your robot from the starting platform to the goal. What lies between? Ahh, that is the harrowing Bridge of Doom. The shape of the centerline of the Bridge of Doom is defined by the following parametric curve:

$$\mathbf{r}(u) = 4 * [0.3960 \cos(2.65(u + 1.4))\mathbf{i} - 0.99 \sin(u + 1.4)\mathbf{j}].\ (u \in [0, 3.2])$$

36.3 (Re)Meet your Neato

You've already achieved passing familiarity with your Neato, however, in this challenge you two will really get acquainted! The Neato moves via differential drive, which we worked with in class Monday. In this challenge you are tasked with piloting your robot across the Bridge of Doom. In order to control your robot, you will be using open-loop control. Open-loop control means that you will determine a sequence of motor commands (e.g., the velocities for each of the Neato's wheels) ahead of time. You will then write a program that sends these motor commands to the robot at the prescribed times irrespective of where the robot is along the path. Despite its simplicity, open-loop control can be quite powerful, and it is up to the task of crossing the Bridge of Doom. That being said, if you are looking for ways to take this challenge to the next level, you can use sensor feedback to modify your path midstream.

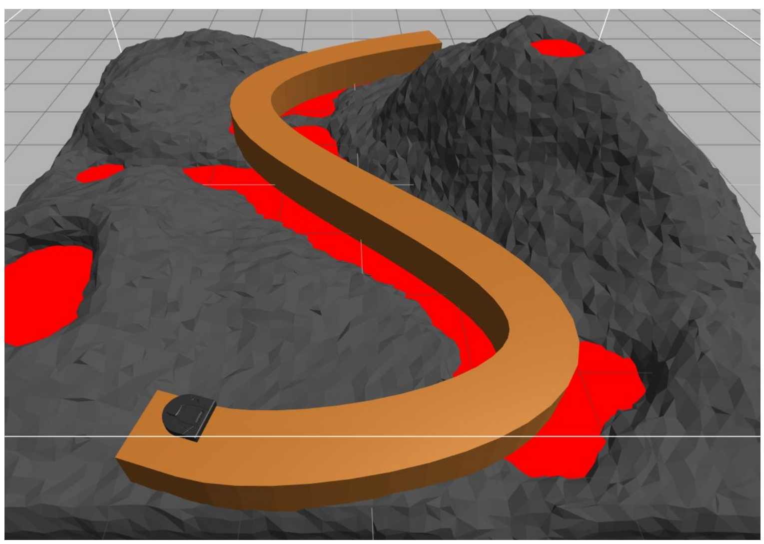


Figure 36.1: The fearless (simulated) NEATO traversing the treacherous Bridge of Doom.

For this challenge, we are considering the maximum wheel speed of the simulated NEATO to be 2.0 m/s. Your commanded wheel speeds should not exceed this value.

36.4 Crossing the Bridge of Doom

Let's reflect on how far we've come towards completing our challenge. We have developed equations for V_L and V_R that achieve a desired linear and angular velocity, and we have validated this model empirically. All that remains is to program our robot to follow a parametric curve.

Exercise 36.1

For the Bridge of Doom, plot the parametric curve that defines the centerline of the bridge. On the same figure, plot the unit tangent and unit normal vectors at several points along the curve. You should have starter code to help with this in the Night 1 assignment.

Exercise 36.2

Compute and plot your robot's left and right wheel velocities as a function of time for the Bridge of Doom. After successfully traversing the bridge (you will do this in a later exercise), add the measured wheel velocities to your plot. The planned (theoretical) velocities should be plotted with a solid line, while the experimental result should be plotted with a dashed line. Make sure your plots include appropriate units, labels, and legends.

Exercise 36.3

Compute and plot your robot's planned linear speed and angular velocity as a function of time for the Bridge of Doom. After successfully traversing the bridge, add the measured linear speed and angular velocity to your plot (**Note:** The wheelbase for the simulated NEATO is d=0.235m). The planned (theoretical) speed and angular velocity should be plotted with a solid line, while the experimental result should be plotted with a dashed line. Make sure your plots include appropriate units, labels, and legends.

Exercise 36.4

Write a program to send the appropriate control signal based on the time elapsed since the start of the path. Be careful about handling the case when $|\mathbf{r}'(t)| = 0$. In this case $\hat{\mathbf{N}}(t)$ is not defined and $\omega(t) = 0$. Note, that you can always slow down or speed up your robot by multiplying t by a constant (we have used the constants α or β in previous assignments for this purpose). Be careful since the maximum speed of each of the robot's wheels is $2.0 m/s$. Take a video of your robot crossing the Bridge of Doom.

To get you started:

1. Pull the latest version of the QEA Robot software by running the following command in PowerShell.

```
docker pull qeacourse/robodocker:spring2020
```

2. To create the the Bridge of Doom environment in the simulator, use the command “NEATO_WORLD=bod_volcano” in PowerShell.

3. The starter code here will initialize the position of the simulate NEATO on the Bridge of Doom. You can build the rest of your robot control program from this starting point.

Exercise 36.5

Map your robot’s predicted and actual path crossing the Bridge of Doom by using the provided code collectDataset_sim.m to collect the wheel encoder data and convert that to coordinates and headings for the robot throughout its perilous journey. Use the Matlab quiver command to plot the predicted and experimental unit tangent vectors at various points along the curve (note: do not include an arrow for every time step or your plot will be too cluttered). The planned (theoretical) path should be plotted with a solid line, while the experimental result should be plotted with a dashed line. Make sure your plots include appropriate units, labels, and legends.

36.5 Writing up Your Work

Exercise 36.6

Prepare a writeup of your work on this challenge. Your writeup should contain the following components.

1. A brief introduction explaining the challenge and including the functional definition of the bridge.

2. A description of your methodology explicitly identifying and explaining the key equations and parameters needed to produce the plots identified above.

3. Each of the plots detailed above with appropriate captions.

4. A link to a youtube video of your robot in action (include a link in your writeup). Bonus points (not really) for high production value like this.

In addition to the writeup, you should also turn in your carefully commented code (you could add a link to a Github repo in your writeup, or upload your MATLAB code files to your Canvas submission).

36.6 Optional Extensions

Exercise 36.7

1. One weakness of the approach that you implemented is that it doesn't take into account the fact that the robot doesn't instantaneously do what you tell it to do. One possible way to remedy this is to monitor the robot's position over time using live readings of the wheel encoders. In class on Day 3 you derived a method for updating the robot's position and orientation given measurements of its wheel velocities. We call this estimate of the robot's position its odometry. By comparing the robot's position as determined by its odometry with the desired position (given by $\mathbf{r}(t)$) you can try to correct your robot's motion to more faithfully follow the path. How you accomplish this exactly is up to your own creativity and analysis skills.

2. In Night 1 of the Duckies module you proposed a set of mathematical curves that were a good approximation for the curves of a fruit or vegetable. Here is your opportunity to have a robot drive a fruit or vegetable shaped trajectory, and plot the resulting path using encoder data. Take a video if you do this!

3. If you'd like, you can traverse the Icy Bridge of Doom. Change the NEATO_WORLD from bod_volcano to bod_ice_bridge. In order to navigate this, you will likely need to use the actual position as feedback to modify your control strategy. The actual position of your Neato is given by the simulator. For an example of how to do this see the motion model code experiment code from Day 2 (basically e-mail us if you want to do this!).

Chapter 37

Day 4: Bridge of Doom Work Day

A good time to work on Bridge of Doom and get all the help you need!

Chapter 38

Night 4: Partial Derivatives, Chain Rule, Max and Min

Learning Objectives

Concepts

- Compute partial derivatives using the chain rule and product rules for differentiation.
- Determine critical points of a continuous function of single variable.
- Use the gradient function to compute critical points of multivariate functions.
- Evaluate whether critical points of a multivariate function are local maxima, minima or otherwise.

Matlab Skills

- Use MATLAB to maneuver a robot through the Bridge of Doom challenge.

Bridge of Doom Challenge

Take some time to complete the challenge.

38.1 Partial Derivatives and the Chain Rule

We met partial derivatives in Robo Night 2, and we are now going to return to this idea to reinforce it and to extend it. Work your way through sections 14.3 and 14.5 from the book `Multivariable Calculus by Stewart`—we've included section 14.6 for completeness, but we will discuss that chapter in a future assignment.

Exercise 38.1

1. Please read Section 14.3 from Stewart on Partial Derivatives. Take notes on important concepts and definitions.
2. Please read Section 14.5 from Stewart on The Chain Rule. Take notes on important concepts and definitions.

   This is new material on extending the notion of the chain rule from functions of one variable to functions of many variables. The main results are captured in the pink boxes labeled 1 through 4 - these are various cases of the chain rule. Again, this text is written for a student who doesn't have linear algebra. As you read these rules, think about how you might use matrix notation to make this cleaner and more compact. At this stage you should ignore the section on Implicit Derivatives - it will be too confusing and take too long.

Do the following exercises (by hand or using a Computer Algebra System)

3. Complete question 1 from 14.5 Exercises.
4. Complete question 5 from 14.5 Exercises.
5. Complete question 11 from 14.5 Exercises.

38.2 Max and Min of Single Variable Functions

In high school you probably spent a good amount of time thinking about the relative maximum or relative minimum of a function of one variable. You've probably met the following idea before:

$x = a$ is a critical point of $f(x)$ if $f'(a) = 0$.

This simply means that a critical point is a point where the derivative is zero.

Exercise 38.2

Find the critical point(s) of $f(x) = x^4 - x^2 + 1$ (can be done by hand or computer algebra system).

Once we have the critical points, there is a straightforward test to determine whether any critical point is a relative maximum or relative minimum.

The critical point $x = a$ is a relative minimum if $f''(a) > 0$. The critical point $x = a$ is a relative maximum if $f''(a) < 0$.

This means that in order for a point to be a relative minimum, the first derivative must be zero and the second derivative must be positive, and vice versa for a relative maximum. If both the first and second derivatives are zero then no conclusion can be drawn without further investigation.

Exercise 38.3

Classify the critical point(s) of $f(x) = x^4 - x^2 + 1$.

If you would like to review this material or have more practice please check out the following sections from Paul's Online Math Notes:

- Critical Points
- Minimum and Maximum Values

38.3 Max and Min of Multivariable Functions

Let's examine the corresponding idea for functions of two variables.

> $(x, y) = (a, b)$ *is a critical point of* $f(x, y)$ *if* $\nabla f(a, b) = \mathbf{0}$. *Recall that the gradient vector is composed of the first-partial derivatives,* $\nabla f = [f_x f_y]$.

This means that in order for a point (a, b) to be a critical point in two dimensions then both partial derivatives need to be zero there.

Exercise 38.4

Determine the critical points of the following functions (can be done by hand or using a Computer Algebra System.)

1. $f(x, y) = 4 + x^3 + y^3 - 3xy$
2. $f(x, y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$

Once we have the critical points, there is another straightforward test to classify the type of critical point. However, when we have more than a single variable, the critical points can be something other than a simple maximum or minimum. This is illustrated in figure below of the $f(x, y) = x^2 - y^2$. At the critical point, $(0, 0)$, $\nabla f = 0$. As you look at $(0, 0)$ does it appear to you as a maximum or a minimum point? To see what is unusual about $(0, 0)$, imagine you were to take a cut of the yz-plane at $x = 0$. What would you see? Then take a cut of the xz-plane at $y = 0$; what would you see in this case?

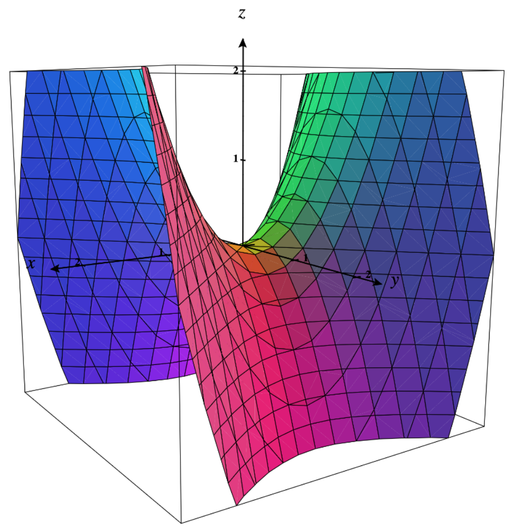


Figure 38.1: The plot of a saddle function. Generated with CalcPlot3D.

The following approach to characterizing critical points is not usually discussed in a multi-variable calculus course because students are not expected to know linear algebra.

Suppose (a, b) is a critical point of $f(x, y)$. The Hessian matrix Hf evaluated at the critical point has real eigenvalues λ_1 and λ_2 (Recall that a symmetric matrix has real eigenvalues). Then the following classifications are possible:

1. *If $\lambda_1 < 0$ and $\lambda_2 < 0$, then the critical point is a relative maximum.*
2. *If $\lambda_1 > 0$ and $\lambda_2 > 0$, then the critical point is a relative minimum.*
3. *If $\lambda_1 > 0$ and $\lambda_2 < 0$ OR $\lambda_2 > 0$ and $\lambda_1 < 0$, then the critical point is a saddle.*
4. *If $\lambda_1 = 0$ or $\lambda_2 = 0$ (or both equal zero) at the critical point then no conclusions can be drawn without further investigation.*

So we see that for a function of two variables that there is a new type of critical point, the saddle - it is a relative maximum in one direction, and a relative minimum in another.

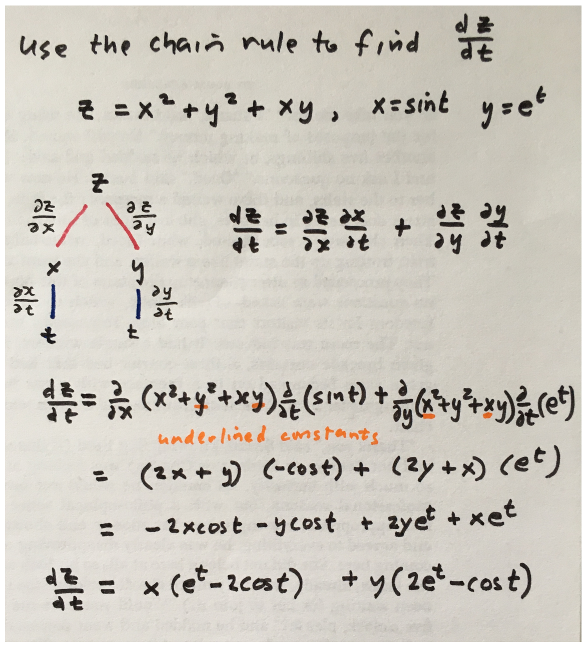
Exercise 38.5

Classify the critical points of the following functions

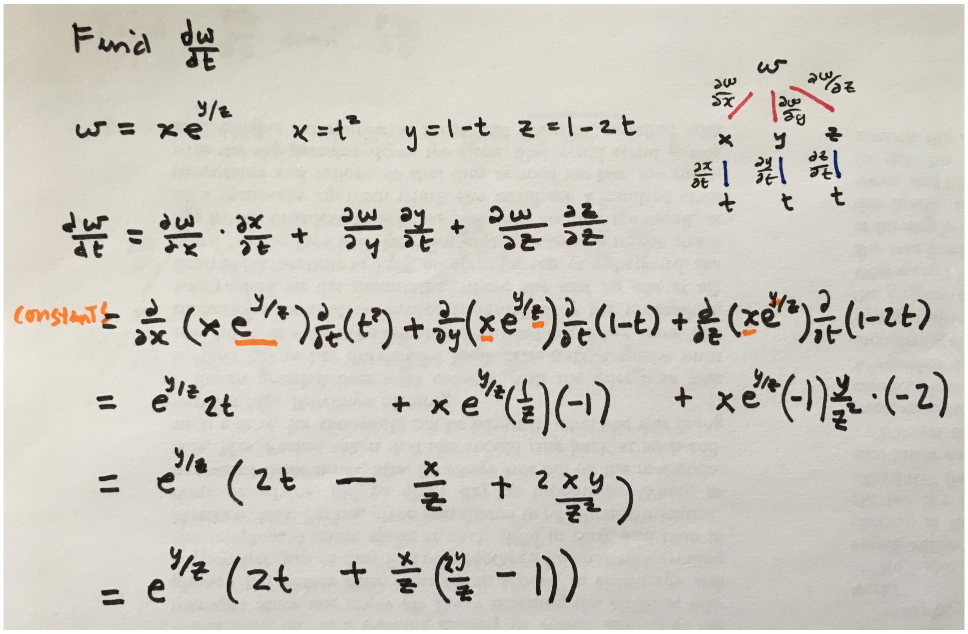
1. $f(x, y) = 4 + x^3 + y^3 - 3xy$
2. $f(x, y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$

Solution 38.1

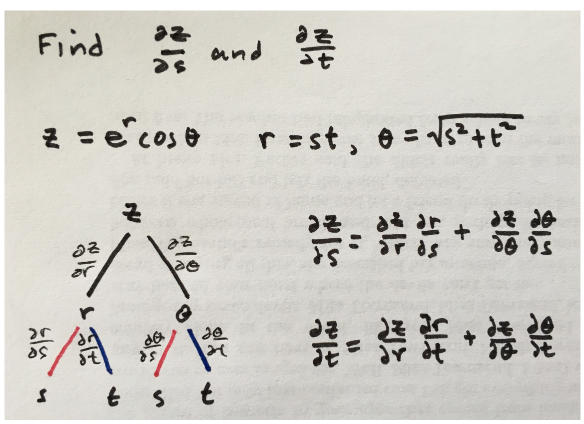
- Complete question 1 from 14.5 Exercises. Notice that z is a function of two variables, x and y; x and y vary with t, so to compute the $\frac{dz}{dt}$, we need to sum portions of $\frac{dz}{dt}$ due to x and the portion due to y.



- Complete question 5 from 14.5 Exercises. Notice that ω is a function of three variables (x, y and z), each varying with t, so to compute the $\frac{d\omega}{dt}$, we need to sum that parts of the total $\frac{d\omega}{dt}$ due to the partial contributions ($\frac{\partial\omega}{\partial t}$) of x, y and z.



- Complete question 11 from 14.5 Exercises. Notice that z is a function of two variables (r and θ), each varying with s and t. To compute $\frac{\partial z}{\partial s}$, for example, we need to sum portions of $\frac{\partial z}{\partial s}$ due to the variation with ∂s in each of r and θ. In this case, $\frac{\partial z}{\partial s}$ are partial derivatives, since the total dz consists of ∂s and ∂t.



$$\frac{\partial z}{\partial s} = \frac{\partial}{\partial r}\left(e^r \underbrace{\cos\theta}_{\text{constants}}\right)\frac{\partial}{\partial s}(st) + \frac{\partial}{\partial \theta}\left(e^r \cos\theta\right)\frac{\partial}{\partial s}\left((s^2+t^2)^{1/2}\right)$$

$$= \cos\theta\, e^r\, t + e^r(-\sin\theta)\left(\tfrac{1}{2}\right)(s^2+t^2)^{-1/2}(2s)$$

$$\frac{\partial z}{\partial s} = e^r\left(t\cos\theta - \frac{s\cdot\sin\theta}{\sqrt{s^2+t^2}}\right)$$

minus sign!

$$\frac{\partial z}{\partial t} = \frac{\partial}{\partial r}\left(e^r \underbrace{\cos\theta}_{\text{constants}}\right)\frac{\partial}{\partial t}(st) + \frac{\partial}{\partial \theta}\left(e^r \cos\theta\right)\frac{\partial}{\partial t}\left((s^2+t^2)^{1/2}\right)$$

$$= \cos\theta\, e^r\, s + e^r(-\sin\theta)\left(\tfrac{1}{2}\right)(s^2+t^2)^{-1/2}(2t)$$

$$\frac{\partial z}{\partial t} = e^r\left(s\cos\theta - \frac{t\cdot\sin\theta}{\sqrt{s^2+t^2}}\right)$$

Solution 38.2

We first take the derivative so that $f'(x) = 4x^3 - 2x$ and then we set it equal to zero so that $4x^3 - 2x = 0$ and then we solve for x. Since x is a common factor we see that the solutions are dictated by $x = 0$ and $4x^2 - 2 = 0$ or $x = \pm\sqrt{2}/2$. This function therefore has three critical points.

Solution 38.3

We've already found the critical points. Let's now determine the second-derivative, and evaluate it at each of the critical points. The second derivative is $f''(x) = 12x^2 - 2$ and evaluating at each of the critical points gives:

$$\begin{aligned} f''(0) &= -2 \\ f''(\pm\sqrt{2}/2) &= 4 \end{aligned}$$

The critical point at $x = 0$ is therefore a relative maximum, while the critical points at $\pm\sqrt{2}/2$ are both relative minima. Graphing the function would confirm this very quickly.

Solution 38.4

1. We already met this function in Robo Night 2 and computed its partial derivatives then. The gradient vector is:
$$\nabla f = \begin{bmatrix} 3x^2 - 3y \\ 3y^2 - 3x \end{bmatrix}$$
We now determine the values of x and y for which both derivatives are zero simultaneously. Setting the first component to zero leads to $y = x^2$ which we can substitute into the second component and set it equal to zero to obtain $x^4 - x = 0$. The solutions are therefore $x = 0$ and $x = 1$ which leads to the points $(0, 0)$ and $(1, 1)$.

2. We've already met this function in Robo Night 2 and computed its partial derivatives then. The gradient vector is:
$$\nabla f = \begin{bmatrix} 6xy - 6x \\ 3x^2 + 3y^2 - 6y \end{bmatrix}$$
We now determine the values of x and y for which both derivatives are zero simultaneously. Setting the first component to zero leads to $x = 0$ or $y = 1$ which we can substitute into the second component and set it equal to zero to obtain two possibilities: $x = 0$ AND $3y^2 - 6y = 0$ OR $y = 1$ AND $3x^2 - 3 = 0$. The first possibility has $y = 0$ ad $y = 2$ as solutions, while the second possibility has $x = \pm 1$ as solutions. There are therefore 4 solutions: $(0, 0)$,$(0, 2)$,$(1, 1)$, and $(-1, 1)$.

Solution 38.5

1. We already know the critical points are $(0, 0)$ and $(1, 1)$. We need to compute the Hessian matrix to classify it. Fortunately, we already computed the second-derivatives in Robo Night 2 so that
$$Hf = \begin{bmatrix} 6x & -3 \\ -3 & 6y \end{bmatrix}$$
   - At $(0, 0)$ we see that
   $$Hf = \begin{bmatrix} 0 & -3 \\ -3 & 0 \end{bmatrix}$$
   which has eigenvalues $\lambda_1 = -3$ and $\lambda_2 = 3$ so that this critical point is therefore a saddle.
   - At $(1, 1)$ we see that
   $$Hf = \begin{bmatrix} 6 & -3 \\ -3 & 6 \end{bmatrix}$$
   which has eigenvalues $\lambda_1 = 9$ and $\lambda_2 = 3$ which implies that this point is a relative minimum.

2. We already know the critical points are $(0, 0)$,$(0, 2)$,$(1, 1)$, and $(-1, 1)$ We need to compute the Hessian matrix to classify it. Fortunately, we already computed the second-derivatives in Robo Night 2 so that
$$Hf = \begin{bmatrix} 6y - 6 & 6x \\ 6x & 6y - 6 \end{bmatrix}$$

- At $(0, 0)$ we see that
$$Hf = \begin{bmatrix} -6 & 0 \\ 0 & -6 \end{bmatrix}$$
which has eigenvalues $\lambda_1 = -6$ and $\lambda_2 = -6$ which implies that this point is a relative maximum.
- At $(0, 2)$ we see that
$$Hf = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$$
which has eigenvalues $\lambda_1 = 6$ and $\lambda_2 = 6$ which implies that $(0, 2)$ is a relative minimum.
- At $(1, 1)$ we see that
$$Hf = \begin{bmatrix} 0 & 6 \\ 6 & 0 \end{bmatrix}$$
which has eigenvalues $\lambda_1 = -6$ and $\lambda_2 = 6$, which implies that this point is a saddle.
- At $(-1, 1)$ we see that
$$Hf = \begin{bmatrix} 0 & -6 \\ -6 & 0 \end{bmatrix}$$
which has eigenvalues $\lambda_1 = -6$ and $\lambda_2 = 6$, so that this point is also a saddle.

Chapter 39
Day 5: Optimization

39.1 Virtual Class Schedule

- 1000-1005: Tech time
- 1005-1015: Introduction to the Day
- 1015-1035: Debrief and Synthesis
- 1035-1115: Introduction to Optimization
- 1115-1130: Coffee
- 1130-1150: The World of Optimization
- 1150-1220: Optimizing your Olin experience - Gross Happiness Index
- 1220-1230: Looking ahead

Learning Objectives

Concepts

- Describe the three elements of an optimization problem: *objective function, decision variable, constraints*
- Reframe a description of a optimization problem in terms of its *objective function* and *decision variable*, subject to *constraints*
- Differentiate between a single or multi-objective optimization problem.

39.2 Debrief and Synthesis/Demos on the Bridge of Doom

- In your breakout groups, celebrate your successes (remember, learning is a success, even if your NEATO ended up in a molten pool of lava!). Share videos if you'd like to.
- As a group, draw a concept map of the main ideas you used in this challenge. Try to make connections to ideas from the previous two modules.
- Discuss what you tried, what your results were, what is still unknown, and how well do you think you characterized the model's accuracy.
- Discuss your process in the Bridge of Doom challenge. What worked and what did not as you engaged with this challenge? What would you do differently next time?

Partial derivatives - Try this if you have time

Exercise 39.1

Consider the figure below, which represents different views of the same $f(x, y)$. For the following critical points, determine whether each element in the Hessian of $f(x, y)$ is positive(+) or negative (-), or something else.

1. $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$
2. $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$
3. $(0, 0)$

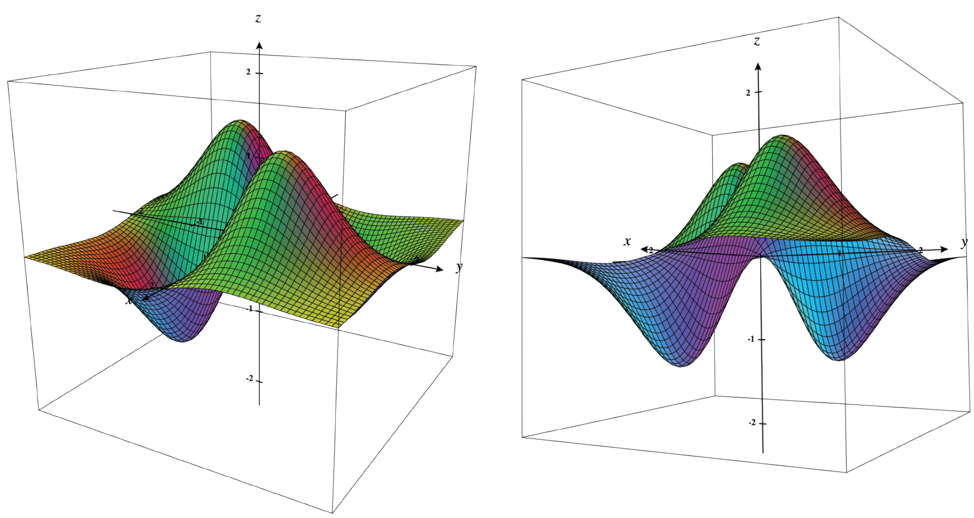


Figure 39.1: Different views of an $f(x, y)$.

39.3 *Introduction to Optimization*

Over the course of the next few days we'll be learning about and applying ideas around optimization to the motion of a NEATO. You might have dealt with some of this before – e.g., maximum or minimum problems in calculus. We'll be broadening these ideas to deal with functions of many variables, as well as applying linear algebra ideas in order to allow us to optimize in situations that are much more complex than those typically found in a calculus textbook.

Optimization is really a type of modeling work: just as in ModSim, you begin by creating a model for your physical system, and then you apply mathematics in order to make a prediction with that model; most optimization problems involving more than one dimension will require computational algorithms and tools. For example, in MATLAB, a set of such tools is in the Optimization Toolbox.

39.4 *The Language of Optimization*

Many optimization problems boil down to this question: Subject to (s.t.) a set of *constraints*, what combination of *decision variables* leads to the best possible value of the *objective function*? An optimization problem, and therefore the solution method that you would use, is defined by the three properties, illustrated in the figure below. The particulars of the problem's structure (e.g., whether there is one or many competing objectives), determines the computational solution strategy.

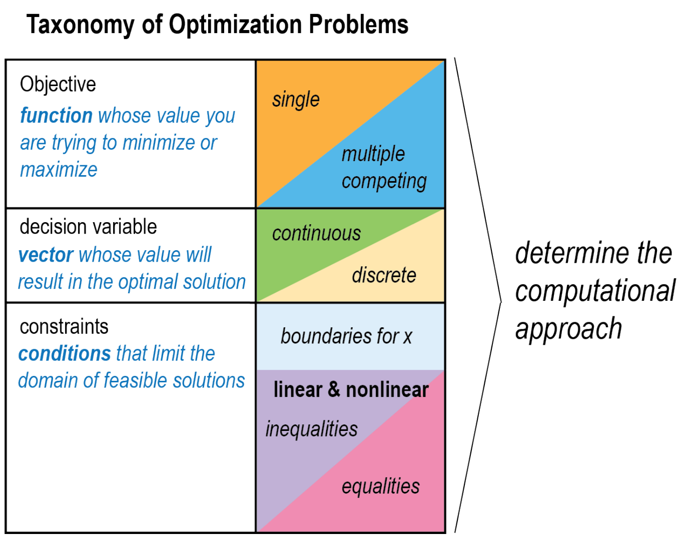


Figure 39.2: A taxonomy of optimization problems. Solving optimization problems involving more than one variable usually requires computational tools.

Objective Function

The objective function is the quantity that you are trying to maximize or minimize. Maybe you're trying to minimize the greenhouse gases (ghg) produced in the manufacture of a product under different processing conditions; maybe you're trying to maximize the fatigue life of a joint; maybe you're trying to minimize the risk of viral spreading under different settings in an optical screening method. In each of these cases, you define a quantity (ghg/product, fatigue life, risk) that you are trying to optimize. And in each of these cases, you would devise a mathematical model that relates the objective quantity to the decision variable(s). Your objective function is a model which takes the decision variable as inputs, and outputs the quantity you're trying to optimize.

Decision Variables

Decision variables are basically the “knobs” you can turn to optimize the objective function; generally you are solving for a vector that represents the ’optimum ’ decision variables. Sometimes decision variables are continuous quantities (e.g., how long a member in a truss is) and other times they must take on discrete values (e.g., the number of cases of Twinkies you order for your business).

Constraints

Optimization is made challenging by uncertainty in the model and limited choices when you “turn the knobs”. For example, while you can make a truss as strong as you want to by adding material, presumably doing so increases both the cost and the mass of the truss – and both of these might be constrained. Often constraints involve inequalities: you might want to maximize the strength of the truss subject to the constraint that the mass is less than 100 kg and the cost is less than 200 dollars. Like objective functions, constraints are mathematical statements that effectively constrain the domain of feasible solutions for the decision variable.

A Familiar Warm-up : Remembering calculus

Here’s an example that may be familiar. Let’s say you are manufacturing a cylindrical package for a food product which will require a volume of 20π cm^3. You have chosen different materials for the ends and the cylinder; each material produces waste in the manufacturing process. The side material wastes 8 gm for every cm^2 of material used; The end material wastes 10 gm for every cm^2 of material used. You’re interested in minimizing the waste through designing the optimum r and h for the cylindrical package.

Exercise 39.2

Create a structure for this optimization problem.

1. List the *objective function*, *decision variable*, and *constraints*. State the *objective function* as an equation that is a function of the *decision variable*.
2. Categorize the *constraints* into limits of the *decision variable*, linear constraints (equalities and inequalities) and nonlinear constraints (equalities and inequalities). Express these as equations or inequalities.
3. Challenge: If you have time, go ahead and solve this optimization problem through hand calculations.

An Example: Setting up the optimization problem

Let’s say Olin is trying to build an optimization model for divesting of fossil fuel energy sources. Olin has a budget to invest in n different renewable energy technologies.

- Each technology displaces the need for coal-based energy sources at a different rate, r_i. For example, if the renewable source displaces 15% of the non-renewable, it's $r = 0.15$.
- Each renewable varies in its mean displacement rate by a standard deviation, σ_i; we consider this a "risk."
- We want to create a diverse portfolio of renewable energy 'investments' that maximize the rate that renewable energies displace non-renewable, subject to a 'tolerable' level of risk. Let's say 10% is tolerable.

Let w_i represent the fraction of the portfolio invested in renewable energy technology i. We can think of the w_i as the weighting of each renewable technology in the total portfolio ($\sum w_i = 1$). The vector $\mathbf{w}$ contains the weighting values.

Then the expected renewable energy return for the portfolio is given by

$$E(\mathbf{w}) = \sum_i w_i r_i$$

and the risk associated with the portfolio is given by

$$\sigma_{total} = \sqrt{\sum_i w_i^2 \sigma_i^2}$$

Exercise 39.3

The optimization problem above could be stated this way: Find the vector $\mathbf{w}$ that maximizes the expected renewable energy return ($\frac{dE}{dw} = 0$) *subject to* the total investments being less than or equal to the money we have to invest ($\sum w_i = 1$) and subject to the risk being less than or equal to a specified amount.

Right now, you don't have enough information to find $\mathbf{w}$.

- Frame the optimization problem in terms of the *objective function*, *decision variable*, and *constraints*.
- Categorize the *constraints* into limits of the *decision variable*, linear constraints (equalities and inequalities) and nonlinear constraints (equalities and inequalities).
- State all of the above in terms of equations or inequalities.

Solving an optimization problem - or trying to...

See the exercise below. For the first two, consider sketching a graph of what the objective function looks like before working toward a solution.

Exercise 39.4

For the renewable energy portfolio problem above, please *perform* (or attempt) the optimization by hand– i.e., find the renewable energy portfolio distribution to maximize renewable rate of return a 10% total risk.

1. For a portfolio with two technologies: $r_1 = 0.1$, $\sigma_1 = 0.1$; $r_2 = 0.2$, $\sigma_2 = 0.05$. *Before starting this, what do you notice about the r's and σ's ? If you plot the $E(\mathbf{w})$, what does it imply as an optimal solution?*

2. For a portfolio with two technologies: $r_1 = 0.1$, $\sigma_1 = 0.1$; $r_2 = 0.2$, $\sigma_2 = 0.3$. *Before starting this, what do you notice about the r's and σ's ? If you plot the $E(\mathbf{w})$, what does it imply as an optimal solution?*

3. For an eight technology portfolio, where $\mathbf{r} = [0.1, 0.13, 0.15, 0.17, 0.2, 0.23, 0.25, 0.28]$ and $\boldsymbol{\sigma} = [0.2, 0.1, 0.4, 0.25, 0.1, 0.35, 0.32, 0.35]$. *Any ideas? Doing this by hand would require a rarefied mathematical prowess–I used a computer algorithm as a "black box" to do the work for me.*

Challenge (for those seeking one): *Optimization in Smile or Facial Recognition*

For the two problems below, please propose the appropriate formal mathematical framing for the optimization that you already performed.

4. Given a set of training data of images which have been tagged as smiling or not, formulate the design of a smile-predictor algorithm as an optimization problem.

5. Given a set of training data of images which have been tagged as individuals, formulate an eigen-face facial recognition algorithm as an optimization problem.

39.5 Coffee [15 minutes]

39.6 *The World of Optimization*

The world of optimization is large and rambling, and if you would like, you can explore this land, say at NEOS Guide. Let the traveler beware: much of this is written by and for computational scientists. In general, optimization problems are defined by the nature of their *objective function (s)*, *decision variable*, and *constraints*. These properties will determine the computational strategy used to solve the problem. The figure below shows three general categories of optimization problems, two of which are out-of-scope for this class, due to their complexity. These examples differ in these general ways,

1. Single v. Multiple Objectives
2. Deterministic v. Stochastic (i.e., Can determine a solution v. probable scenarios)
3. Continuous v. Discrete
4. Unconstrained v. Constrained
5. Linear v. Nonlinear Relationships

Exercise 39.5

How would you classify the optimization problem of the cylindrical package?

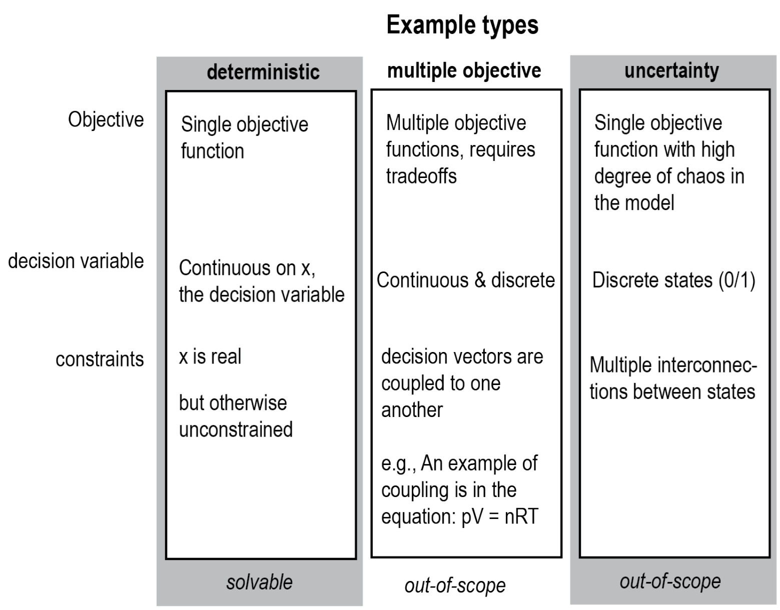


Figure 39.7: Examples of three general categories of optimization problems.

39.7 Optimizing your Olin experience - Gross Happiness Index

Now that we know a little about optimization, we are going to have you frame and design an optimization problem of your own as teams. We want you to have an optimal educational experience and fully realize that the current world conditions do not allow us to do that. However, let's consider the future and design together.

Before we start optimizing your engineering education experience, we will remember the National Society of Professional Engineer's Creed, to recall the commitment of the engineering profession.

Exercise 39.6

Create a short list of espoused priorities that you see in the National Society of Professional Engineer's Creed. Order these in order of highest to lowest; we will use this list later.

Next, we will introduce you to an index that the nation of Bhutan is using as a measure for its government's success: The Gross National Happiness Index (GNH). This particular index is a composite measure of the nation's Gross Happiness Index (GHI). Incidentally, a mantra of systems thinking is "We value what we measure and measure what we value." The U.S. has historically used the Gross Domestic Product (the annual monetary value of all finished goods, whether these goods are in fact socially 'good' or 'bad.'). The GHI purports to measure one's gross happiness as defined by nine domains of well-being,

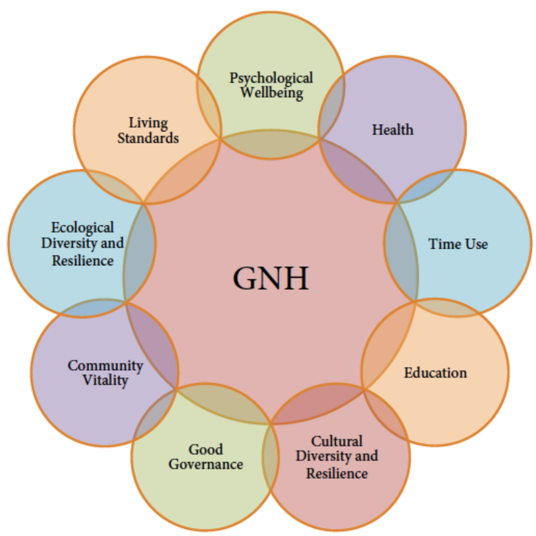


Figure 39.8: Nine domains of Bhutan's Gross Happiness Index.

- Living Standards - *material comforts*
- Health - *physical and mental*

- Education - *types, values & skills*
- Good governance - *perception of government functions*
- Ecological diversity and resilience - *environmental health*
- Time use - *work, play, sleep, life balance*
- Psychological well-being - *quality of life, spirituality*
- Cultural diversity and resilience - *traditions*
- Community vitality - *relationships, interactions*

Exercise 39.7

As a group, use the framework of GHI to design an index to optimize during your time at Olin. In the last 20 minutes of class today, you will present your design to a QEA faculty member. Aim to have a rough prototype of a design to optimize your Olin Gross Happiness Index.

1. First, do your best identify the type of optimization problem (*deterministic*, *multiobjective*, *uncertain*) and its properties (*objective function(s)*, *decision variable*, and *constraints*) . It may help to peruse the specifics of the GHI survey here.
2. Next, make a list of your 'professional' priorities and compare it to the espoused list in the Engineering Ethics Creed. How do these compare? Are they aligned? (They do not have to be any particular way.).
3. Next, choose a particular domain (or make up your own) and treat this as an objective function. What would you use for your decision variable, what kind of constraints (boundaries, linear or non-linear equalities/inequalites) would you develop? *There are no wrong answers to this design activity. In a way, this is an opportunity to communicate to a QEA faculty member what constitutes your happiness at Olin.*
4. How does it feel to think about mathematically optimizing your Olin Gross Happiness?

Solution 39.1

Consider the figure below, which represents different views of the same $f(x, y)$. For the following critical points, determine whether each element in the Hessian of $f(x, y)$ is positive(+) or negative (-), or something else. Recall that the H f is a matrix of the second partial derivatives of f. It has the form,

$$Hf = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

This question is essentially asking you to determine the curvature at the different critical points.

1. $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ You can see that this is a local maximum. Therefore, all the curvatures are negative (-)

2. $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ You can see that this is a local minimum. Therefore, all the curvatures are positive (+)

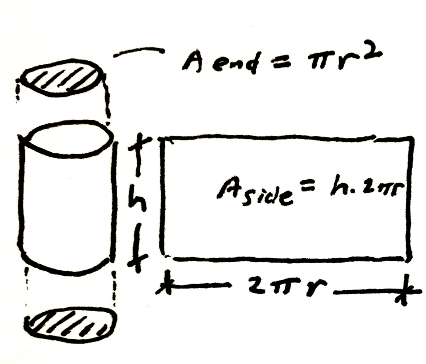
3. $(0, 0)$ This is a very strange point. It sort of looks like a saddle, but it's not. For this point, all the values of Hf are 0.

Solution 39.2

As a optimization problem, its structure is:
objective: *minimize waste*, w (units: gm)

$$w = 10\frac{gm}{cm^2} \cdot A_{ends} \cdot 2 + 8\frac{gm}{cm^2} \cdot A_{side}$$

$$w = 10\frac{gm}{cm^2} \cdot \pi r^2 \cdot 2 + 8\frac{gm}{cm^2} \cdot 2\pi rh$$



and, making sure that r and h are measured in cm, the objective function to minimize is

$$w = 20\pi r^2 + 16\pi rh$$

decision variable: r (units: cm)

> *We could equally choose h as a decision variable; r and h are coupled through the constraints.*

constraints: lower and upper bounds of the decision variable: $0 < r < 20$

> *This is a rough range. If we were using a computing algorithm, this range would help to keep the computer iterating possible solutions that are feasible.*

linear constraints expressed as equalities: *none*. (An example of one: $r = 2h$.)
linear constraints expressed as inequalities: *none*. (An example of one: $r > h$.)
nonlinear constraints expressed as equalities: *none*. (An example of one: $r^2 = 4$.)
nonlinear constraints expressed as inequalities: $V = \pi r^2 h \geq 20\pi$.

To optimize (i.e., minimize in this case) w, we substitute for h in w, and find the critical point where $\frac{dw}{dr} = 0$

$$w = 20\pi r^2 + 16\pi r \frac{20}{r^2}$$

$$\frac{dw}{dr} = 2 \cdot 20\pi r + (-1)(16 \cdot 20)\pi \frac{1}{r^2} = 0$$

This reduces to,

$$r^3 - 8 = 0$$

$$r = 2$$

and

$$h = \frac{20}{r^2} = \frac{20}{2^2} = 5$$

As a check, we can plot the objective function w as function of r, which indeed indicates that $r = 2$ is a minimum in w. We can also compute the volume to see if it meets the constraints.

$$V = \pi r^2 h = \pi 2^2 \cdot 5 = 62.8 \geq 20\pi$$

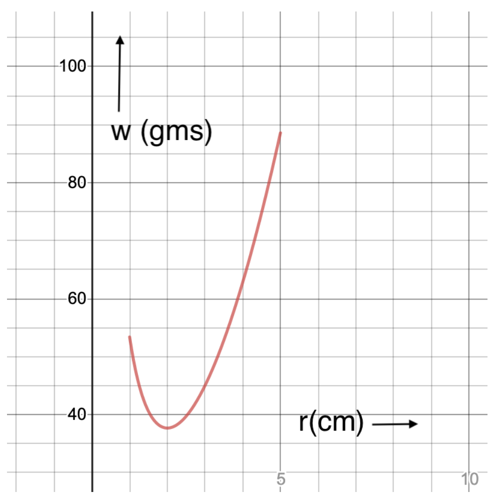


Figure 39.3: The objective function w v. radius

Solution 39.3

The structure of this optimization problem is:
objective: *maximize rate of renewables displacing non-renewables, E* (units: $\frac{kWh_R}{kWh_{NR}}$)

$$E(\mathbf{w}) = \sum_i w_i r_i$$

decision variable: $\mathbf{w} = [w_i, ..w_n]$ (units: fraction relative to total budget)
constraints: lower and upper bounds of the decision variable: $0 < [w_i, ..w_n] < 1$
linear constraints expressed as equalities: $\sum w_i = 1$.

linear constraints expressed as inequalities: *none*.
nonlinear constraints expressed as equalities: *none*.
nonlinear constraints expressed as inequalities: $\sigma_{max} = \sqrt{\sum_i w_i^2 \sigma_i^2} \leq 0.10$.

Solution 39.4

For the renewable energy portfolio problem above, please *perform* (or attempt) the optimization by hand– i.e., find the renewable energy portfolio distribution to maximize renewable rate of return a 10% total risk.

1. For a portfolio with two technologies: $r_1 = 0.1$, $\sigma_1 = 0.1$; $r_2 = 0.2$, $\sigma_2 = 0.05$. *Before starting this, what do you notice about the r's and σ's ? If you plot the $E(\mathbf{w})$, what does it imply as an optimal solution?*

   You can see that the renewable technology 2, having a higher rate of return, also has a lower risk that is less than our $\sigma_{max} = 0.1$. An apparently logical conclusion is that if we only invest in technology 2, we meet our criteria of maximum return for $\sigma_T = 0.1$.

   For all solutions here, we use the structure of the problem that you developed in the previous exercise. The objective function, $E(\mathbf{w})$, becomes,

$$
\begin{aligned}
E(\mathbf{w}) &= w_1 \cdot r_1 + w_2 \cdot r_2 \\
&= 0.1w_1 + 0.2w_2 \\
&= 0.1w_1 + 0.2(1 - w_1) \\
&= 0.1w_1 + 0.2(1 - w_1) \\
&= 0.2 - 0.1w_1
\end{aligned}
$$

   In this problem, we are looking to find the value of $\mathbf{w} = [w_1, w_2]$, that maximizes $E(\mathbf{w})$.

   When we plot the $E(\mathbf{w})$, we see the $E([0, 1])$ is the obvious maximum. If we attempt to use our calculus methods of taking $\frac{dE}{dw_1}$ and setting it equal to 0, we are going to get a non-sense result, since E is linear with w_1. But as you can see from the graph, the maximum E is at $w_1 = 0$. For such an non-diversified portfolio, the measure of 'risk' is 0.05, or 5%, which meets our constraint of $\sigma_T \leq 0.1$. If we were instead minimizing risk, you can see that we would get a different answer, $\mathbf{w} = [0.2, 0.8]$

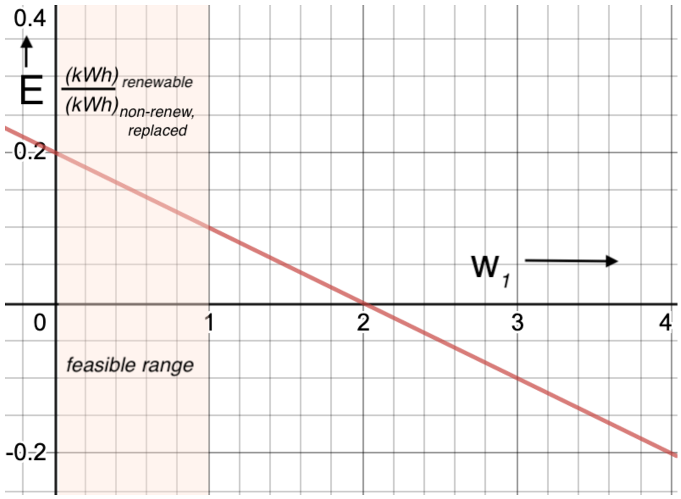


Figure 39.4: The objective function $E(\mathbf{w})$ v. w_1.

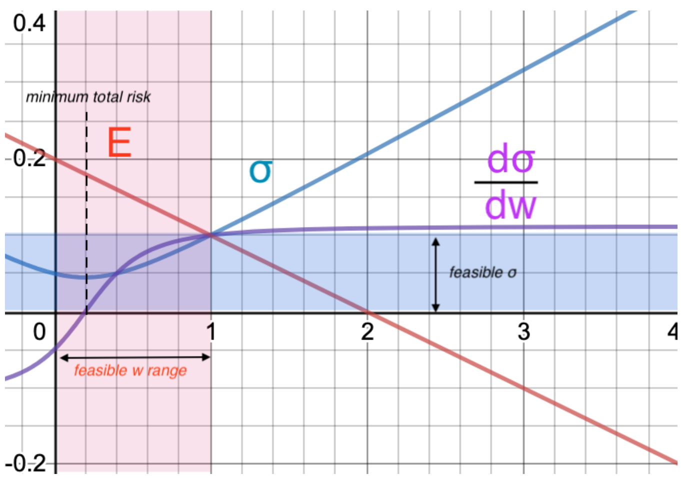


Figure 39.5: Risk minimized at $w_1 = 0.2$.

2. For a portfolio with two technologies: $r_1 = 0.1$, $\sigma_1 = 0.1$; $r_2 = 0.2$, $\sigma_2 = 0.3$. *Before starting this, what do you notice about the r's and σ's ? If you plot the $E(\mathbf{w})$, what does it imply as an optimal solution?*

   In this case, the higher return renewable (2) also has a higher risk–that is more usual. $E(\mathbf{w})$ has the same relationship as in the first problem.. For this case, there is a narrow range of w_1 that will satisfy the condition of $\sigma_T \leq 0.1$. We choose $\mathbf{w}$ based on the maximum E among the feasible solutions where $\sigma_T \leq 0.1$: $\mathbf{w} = [0.8, 0.2]$.

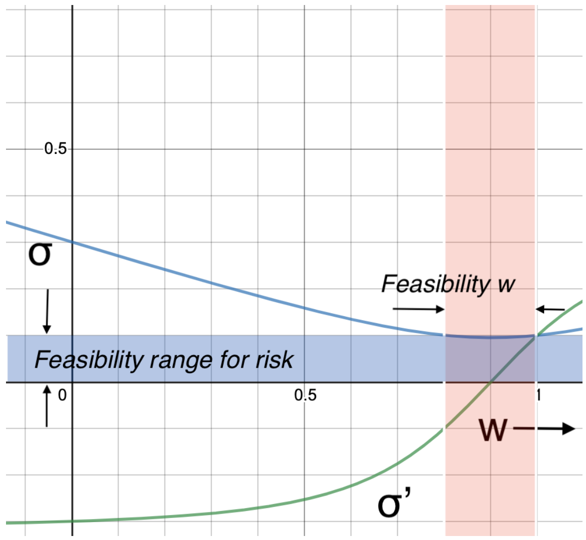


Figure 39.6: Risk acceptable for $0.8 \leq w_1 \leq 1$.

3. For an eight technology portfolio, where $\mathbf{r} = [0.1, 0.13, 0.15, 0.17, 0.2, 0.23, 0.25, 0.28]$ and $\boldsymbol{\sigma} = [0.2, 0.1, 0.4, 0.25, 0.1, 0.35, 0.32, 0.35]$. *Any ideas? Doing this by hand would require a rarefied mathematical prowess–I used a computer algorithm as a "black box" to do the work for me.*

This is a gnarly problem that requires computational methods. If you are interested in such optimization problems, you can find a really clear walk through on how to set it up for MATLAB's solver at APMonitor.com. This requires MATLAB's Optimization Toolbox. Here is a solution. This solution uses MATLAB code that identifies the objective function, the decision variable and constraints. If interested, you can use this code, optimizationStarter.m and the associated non-linear constraint code, nlcon.m.

Solution 39.5

Choosing the dimensions of cylinder package that minimizes waste is a *deterministic* optimization problem with a single, continuous objective function. This problem is constrained by the minimum volume; this constraint happens to be non-linear.

Solution 39.6

Of the priorities that are named in the Creed, this is a list from highest to lowest, although only the first is clearly the top priority:

1. The public welfare
2. The honor and standing of the profession over personal advantage
3. Personal advantage
4. Service over profit

5. Profit

Solution 39.7

You may recognize this as a multiobjective optimization involving a combination of continuous and discrete variables. It is quite complicated as something to "optimize." Is is less complicated as something to measure. If you are totally into the idea of multiobjective optimization, you might want to look into A tutorial on multiobjective optimization: fundamentals and evolutionary methods, by Emmerich and Deutz and declare yourself a computing major.

Chapter 40

Night 5: Optimization and Gradient Ascent

Learning Objectives

Concepts

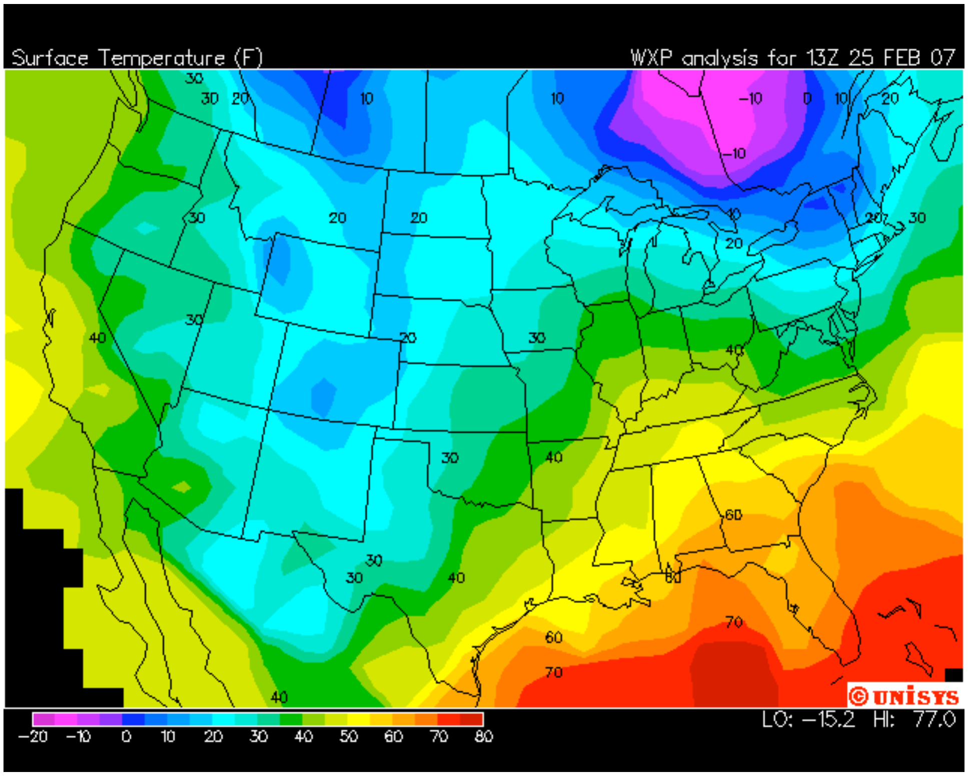
- Describe the meaning of the $\nabla f(x, y)$ on a contour map
- From a map of a vector field of $\nabla f(x, y)$, sketch contour lines of equal $f(x, y)$ value
- From a contour map of $f(x, y)$ sketch the path of steepest ascent.
- Determine a discrete approximation to a path of steepest ascent for a given function $f(x, y)$.

MATLAB Skills

- Use the concepts of optimization to program the NEATO to ascend a gradient.

Conceptual Exercise: The Leisure Seeker

The map below gives the temperature across the United States on a certain winter day. Regions of the same color have the same temperature: violet represents the coldest areas, and temperatures rise as the colors traverse the spectrum from indigo to blue to green to yellow to orange to red.



Exercise 40.1

Locate Chicago on the map and mark it with a dot. The weather in Chicago is freezing in winter, so a resident of the city decides to embark on a journey in search of the sun. From Chicago, she wants to travel in the direction in which the temperature rises most quickly. As her journey proceeds, she decides to keep traveling in the direction in which the weather warms up most quickly: wherever she is at any moment, she moves in the direction of fastest temperature rise.

1. Make a rough sketch of the route she takes.
2. Where does she end up, assuming that she doesn't leave the United States?
3. What would happen if her friend started in Billings, Montana? Where would he end up?

40.1 Readings, Videos, and Conceptual Questions - Partials and Gradients

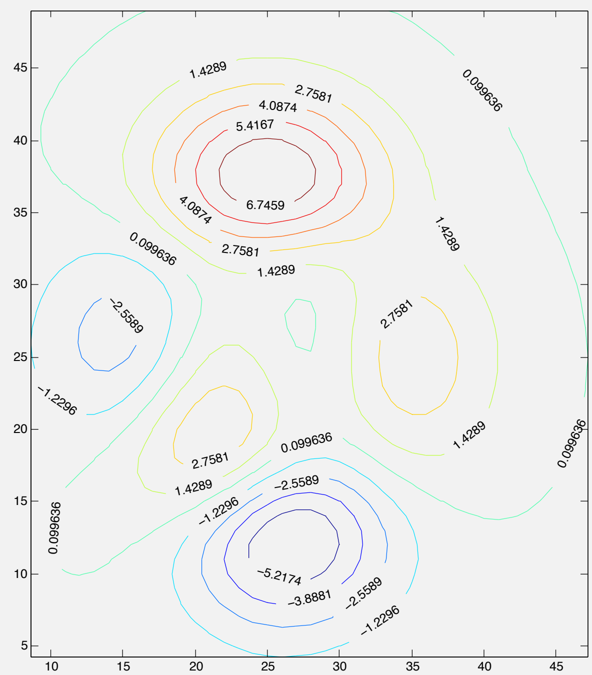
At this link you will find a set of readings and videos about partial derivatives, the gradient, and the Hessian. We have already met many of these concepts in Night 4 but some of it is new. Read the text and/or watch these videos, and then write short qualitative answers to the following questions:

Exercise 40.2

1. What is meant by f_x? By $\frac{\partial^2 f}{\partial x^2}$? By $D_u f$? By ∇f?
2. In Stewart, the idea of gradient is discussed primarily in two dimensional and three dimensional "physical" spaces, using $\hat{\mathbf{\imath}}$, $\hat{\mathbf{\jmath}}$, and so forth. But more generally, you can have a gradient of a function of any number of variables. Give a real-world example of a gradient for a situation that involves more than 3 variables.
3. "The gradient always points in the direction of fastest increase." Can you think of physical examples where there is *more than one* direction of fastest increase? What's going on here?
4. "The gradient is always normal to level curves/surfaces." Explain.
5. "The directional derivative in the direction of $\mathbf{u}$ is given by $\nabla f \cdot \hat{\mathbf{u}}$." Why does this make sense?
6. If the gradient is zero, does that imply that you are at a max or a min? Why or why not?
7. What does it mean for $\frac{\partial^2 f}{\partial x^2} > 0$? What about $\frac{\partial^2 f}{\partial x \partial y} > 0$?

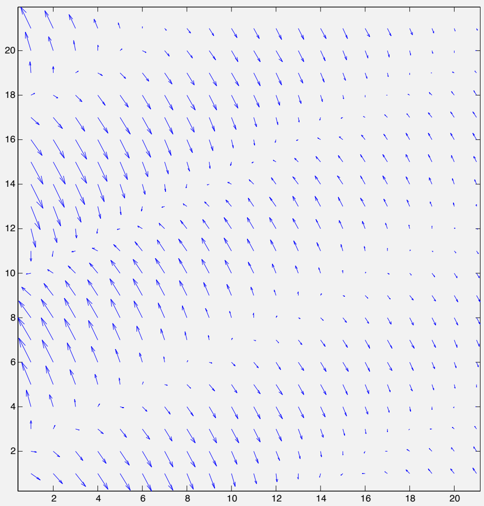
Exercise 40.3

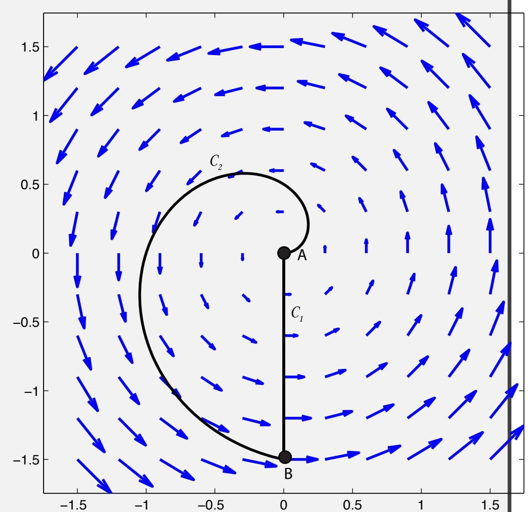
The diagram below shows a contour plot for a function. Sketch in the gradient field.



Exercise 40.4

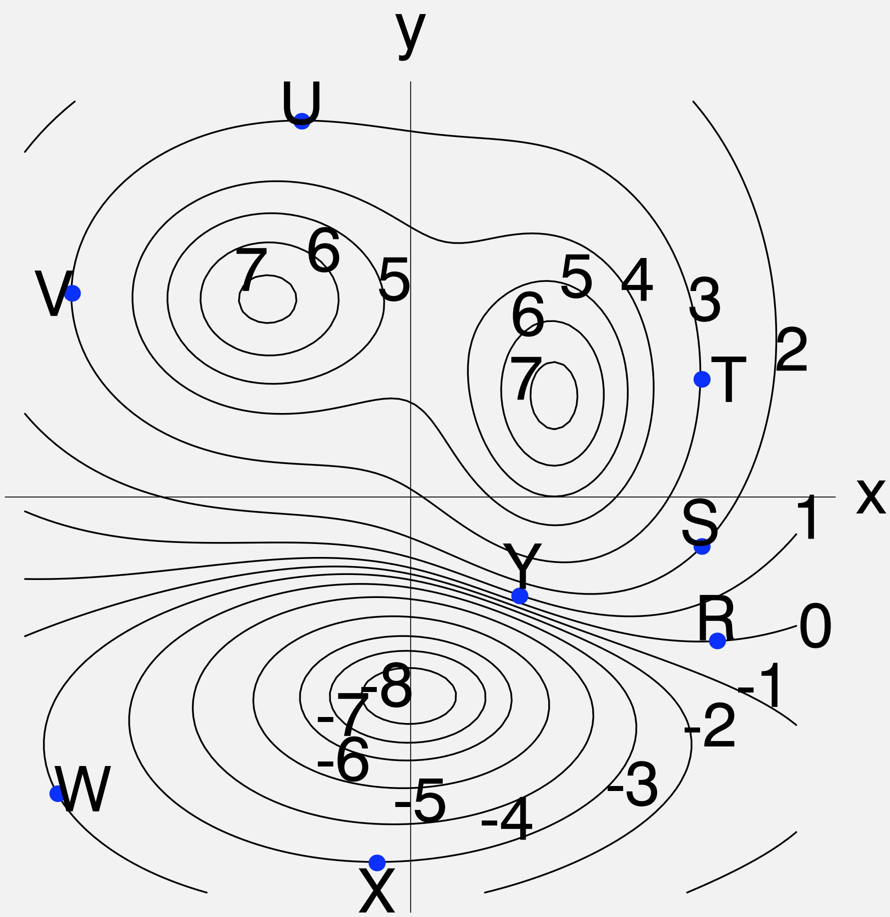
The diagrams below show two vector fields. One is the gradient of a function of two variables; the other is not. Which one could be a gradient? Why? Sketch in the level curves for the one that works.





Exercise 40.5

The diagram below shows some level curves of a function $g(x, y)$. The numbers indicate the g-values of these level curves, and the letters indicate points on the level curves. Note that point Y is on the level curve $g = 1$ and point W is on the level curve $g = -2$.



1. What are the signs of the partial derivatives $\frac{\partial g}{\partial x}$ and $\frac{\partial g}{\partial y}$ at each of the points marked (R,S,T,U,V,W,X and Y)?
2. At which of the points marked does the gradient vector ∇g have the greatest magnitude? Explain.
3. Let
$$\mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$
The directional derivative $D_{\hat{\mathbf{u}}} g$ is zero at exactly one of the points marked. Which point is it?

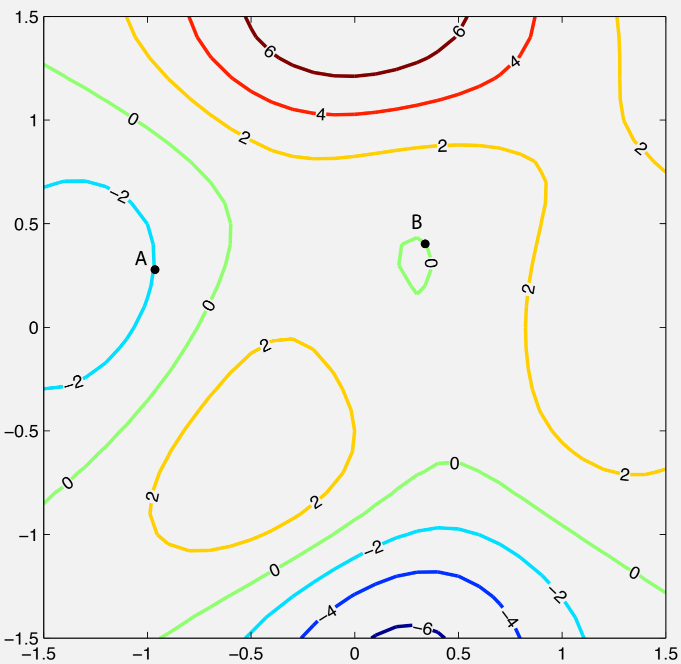
40.2 Readings, Videos, and Conceptual Questions - Optimization with Gradient Ascent

At this link you will *also* find a set of readings and videos about gradient ascent (or descent - same thing, but for a sign!). Read the text from Giordano and watch the videos. If you are interested, you can also read the stuff about conjugate gradient ascent, which is cool and leverages eigenstuff - but is optional!

Exercise 40.6

The figure below shows a contour plot with two points marked (A and B). For both points,

1. Draw the path that gradient ascent would use if the step size was small (following the approach in the first video).
2. Draw the path that gradient ascent would follow if the algorithm is implemented as shown in the second video.



40.3 Gradient Ascent

As you've just been learning, Gradient Ascent (or descent) is a technique to determine the maximum or minimum of a function of many variables by taking steps in the direction of the gradient (or negative gradient). If the height of a surface is described by $z = f(\mathbf{r})$, where $\mathbf{r}$ is the position vector in the plane, and we begin at $\mathbf{r}_0$, the points determined by Gradient Ascent are given by

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \lambda_i \nabla f(\mathbf{r}_i),\ i = 0, 1, 2, \ldots$$

where λ_i is the relative size of the step that we take in the direction of the gradient. There are various schemes for choosing these, and one of the simplest is to determine the next step with a simple proportionality

$$\lambda_{i+1} = \delta \lambda_i$$

where both δ and λ_0 are thoughtfully chosen for the problem at hand. We are going to develop a method and implementation to drive your NEATO on the floor of the classroom in a way that physically realizes the method of Steepest Ascent. First, we are going to introduce the mountain you will climb, and you will think through the steps involved.

Exercise 40.7

The mountain you will "climb" is defined as follows

$$f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4$$

where x, y, and f are measured in feet. You are required to start at $(1, -1)$, and work your way to the top by method of steepest ascent.

1. Visualize the contours of this function on the domain $(-3, 1) \times (-3, 1)$.
2. Draw the path of steepest ascent if we were moving continuously from a starting point at $(1, -1)$.
3. Find the gradient of this function.
4. Assuming $\mathbf{r}_0 = (1, -1)$, what is the initial gradient at $\mathbf{r}_0$? What would be a reasonable choice for λ_0 so that $\mathbf{r}_1$ is not too far from the continuous path? Plot $\mathbf{r}_1$ on your contour plot.
5. Assuming you place your NEATO at $(1, -1)$ pointing along the y-axis, how much do you have to rotate it in order to align it with the gradient at $\mathbf{r}_0$? What would be a reasonable angular speed?
6. Assuming that you are going to drive your NEATO at $0.1 m/s$, how long would you drive in order to reach $\mathbf{r}_1$? (Careful with unit changes!)
7. What is the gradient at $\mathbf{r}_1$? What value of δ should you use so that λ_1 and $\mathbf{r}_2$ are reasonable? Plot $\mathbf{r}_2$ on your contour plot.
8. Assuming your NEATO is now at $\mathbf{r}_1$, how much do you have to rotate it in order to align it with the new gradient? What would be a reasonable angular speed?

9. Assuming that you are going to drive your NEATO at $0.1m/s$, how long would you drive in order to reach $\mathbf{r}_2$?

40.4 Gradient Ascent in MATLAB

In this exercise we would like you to implement gradient ascent in MATLAB. Recall that if $f(\mathbf{r})$ is a scalar function of a position vector $\mathbf{r}$, the points determined by gradient ascent are given by

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \lambda_i \nabla f(\mathbf{r}_i),\ i = 1, \ldots$$

where λ_i is the relative size of the step that we take in the direction of the gradient. There are various schemes for choosing these—one of the simplest is a proportionality

$$\lambda_{i+1} = \delta \lambda_i$$

where both δ and λ_0 are thoughtfully chosen for the problem at hand.

Exercise 40.8

1. Develop pseudo-code for the algorithm. Keep it general: f is a general scalar function of a vector $\mathbf{r}$. You are going to want to use a loop - a "while" loop would be a good choice - what would be a reasonable stopping criterion?
2. When you are happy with your pseudo-code, develop a script or function that:
   a) Automatically determines the discrete points $\mathbf{r}_1$, $\mathbf{r}_2$, ... given an initial point $\mathbf{r}_0$.
   b) Can be tuned by varying δ and λ_0.

You should implement your method on the function that we met earlier

$$f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4$$

and in order to validate your approach you will want to visualize the contours and the discrete points.

40.5 Flatland Challenge

You will develop a method and implementation to drive your NEATO on the (virtual) floor of the classroom in a way that physically realizes the method of Steepest Ascent. The mountain you will "climb" is defined as follows: (this should look somewhat familiar—when have you seen this equation before and what does it describe?)

$$z = f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4$$

where x, y, and z are measured in feet. You are required to start your NEATO at $(1, -1)$ pointing in the $+y$ direction, and work your way to the top. In order to position your NEATO with the correct orientation, you can use the `placeNeato.m` function, which we have provided.

Exercise 40.9

1. Decompose this problem. What are the steps involved? You should have a decomposition that clearly explains the process of driving the NEATO along a discrete approximation to the path of steepest ascent.

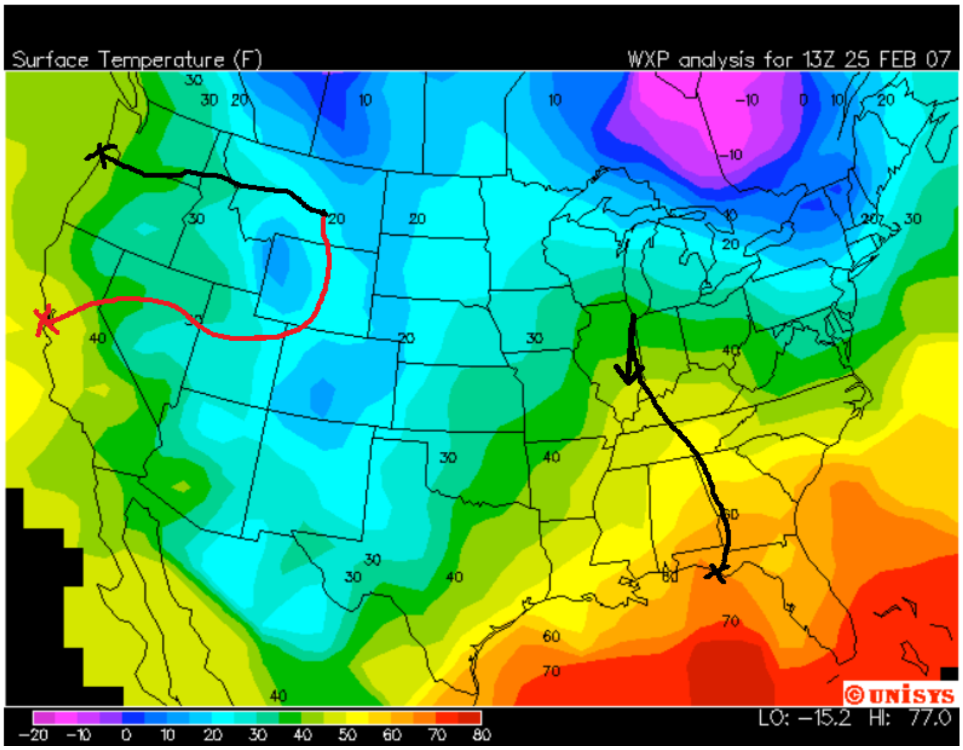
2. Now develop the code for the NEATO.

3. Now drive your NEATO! You will startup the Neato simulator using the `flatland` environment. If you are adapting the command from the Meeto Your Neato page, you would run the following command.

```
docker stop neato; docker rm --force neato; docker run --rm --name=neato
    --sysctl net.ipv4.ip_local_port_range="32401 32767" -p 11311:11311 -p
    8080:8080 -p 32401-32767:32401-32767 -e NEATO_WORLD=flatland_no_spawn
    -it qeacourse/robodocker:spring2020
```

Solution 40.1

1. See Fig. 1.



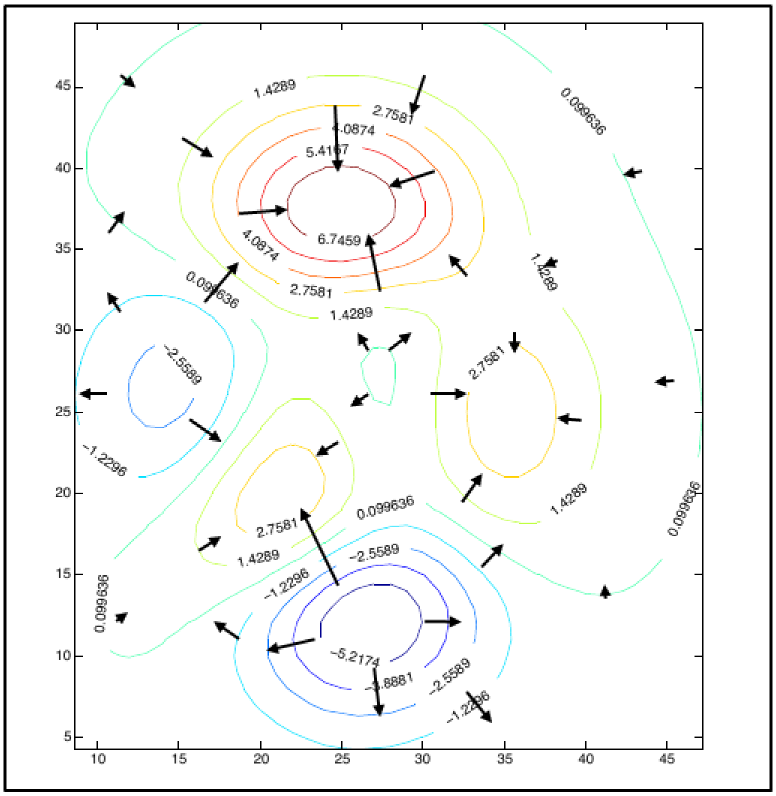
2. Near Panama City, FL
3. Hard to know exactly without more isotherms (contours of constant temperature), but maybe the SF Bay Area or the Portland area.

Solution 40.2

1. The partial derivative with respect to x, meaning the slope in the x direction (where every other independent variable is held constant); the second partial derivative with respect to x, meaning the curvature in the x direction (where every other independent variable is held constant); the directional derivative with respect to vector u, meaning the slope in the u direction; the gradient, meaning the maximum slope in its corresponding direction.
2. Any kind of optimization problem could have a gradient of the objective function in many dimensions (decision variables).
3. At the bottom of a cone, the increase is equally fast in all directions.

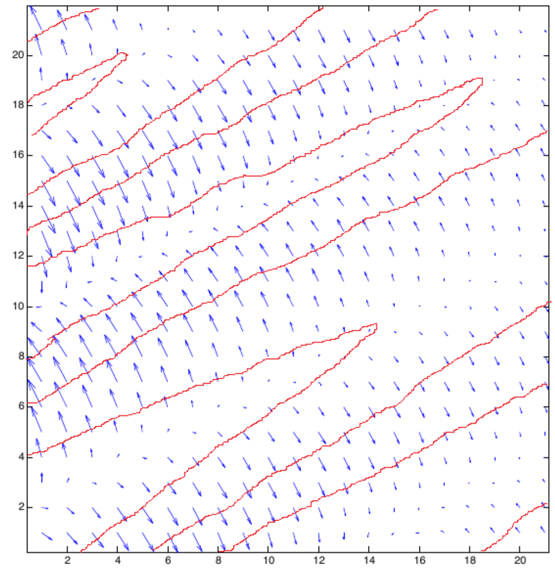
4. The gradient points in the direction of most rapid increase—this direction must be perpendicular to the level curves of the function since the function is unchanged on a level curve.
5. The gradient could be decomposed into a component in $\hat{\mathbf{u}}$ direction and a component normal to that. So the rate of change of the function in the $\hat{\mathbf{u}}$ direction is $\nabla f \cdot \hat{\mathbf{u}}$.
6. No, because you could be at a saddle point, for example.
7. Upward curvature in the x-direction; slope in x-direction increases with increasing y (and vice versa).

Solution 40.3



Solution 40.4

The one on the left could be a gradient; the right one could not because a path of gradient ascent would go in a circle. That is a problem because any point would have two gradient values associated with it—there is no way that it can come from the derivatives of a function. See Fig. 40.1 for a sketchy sketch of level curves for the left one.



Solution 40.5

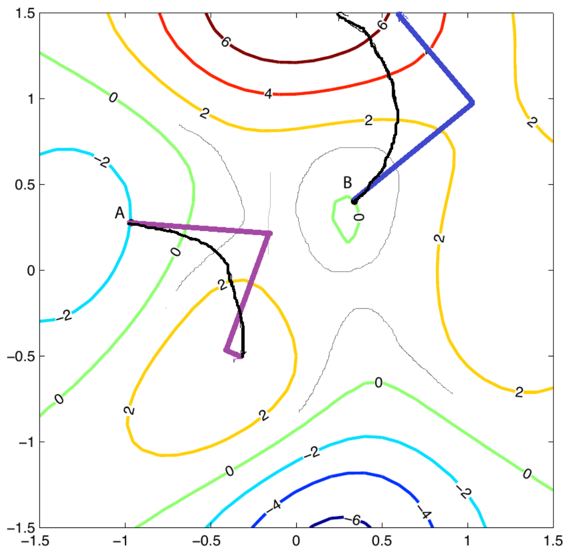
1.

Point	Sign of g_x	Sign of g_y
R	- or 0	+
S	-	+
T	-	0
U	0	-
V	+	0
W	-	-
X	0	-
Y	+	+

2. At Y because the contours are closest together, indicating the greatest change in the function value per unit distance normal to the level curves.

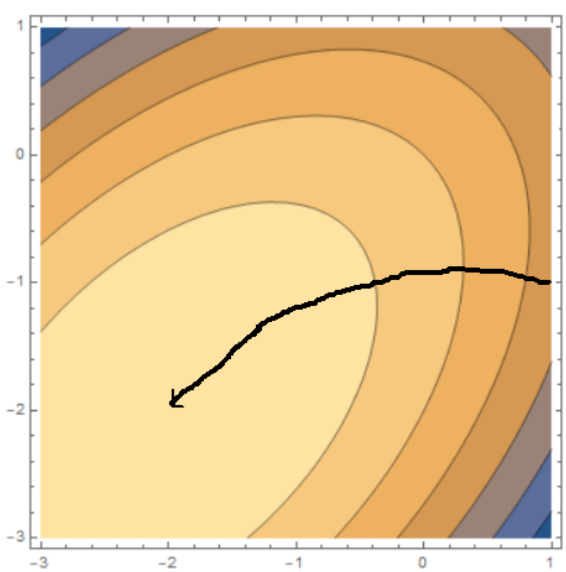
3. Point S, where the contour line is parallel to **u**.

Solution 40.6



The black path is the small step size ascent; the purple and blue paths are the walk-straight-until-you-start-to-go-downhill method. But it's a little hard to predict the path from point A given the large spacing of contours...other reasonable paths will be accepted.

Solution 40.7

1. 

2. See the path on the contour plot.

3. $(-2 - 2x + y, -2 + x - 2y)$

4. Initial gradient is (-5,1), with a magnitude of roughly 5.1. To move about 0.5 feet, the initial multiplier λ_0 should be about 0.1 feet. This gives $\mathbf{r}_1 = \mathbf{r}_0 + \lambda_0 \nabla f(\mathbf{r}_0) = (0.5, -0.9)$.
5. We have to rotate from the y-axis to the direction of the gradient (-5,1) - this is roughly 1.37 rad (78.7 degrees) CCW. 0.5 rad/s would not exceed 0.3 m/s if rotated in place.
6. The time taken to reach $\mathbf{r}_1$ is determined by $t = \frac{|\mathbf{r}_1 - \mathbf{r}_0|}{V} \frac{1\text{m}}{3.28\text{ft}}$, where we remember to keep our units the same. The distance traveled is roughly $\sqrt{0.26} = 0.5$ feet which is roughly 0.15 m. At a speed of 0.1 m/s this would take 1.5 seconds.
7. The approach to these is similar to the first step.
8. The approach to these is similar to the first step.
9. The approach to these is similar to the first step.

Chapter 41

Day 7: LIDAR in Polar and Cartesian Coordinates

41.1 Virtual Class Schedule

- 1000-1005: Tech time
- 1005-1015: Introduction and LIDAR
- 1015-1100: Polar and Cartesian Coordinates
- 1100-1110: Coffee
- 1110-1145: Least Squares for Line Fitting
- 1145-1230: Principal Component Analysis for Line Fitting

Learning Objectives

Concepts

- Convert vectors from 2D Cartesian to Polar coordinates (and vice versa).
- Explain how PCA produces a line of best fit to a 2D dataset.

MATLAB skills

- Eliminate 0-entries from LIDAR data.
- Plot LIDAR data on a polar coordinate system.
- Generate best-fit lines to LIDAR data using linear regression and principal component analysis.

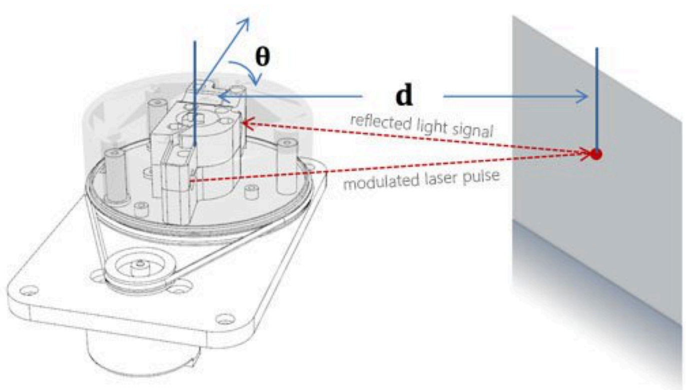
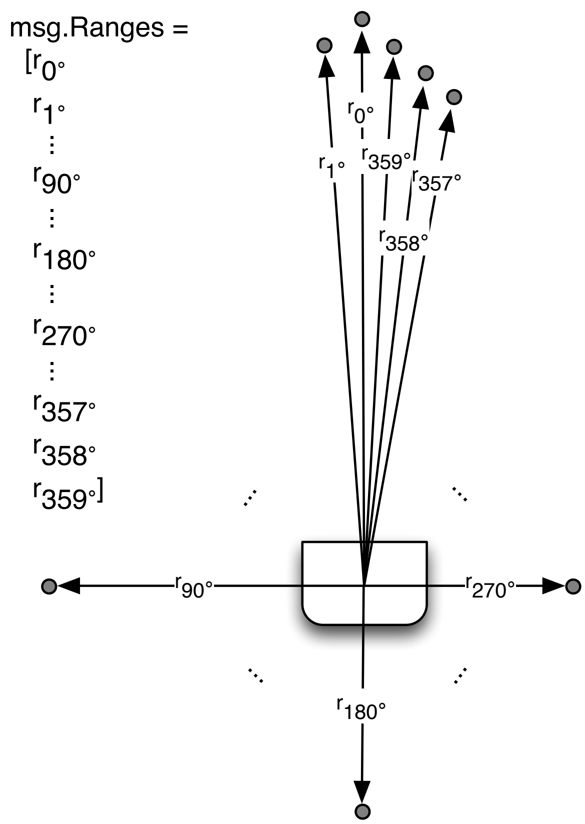


Figure 41.1: A schematic of the Neato's laser scanner. Due to the relative position of the emitter and detector, the position of the reflected light changes predictably as a function of d. The laser scanner spins with a rate of 5Hz. The angle of the scanner is denoted as θ.

Laser scanners are becoming an increasingly important sensor in modern robotics. A key accelerator of this trend is the development of self-driving cars, which use laser scanners as their principal source of information about the world around them (e.g., check out this video of a 3D map of a highway in the San Francisco Bay Area).

Your Neato comes with a much more basic laser scanner than the long-range 3D scanners you see in self-driving cars. The Neato's laser scanner provides an estimate of the distance to nearby features in the environment (e.g., obstacles) at a rate of 5 Hz, with an angular resolution of 1 degree, and a maximum usable range of about 3 meters. Under the hood, the sensor works by emitting a laser and then detecting the reflected laser light using a camera. From the known geometry of the camera / laser pair and the position of the reflected laser beam in the camera image, the depth of the obstacle can be recovered (see Figure 41.1). Further, the entire laser / camera assembly spins, providing an estimate of the distance to features in the environment in a full 360-degree sweep around your robot.



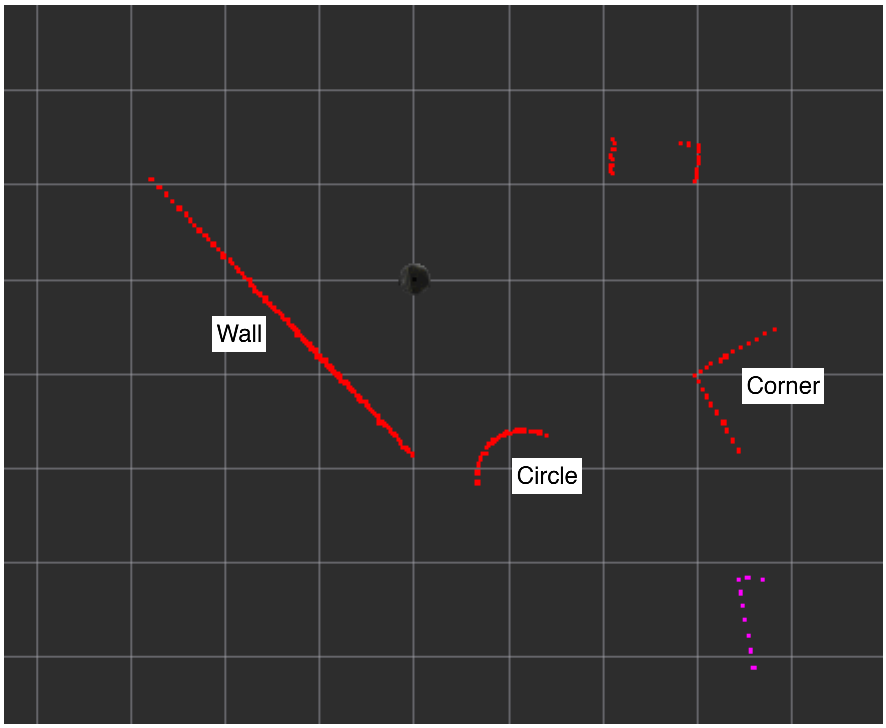


Figure 41.3: A laser scan with high-level features identified.

Figure 41.2: A schematic of the laser scan data. In the figure, r_{θ° indicates the distance in meters to a detected object at a bearing of θ degrees. $msg.Ranges$ shows how each of these distances maps to a particular index in the array. The array holds the distances, and the angles are implicitly encoded by the position in which they appear in the array. If no detection is made at a particular bearing, the value 0.0 is used instead. Note: the angles of the measurements around $\theta = 0^\circ$ have been exaggerated for visualization purposes (i.e. the angles shown are larger than a single degree).

Sometimes it is useful to take the raw points detected by the laser scanner and interpret them as geometrical objects (e.g., lines, curves, polygons). For instance, given the laser scan in Figure 41.3 we may wish to automatically interpret the laser scan data (i.e. pick out lines, circles, etc.).
Conveniently, in modules 1 and 2 we've built a lot of the machinery needed to do just this (aren't we sneaky?). Here, you'll have a chance to bring it all together.

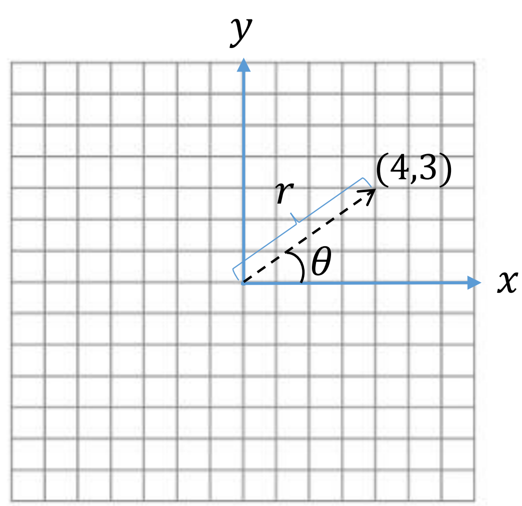


Figure 41.4: Illustration of vector in polar and Cartesian coordinates.

Polar and Cartesian Coordinates [1 hr]

The output data from many sensors, including LIDAR data, is typically presented in polar coordinates. In many situations, however, it is more convenient to represent the data in Cartesian coordinates. In this set of exercises, you will get some practice converting polar to Cartesian coordinates and vice-versa.

In Cartesian coordinates, a point in 2-D space can be described as a sum of unit vectors in the $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ directions, which are unit vectors in the directions of the x and y axes respectively.

Consider the point $(4, 3)$ illustrated in Figure 41.4. You can write the vector representing this point as a sum of $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ as follows

$$\begin{bmatrix} 4 \\ 3 \end{bmatrix} = 4\hat{\mathbf{i}} + 3\hat{\mathbf{j}} \tag{41.1}$$

We can also write this point in terms of polar co-ordinates as (r, θ), where

$$r = \sqrt{4^2 + 3^2} = 5 \tag{41.2}$$

$$\theta = \tan^{-1}\left(\frac{3}{4}\right) \tag{41.3}$$

In general a vector (x, y) in Cartesian coordinates can be expressed in polar coordinates as (r, θ) where

$$r = \sqrt{x^2 + y^2} \tag{41.4}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \tag{41.5}$$

To go from polar to cartesian coordinates, we can use the following formulas

$$x = r\cos\theta$$
$$y = r\sin\theta$$

Besides expressing a vector as a linear combination of the $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ vectors, we can also define two new unit vectors $\hat{\mathbf{r}}$ and $\hat{\boldsymbol{\theta}}$ defined as follows

$$\hat{\mathbf{r}} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{41.6}$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \tag{41.7}$$

Note that these vectors are not absolute with respect to the origin of the cartesian coordinates. While it does seem strange to represent unit vectors in this manner, this kind of representation will prove useful in the near future when we are dealing with dynamics of rotating bodies (i.e. you will not directly use this until next semester, but we thought we should give you a preview).

Exercise 41.1

Convert the following from Cartesian to polar coordinates.

1. $(x, y) = (-8, 6)$
2. $(x, y) = (5, 12)$

Exercise 41.2

Convert the following from polar to Cartesian coordinates.

1. $(r, \theta) = \left(2, \frac{\pi}{6}\right)$
2. $(r, \theta) = \left(2, -\frac{\pi}{3}\right)$

Exercise 41.3

What is $\hat{\mathbf{r}}^T \hat{\boldsymbol{\theta}}$?

Exercise 41.4

In words, describe the $\hat{\mathbf{r}}$ and $\hat{\boldsymbol{\theta}}$ vectors.

Exercise 41.5

Now let's look at some LIDAR data that was collected from the Chamber of Emptiness™, which is a desolate room with one wall. Load the data in *scan1.mat*, which should contain the variables *r* for range (distance from the scanner in meters) and *theta* for the angle from the front of the robot (see Figure 41.3).

1. Plot these data using: `plot(theta,r,'ks')`
2. Plot the data in polar coordinates using:
   `polarplot(deg2rad(theta),r,'ks','MarkerSize',6,'MarkerFaceColor','m')`
3. You should see two subsets of data (a line and a parabola in the plot from part a). Where are these two subsets represented in the polarplot?
4. When the LIDAR does not detect any object at a given angle, it outputs a 0 for the value of *r* at that angle (this is a common problem in the Chamber of Emptiness). Create new variables *theta_clean* and *r_clean* which represent the angles and values of *r* when *r* is not zero.

   Note: It may be useful to use the Matlab syntax "∼=", which means "not equal." The `find()` function can also be helpful here.
5. Now convert *r_clean* and *theta_clean* to Cartesian coordinates and plot them using the *plot()* function.

Note: A working code snippet to accomplish these tasks is in the solutions.

41.2 Coffee [10 minutes]

41.3 Fitting Models to Laser Scan Data

Least Squares for Line Fitting [35 minutes]

The first structure we would like to detect in the laser scan data is a line. We've seen the notion of a line of best fit a couple of times in Module 1, and most recently in Night 1 of Module 2. Let's take a moment to review the process of fitting a best fit line to a set of data using techniques from linear algebra (often known as linear regression).

Assuming we have converted our LIDAR measurements to Cartesian coordinates, we can assemble all of the x-coordinates in a column vector x and all of the y-coordinates in a column vector y. Let's now find the best-fit straight-line through these points, i.e. let's find the parameters m and b so that the straight-line defined by

$$y = mx + b \tag{41.8}$$

fits the points as well as possible.

The approach we take is motivated by our work in linear algebra. If we pack all of the x-coordinates into a vector $\mathbf{x}$ and all of the y-coordinates into a vector $\mathbf{y}$ then we would like to satisfy the vector equation

$$\mathbf{y} = m\mathbf{x} + b \tag{41.9}$$

as well as we can. Notice that there are many equations here (one for each point) and only two unknown parameters. An exact solution is impossible (unless the points happen to lie on a line) and so we use orthogonal projection to find the best solution. Let's define a matrix $\mathbf{A}$ and parameter vector $\mathbf{p}$ so that the vector equation for a straight-line becomes

$$\mathbf{A}\mathbf{p} = \mathbf{y} \tag{41.10}$$

where $\mathbf{A}$ and $\mathbf{p}$ are given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{x} & \mathbf{1} \end{bmatrix}, \mathbf{p} = \begin{bmatrix} m \\ b \end{bmatrix}$$

Notice that there is a coefficient of "1" in front of the "b" term so we had to create a column vector and fill it with 1's.

Recall that to find the best solution we multiply by $\mathbf{A}^T$,

$$\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y} \tag{41.11}$$

and solve this linear system for $\mathbf{p}$. In MATLAB we can simply use the backslash operator as follows (assuming we already defined x and y as earlier), since it will return the best fit solution when $\mathbf{A}$ has more rows than columns:

```
A = [x ones(length(x),1)]
p = A\y
```

Exercise 41.6

For each of the laser scans below (the data has been cleaned to remove zero values), please consider, qualitatively, what the "best fit line" should look like. To do this, follow these steps:

1. One person in your group should share their screen so the plots are visible to the whole group. Try to have only the "Clean Data" plot visible, not the second plot which includes the linear regression fit.

2. Each member of the group should draw in their guess of the best fit line using the Zoom annotate tools (use the line tool, not the free-hand drawing tool).

3. Each member of the group should explain why they chose the line they did, and how their proposed line minimizes the vertical error between the experimental data and the proposed fit.

4. Once each group member has discussed their fit, scroll down the document so that the linear regression solution plot is underneath your annotated lines. Compare your qualitative best fit line with the linear regression solution.

   a) How close was your guess to the linear regression solution?

   b) Was your fit or the algorithmic fit "more accurate"? What metric of accuracy are you using when you make this decision?

   c) For cases where your qualitative best fit and the linear regression solution didn't match, why do you think that is? You might want to consider the presence of outliers, how vertically aligned data impacts linear regression, etc.

   d) Linear regression seeks to minimize the vertical error between the data and the best fit solution- is that optimal for the data set? What is another approach that could lead to a "better" fit.

Note: These plots were created using the code here. An excerpt for scan1 is below. All of the data is here.
Note 2: How you visualize data is very important, and greatly impacts our qualitative idea of what is a "better" fit. This is made very apparent for scan2 below. I have created the scan2 plot using the "axis equal" command in Matlab, which ensures an equal range in x and y. Try plotting the data without that command or with the "axis square" command. Would you draw the same fit line?

```
close all, clear all, clc

%load data from Scan1
load scan1.mat
r1=r;
theta1=theta;

%eliminate zeros
index1=find(r1≠0); %note the "not equal" is a tilde followed by an equal
r1_clean=r1(index1);
theta1_clean=theta1(index1);
figure;
polarplot(deg2rad(theta1_clean),r1_clean,'ks','MarkerSize',6,'MarkerFaceColor','m')

[x1,y1]=pol2cart(deg2rad(theta1_clean),r1_clean);
figure;
plot(x1,y1,'ks')
title('Scan 1- Clean Data')
xlabel('[m]')
ylabel('[m]')

%We can use linear regression to find the best fit line
X1=[x1 ones(length(x1),1)];
beta1=(X1'*X1)\(X1'*y1);
```

```
y1_fit=x1*beta1(1)+beta1(2);

figure;
plot(x1,y1,'ks')
hold on
plot(x1,y1_fit,'r','linewidth',2)
legend('Clean Data','Linear Regression')
title('Scan 1- Clean Data and Fit Lines')
xlabel('[m]')
ylabel('[m]')
```

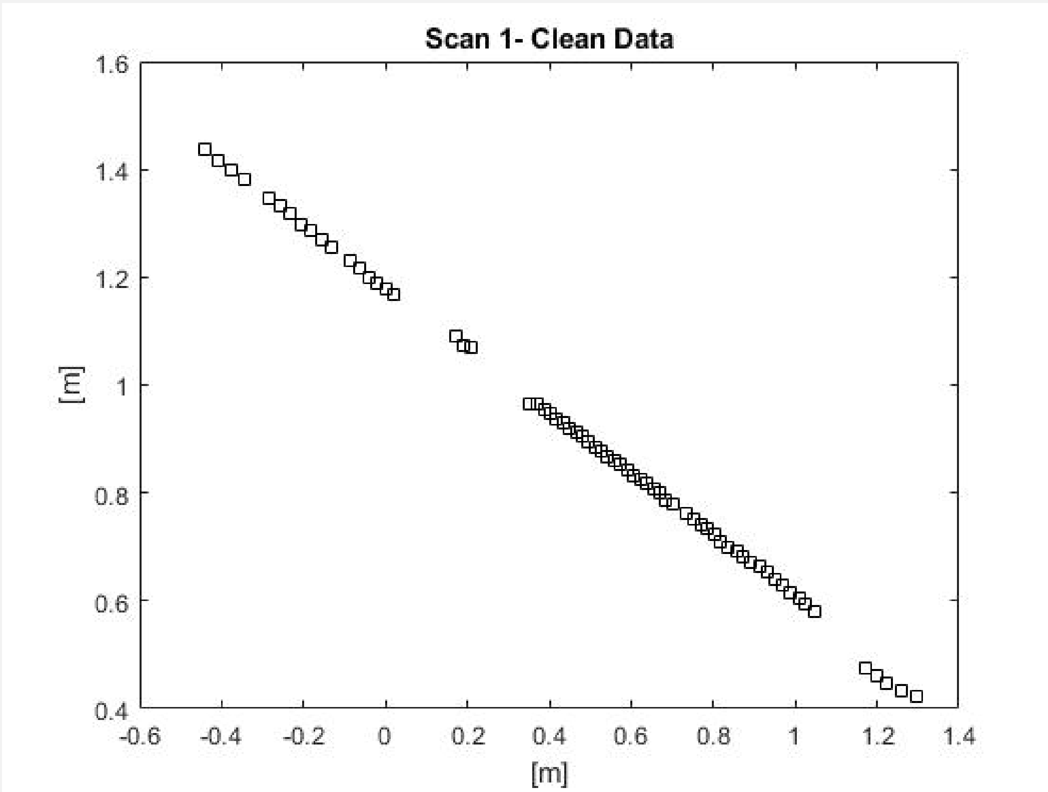


Figure 41.5: Clean data from Scan1. Please add your fit lines using the Zoom annotation tools.

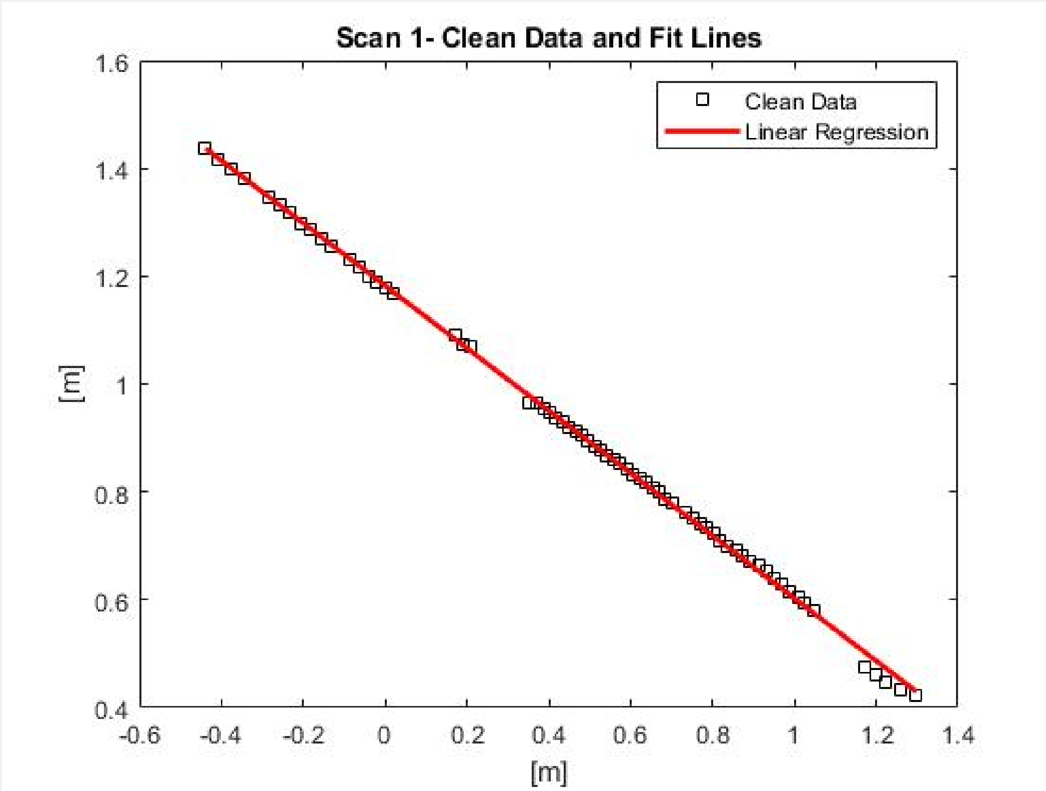


Figure 41.6: Best fit using linear regression for scan1.

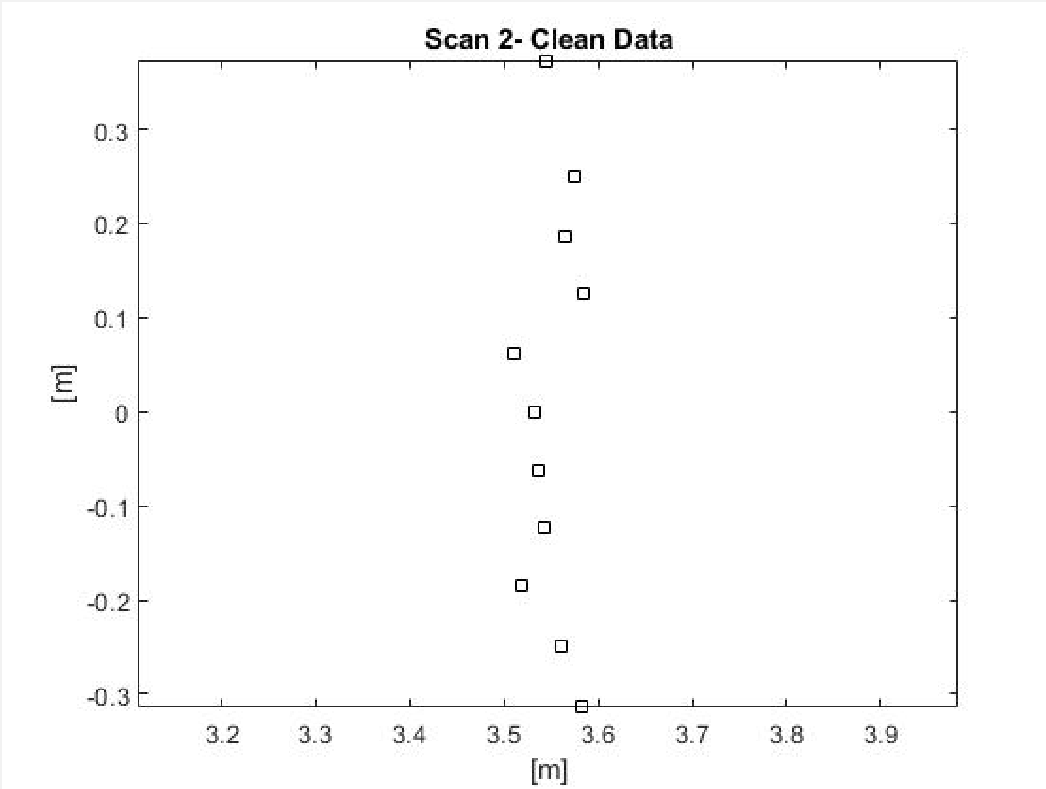


Figure 41.7: Clean data from Scan2. Please add your fit lines using the Zoom annotation tools.

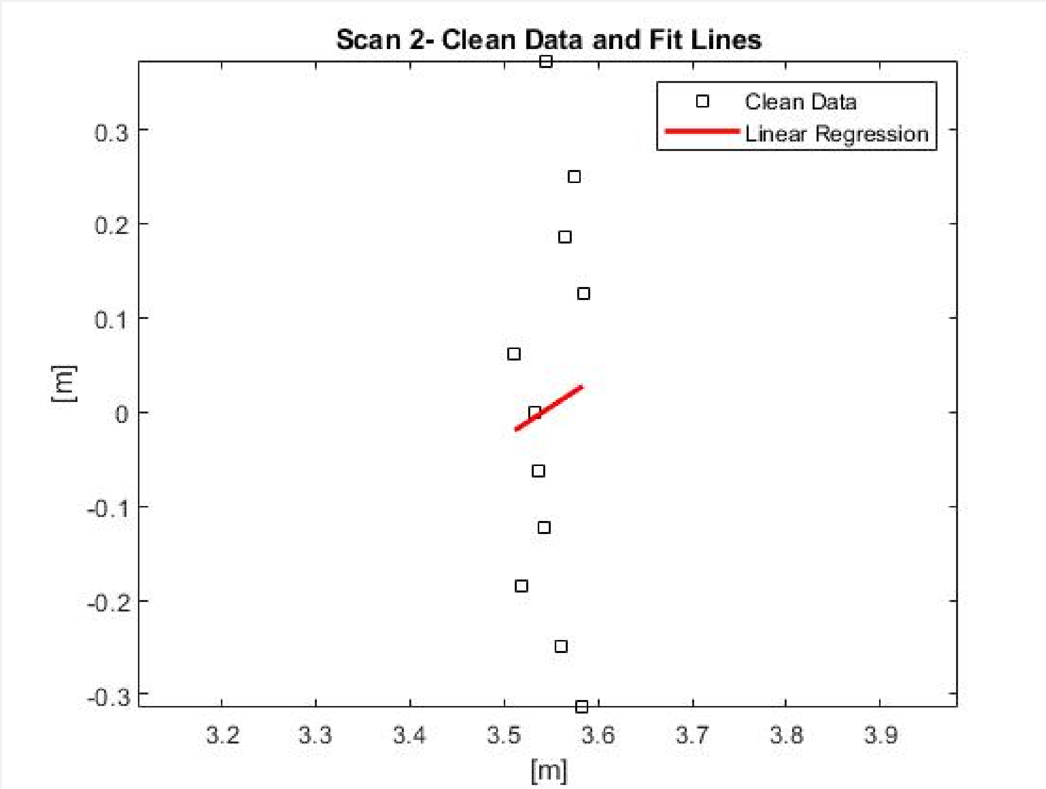


Figure 41.8: Best fit using linear regression for scan2.

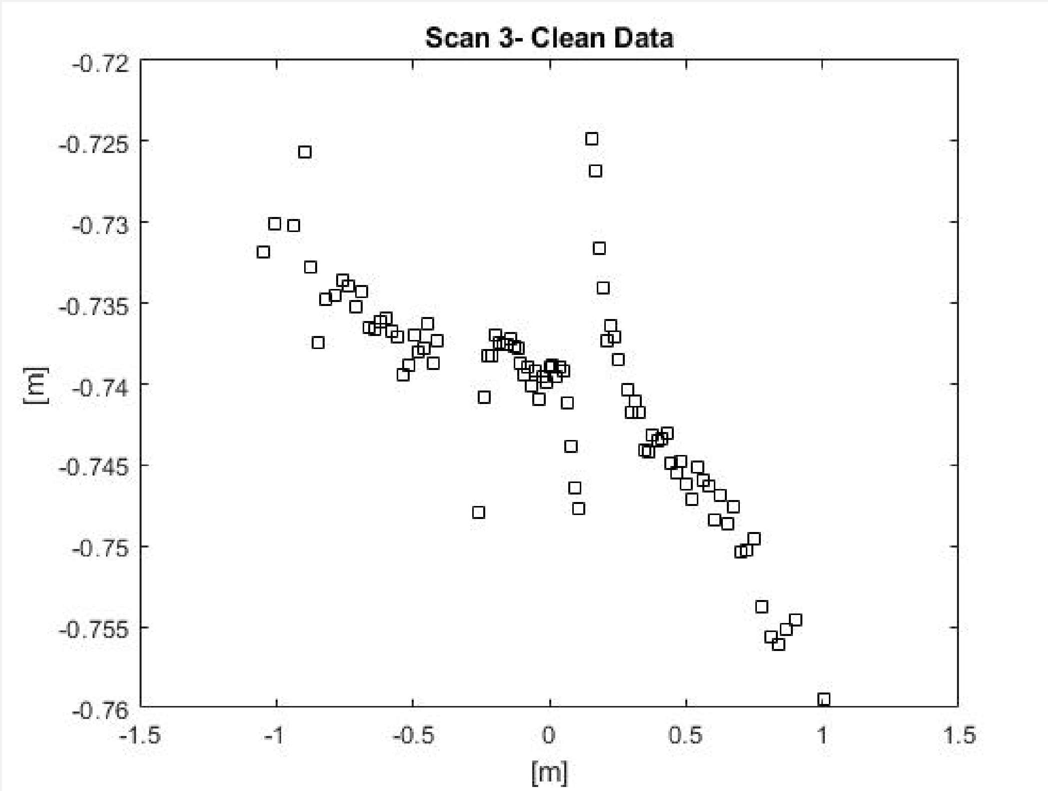


Figure 41.9: Clean data from Scan3. Please add your fit lines using the Zoom annotation tools.

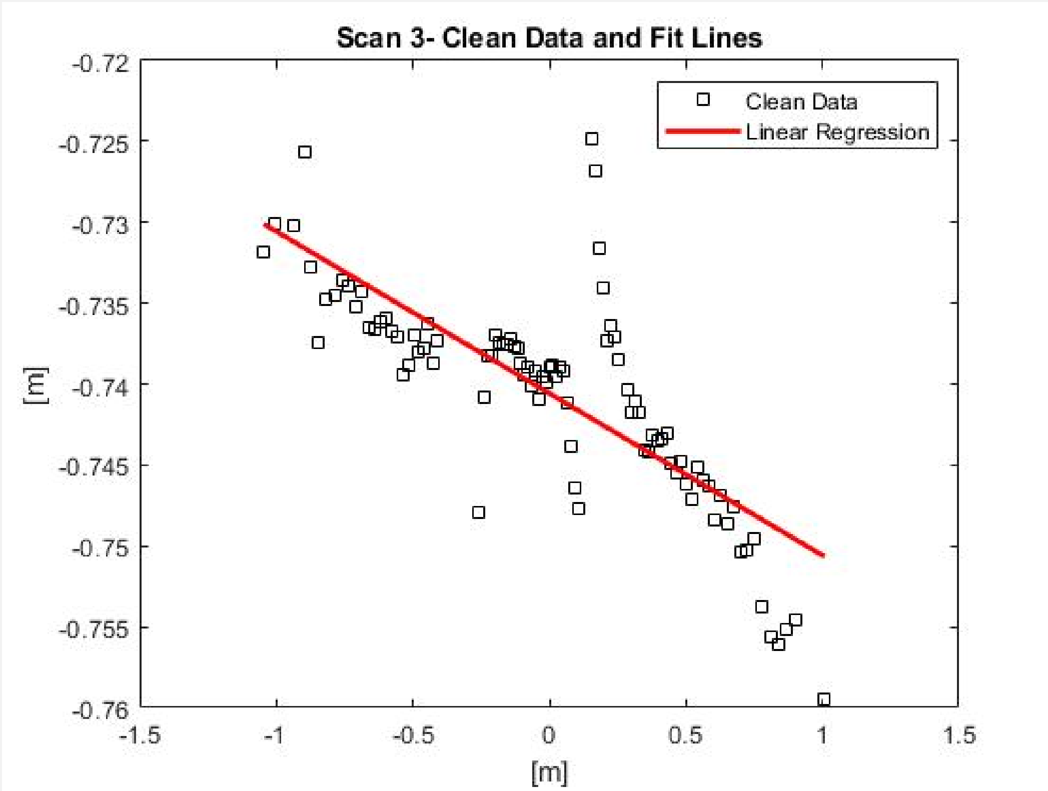


Figure 41.10: Best fit using linear regression for scan3.

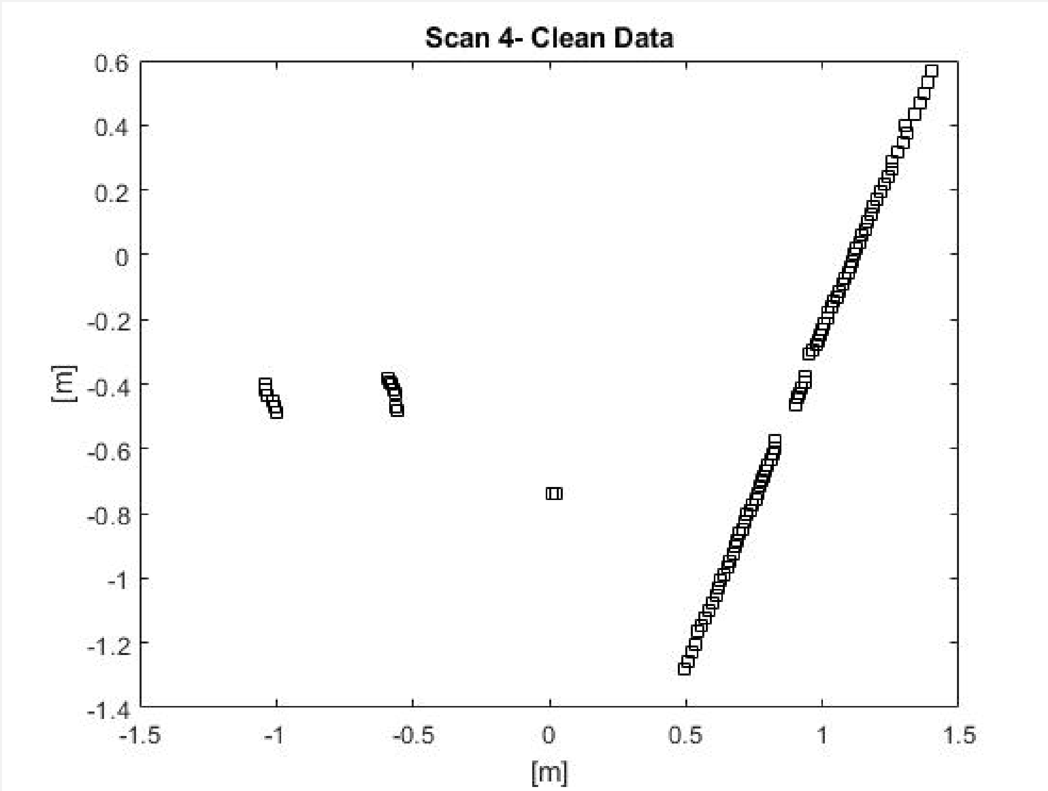


Figure 41.11: Clean data from Scan4. Please add your fit lines using the Zoom annotation tools.

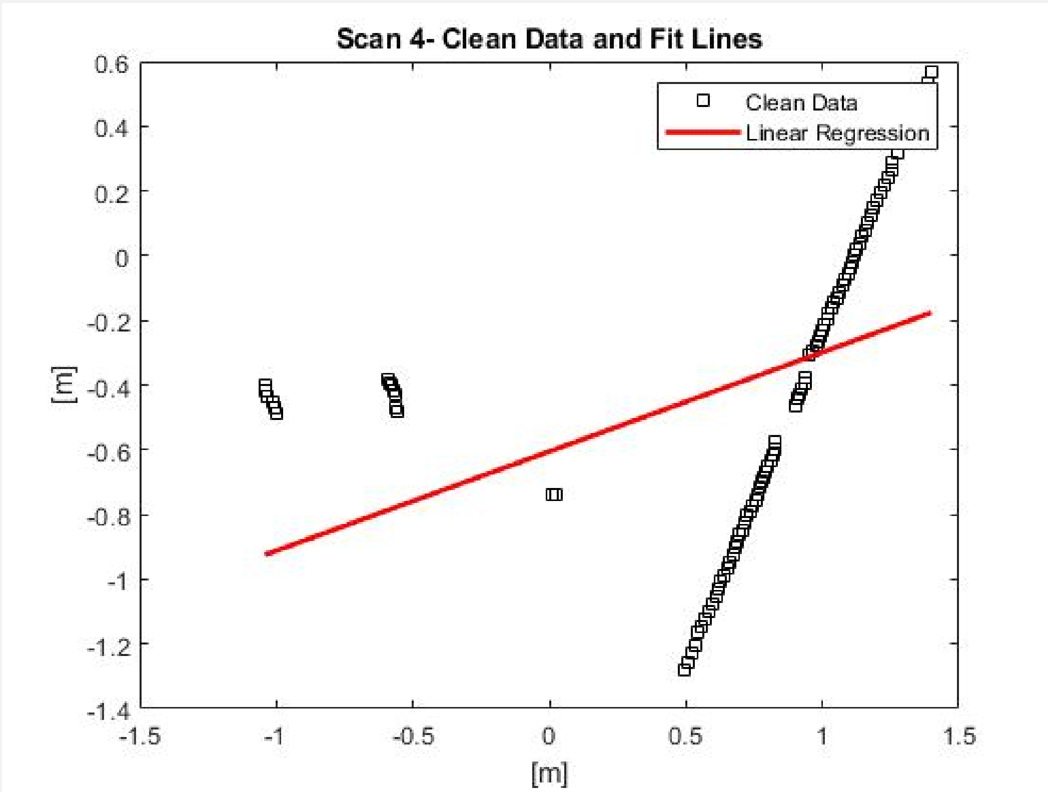


Figure 41.12: Best fit using linear regression for scan4.

Principal Component Analysis for Line Fitting [45 minutes]

Linear regression defines the line of best fit in terms of the sum of squared distances, measured vertically, between the data points and the line (see Figure 41.13). However, when determining the best fitting line for a laser scan, there's nothing special about the vertical direction. In fact, it would make as much sense to minimize the sum of the squared horizontal distances. In this context, a much more natural way to think about the line of best fit is to minimize the perpendicular distances between the line and the data points (see Figure 41.13).

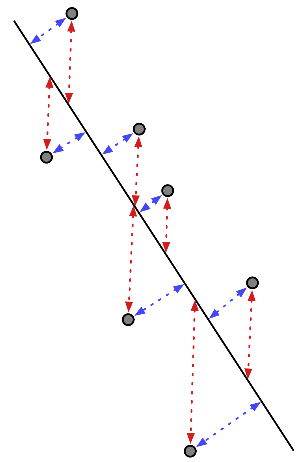


Figure 41.13: Laser scan points (gray dots), a proposed fitted line, and two different notions of error. The red dashed lines provide the vertical distance (which is used in linear regression). The blue dashed lines provide the perpendicular distance which is a more natural way to define line of best fit in the context of fitting lines to laser scan data.

Exercise 41.7

Think about how to use principal component analysis to determine the line of best fit that minimizes the sum of squared perpendicular distances as shown by the blue dashed lines in Figure 41.13. You might recall that in PCA, we found the directions of greatest variation in an (x, y) dataset by first creating a matrix of mean-centered data,

$$\mathbf{A} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} x_1 - \mu_x & y_1 - \mu_y \\ x_2 - \mu_x & y_2 - \mu_y \\ x_3 - \mu_x & y_3 - \mu_y \\ \vdots & \vdots \\ x_N - \mu_x & y_N - \mu_y \end{pmatrix} \tag{41.12}$$

and then finding the eigenvectors of the co-variance matrix, $\mathbf{R} = \mathbf{A}^T\mathbf{A}$.

1. Describe conceptually why you think this algorithm will achieve the desired goal.
2. Write pseudocode to describe the steps you will need to take. You should consider what an eigenvector represents. (Hint: if you translate your data, be sure to translate it back.)

Exercise 41.8

In MATLAB, do the following for the Cartesian data for scans scan2.mat and scan3.mat.
Note: Make use of the code snippets above to remove zeroes and convert the data from polar to Cartesian coordinates. To find the best fit line using PCA, the `pca` function in Matlab will be useful. Remember you can always use `help pca` or `doc pca` to learn the usage details of the `pca` function. Once you have attempted the Matlab solution as a group, a working solution is here.

1. Compute the line of best fit using PCA.
2. Plot the data points, line of best fit from PCA, and line of best fit from linear regression. Be sure to note which line is which.
3. How does the fit using PCA differ from your qualitative fit from above, and from the fit using linear regression? Why do you think this is? Discuss how the difference between linear regressions minimizing vertical error, and PCA identifying the direction of greatest (and least) variation impacts the fit lines.

Solution 41.1

1. θ=2.5 radians, R=10
2. θ=1.18 radians, R=13

Solution 41.2

1. x=1.73, y=1.00
2. x=1, y=-1.73

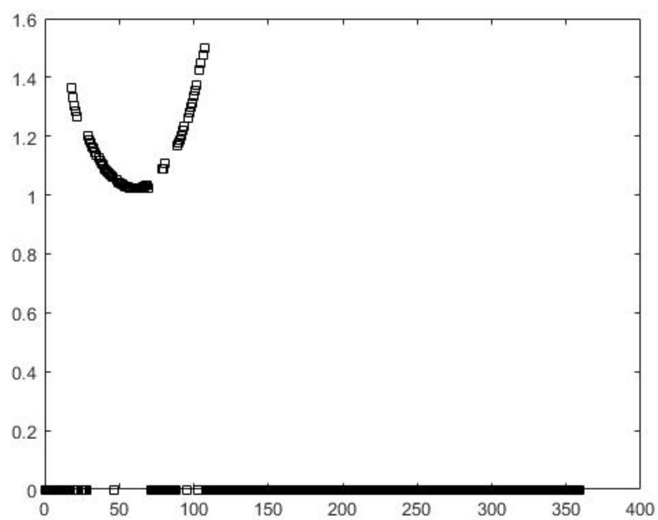
Solution 41.3

$\hat{r}$ and $\hat{\theta}$ are orthogonal, so $\hat{r}^T\hat{\theta}$=0.

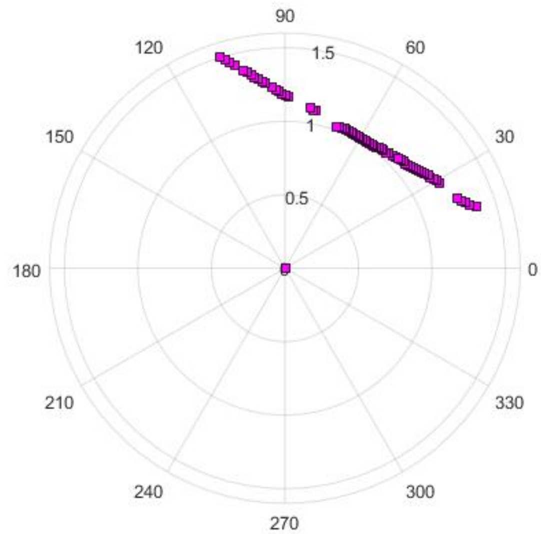
Solution 41.4

The $\hat{r}$ vector has length 1 and points in the direction of $\vec{r}$, and $\hat{\theta}$ has length 1 and points in the direction of increasing θ, orthogonal to $\hat{r}$.

Solution 41.5



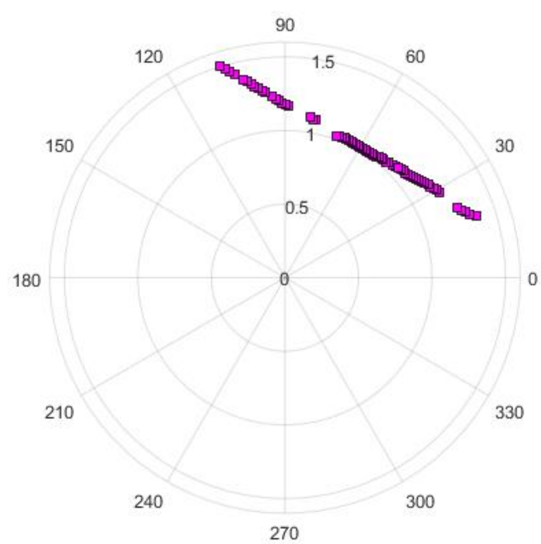
1.

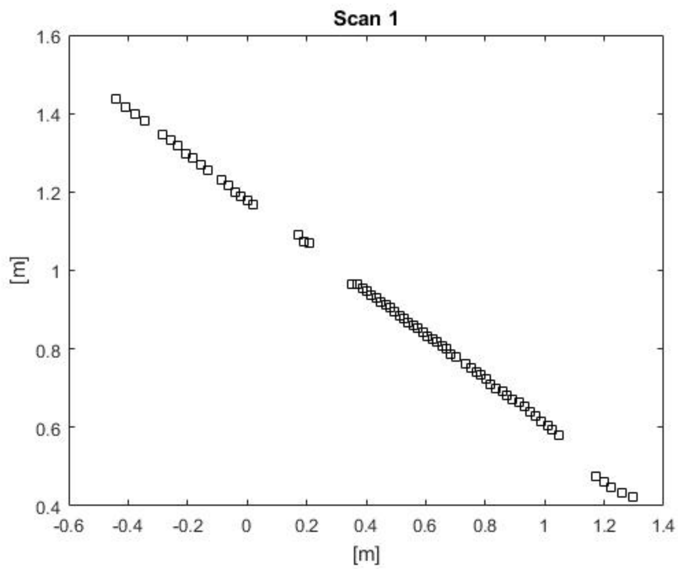


2.

3. The samples making up the line in part a are all at zero distance, and appear as a cluster and the center of the polar plot. The parabola from part a becomes the line in part b.

4. one implementation:
   index=find(r∼ =0);
   r_clean=r(index);
   theta_clean=theta(index);



5. 

An example code section that could do this whole process:

```
%load data from Scan1
load scan1.mat
r1=r;
theta1=theta;

%eliminate zeros
index1=find(r1≠0); %Note, the "not equal" sign in Matlab is a tilde followed by ...
    an equal.
r1_clean=r1(index1);
theta1_clean=theta1(index1);
figure;
polarplot(deg2rad(theta1_clean),r1_clean,'ks','MarkerSize',6,'MarkerFaceColor','m')

[x1,y1]=pol2cart(deg2rad(theta1_clean),r1_clean);
figure;
plot(x1,y1,'ks')
title('Scan 1- Clean Data')
xlabel('[m]')
ylabel('[m]')
```

Solution 41.7

1. We can recall from Module 1 that PCA identifies the direction of greatest variation in a data set (this direction is given by the eigenvector associated with the largest eigenvalue). Conceptually, the direction of *greatest* variation should be orthogonal to the direction of *least* variation. So, the direction perpendicular, or orthogonal, to the line found using PCA should minimize variation, and therefore minimize the sum of squared perpendicular distances.
2. a) Convert data from polar to cartesian coordinates.

b) Calculate the x and y means for the dataset.

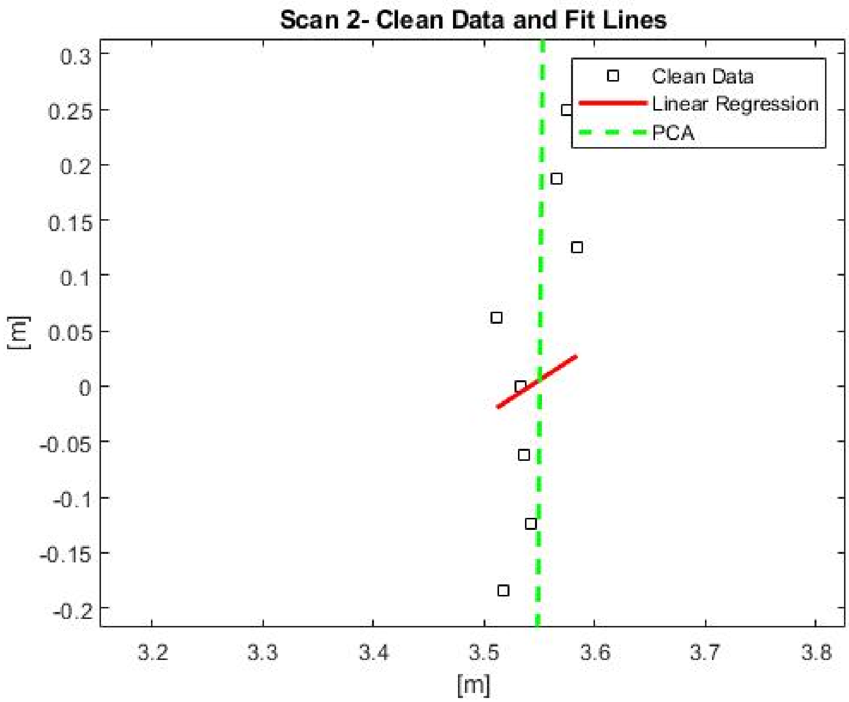
c) Use the x and y means to center the data.

d) Find the singular value decomposition (SVD).

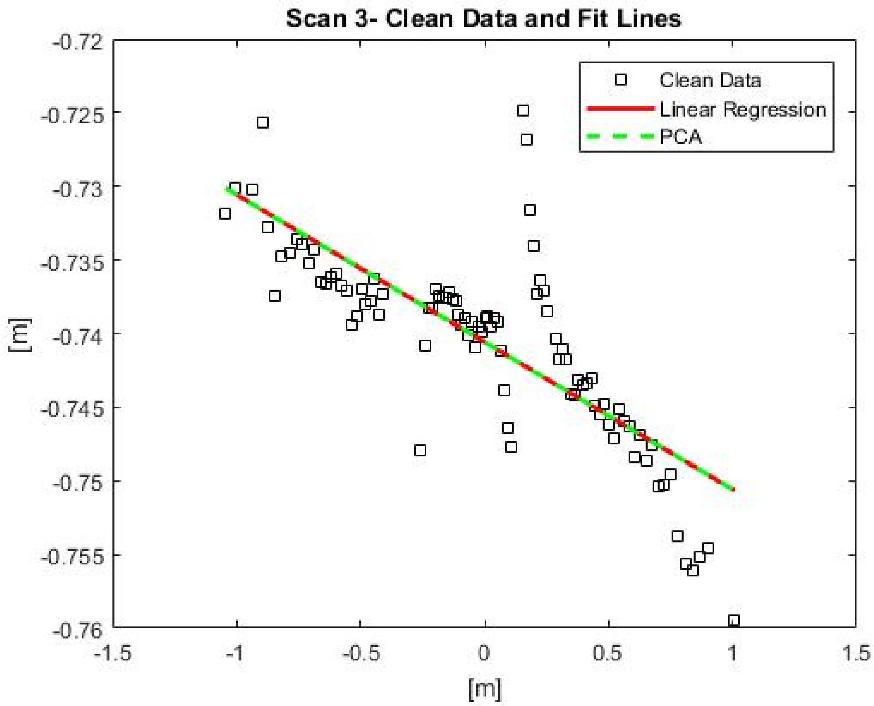
e) Identify the direction of greatest variation in the dataset as the eigenvector associated with the largest eigenvalue.

f) Translate the data back to its original positions.

Solution 41.8



Clean data and best fit lines using linear regression and PCA for Scan 2.



Clean data and best fit lines using linear regression and PCA for Scan 3.

Chapter 42

Night 7: Frames of Reference and LIDAR

Learning Objectives

Concepts

- Create a matrix that rotates a vector from one 2D coordinate system to another that shares its origin but is rotated an angle, θ, clockwise, relative to the original coordinate system.
- Create a matrix that translates the origin of a 2D coordinate system to another known point.

MATLAB skills

- Compute the vector coordinates of a vector in a global matrix to a rotated coordinate system with a translated origin.
- Determine the location of the NEATO in the global frame in response to LIDAR data.

Overview

In this overnight activity you are going to think about transforming points between different frames of reference, and how to apply this data collected by the LIDAR on the NEATO in order to create a "map" of the room. Rather than translating and rotating points, we will be using translation and rotation matrices to express points in different frames of reference. For example, imagine we want to refer to a point (a,b,c) in a global frame (G), from within a different frame (M). Our task would be to translate the origin and transform the G-frame to the M-frame. How would we do that? Let's first remember how we might transform the frames when they share an origin.

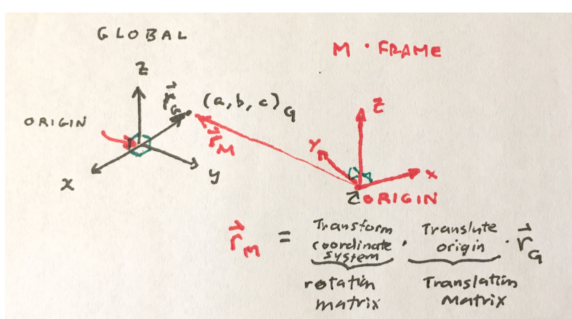


Figure 42.1: The global frame of reference (G) differs in origin and shape from the M-frame.

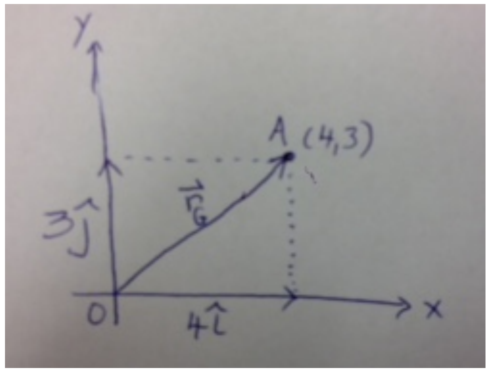


Figure 42.2: The coordinates of the point $(4, 3)$ are the projections of the position vector onto the relevant basis vectors.

42.1 Frames of Reference

In robotics and indeed many other applications, it is often useful to express points using different coordinate systems. You have already done this to a certain extent in Module 1 when you expressed vectorized images as linear combinations of Eigenfaces. In robotics applications, it is beneficial to express points relative to a fixed origin (e.g. a point designated as the origin in a room) and orthogonal basis vectors. In other cases, it is convenient to express points relative to an origin on the robot itself, with one basis vector in the direction of motion, and the others in orthogonal directions. Here, we will first develop some tools to translate points from one coordinate system to another, and apply these tools in the context of the NEATO. You will then be able to take points expressed in a coordinate system centered on the robot (e.g. from sensor data), and represent them in terms of a fixed coordinate system (e.g. with a corner of a room as the origin). We will work in 2D, but we can just as easily extend this discussion to higher dimensions.

Coordinate Systems with the Same Origin

A coordinate system consists of an **origin** and a set of **basis vectors**. Recall that a set of vectors form a basis if they are linearly independent—if they are mutually orthogonal then we have an orthogonal coordinate system.

The standard basis vectors for 2D are usually labelled $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$, but they are sometimes written as $\mathbf{e}_1$ and $\mathbf{e}_2$, or $\mathbf{e}_x$ and $\mathbf{e}_y$, or $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. From now on, we will assume that the global reference frame (frame G) is defined by the standard basis vectors, but just to be very clear we will use a subscript and write $\hat{\mathbf{i}}_G$ and $\hat{\mathbf{j}}_G$ as the basis vectors of the global frame G.

Points in 2D are expressed in terms of the basis vectors, and we refer to the component of the vector as the **coordinates** of the point. For example, the point A in Figure 1 has coordinates $(4, 3)$ when expressed in terms of the standard basis vectors. We might also write the position vector of A as

$$\mathbf{r}_G = 4\hat{\mathbf{i}}_G + 3\hat{\mathbf{j}}_G$$

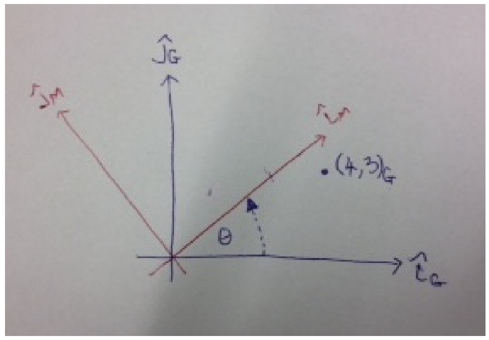


Figure 42.3: The frame M has the same origin, but the basis vectors are rotated by an angle of θ. The coordinates of the point $(4, 3)_G$ can be expressed in terms of the new frame M.

or express it as a row or column vector

$$\mathbf{r}_G = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

All of these refer to the same point, but the first and last representations imply the basis vectors, while the second representation makes it explicit. We use the notation $\mathbf{r}_G$ to be clear that this is the position vector of point A when expressed in the frame G. Notice that the coordinates (x_G, y_G) of the point A are simply the projection of the position vector onto the relevant basis vectors,

$$x_G = \mathbf{r}_G \cdot \hat{\mathbf{i}}_G,\ y_G = \mathbf{r}_G \cdot \hat{\mathbf{j}}_G$$

Since these basis vectors are mutually orthogonal the coordinates are simply $(4, 3)$ as expected, but again for clarity we will write $(4, 3)_G$ to mean that these are the coordinates in the frame G. You may recall that we met this concept in detail back in module 1 when we learned about *decomposition*.
How do we express the same point in terms of a new set of basis vectors, $\hat{\mathbf{i}}_M$ and $\hat{\mathbf{j}}_M$? Mathematically, we are trying to express the vector $\mathbf{r}_M$ as a linear combination of these vectors

$$\mathbf{r}_M = \mathbf{x}_M \hat{\mathbf{i}}_M + y_M \hat{\mathbf{j}}_M$$

where the coordinates of the point are now (x_M, y_M), i.e. the x and y coordinates of the point in the frame of reference M. See Figure 2.
The components of the point A expressed in this coordinate system is again the projection of the position vector in the frame G onto each basic vector of frame M in turn,

$$x_M = \mathbf{r}_G \cdot \hat{\mathbf{i}}_M,\ y_M = \mathbf{r}_G \cdot \hat{\mathbf{j}}_M$$

Since $\mathbf{r}_G = x_G \hat{\mathbf{i}}_G + y_G \hat{\mathbf{j}}_G$ we see that

$$x_M = x_G \hat{\mathbf{i}}_G \cdot \hat{\mathbf{i}}_M + y_G \hat{\mathbf{j}}_G \cdot \hat{\mathbf{i}}_M,\ y_M = x_G \hat{\mathbf{i}}_G \cdot \hat{\mathbf{j}}_M + y_G \hat{\mathbf{j}}_G \cdot \hat{\mathbf{j}}_M$$

Whilst this looks cumbersome, it becomes a lot clearer when we use the following matrix-vector formulation

$$\begin{bmatrix} x_M \\ y_M \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{i}}_G \cdot \hat{\mathbf{i}}_M & \hat{\mathbf{j}}_G \cdot \hat{\mathbf{i}}_M \\ \hat{\mathbf{i}}_G \cdot \hat{\mathbf{j}}_M & \hat{\mathbf{j}}_G \cdot \hat{\mathbf{j}}_M \end{bmatrix} \begin{bmatrix} x_G \\ y_G \end{bmatrix}$$

This matrix is the transformation matrix from the global reference frame (frame G) to the new reference frame (frame M), and we will often use the notation $\mathbf{R}_{MG}$ when referring to this transformation matrix

$$\mathbf{r}_M = \mathbf{R}_{MG}\mathbf{r}_G$$

For example, consider the frame M shown in Figure 2, which is simply the global frame rotated counter-clockwise by an angle θ. The basis vectors in this frame are

$$\hat{\mathbf{i}}_M = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \ \hat{\mathbf{j}}_M = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

which means that the transformation matrix from frame G to frame M is

$$\mathbf{R}_{MG} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

which is almost identical to the rotation matrix we met in Module 1. Rather than rotating the point through θ degrees, it rotates the coordinate system through θ degrees and expresses the point in terms of this new coordinate system. If $\theta = \pi/4$, the point $(4, 3)_G$ is expressed as

$$\begin{aligned} \mathbf{r}_M &= \begin{bmatrix} 1/\sqrt{2} & 1\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} \\ \Rightarrow \mathbf{r}_M &= \begin{bmatrix} 7/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} \end{aligned}$$

or equivalently $(7/\sqrt{2}, -1/\sqrt{2})_M$.
What if we have the coordinates of a point in frame M, and we wish to express them in frame G? Referring to the transformation matrix we developed earlier, we can express this using the matrix inverse

$$\mathbf{r}_G = \mathbf{R}_{MG}^{-1}\mathbf{r}_M$$

Since this inverse must be the transformation that takes points from frame M to frame G it must be true that

$$\mathbf{R}_{GM} = \mathbf{R}_{MG}^{-1}$$

Transforming from frame M to frame G corresponds to a clockwise rotation of θ so that $\mathbf{R}_{GM}$ must be the transpose of $\mathbf{R}_{MG}$,

$$\mathbf{R}_{GM} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

For example, consider the point $(1, 2)_M$ in frame M. The coordinates of this point in frame G must be

$$\begin{aligned} \mathbf{r}_G &= \begin{bmatrix} 1/\sqrt{2} & -1\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ \Rightarrow \mathbf{r}_G &= \begin{bmatrix} -1/\sqrt{2} \\ 3/\sqrt{2} \end{bmatrix} \end{aligned}$$

or simply $(-1/\sqrt{2}, 3/\sqrt{2})_G$.

Exercise 42.1

The frame M is a counterclockwise rotation of the global frame G by $\pi/3$ radians.

1. Draw the basis vectors for frame G and frame M.
2. Plot the frame G coordinates $(2, -1)_G$. Now express the frame G coordinates $(2, -1)_G$ in the frame M, and confirm that this is the same point by plotting it using the frame M.
3. Plot the frame M coordinates $(3, -2)_M$. Now express the frame M coordinates $(3, -2)_M$ in the frame G, and confirm that this is the same point by plotting it using the frame G.

Coordinate Systems with a Different Origin

In addition to defining new basis vectors, we often encounter situations in which we use a new origin. In Figure 3 we define a global frame with basis vectors $\hat{\mathbf{i}}_G$ and $\hat{\mathbf{j}}_G$, and origin O_G. We also define a frame M with basis vectors $\hat{\mathbf{i}}_M$ and $\hat{\mathbf{j}}_M$, and origin O_M. How do we transform points from frame G to frame M, and vice versa?

Fortunately, we already met the concept of translating points in Module 1, and here we will utilize these ideas to translate the origin before rotating the basis vectors. Recall from Module 1 that in order to translate a point we can use the translation matrix

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

where t_x and t_y are the components of the translation, and the translation matrix acts on the vector

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

For example, let's consider the case when the origin of frame M is located at $(a, b)_G$. The coordinates of this point in frame M must be, by definition, $(0, 0)_M$. The components of the translation must therefore be $-a$ and $-b$ and the translation matrix that moves the origin of frame G to frame M is then

$$\mathbf{T}_{MG} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$$

and the translation matrix that moves the origin of frame M to frame G

$$\mathbf{T}_{GM} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

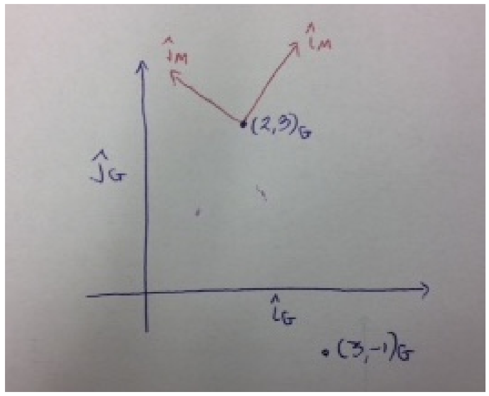


Figure 42.4: The frame M has an origin at $(2,3)_G$, and the basis vectors are rotated by an angle of θ. The coordinates of the point $(3,-1)_G$ can be expressed in terms of the new frame M.

In order to be consistent, we should adapt our rotation matrix so that it acts on a vector with 1 in the third slot

$$\mathbf{R}_{MG} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{R}_{GM} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We are now ready to transform a point from frame G to frame M, by first translating the origin of frame G to the origin of frame M, and then rotating the basis vectors from frame G to frame M. The position vector of an arbitrary point is then

$$\mathbf{r}_M = \mathbf{R}_{MG}\mathbf{T}_{MG}\mathbf{r}_G$$

We can, if we choose, combine the translation with the rotation into a general transformation, but we don't have to, and there are some advantages to keeping the distinction clear.
Transforming back from frame M to frame G would be accomplished with

$$\begin{aligned} \mathbf{r}_G &= (\mathbf{R}_{MG}\mathbf{T}_{MG})^{-1}\mathbf{r}_M \\ \Rightarrow \mathbf{r}_G &= \mathbf{T}_{MG}^{-1}\mathbf{R}_{MG}^{-1}\mathbf{r}_M \end{aligned}$$

We've already seen that the inverse of the rotation matrix is just the transpose of the original and thus $(\mathbf{R}_M G)^{-1} = R_{GM}$. This makes sense. To rotate back from frame M to frame G we use the R_{GM}. Furthermore, the inverse of the translation matrix $\mathbf{T}_{MG}$ is just the translation matrix $\mathbf{T}_{GM}$. Notice, however, that we first apply the inverse rotation and then the inverse translation,

$$\mathbf{r}_G = \mathbf{T}_{GM}\mathbf{R}_{GM}\mathbf{r}_M$$

For example, let's express the point $(3,-1)_G$ in frame M, which has its origin at $(2,3)_G$, with basis vectors rotated counterclockwise by $\pi/4$. The coordinates in frame M are therefore

$$\begin{aligned}
\mathbf{r}_M &= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} \\
\Rightarrow \mathbf{r}_M &= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix} \\
\Rightarrow \mathbf{r}_M &= \begin{bmatrix} -3/\sqrt{2} \\ -5/\sqrt{2} \\ 1 \end{bmatrix}
\end{aligned}$$

which means the coordinates are $(-3/\sqrt{2}, -5/\sqrt{2})_M$. Let's check to see if we can transform this point back from frame M to frame G. The coordinates in frame G are therefore

$$\begin{aligned}
\mathbf{r}_G &= \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3/\sqrt{2} \\ -5/\sqrt{2} \\ 1 \end{bmatrix} \\
\Rightarrow \mathbf{r}_G &= \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix} \\
\Rightarrow \mathbf{r}_G &= \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix}
\end{aligned}$$

which is just where we started!

Key Points

- A frame of reference is defined by an origin and a coordinate system.
- A coordinate system is defined by a set of basis vectors.
- The coordinates of a point correspond to the components along each basis vector of a position vector from the origin to the point.
- The coordinates of a point are therefore dictated by the frame of reference.
- The notation $\mathbf{r}_G$ or $(x,y)_G$ refers to the coordinates of a point in frame G.
- Points can be transformed from one frame to another using matrix multiplication.
- If frame M has origin located at $(a,b)_G$, and has basis vectors rotated counterclockwise by θ, then the transformation from frame G to frame M is

$$\mathbf{r}_M = \mathbf{R}_{MG}\mathbf{T}_{MG}\mathbf{r}_G$$

where $\mathbf{T}_{MG}$ is the matrix that translates the origin of frame G to frame M, and $\mathbf{R}_{MG}$ is the matrix that rotates the basis vectors of frame G to frame M:

$$\mathbf{R}_{MG} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{MG} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$$

- Transforming back from frame M to frame G involves the inverse of these matrices

$$\mathbf{r}_G = \mathbf{T}_{MG}^{-1}\mathbf{R}_{MG}^{-1}\mathbf{r}_M$$

- These concepts and quantities can be used to transform LIDAR data to the room frame in order to create a map.

Exercise 42.2

The frame M is a counterclockwise rotation of the global frame G by $\pi/3$ radians, and has its origin at $(-3, 1)_G$.

1. Draw the origin and basis vectors for frame G and frame M.
2. Plot the frame G coordinates $(2, -1)_G$. Now express the frame G coordinates $(2, -1)_G$ in the frame M, and confirm that this is the same point by plotting it using the frame M.
3. Plot the frame M coordinates $(3, -2)_M$. Now express the frame M coordinates $(3, -2)_M$ in the frame G, and confirm that this is the same point by plotting it using the frame G.

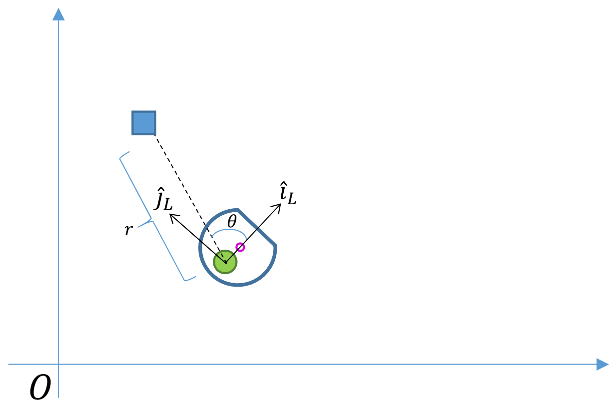


Figure 42.5: Illustration of NEATO with different coordinate systems with origin at the center of rotation of the LIDAR, and in the origin of a fixed frame of reference (e.g. origin of the room).

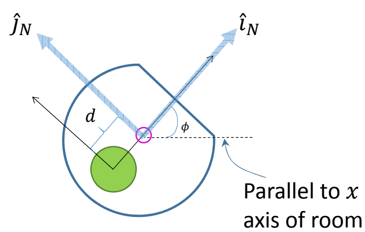


Figure 42.6: Illustration of NEATO with origin at center of rotation.

42.2 *Application to the NEATO*

The LIDAR reading provides a range and angle with respect to the center of rotation of the LIDAR sensor on the NEATO, with the angle measured relative to the front of the NEATO as illustrated by θ in Figure 42.5. The square object is located at a distance r and angle θ as measured by the LIDAR on the Neato.
For further reference, consider Figure 42.6 which indicates the location of the LIDAR relative to the center of rotation of the Neato. The center of rotation of the Neato is indicated by the magenta circle and is the origin of two orthogonal unit vectors $\hat{\mathbf{i}}_N$ and $\hat{\mathbf{j}}_N$ (thicker, light blue arrows). For the physical Neato, you could measure the distance d between the origin of the reference frame based on the LIDAR and the origin of the reference frame based on the Neato's center of rotation. For the simulated Neato, the origin of the frame based on the LIDAR is located at position $-0.084m\,\hat{\mathbf{i}}_N + 0m\,\hat{\mathbf{j}}_N$ relative to the origin of the Neato coordinate system.
The orientation of the Neato relative to the absolute horizontal axis of the room is indicated by the angle ϕ. Depending on your application, you may wish to express the position of the object (the box, in this case) in terms of a coordinate system relative to the center of rotation of the LIDAR sensor with unit vectors $\hat{\mathbf{i}}_L$

and $\hat{\mathbf{j}}_L$, center of rotation of the Neato with the unit vectors $\hat{\mathbf{i}}_N$ and $\hat{\mathbf{j}}_N$, or the global frame of reference indicated by the origin marked "O" and the blue arrows for which the unit vectors are $\hat{\mathbf{i}}_G$ and $\hat{\mathbf{j}}_G$.

Exercise 42.3

1. Suppose that the LIDAR returns a value of (r, θ) when scanning an object. With reference to Figure 42.5, please express the location of the object with respect to the LIDAR frame L.
2. With reference to Figures 42.5 and 42.6, please express the location of the object with respect to the NEATO frame N
3. Please express the location of the square object in the global frame G. Assume that the center of rotation of the NEATO is located at $(x_N, y_N)_G$

Exercise 42.4

Advice for this exercise.

- Before doing any coding, make sure you have a good sense of the chain of transformations needed to express the points in the global frame (this was the focus of the preceding exercise).
- Write out some pseudocode for how you are going to enact these transformations.
- Consider writing your code using a proper MATLAB script (.m). You may want to create a function that takes as input scan data and information about the location where the scan came from and returns the points in the global frame.
- Create intermediate visualizations (e.g., scatter plots) as you move from the polar representation in the LIDAR frame all the way to the global frame. Scrutinize these intermediate plots to make sure they make sense.

Now, we will use these techniques to take LIDAR data and build a map with respect to a fixed co-ordinate frame. In the classroom, we have defined an origin, x, and y axes, as well as placed objects on the floor at fixed locations in the Gauntlet. **Your job is to build a map of the Gauntlet when the NEATO is placed at different positions and orientations.** The map will be built using the global coordinate frame and unit vectors in the $\hat{\mathbf{i}}_G$ and $\hat{\mathbf{j}}_G$ directions. Recall that the LIDAR data is provided to you in polar co-ordinates using the coordinate frame with the origin at the center of the LIDAR sensor and the basis vectors pointing in the forward direction of the LIDAR, and 90^0 counter-clockwise from it.
To start the Gauntlet world, use the following Docker command. Once your simulator is running, you can connect to it through MATLAB using the normal procedures.

```
docker stop neato; docker rm --force neato; docker run --rm --name=neato --sysctl
    net.ipv4.ip_local_port_range="32401 32767" -p 11311:11311 -p 8080:8080 -p
    32401-32767:32401-32767 -e NEATO_WORLD=gauntlet_no_spawn -it
    qeacourse/robodocker:spring2020
```

1. Read through the parts (b)-(e) below. For the sake of logistics, you can collect all of your data (four different placements of the Neato) and then construct the plots. You can use the code in collectScans.m or write your own code to collect these data. You can move your virtual Neato using the code that you've used previously in this module placeNeato.m. Note that the code allows you to specify a position and heading vector for the Neato.

2. By default the NEATO starts at the origin of the global frame facing in the direction of the $\hat{\mathbf{i}}_G$ axis. Note that if you needed to reposition the Neato at this location, you can run the MATLAB command `placeNeato(0,0,1,0)`. In this position and orientation, collect data from the LIDAR. Express the LIDAR data in the fixed co-ordinate frame. Plot the data in MATLAB using the fixed reference frame and compare it to the locations of the objects in the Gauntlet.

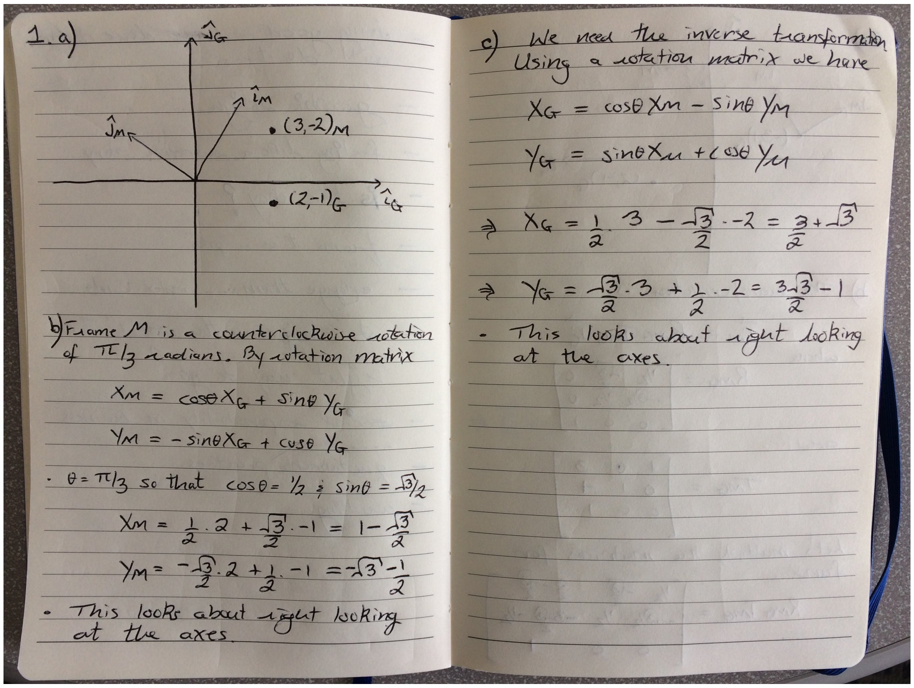
3. With the NEATO at the origin, rotate it through some angle ϕ (you can just pick your favourite angle here) and repeat what you did for the previous part (i.e., collect LIDAR data, translate to the fixed co-ordinate frame, plot in MATLAB, compare to the locations of the physical objects in the Gauntlet). In order to rotate the Neato by an angle ϕ you can run the MATLAB command `placeNeato(0, 0,cos(phi),sin(phi))`.

4. Now, move the NEATO to a different location (pick your favourite location in the Gauntlet), but keep it pointing in the same direction as the $\hat{\mathbf{i}}_G$ axis, and repeat what you did for the previous part. If you want to move the Neato to position $(a, b)_G$ but still have it $\hat{\mathbf{i}}_N$ oriented along $\hat{\mathbf{i}}_G$, you can run the MATLAB command `placeNeato(a,b,1,0)`. Make sure your Neato doesn't crash into anything when you move it. Note that it is a bit difficult to see where the axes of the Gauntlet coordinate system are located, but the grid shown in the visualizer is $1m$ by $1m$.

5. Now, move the NEATO to a different location and pointing in some arbitrary direction, and repeat what you did for the previous part.

6. Plot the locations of all four scans in the fixed (global reference frame) on the same plot (this is a great way to see if your code is correct). If all went well, the scans should mesh together well.

Solution 42.1



b) Frame M is a counterclockwise rotation of $\pi/3$ radians. By rotation matrix

$$X_M = \cos\theta X_G + \sin\theta Y_G$$

$$Y_M = -\sin\theta X_G + \cos\theta Y_G$$

- $\theta = \pi/3$ so that $\cos\theta = 1/2$; $\sin\theta = \sqrt{3}/2$

$$X_M = \frac{1}{2}\cdot 2 + \frac{\sqrt{3}}{2}\cdot -1 = 1 - \frac{\sqrt{3}}{2}$$

$$Y_M = -\frac{\sqrt{3}}{2}\cdot 2 + \frac{1}{2}\cdot -1 = -\sqrt{3} - \frac{1}{2}$$

- This looks about right looking at the axes.

c) We need the inverse transformation. Using a rotation matrix we have

$$X_G = \cos\theta X_M - \sin\theta Y_M$$

$$Y_G = \sin\theta X_M + \cos\theta Y_M$$

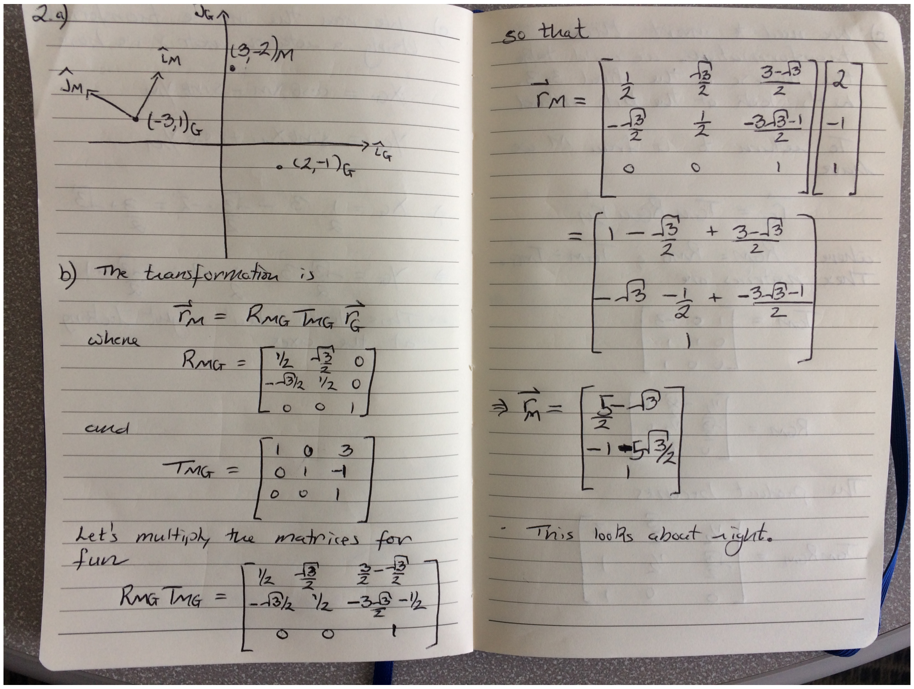
$$\Rightarrow X_G = \frac{1}{2}\cdot 3 - \frac{\sqrt{3}}{2}\cdot -2 = \frac{3}{2} + \sqrt{3}$$

$$\Rightarrow Y_G = \frac{\sqrt{3}}{2}\cdot 3 + \frac{1}{2}\cdot -2 = \frac{3\sqrt{3}}{2} - 1$$

- This looks about right looking at the axes.

Solution 42.2

1. Parts 1 and 2:



2. a)

b) The transformation is

$$\vec{r}_M = R_{MG} T_{MG} \vec{r}_G$$

where

$$R_{MG} = \begin{bmatrix} 1/2 & \frac{\sqrt{3}}{2} & 0 \\ -\sqrt{3}/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$T_{MG} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

Let's multiply the matrices for fun

$$R_{MG} T_{MG} = \begin{bmatrix} 1/2 & \frac{\sqrt{3}}{2} & \frac{3}{2} - \frac{\sqrt{3}}{2} \\ -\sqrt{3}/2 & 1/2 & -\frac{3\sqrt{3}}{2} - 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

so that

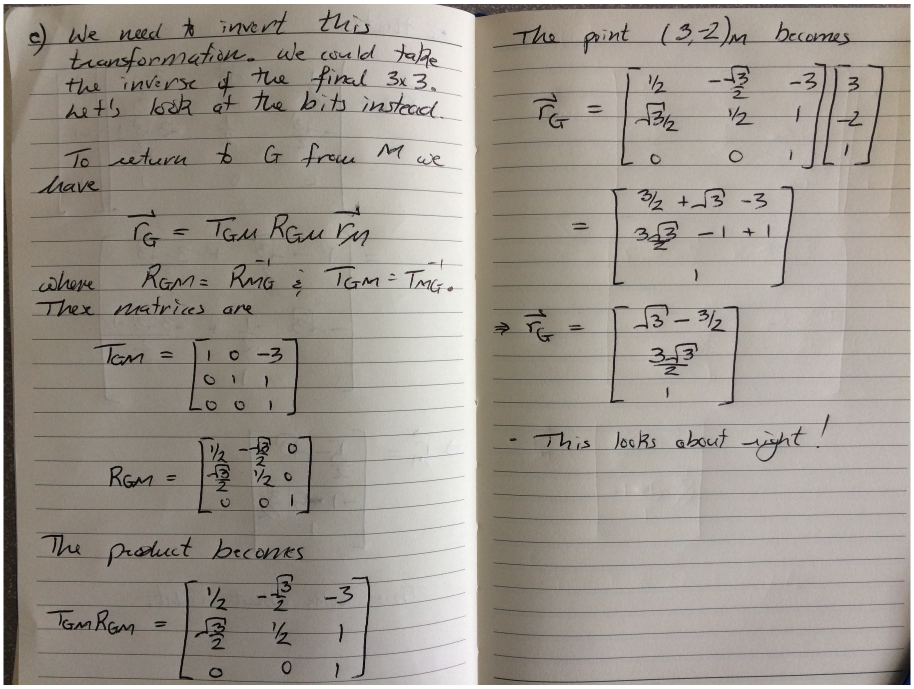
$$\vec{r}_M = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{3-\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{-3\sqrt{3}-1}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

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

$$\Rightarrow \vec{r}_M = \begin{bmatrix} \frac{5}{2} - \sqrt{3} \\ -1 - 5\sqrt{3}/2 \\ 1 \end{bmatrix}$$

- This looks about right.

2. Part 3:



c) We need to invert this transformation. We could take the inverse of the final 3x3. Let's look at the bits instead.

To return to G from M we have

$$\vec{r}_G = T_{GM} R_{GM} \vec{r}_M$$

where $R_{GM} = R_{MG}^{-1}$ & $T_{GM} = T_{MG}^{-1}$. These matrices are

$$T_{GM} = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{GM} = \begin{bmatrix} 1/2 & -\frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The product becomes

$$T_{GM} R_{GM} = \begin{bmatrix} 1/2 & -\frac{\sqrt{3}}{2} & -3 \\ -\frac{\sqrt{3}}{2} & 1/2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

The point $(3,-2)_M$ becomes

$$\vec{r}_G = \begin{bmatrix} 1/2 & -\frac{\sqrt{3}}{2} & -3 \\ \sqrt{3}/2 & 1/2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3/2 + \sqrt{3} - 3 \\ \frac{3\sqrt{3}}{2} - 1 + 1 \\ 1 \end{bmatrix}$$

$$\Rightarrow \vec{r}_G = \begin{bmatrix} \sqrt{3} - 3/2 \\ \frac{3\sqrt{3}}{2} \\ 1 \end{bmatrix}$$

- This looks about right!

Solution 42.3

1. We will denote the location of the object in the LIDAR frame L as $\mathbf{r}_L$. If we measure the polar coordinates of the object then its location is $\mathbf{r}_L = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}$.

2. We will denote the location of the object in the NEATO frame as $\mathbf{r}_N$. It is a translation with respect to the LIDAR origin. The location of the object is now $\mathbf{r}_N = \begin{bmatrix} r\cos\theta - d \\ r\sin\theta \end{bmatrix}$.

3. The NEATO is translated and rotated with respect to the room. The transformation matrices are

$$\mathbf{T}_{GN} = \begin{bmatrix} 1 & 0 & x_N \\ 0 & 1 & y_N \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{GN} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplying these out and transforming the coordinates of the object in the NEATO frame leads to

$$\mathbf{r}_G = \begin{bmatrix} r\cos(\theta+\phi) - d\cos\phi + x_N \\ r\sin(\theta+\phi) - d\sin\phi + y_N \end{bmatrix}$$

where we have used a trig formula.

Solution 42.4

Our solutions are available in the following two MATLAB files.

- collectScansSolution.m: after connecting to the simulator, run this function to position the robot and collect the scan data. The data is stored in a `.mat` file for subsequent process.
- makeGauntletMapSolution.m: after running `collectScansSolution`, you can run this function transform each scan to the global frame and plot them on a single plot.

Chapter 43

Day 8: The RANSAC Algorithm and Finding Lines

43.1 Schedule

- 1000-1005: Tech Time
- 1005-1020: Review and Preview
- 1020-1050: RANSAC Overview
- 1050-1100: Coffee
- 1100-1145: Pseudo-coding
- 1145-1230: Implementation

Learning Objectives

From LIDAR data, (r, θ), that contains a linear object and outliers,

- Design the steps for a RANSAC algorithm to find a line in a laser scan.
- Create an algorithm to sort the data points that fit the line and those that do not.

Challenge: Find multiple lines using the RANSAC algorithm from a LIDAR data set with multiple linear object signals.

43.2 Review and Preview [15 minutes]

We will start all together in the main session for any announcements, review of the Night 7 material, and a brief intro to important concepts in Day 8.

43.3 RANSAC [20 minutes]

Today we'll be learning a technique for optimization in the presence of outliers. The algorithm is called Random Sample Consensus, RANSAC for short, and it has applicability to a wide variety of problems in robotics and computer vision. In The Gauntlet™ we'll be using it to find lines in laser scan data even when multiple structures are present (e.g., multiple walls, lines and circles).

Motivating Example

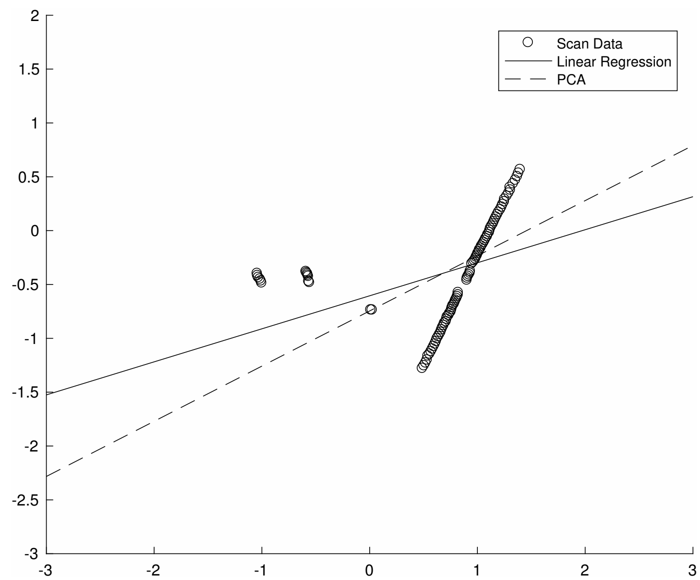


Figure 43.1: The lines of best fit as computed by linear regression and PCA. Due to the mixed structures in the data, the results are very poor.

In the overnight, you applied line fitting techniques to four different laser scans. The linear regression method worked well in some cases, but it had some clear shortcomings (e.g., when the scan points were oriented vertically). The PCA algorithm was able to overcome some of these limitations, however, there were cases when even PCA failed. For instance, Figure 43.1 shows the results of applying both linear regression and PCA to scan4.mat. Due to the fact that there are outliers (i.e. points that do not lie on the line), both linear regression and PCA find lines that do not correspond at all to the line clearly visible in the scan.

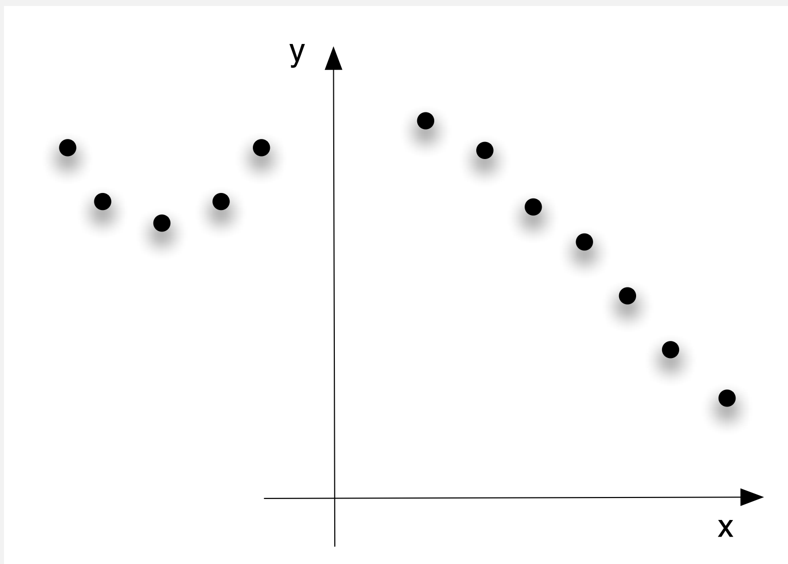
We can conclude from this example that the methods of line fitting that we've learned thus far are effective yet brittle (i.e. they fail when conditions aren't ideal). Motivated by this observation, today we'll be learning how to use the RANSAC algorithm to filter outliers *before* applying one of the line fitting methods you explored in the overnight.

Robust Optimization and the RANSAC Algorithm

The mathematical field of robust optimization provides us with techniques for optimizing functions (such as MSE) that are robust in the presence of outliers. Today, you'll explore a very powerful technique for robust optimization called Random Sample Consensus (or RANSAC for short). The algorithm works by choosing a small subset of data points to fit a model, determining whether or not a significant proportion of the data is consistent with the fitted model, and then repeating this process until a satisfactory model is found. For instance, let's suppose we want to find a line in this laser scan.

Exercise 43.1

1. Qualitatively, how many unique structures do you see in this laser scan?
2. Using the Zoom annotation tools, discuss and draw in your "best fit" line(s).
3. Based on your experience with linear regression and PCA during Day 7, where do you expect these algorithms would place a fit line? Go ahead and sketch them as well.



The RANSAC algorithm starts by randomly choosing a minimal subset of data points required to define a line (which of course is 2). Given these 2 points, we define our candidate line as the line that passes through both of those points (this is our "model"). Next, we divide *all* of the points in our laser scan into two groups. The first group, called *inliers*, consists of points in the laser scan that are sufficiently close to our candidate line. The second group of points, called *outliers*, consists of points in the laser scan that are far from the candidate line. We are free to choose the notion of what it means to be close or far from our candidate line. Based on our experience in the overnight, the perpendicular distance seems like a reasonable choice to measure the closeness of a point to a line (recall that this is what PCA uses). Therefore, we'll use the perpendicular distance as a metric for deciding whether a point is an inlier or an outlier. Figure 43.2 shows an example of applying this procedure. The inliers are marked in either white (for the two points used to make the line) or blue. The inliers are the points that fall within a perpendicular distance d from the line (d is a threshold value that you specify to RANSAC). The outliers (those that fall outside of a distance d of the line) are marked in red. While the line defined by the two randomly chosen points results in some inliers, the number of inliers (4 including the points used to make the line) is perhaps not as large as it could be.

Exercise 43.2

1. On Figure 43.2, draw a new best fit line and threshold lines that would optimize the number of inliers vs. outliers. Identify which points would serve as the endpoints of this line.

2. How did you choose your line/endpoints? Discuss possible strategies to accomplish this task.

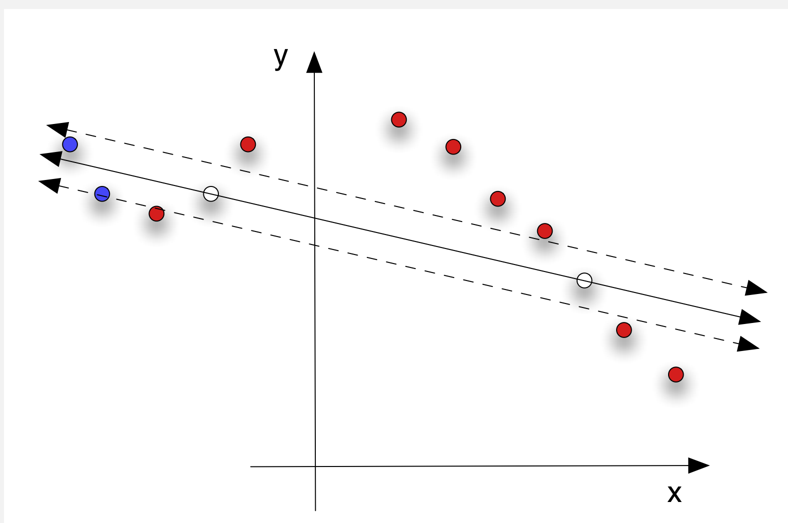


Figure 43.2: The white points represent the randomly chosen subset of two points to define the line. The solid line passing through them is the resultant line. The two dashed lines correspond to the inlier threshold (defined by measuring a specified distance d perpendicularly from the line). The points that are inliers are marked in blue and the outliers are red.

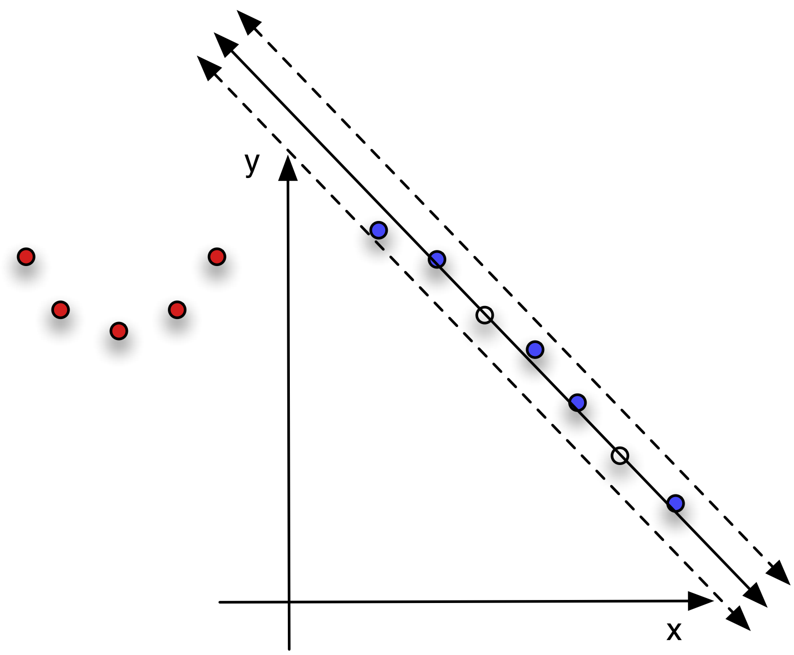


Figure 43.3: The white points represent the randomly chosen subset of points and to define the line. The solid line passing through them is that resultant line. The two dashed lines correspond to the inlier threshold (defined by measuring a specified distance d perpendicularly from the line). The points that are inliers are marked in blue and the outliers are red.

Next, we choose a new random set of 2 points, define the line passing through those points as our candidate line, and determine inliers and outliers. If this new candidate line results in more inliers than the first one we tried, we save the line for later. We repeat the procedure (choosing two random points, determining inliers, and testing to see if the number of inliers is the highest we've found so far) n times, where n is a parameter that you specify to the RANSAC algorithm. If all goes well, one of these n lines results in something like Figure 43.3. For the randomly chosen points shown in the figure, the number of inliers is 7 which looks to be about as good as we can do with this scan data.

As a final consideration, in the case of fitting lines to laser scan data, we want to be able to determine not just where the lines are in the scan, but also where the lines begin and end. This is crucial since the lines we are finding correspond to things like the beginning and end of walls or obstacles. There are a few ways to approach this task, and we'd like you to come up with your own method as part of today's activities. We are certainly here to scaffold this, but wrestling with this a bit will help to build your intuition about the geometry of the problem.

43.4 *Coffee Break [10 minutes]*

43.5 *Converting from Intuition to Pseudocode [45 minutes]*

Let's take stock of where we are. You should now have a pretty good idea about how RANSAC works on a conceptual level. Next, you'll be working to take this intuition and convert it into pseudocode.

When writing pseudocode it helps to start by stating your high-level conceptual understanding of how to solve the problem, and then, work towards progressively more concrete statements of how to solve the problem. Motivated by this idea, here's a possible procedure you can use with your partner to generate pseudocode.

1. **Clear up conceptual misunderstandings:** You just read through some text that described the RANSAC algorithm. Were there any parts that didn't make sense at a conceptual level? If so, make sure you work through these with your partner. If you can't figure out one of your questions, let us know! We're here to help.

2. **Simulate the algorithm:** at the whiteboard, simulate the steps that RANSAC would go through to find a line in a laser scan. You can do this through a combination of pictures (like the ones in Figure 43.3) and explanatory text (e.g., select points at random). As you go through this process you may find that you don't understand some of the details as well as you thought. This is another chance to ask for help from the teaching team.

3. **Write out the major steps:** Next, write the sequence of major steps the algorithm should perform. By this point your descriptions should be getting more precise (although not necessarily more detailed). Deciding how big to make each step is a bit of a balancing act. You want to avoid sequences such as "1. Do the thing 2. ??? 3. Profit", but you certainly don't want to have a 25 step process. Shoot for something on the order of 5-7 steps. Each of these steps can later be subdivided into smaller pieces.

4. **Figure out your functions:** the major steps you've defined can be thought of as the functions that you will write to implement your algorithm. For instance, you may have come up with a step called "compute inliers and oultiers". The fact that you identified this as a major step for your algorithm suggests that making a function that performs this computation is probably a good idea. At the whiteboard write out a list of the functions you will create when implementing RANSAC. Make sure

to describe what each of these functions expects as input, what it will generate as output, and what it does.

5. **Write your pseudocode:** Next, write pseudocode for each of the functions you've identified as well as pseudocode that stitches these functions together to implement RANSAC. Your pseudocode should be written in natural language (avoid using actual MATLAB syntax in your pseudocode), yet precise enough that there is little ambiguity about how it could be translated into actual MATLAB code. To better make this last point, here are two different potential ways to write pseudocode for finding the maximum element in a list of values.

```
Input: a list of numbers L
for each number x in the list L
    if x is the highest value so far
        remember x
return the highest value we found
```

This pseudocode has some good properties. It is written in natural language and it specifies the inputs of the function. On the negative side there is still a good deal of ambiguity here. How do I test if "x is the highest value so far"? How do I "remember x"? Here's a version that improves the pseudocode in this respect.

```
Input: a list of numbers L
initialize a variable called maxval to the value negative infinity
for each number x in the list L
    if x is greater than maxval
        assign the value of x to maxval
return the variable maxval
```

This new version of the pseudocode can be unambiguously translated to a computer program.

Exercise 43.3

Write pseudocode to implement the algorithm described in Section 43.3. Your pseudocode should be capable of going from a polar coordinate representation of a laser scan to the endpoints, in Cartesian space, of a line segment that corresponds to a line in the laser scan.

43.6 Implementing Your Algorithm [50 minutes]

In this section you'll be translating your pseudo-code into MATLAB.

Exercise 43.4

Implement RANSAC using the pseudocode you wrote in the previous exercise. Test your algorithm on the data in scan4.mat. As a suggestion, you should define a top-level function called *robustLineFit* that takes as input a polar representation of the laser scan, the threshold d to use to determine whether a scan point is an inlier, and n the number of random lines to try. Once you've implemented your algorithm, experiment with d and n to understand their effect.
Debugging and implementation tips:

1. Visualize, visualize, visualize. For instance, make sure your procedure for determining inliers is correct, you should plot the fitted line, the inliers, and the outliers. Make sure to use different colors to plot the inliers versus outliers.

2. Develop incrementally. Build your program bit by bit. Experiment in the command window before writing code in your MATLAB script.

3. Set breakpoints. This can be accomplished by either using the *keyboard* statement or by creating a stop sign by clicking to the left of the line of MATLAB code. These breakpoints are particularly useful in two situations. The first situation is obvious – when attempting to debug code. The second situation is when you are about to write an intricate section of code. In this case, set the breakpoint where the code will eventually go. Run your existing (but incomplete) code. When MATLAB stops at the break point, you can prototype your solution in the command window before adding it into your script.

Recommended Extension: If all has gone well, you should have a beautiful line segment fit to the data in *scan4.mat*. Unfortunately for you, The Gauntlet™ is a bit more complicated than the Chamber of Emptiness™. Next you'll be updating your code to find multiple lines.

Exercise 43.5

First, write pseudocode to find multiple lines in a laser scan. When doing this you should HEAVILY leverage what you did in exercise 3. For instance, if you have a function called *robustLineFit* which takes a scan and computes the best fitting line segment, you can call it repeatedly to find multiple lines. Specifically, each time you call your *robustLineFit* code, you'll determine a line segment in scan. After determining a line segment, you should remove the inlier points for that line segment from the scan data. By removing these scan points each subsequent line you find will consist of points that were outliers with respect to the lines you previously found. Implement your approach and try it on the data stored in playpensample.mat.
Debugging and implementation tips:

1. In order to write this in a sane fashion, your functions will need to be solid. Make sure you are confident in each function before building on it.

2. You may consider modifying your *robustLineFit* to return multiple pieces of information

(instead of just the line segment of best fit). For instance, if you return the inlier points and outliers points separately your life will be a lot easier. If you're not familiar with returning multiple outputs from a MATLAB function, see this page.

Challenge Problem: If all went well you should now be finding all the walls in a fairly complex laser scan. You may have noticed some warts in the lines that your code finds. The biggest issue is that the line segments that are found sometimes have large gaps in them. This causes you to find lines that span across gaps in the environment. Later in the challenge you'll be using these line segments to determine a path through the environment, so it is important that the gaps are not covered over by spurious lines.

Exercise 43.6

Modify your *robustLineFit* code to avoid fitting line segments with large gaps. To do this, define a procedure for computing the largest gap in a candidate line segment. Start out by working on the board to build intuition, and only when you have a good sense of how to solve the problem, implement things in MATLAB.

Solution 43.4

An example solution using the RANSAC algorithm with d=0.1m and n=30 iterations is shown in Figure 43.4. A heavily commented, working RANSAC function is here.

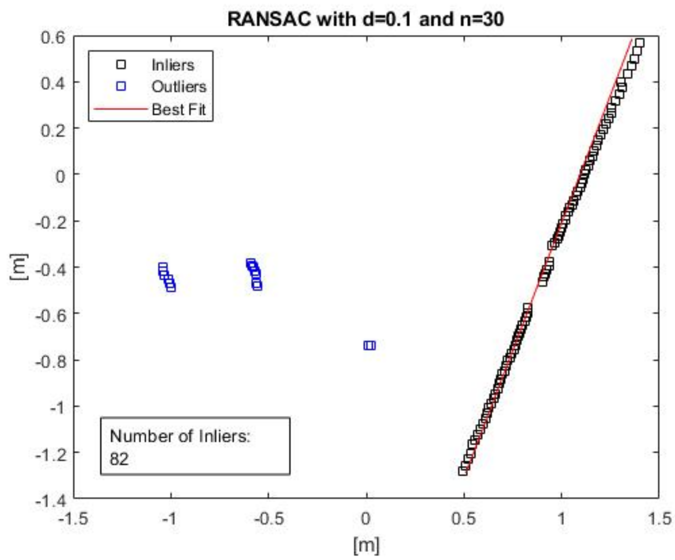


Figure 43.4: Best fit to Scan4 using RANSAC with d=0.1m and n=30.

Solution 43.5

A solution is here.

Solution 43.6

In the solution here, gaps are identified using the code:

```
%we also want to check that there are no big gaps in our walls. To do
   %this, we are first taking the distance of each inlier away from an
   %endpoint (diffs) and projecting onto the best fit direction. We then
   %sort these from smallest to largest and take difference to find the
   %spacing between adjacent points. We then identify the maximum gap.
   biggestGap = max(diff(sort(diffs(inliers,:)*v/norm(v))));
```

Chapter 44

Night 8: Scalar and Vector Fields

Learning Objectives

Concepts

1. Differentiate between scalar and vector fields.
2. Determine the Jacobian, the divergence, and the curl of a vector field.
3. Build a scalar potential field using sources and sinks.

44.1 Scalar and Vector Fields - Definitions

Scalar and vector fields are mathematical objects which are ubiquitous across many disciplines. You have already been working with both of these concepts already, but now is a good time to formalize the definitions and deepen our understanding of these ideas.

A **scalar field** is a function which is defined across a space of inputs and outputs a scalar value. The input space can be of any dimensionality. As an example, a function which defines the temperature profile at all points in the air in the QEA classroom, $T(x, y, z)$ is a scalar field with a three dimensional input and a one dimensional output. The word "field" tells us that the input is a vector, and the word "scalar" tells us that the output is, well, a scalar.

Another example of a scalar field is the elevation of a mountain as a function of two-dimensional position, $H(x, y)$. A scalar field then is a function f which accepts a vector as an input and outputs a scalar—we can either use the notation $f(x, y, z)$ which specifies the components of the input, or the more general notation $f(\mathbf{r})$. If we want to be even more specific, we can write $f : \mathbf{R}^N \to \mathbf{R}$, which is shorthand notation for f takes real values in N dimensions as input and outputs a real value.

A **vector field** is a function which is defined across a space of inputs and outputs a vector. An example of this would be the velocity of the air currents at all points in the QEA classroom $\mathbf{v}(x, y, z) = v_x\hat{\mathbf{i}} + v_y\hat{\mathbf{j}} + v_z\hat{\mathbf{k}}$, where v_x is the component of the velocity in the x-direction etc. (Although this is a widely used notation, it can be confusing because v_x is also the notation used to define the partial derivative of v with respect to x!) Another example is the gradient function of the surface of a mountain $\nabla H = \frac{\partial H}{\partial x}\hat{\mathbf{i}} + \frac{\partial H}{\partial y}\hat{\mathbf{j}}$. A vector field then is a function $\mathbf{F}$ which accepts a vector as an input and outputs a vector of the same number of dimensions—again we can either use the notation $\mathbf{F}(x, y, z)$, or the more general notation $\mathbf{F}(\mathbf{r})$, where $\mathbf{F} : \mathbf{R}^N \to \mathbf{R}^N$.

Exercise 44.1

Which of the following quantities could be considered a scalar field or vector field (or neither).

1. The temperature at all points in your room right now.
2. The temperature gradient at all points in your room right now.

3. The velocity of a bird in flight over the course of a day.
4. The population density at all points in the USA right now.
5. The wind velocity at all points in the USA right now.

Exercise 44.2

Generate a list of at least 3 quantities and classify them as a scalar field or a vector field.

44.2 *Visualizing Scalar and Vector Fields*

We've already spent quite a bit of time visualizing scalar and vector fields. Recall that for scalar fields we often plot contours, and for vector fields we plot, well, a set of vectors defined at lots of points in space.

Exercise 44.3

Use the favorite tool of your choice (Matlab, WolframAlpha, etc) to visualize the following scalar and vector fields.

1. $f(x, y) = (x-1)^2 + (y-2)^2$
2. $f(x, y, z) = x^2 + 2y^2 + 3z^2$
3. $\mathbf{F}(x, y) = (x^2 - y)\hat{\mathbf{i}} + (x + y^2)\hat{\mathbf{j}}$
4. $\mathbf{F}(x, y, z) = (2y - z)\hat{\mathbf{i}} + (x + y^2 - z)\hat{\mathbf{j}} + (4y - 3x)\hat{\mathbf{k}}$

44.3 *Jacobian, Divergence, and Curl of Vector Fields*

We've already met partial derivatives, and the gradient operator $\nabla = \begin{bmatrix} \partial_x & \partial_y & \partial_z \end{bmatrix}$, which is the multi-dimensional equivalent of the derivative. We can apply the gradient to vector fields in a number of ways. The material here is a very brief introduction and is focused on introducing terminology, not interpretation - that will come later!

The Jacobian of a vector field

Given a vector field $\mathbf{F}(x,y,z) = X(x,y,z)\hat{\mathbf{i}} + Y(x,y,z)\hat{\mathbf{j}} + Z(x,y,z)\hat{\mathbf{k}}$ its **Jacobian** is the matrix of partial derivatives

$$J\mathbf{F} = \begin{bmatrix} \frac{\partial X}{\partial x} & \frac{\partial X}{\partial y} & \frac{\partial X}{\partial z} \\ \frac{\partial Y}{\partial x} & \frac{\partial Y}{\partial y} & \frac{\partial Y}{\partial z} \\ \frac{\partial Z}{\partial x} & \frac{\partial Z}{\partial y} & \frac{\partial Z}{\partial z} \end{bmatrix}$$

For example, the derivative of the vector field $\mathbf{F}(x,y,z) = x^2\hat{\mathbf{i}} + xyz\hat{\mathbf{j}} + (x^2 - z^2)\hat{\mathbf{k}}$ is the matrix

$$J\mathbf{F} = \begin{bmatrix} 2x & 0 & 0 \\ yz & xz & xy \\ 2x & 0 & -2z \end{bmatrix}$$

. We often simply refer to the Jacobian matrix as the "derivative" of the vector field, but that can be confusing.

Exercise 44.4

Find the Jacobian of the following vector fields.

1. $\mathbf{F}(x,y,z) = (3x+4y)\hat{\mathbf{i}} + (4y-5z)\hat{\mathbf{j}} + (x+y-z)\hat{\mathbf{k}}$
2. $\mathbf{F}(x,y,z) = yz\hat{\mathbf{i}} + xz\hat{\mathbf{j}} + xy\hat{\mathbf{k}}$

The Divergence of a Vector Field

Consider a vector field given in Cartesian coordinates as $\mathbf{F}(x,y,z) = X(x,y,z)\hat{\mathbf{i}} + Y(x,y,z)\hat{\mathbf{j}} + Z(x,y,z)\hat{\mathbf{k}}$. The **divergence** of $\mathbf{F}$ is the scalar field

$$\mathrm{div}\mathbf{F} = \frac{\partial X}{\partial x} + \frac{\partial Y}{\partial y} + \frac{\partial Z}{\partial z}$$

For example, the divergence of $\mathbf{F} = \sin(x)\cos(y)\hat{\mathbf{i}} + x^2yz\hat{\mathbf{j}} + (\sin(z) - x^2)\hat{\mathbf{k}}$ is the quantity $\mathrm{div}\mathbf{F}(x,y) = \cos(x)\cos(y) + x^2z + \cos(z)$.

Exercise 44.5

Find the divergence of the following vector fields

1. $\mathbf{F}(x,y) = \cos(x)\sin(y)\hat{\mathbf{i}} - \sin(x)\cos(y)\hat{\mathbf{j}}$
2. $\mathbf{F}(x,y,z) = y\sin(z)\hat{\mathbf{i}} - x\sin(z)\hat{\mathbf{j}} + \cos(z)\hat{\mathbf{k}}$

Exercise 44.6

Show that the divergence of $\mathbf{F}$ is the dot product of the gradient operator with the vector field, i.e.

$$\text{div}\mathbf{F} = \nabla \cdot \mathbf{F}$$

The Curl of a Vector Field

Consider a vector field given in Cartesian coordinates as $\mathbf{F}(x,y,z) = X(x,y,z)\hat{\mathbf{i}}+Y(x,y,z)\hat{\mathbf{j}}+Z(x,y,z)\hat{\mathbf{k}}$. The **curl** of $\mathbf{F}$ is the vector field

$$\text{curl}\mathbf{F} = (\frac{\partial Z}{\partial y} - \frac{\partial Y}{\partial z})\hat{\mathbf{i}} - (\frac{\partial Z}{\partial x} - \frac{\partial X}{\partial z})\hat{\mathbf{j}} + (\frac{\partial Y}{\partial x} - \frac{\partial X}{\partial y})\hat{\mathbf{k}}.$$

Exercise 44.7

Compute the curl of the following vector fields

1. $\mathbf{F}(x,y) = \cos(x)\sin(y)\hat{\mathbf{i}} - \sin(x)\cos(y)\hat{\mathbf{j}}$
2. $\mathbf{F}(x,y,z) = y\sin(z)\hat{\mathbf{i}} - x\sin(z)\hat{\mathbf{j}} + \cos(z)\hat{\mathbf{k}}$

Exercise 44.8

Show that the curl of $\mathbf{F}$ is the cross product of the gradient operator with the vector field, i.e.

$$\text{curl}\mathbf{F} = \nabla \times \mathbf{F}$$

44.4 Potential Fields using Sources and Sinks

In an earlier assignment we used the method of steepest ascent to navigate the NEATO to the top of a "mountain", which we defined using a scalar field $H(x,y)$.

In the Gauntlet challenge we are going to use targets (bucket of benevolence) and objects (walls, boxes, etc) to create an artificial mountain landscape, and we will navigate the NEATO using the **method of steepest descent** so as to avoid the obstacles and hone in on the target. (We use steepest descent in order to connect this with ideas from physics.)
To accomplish this task, we are going to introduce a particular scalar field $V(x, y)$ which can be used to create useful mountain landscapes. It is known as a **potential** field.

Exercise 44.9

Let $V(x, y) = \ln\sqrt{x^2 + y^2}$. Note that V is not defined at the origin!

1. What is V in polar coordinates (note: this question is just here to make you realize that it's actually a pretty function, and to make it easier to visualize)?
2. Create a contour plot of V.
3. Determine the gradient field defined by ∇V, and visualize it.
4. How would a NEATO performing gradient descent behave if you put it on a Flatland defined by V and started it at $(3, 3)$?

Exercise 44.10

Now consider $V(x, y) = \ln\sqrt{x^2 + y^2} - \ln\sqrt{(x-1)^2 + y^2}$. Notice that there is a strong "trough" at $(0, 0)$ and a strong "peak" at $(1, 0)$. The sign in front of the logarithm determined whether it is a peak or a trough.

1. Create a contour plot of V.
2. Determine the gradient field defined by ∇V, and visualize it.
3. How would a NEATO performing gradient descent behave if you put it on a Flatland that looked like V and started it at $(3, 3)$?

Exercise 44.11

In the last question, a NEATO would be attracted by the trough at $(0, 0)$ but repelled by the peak at $(1, 0)$. We will refer to the point at $(0, 0)$ as a "sink" and the point at $(1, 0)$ as a "source".

1. Define a scalar field $V(x, y)$ that has a sink at $(0, 0)$, and sources at $(1, 0)$ and $(2, 3)$.

2. Visualize the scalar field V and the gradient field ∇V.
3. How would a NEATO performing gradient descent behave if you put it on a Flatland that looked like V and started it at $(1, 2)$?

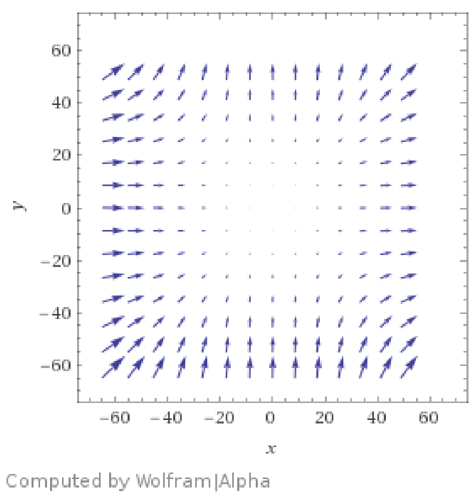
Solution 44.1

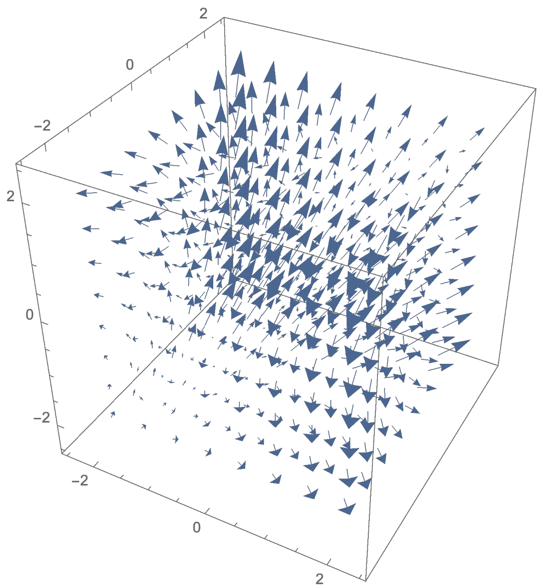
1. $T(x, y, z)$: scalar field because vector in and scalar out
2. $\nabla T(x, y, z)$: vector field because vector in and scalar out
3. $\mathbf{v}(t)$: neither because scalar in and vector out
4. $\rho(x, y)$: scalar field because vector in and scalar out
5. $\mathbf{V}(x, y)$: vector field because vector in and vector out

Solution 44.2

Hopefully lots of examples here!

Solution 44.3

1. The contours will be circles centered at $(1, 2)$.
2. The contours will be ellipsoids (axes 1,2,3) centered at the origin.
3. 

4. 

Solution 44.4

1. $J\mathbf{F} = \begin{bmatrix} 3 & 4 & 0 \\ 0 & 4 & -5 \\ 1 & 1 & -1 \end{bmatrix}$

2. $J\mathbf{F} = \begin{bmatrix} 0 & z & y \\ z & 0 & x \\ y & x & 0 \end{bmatrix}$

Solution 44.5

1. $\nabla \cdot \mathbf{F} = -\sin(x)\sin(y) + \sin(x)\sin(y) = 0$

2. $\nabla \cdot \mathbf{F} = 0 + 0 - \sin(z) = -\sin(z)$

Solution 44.6

$$\nabla \cdot \mathbf{F} = \begin{bmatrix} \partial_x & \partial_y & \partial_z \end{bmatrix} \cdot \begin{bmatrix} X & Y & Z \end{bmatrix} = \frac{\partial X}{\partial x} + \frac{\partial Y}{\partial y} + \frac{\partial Z}{\partial z}$$

Solution 44.7

1. $\nabla \times \mathbf{F} = -2\cos(x)\cos(y)\hat{\mathbf{k}}$

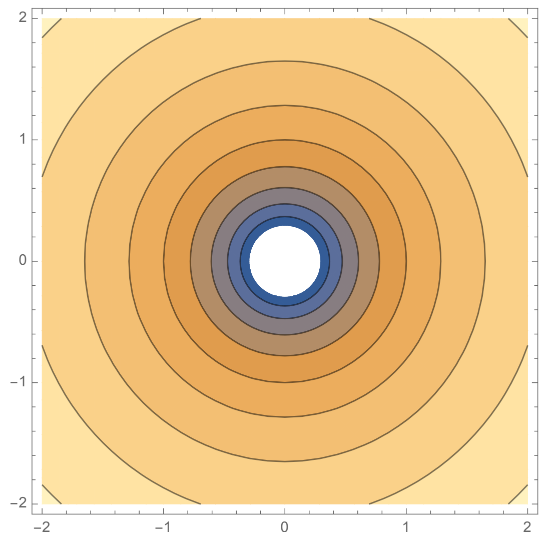
2. $\nabla \times \mathbf{F} = x\cos(z)\hat{\mathbf{i}} + y\cos(z)\hat{\mathbf{j}} - 2\sin(z)\hat{\mathbf{k}}$

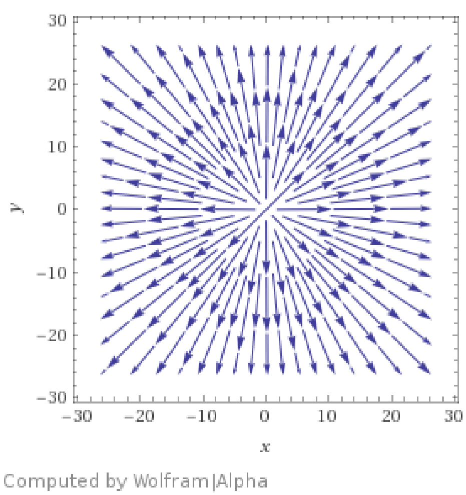
Solution 44.8

$$\nabla \times \mathbf{F} = \begin{bmatrix} \partial_x & \partial_y & \partial_x \end{bmatrix} \times \begin{bmatrix} X & Y & Z \end{bmatrix} = (\frac{\partial Z}{\partial y} - \frac{\partial Y}{\partial z})\hat{\mathbf{i}} - (\frac{\partial Z}{\partial x} - \frac{\partial X}{\partial z})\hat{\mathbf{j}} + (\frac{\partial Y}{\partial x} - \frac{\partial X}{\partial y})\hat{\mathbf{k}}$$

Solution 44.9

1. In polar coordinates $V(r, \theta) = \ln(r)$, so the contours of V should be circles.

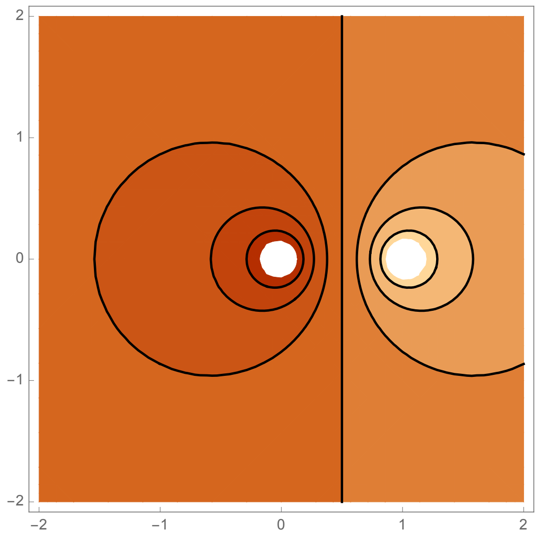


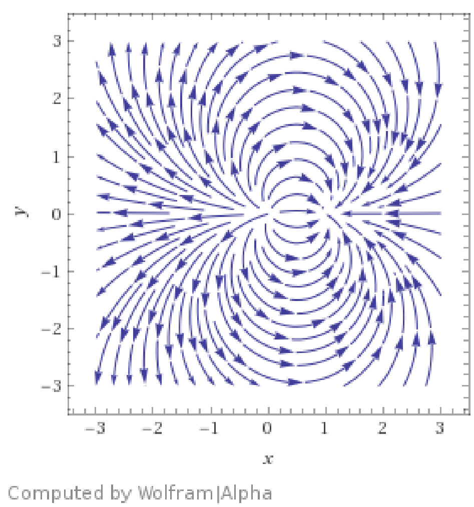
2. 

3.

4. It would move straight toward the origin (gradient descent)!

Solution 44.10

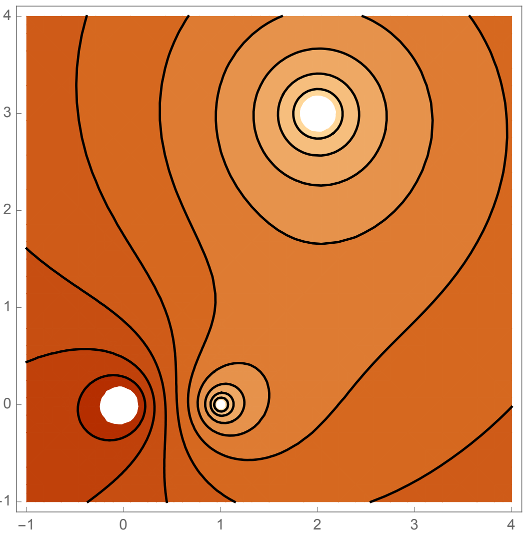
1. 

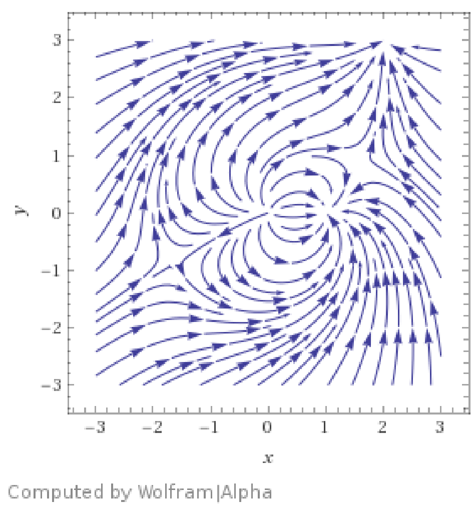
2. 

3. It would move away from $(1, 0)$ and eventually find its way to $(0, 0)$.

Solution 44.11

1. $V(x, y) = \ln\sqrt{x^2 + y^2} - \ln\sqrt{(x-1)^2 + y^2} - \ln\sqrt{(x-2)^2 + (y-3)^2}$

2. 

3. 

4. It would move away from $(1, 2)$ and eventually find its way to $(0, 0)$.