Math 305: Linear Algebra Labs

Created by: Professors Amanda Harsy and Michael Smith

July 12, 2024

.

Contents

Pre-Lab Assignments

- Lab 1: No pre-lab assignment.

- Lab 2: Read page 13 before lab.

- Lab 3: Read and answer the questions from Section 3.1 (through Colley Method -stop at 3.2). Turn in pre-lab questions with Lab 3 report.

- Lab 4: Read The Transmission Radiograph and Tomography (A Simplified Overview) and answer the questions on first 2 pages of pre-lab. Turn these in with Lab 4 report.

- Lab 5: Watch the video `https://youtu.be/ysOEnGg_rz0` and then answer the following questions from Section 5.1.

- Lab 6: Read through Common Misconceptions and answer the first 5 pre-lab questions before class.

- Lab 7: Complete the pre-reading in Section 7.1.

- Extra Lab 8: Read and answer questions from Section 8.1.

- Extra Lab 9: Read and answer questions from Section 9.1.

References

Radiography and Tomography Labs and Diffusion Labs are from the ImageMath.org project created by Dr. Tom Asaki, Dr. Heather Moon, and Dr. Marie Snipe. This material is based upon work supported by the National Science Foundation under grants DUE-1503929, DUE-1642095, DUE-1503870, and DUE-1503856. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Copy of Consent Form: Below is a copy of the consent form from our study of our Labs from this course:

I agree to participate in the study entitled, "The Impact of the Use of Application-focused Labs in Linear Algebra" being conducted by Mr. Michael Smith and Dr. Amanda Harsy, mathematics professors conducting research through Lewis. I have been informed that the purpose of the study is to research the impact and usefulness of incorporating labs in linear algebra courses. The data gathered through these surveys will also be used as feedback to improve the labs for our Linear Algebra students.

I understand that if I agree to participate in this study, I will be asked to complete a total of seven 10-15 minute surveys administered after each lab.

I understand that all information gathered during this study will be kept confidential by the researchers. The professors disseminating the data will be given no identifying information, so my answers to the prompts will be anonymous. No names are needed and for the research and no identifying information will be shared. Data collected will be kept on a password-protected computer for three years and then deleted.

I realize that Lewis University does not provide compensation for, nor do the institutions carry insurance to cover injury or illness incurred as a result of participation in this research project.

I understand that my consent to participate in this project does not constitute a waiver of any legal rights or redress I might have as a result of my participation, and I acknowledge that I have received a copy of this consent form.

I am aware that my participation is voluntary and I may withdraw at any time without penalty or prejudice. My decision whether or not to participate will not jeopardize my future relations with Lewis University and will not jeopardize my success in my current and future mathematics courses. I understand that I may refuse to answer any question that I do not wish to answer. If I have any additional questions concerning this study, I may contact Mr. Michael Smith at msmith42@lewisu.edu or Dr. Amanda Harsy at harsyram@lewisu.edu. I understand that if I wish further information regarding my rights as a research subject, I may contact the Lewis University Institutional Review Board Administrator, Dr. William Chura, Associate Provost for Research and Faculty Development, churawi@lewisu.edu, 815-836-597. This study has been approved by the Institutional Review Board at Lewis University.

1 Lab 1: Introduction to Computer Graphics and Matrix Algebra

"Did you know that the first Matrix was designed to be a perfect human world? " -Agent Smith, *The Matrix*

Definition 1.1. *An $m \times n$ **matrix** is a rectangular ordered arrangement (an array) of numbers with m rows and n columns.*

Examples:

$$A = \begin{bmatrix} 2 & 7 & 9 \\ 0 & 1 & 5 \end{bmatrix} = \begin{pmatrix} 2 & 7 & 9 \\ 0 & 1 & 5 \end{pmatrix}, \qquad B = \begin{bmatrix} 1 & 5 \\ 4 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 \\ 20 \\ 3 \\ 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix}.$$

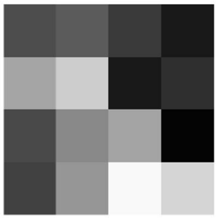
1. What are the dimensions of the matrices above?

1.1 Connection to Graphics

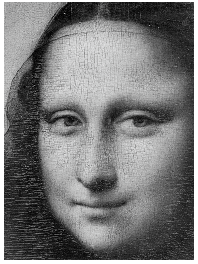
Grayscale images are stored in an ordered array or matrix. Consider

$$A = \begin{bmatrix} 78 & 92 & 59 & 26 \\ 166 & 206 & 26 & 48 \\ 74 & 138 & 165 & 3 \\ 66 & 151 & 250 & 214 \end{bmatrix}.$$

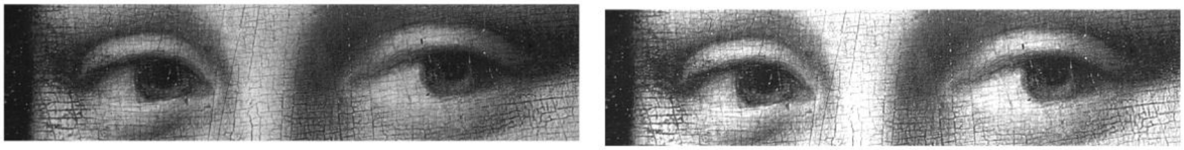
Each element corresponds to a grayscale pixel. Grayscale values vary from 0 (black) to 255 (white). The image represented by matrix A's upper lefthand corner pixel has a grayscale of 78. The matrix A corresponds to the image:



We can use scalar multiplication to lighten or dim an image. Consider the grayscale image of the Mona Lisa.



We can lighten her eyes by multiplying the matrix of pixels for the Mona Lisa by 1.4.



We can also invert images. If we want a white pixel (255) to become black (0), and black (0) to become white. We want to the points (255,0), (0,255) which correspond to the line $y = -x + 255$. We can turn this into a matrix equation. If we let M be the matrix of the Mona Lisa and B be the matrix which has every entry as 255, then our new matrix $G = -M + B$ which will give us:

2. If we wanted to invert the matrix for $A = \begin{bmatrix} 78 & 92 & 59 & 26 \\ 166 & 206 & 26 & 48 \\ 74 & 138 & 165 & 3 \\ 66 & 151 & 250 & 214 \end{bmatrix}$, what would be our resulting matrix?

1.2 Introduction to Octave

We are now going to introduce a multi-paradigm numerical computing environment and proprietary programming language called MatLab which will be very useful in this course. MatLab can be used in our computer labs, but a free version of MatLab, called Octave, can be used online.
Instructions for using octave-online.net

- Open octave-online.net in a web browser. Close (or read and close) the introduction splash window.
- Click the three bars in the upper right of the screen and sign in.
- The rightmost window is the command prompt terminal. You will type your commands here. You also have the option of creating your own scripts and functions to save and run.

3. Experiment with the Command Window: Type 1+1 and hit enter. Type 10*7*11. You can create the matrix $A = \begin{bmatrix} 2 & 7 & 9 \\ 0 & 1 & 5 \end{bmatrix}$ by typing [2 7 9; 0 1 5]. You can define the matrix by typing $A =$ [2 7 9; 0 1 5]. What is the syntax for defining a matrix $B = \begin{bmatrix} 1 & 5 \\ 4 & 1 \end{bmatrix}$?

4. We are now going to experiment with inverting pictures. Go to Blackboard and download the Lab 1 matlab file. **For the lab.m file, you will have to click save link as (and then save the file as a .m file** Use the picture provided or add your own picture. (If using your own picture, you will need to rename the image in the octave code and may need to resize your picture -make it smaller. Resizing a picture can be done several ways -one easy way is to open your photo up in paint and resize it there.).
a) Change the numbers for the following (remember to hit save before you hit run):

redSlope = 1;	greenSlope = -.75;	blueSlope = -.50;
redIntercept = -80;	greenIntercept = 150;	blueIntercept = 200;

b) Upload your own image if you haven't already or change the settings to change the colors from part a. Save or snip a picture of the inverted image you made! Post your "inverted photo" in the Blackboard discussion board for Applied Linear Algebra under the discussion Lab 1. Y

Note One issue you may have with the code is the following: If your browser is opening the jpg and mfile for the Inversion problem in another tab, you can right click on the file (before you download it form Bb) and click on Save Link as. Use this to save it to your computer and then you can drag it to the Octave online or use the Upload button to the right under "file" which you can upload the mfile AND the jpg.

5. **Bonus Question:** From *The Matrix:* Neo: "What is the Matrix?"
a) Trinity: "The answer is out there,... it's looking for you, and it will find you if you want it to."
b) Morpheus: "The Matrix is everywhere. It is all around us. Even now, in this very room... It is the world that has been pulled over your eyes to blind you from the truth."
c) Trinity: "The Matrix isn't real."
d) Morpheus: "The Matrix is a system, Neo. That system is our enemy."
e) Morpheus: "Control. The Matrix is a computer-generated dream world built to keep us under control in order to change a human being into this."
f) Morpheus:"Unfortunately, no one can be told what the Matrix is. You have to see it for

yourself."
g) None of the above, instead it is .... (fill in your answer below)

1.3 Matrices Algebra

1.4 Addition

In this course, we will be using matrices a lot and like many other mathematical objects, we like to be able to combine them. The first way we ever learned to combine numbers was using addition. Matrix addition is so easy it could almost become an afterthought[1]:

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & -1 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 4 & 8 \\ -5 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1+0 & 2+4 & 3+8 \\ 6+(-5) & -1+1 & 4+2 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 11 \\ 1 & 0 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & -1 & 4 \end{bmatrix} - \begin{bmatrix} 0 & 4 & 8 \\ -5 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1-0 & 2-4 & 3-8 \\ 6-(-5) & -1-1 & 4-2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & -5 \\ 11 & -2 & 2 \end{bmatrix}$$

1.5 Multiplication

Being able to add matrices encourages us to consider multiplication of matrices. Please watch my echopen lecture on Matrix Multiplication which is located in the echopen lecture Bb folder.If you are still having trouble understanding Matrix Multiplication, watch these videos: https://www.youtube.com/watch?v=aKhhYguY0DQ and https://www.youtube.com/watch?v=OAh573i_qn8

For each of the exercises below, use the matrices

$$P = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 3 & -1 \\ 2 & 2 & 0 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 2 \\ 1 & 2 & 0 \\ 3 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 0 & -1 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 2 & 3 & 4 \\ 1 & -1 & -3 & -1 \\ -1 & -2 & 0 & 1 \end{pmatrix},$$

$$\text{and} \quad S = \begin{pmatrix} 0 & 0 & -1 & 0 & 2 & 1 \\ -3 & -1 & -2 & 1 & 2 & 0 \\ 2 & -1 & 0 & 1 & 2 & -1 \end{pmatrix}$$

[1] If you still want some guide with this, watch the Kahn Academy Video on Adding and Subtracting Matrices: https://www.youtube.com/watch?v=WR9qCSXJlyY

6. Compute the following sums of matrices. If the addition is not defined, state how you know.

(a) $P+Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 3 & -1 \\ 2 & 2 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 2 \\ 1 & 2 & 0 \\ 3 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 0 & -1 \end{pmatrix} =$

(b) $R+P = \begin{pmatrix} 0 & 2 & 3 & 4 \\ 1 & -1 & -3 & -1 \\ -1 & -2 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 3 & -1 \\ 2 & 2 & 0 & 0 \end{pmatrix} =$

(c) $P+R = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 3 & -1 \\ 2 & 2 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 3 & 4 \\ 1 & -1 & -3 & -1 \\ -1 & -2 & 0 & 1 \end{pmatrix} =$

(d) $Q+S = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 2 \\ 1 & 2 & 0 \\ 3 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 0 & -1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & -1 & 0 & 2 & 1 \\ -3 & -1 & -2 & 1 & 2 & 0 \\ 2 & -1 & 0 & 1 & 2 & -1 \end{pmatrix} =$

(e) What must be true about matrices in order to add them?

(f) What addition property do you notice about matrix addition?

7. Compute $3P$ and $2Q$. If the scalar product is not defined, state how you know.

(a) $3P = 3 \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 3 & -1 \\ 2 & 2 & 0 & 0 \end{pmatrix} =$

(b) $2Q = 2 \cdot \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 2 \\ 1 & 2 & 0 \\ 3 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 0 & -1 \end{pmatrix} =$

(c) Will scalar multiplication ever not be defined for a given $m \times n$ matrix?

8. Let $A = \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix}$, $B = \begin{bmatrix} -2 & -2 \\ 3 & -5 \\ -4 & 5 \end{bmatrix}$, $C = \begin{bmatrix} 1 & -1 & 0 \\ 2 & 0 & -3 \end{bmatrix}$, $D = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$. Compute the following matrix products. If the matrix product is not defined, state how you know.

(a) $A \cdot B = \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} \cdot \begin{bmatrix} -2 & -2 \\ 3 & -5 \\ -4 & 5 \end{bmatrix} =$

(b) $B \cdot A = \begin{bmatrix} -2 & -2 \\ 3 & -5 \\ -4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} =$

(c) $A \cdot C = \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ 2 & 0 & -3 \end{bmatrix} =$

(d) $D \cdot A = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix} \cdot \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} =$

(e) $A \cdot D = \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix} =$

(f) $A \cdot B \cdot C = \begin{bmatrix} 4 & -4 & 5 \\ 4 & -4 & -3 \end{bmatrix} \cdot \begin{bmatrix} -2 & -2 \\ 3 & -5 \\ -4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ 2 & 0 & -3 \end{bmatrix} =$

Hint: use a previous answer.

9. If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: `http://bit.ly/linearalgebralabs2020`

Bonus Questions on next page!

1.6 Bonus Application: Convex Combinations

Now that we know about vectors and how images can be represented by vectors, we would like to show a neat application of vectors in computer graphics. The following questions are bonus and are not required for this first lab. First listen to the echopen lecture on Convex Combinations posted in the Lab 1 section of Blackboard. Then look at the slides posted in the same section in Bb. These are slides we have truncated from freely shared materials from *Coding the Matrix*'s website: `http://codingthematrix.com/` and Tim Chatier's (The author of *When Life is Linear*) Website `http://academics.davidson.edu/math/chartier/` . We are using their slides because they have nice visualizations of these applications.

6. Calculate the convex combination of $a \begin{bmatrix} 6 \\ -2 \end{bmatrix} + b \begin{bmatrix} -4 \\ 8 \end{bmatrix}$
   Remember $b = 1 - a$ for convex combinations.

   (a) for $a = 0$

   (b) for $a = .25$

   (c) for $a = .5$

   (d) for $a = 1$

7. What two applications of convex combinations did the slides demonstrate? Can you think of any other possible uses for convex combinations?

2 Lab 2: Radiography and Tomography Part 1

2.1 Pre- Lab 2 (Rad/Tom Lab Part 1 Introduction)

Definition 2.1. *An **image** is a finite ordered list of real values with an associated geometric array description.*

Examples:

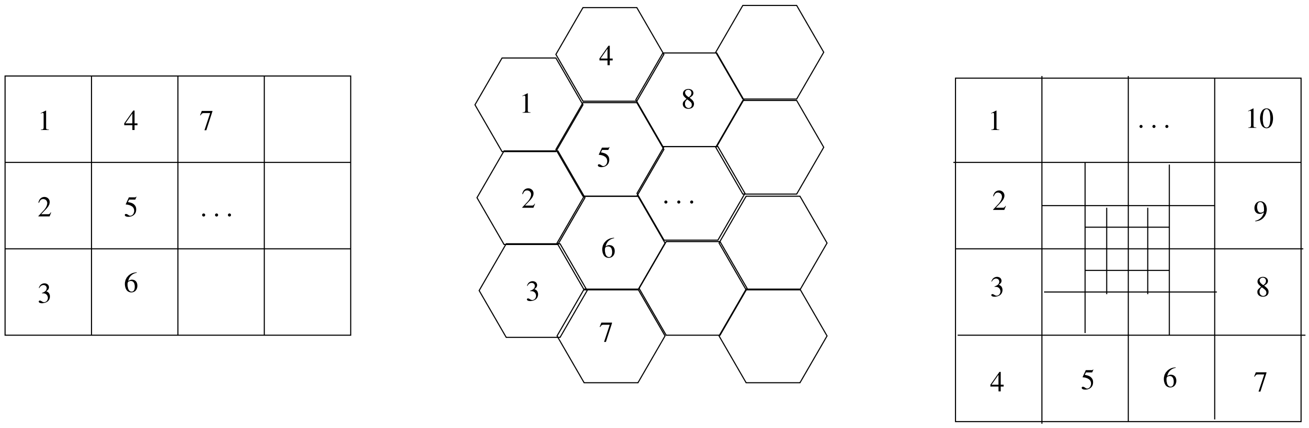


Figure 1: Examples of image arrays. Numbers indicate example pixel ordering.

Addition: We can add 2 images together to get another image!

Definition 2.2. ***The sum of two images*** *is the image that results by adding corresponding values of the ordered list.*

Definition 2.3. ***The sum of two images*** *is the image that results by pixel-wise addition.*

Definition 2.4. *Given two images x and y with (ordered) intensity values $(x_1, x_2, ..., x_n)$ and $(y_1, y_2, ..., y_n)$, respectively, the sum, written $z = x + y$ is the image with intensity values $z_i = x_i + y_i$ for all $i \in \{1, 2, ..., n\}$.*

Scalar Multiplication: We can multiply an image by a scalar (number) to get a new image!

Definition 2.5. ***A scalar times an image*** *is the image which results from multiplication of each of the intensity values (or values in the ordered list) by that scalar.*

Definition 2.6. *A scalar times an image is the image that results by pixel-wise scalar multiplication.*

Definition 2.7. *Given scalar a and image x with (ordered) intensity values $(x_1;, x_2, ..., x_n)$, the scalar product, written $z = ax$ is the image with intensity values $zi = ax_i$ for all $i \in \{1, 2, ..., n\}$.*

2.2 In-Class: Introduction to Radiography and Tomography

Radiography is an imaging technique used to view internal structures of the body that uses electromagnetic radiation (x-rays) by exposure of film specially sensitized to x-rays or gamma rays. **Tomography** is any method that produces images of single tissue planes by computer processing -think "CAT scan" (computerized axial tomography). This scan is divided into a square matrix of pixels.
Why is it good: noninvasive and has high contrast resolution
Source:`http://medical-dictionary.thefreedictionary.com/tomography`

If you want to try this out on your own, go to the course's blackboard site. Click the Labs link and download the "Source Code for Labs." Save it to a folder you will be able to find later. Open Matlab and make sure you are in the correct folder. Note this only works in MatLab.

RadTomExamples(1) shows a sequence of horizontal slices through a brain. The first frame is somewhere near the level of the nose and each subsequent frame (181 in all) is higher than the previous one. At the end of the "movie," you will see a rectangular object which is a stabilizer for keeping the subject's head from moving during the procedure. These slices are examples of desired images of the brain one might want to use to identify or evaluate abnormalities. Gray values are proportional to density: bone shows up bright white with soft tissue exhibiting various shades of gray. We cannot physically slice a person's head to get such information. What we can do is take images such as what happens in a CAT scan and use these to produce the slices.

RadTomExamples(2) shows a sequence of 30 representative CAT scan images that are actually taken in such a radiographic exam. Each view is taken from a different angle around the head in increments of 6 degrees. Each view is perpendicular to the horizontal slices that we just viewed. These radiographs show grayscale values that contain some Poisson instrumentation noise. Longer exposures have less relative noise as long as the detector array can accurately handle the x-ray intensity.

RadTomExamples(3) shows the 30 views that were shown in the CAT scan movie. In an actual CAT scan there are typically hundreds if not over a thousand scans taken in order to produce the brain images that we saw originally. **Question:** Why might we want to take fewer than hundreds of views?

RadTomExamples(4) shows the series of brain slice reconstructions using a relatively simple left-inverse method (this is developed and implemented by students in Lab 5). The reconstruction uses 120 low noise (high exposure) radiographs from unique angles around the head. Notice that the reconstructions are noisier than the radiographs. This noise amplification effect is typical in inverse problems with noisy data.

RadTomExamples(5) shows a comparison of reconstructions using hundreds of very low noise (very high exposure) radiographs and advanced Linear Algebra methods (top) and reconstructions from the 120 low noisy radiographs. The first set of reconstructions is the ideal reconstruction (very close to the original data). The upshot of this comparison is that the left-inverse method used for the second reconstruction (developed and implemented by students in a later lab) is not necessarily the best inversion method when the data is noisy and when we have some prior knowledge of the object (brain) details.

RadTomExamples(6) shows the series of brain slice reconstructions using only the 30 low noise radiographs shown previously. This reconstruction uses a noise-sensitive pseudo-inverse method based on the singular value decomposition of the radiographic transformation (this is developed and implemented by students in later labs).

Question: What are the goals of a researcher in this mathematical field?

RadTomExamples(7) shows a comparison of three reconstructions methods: (top) noise-sensitive pseudo-inverse on hundreds of low-noise radiographs, (middle) left-inverse on 120 low-noise radiographs, (bottom) noise-sensitive pseudo-inverse on 30 low-noise radiographs.

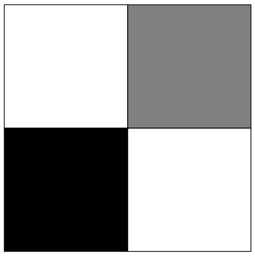
RadTomExamples(8) shows the series of brain slice reconstructions using only the 30 low-noise radiographs shown previously. This reconstruction uses a noise-sensitive pseudo-inverse method and allows radiographic transformation null space contributions that satisfy prior-knowledge constraints.

RadTomExamples(9) shows a comparison of all four reconstruction methods: (top) noise-sensitive pseudo-inverse on hundreds of low-noise radiographs, (second) left-inverse on 120 low-noise radiographs, (third) noise-sensitive pseudo-inverse on 30 low-noise radiographs, (bottom) noise-sensitive pseudo-inverse on 30 low-noise radiographs with transformation null-space enhancements. In the course of the parts of the Radiography and Tomography Labs, we will learn how to perform each of these types of brain reconstructions. Although we will not get to the later parts of the lab, we want to show the the results of these reconstructions in order to demonstrate that more sophisticated reconstructions involving further study of linear algebra are available to deal with noise and sparser data.

2.3 Lab 2: Radiography and Tomography: Part 1

Grayscale Images

A grayscale image can be associated with a set of numbers representing the brightness of each pixel. For example, the 2 × 2 pixel image below can be associated with the corresponding array of numbers, where the black squares represent pixels with intensity 0 and the white squares represent pixels with intensity 16:



~

16	8
0	16

Note that this interpretation gives us a way to think about multiplying an image by a constant and adding two images. For example, suppose we start with the three images (A, B, and C) below.

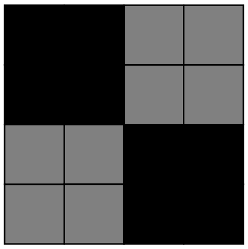


Image A

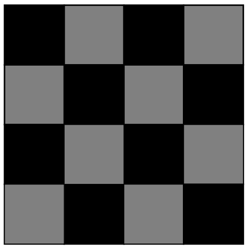


Image B

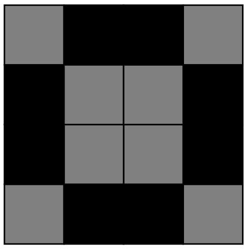


Image C

Then multiplying Image A by 0.5 results in Image 1 below. Note that the maximum intensity is now half what it previously was, so the all pixels have become darker gray (representing their lower intensity). Adding Image 1 to Image C results in Image 2 below; so Image 2 is created by doing arithmetic on Images A and C.

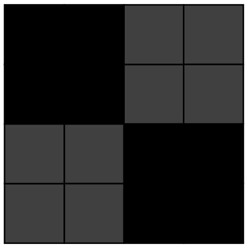


Image 1

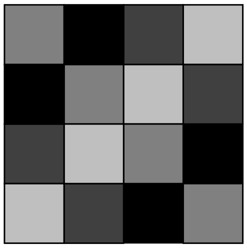


Image 2

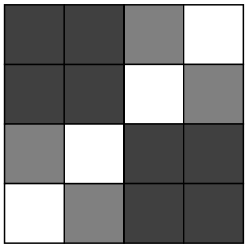


Image 3

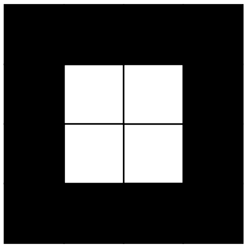


Image 4

Exercise 1: Write Image 3 and Image 4 using arithmetic operations of Images A, B, and C.

Images in Matlab

We’ll now see how to input matrices, convert them into grayscale images, and display them in Matlab (or Octave). If you do not have access to Matlab or Octave through academic or personal computing resources then you may complete this Lab using the online service octave-online.net (instructions at the end of this writeup).

Exercise 2 Input these lines into the command window of Matlab. Note that ending a line with a semicolon suppresses the output. If you want to show the result of a computation, delete the semicolon at the end of its line. Briefly describe what the output of each of these lines of code gives you. Note: In order to display an image, we have to know the minimum and maximum pixel intensities. The imshow function allows the user to specify these values.

```
M_A = [0 0 8 8; 0 0 8 8; 8 8 0 0; 8 8 0 0];
M_B = [0 8 0 8; 8 0 8 0; 0 8 0 8; 8 0 8 0];
M_C = [8 0 0 8; 0 8 8 0; 0 8 8 0; 8 0 0 8];
figure;
subplot(1,3,1), imshow(M_A, [0, 16]),title(’Image A’);
subplot(1,3,2), imshow(M_B, [0, 16]),title(’Image B’);
subplot(1,3,3), imshow(M_C, [0, 16]),title(’Image C’);
```

Exercise 3: Enter the following lines of code one at a time and state what each one does.

```
M_A
M_1 = .5*M_A
M_2 = M_1 + M_C
figure;
subplot(1,2,1), imshow(M_1, [0, 16]), title(’Image 1’);
subplot(1,2,2), imshow(M_2, [0, 16]), title(’Image 2’);
```

Exercise 4 Now write your own lines of code to check your conjectures to get Images 3 and 4. How close are these to Images 3 and 4? Be sure to print out your code and submit it with this lab.

Exercise 5: Please give some thoughts on the following questions.

(a) What happens when pixel intensities in an image exceed the display range as input into the imshow function?

(b) How should we interpret pixel intensities that lie outside our specified range?

(c) What algebraic properties does the set of all images have in common with the set of real numbers?

Survey Feedback Question: If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: http://bit.ly/linearalgebralabs2020

Instructions for using octave-online.net

1. Open octave-online.net in a web browser. Close (or read and close) the introduction splash window.
2. Optional: Click the three bars in the upper right of the screen and sign in. This will be necessary if you complete the next Lab using this service.
3. The rightmost window is the command prompt terminal. You will type your commands here. You also have the option of creating your own scripts and functions to save and run.
4. Now you are ready to complete the assignment!

3 Lab 3: March Mathness: Using Linear Algebra for Predictive Modeling

3.1 Pre-Lab

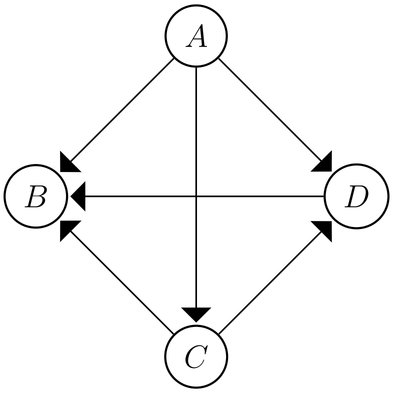
Linear Algebra can be used in sports analytics. In this lab, we will discuss how we can use linear algebra to develop sports ranking systems. The most basic way to ranks teams is through win percentage.
Recall Win Percentage= $\dfrac{\text{\# of games won}}{\text{\# of games played}} = \dfrac{w_i}{t_1}$ for each team i.
Why would basing a team's ranking only by win percentage not necessarily be the best method for sports ranking?

On reason this may not work well is that there is no way to distinguish strength of schedule. Team A may only play and win 1 game and thus has a win percentage of 100% while Team B may play 100 games and win 99 of them which would give them a win percentage of 99%. If we based teams only by win percentage, we would rank team A over team B. Is this fair? Similarly if team C plays and wins all 100 games, they have the same win percentage as team A. Is it fair to give them the same ranking? How can we distinguish between a team that is undefeated with a difficult schedule verses an undefeated team in a weaker conference? It turns out linear algebra can help.

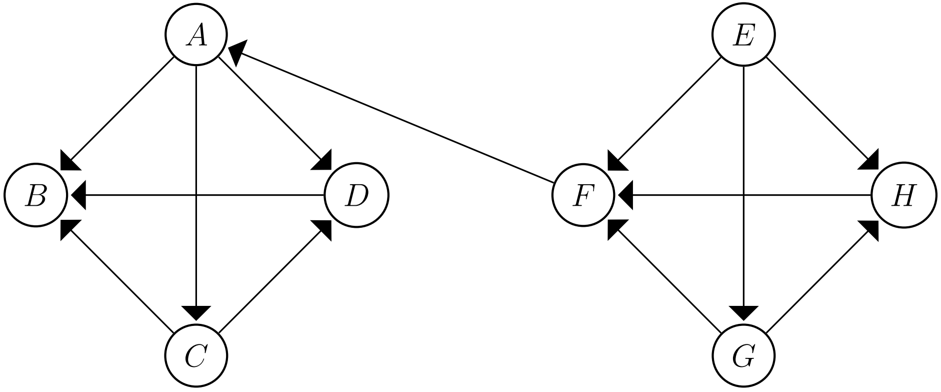
3.1.1 Directed Graphs

Before we discuss two different ranking methods, first we will talk about a way to organize sports data. Suppose we have 4 teams, Team A, B, C, & D. Team A beats B and D, C beats D and B, and Team D beats team B. You can represent this situation using a *directed graph* with the teams making up the vertices of the graph. This graph is sometimes called the "The Poor Sportsmanship Graph", as the winner is rudely pointing at the loser.

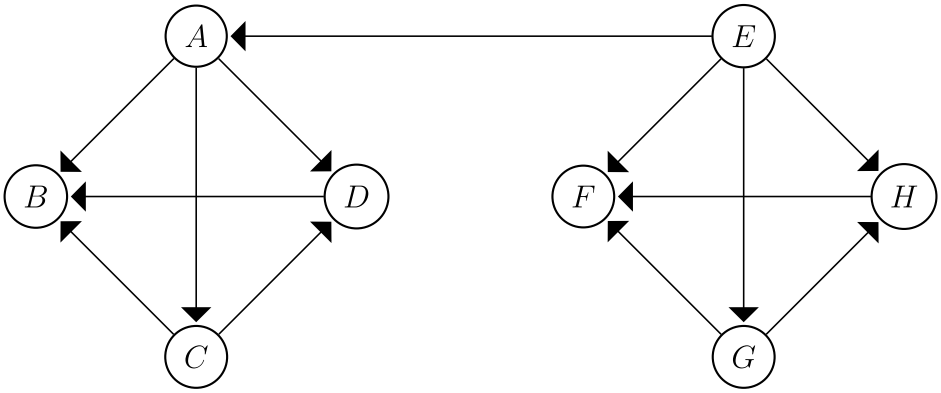
1. How would you rank this system of teams (from first place to last place)?



2. How would you rank this system of teams?



3. How would you rank this system of teams?



4. What challenges come up when trying to create these rankings? What other information would you like to know about these teams and systems?

Our goal, as sports analysts, is to determine not only who is number one, but who is number two, three, four, etc. Sports ranking is similar to Google's task of ranking pages, but is much tougher because the outcome of games require both skill and luck.

3.2 Lab 3: In class portion

3.2.1 The Colley Method

As part of his mathematics PhD dissertation, Dr. Wesley Colley developed a method to rank teams in a way that incorporates strength of schedule which can make it a little more accurate than win percentage. At the start of the season, a team's winning percentage would 0/0 which is *undefined!* The Colley Method says an untried team should have a rating of 50%, 50-50 chance the team will win or lose. Colley also wants to have the average of all rating be roughly 0.5 as well. So we adjust how we calculate our win percentage.
Regular Winning Percentage is calculated by $\dfrac{w_i}{t_i}$, where w_i is the total number of wins for Team i and t_i is the total number of games placed by Team i.
The Colley Winning Percentage is calculated as $\dfrac{w_i+1}{t_i+2}$. Since, after each game, the denominator increases by 1, the losers rating will decreasing and the winners rating will increase after each game.

To set up his system, Colley sets the rating for each team, r_i, equal to its Colley win percentage. That is, $r_i = \dfrac{w_i+1}{t_i+2}$. In order to incorporate strength of schedule, Colley does the following algebraic manipulation (note l_i denotes the total number of losses for Team i):

$$r_i = \frac{w_i+1}{t_i+2} \tag{1}$$

$$(2+t_i)r_i = 1+w_i \tag{2}$$

$$(2+t_i)r_i = 1+\frac{w_i}{2}+\frac{w_i}{2}+\frac{l_i}{2}-\frac{l_i}{2} \tag{3}$$

Note we use the fact that $w_i = \frac{w_i}{2}+\frac{w_i}{2}$ and $\frac{l_i}{2}-\frac{l_i}{2}=0$. Rearranging our equation, we now get:

$$(2+t_i)r_i = 1+\frac{w_i-l_i}{2}+\frac{w_i+l_i}{2} \tag{4}$$

Note that $\dfrac{w_i+l_i}{2}$ is just the total number of games played by team i divided by 2. This is where Colley made an adjustment to his system in order to incorporate strength of schedule. He does this by replacing $\dfrac{w_i+l_i}{2}$ with the sum of the ratings of the teams played by team i. Thus our new Colley system has the following set up. For each Team i, we get the following equation:

$$(2+t_i)r_i = 1+\frac{w_i-l_i}{2}+S, \tag{5}$$

where S is the sum of the ratings of teams played by Team i.

Example 3.1. *If Team A beat Team B, Team C beat Team A, Team C beat team B, and Team B beat Team C, using the Colley Method we would have the following system of equations:*

$$
\begin{aligned}
(2+2)r_A &= 1 + \frac{1-1}{2} + r_B + r_C \\
(2+3)r_B &= 1 + \frac{1-2}{2} + r_A + 2r_C \\
(2+3)r_C &= 1 + \frac{2-1}{2} + r_A + 2r_B
\end{aligned}
$$

Rearranging the system gives us:

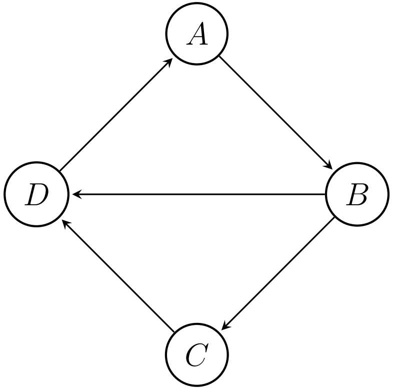
$$
\begin{aligned}
4r_A - r_B - r_C &= 1 \\
-r_A + 5r_B - 2r_C &= 0.5 \\
-r_A - 2r_B + 5r_C &= 1.5
\end{aligned}
$$

This gives us a nice symmetric matrix system, $Cr = b$:

$$
\begin{bmatrix} 4 & -1 & -1 \\ -1 & 5 & -2 \\ -1 & -2 & 5 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1.5 \end{bmatrix} \tag{6}
$$

Solving this system gives us $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.42857 \\ 0.57143 \end{bmatrix}$. We order the team rankings by the value of the ratings with the largest rating corresponding with the top ranked team. Putting the ratings in order from largest to smallest, we get the following ranking: Team C first, Team A second, and Team B third.

(a) Use the Colley method to come up with a system of equations for the following season.



(b) Use the Colley method to come up with a rating for each team, then rank them.

(c) What were some of the strengths and weaknesses of this method?

3.2.2 Adding Weights to Colley Systems

We can weight this rating system to give us a more personalized rating. The key now is that we count total weighted games.

Example 3.2. *Suppose we have the teams A, B, and C. During the early part of the season, Team C beats A. During the middle of the season, Team B beats C and Team A beats B. Near the end of the season, Team A beats B. Create and determine the weighted Colley rating by weighting the time of the game. Use the weighting where the first part of the season is weighted by 0, the middle by 1, and the end of the season by 1.5.*

We will start by drawing the season.

We will now need to write our Colley equations, using the weightings for the various wins and losses.

Setting up the Colley system then gives us:

$$\begin{bmatrix} 4.5 & -2.5 & 0 \\ -2.5 & 5.5 & -1 \\ 0 & -1 & 3 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 2.25 \\ 0.25 \\ 0.5 \end{bmatrix} \tag{7}$$

Solving this system gives us $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 0.745 \\ 0.441 \\ 0.314 \end{bmatrix}$ which gives us the ranking Team A, Team B, and Team C.

(a) Now use a weighted the Colley Method which ranks games won (loss) early in the season as 0.5 wins (losses), games won in the mid-season ranked as 1 win (loss) , and games won late in the season as 2 wins (losses) to rank the following system:
During the Early Part of the Season:
Team B beats C once. Team D beats E once and C once. Team E beats B once.
During the Middle Part of the Season:
Team A beats C once. Team D beats B once. Team E beats C once
During the Later Part of the Season:
Team A beats E once and C once. Team E beats B once.

Draw a directed graph of this the season. **Note that these weightings are different than the example.**

(b) Write the system of equations for this season using the weighted Colley method.

(c) Solve the system to find the ratings for each team, and use those ratings to rank them.

3.2.3 A Way to Use Scores - The Massey System of Ranking

Suppose we want to use a system of rating that incorporates the scores of the games. On the other hand, Kenneth Massey's method of sports rating creates a system of equations which is based on the score of the games played. Massey originally created this method for ranking college football teams. Using the Massey Method, for each game, the difference in the ratings correlates to the point differential. This is represented as a system: $r_i - r_j = b_k$ where r_i is the rating for team i and r_j is the rating for team j and b_k is the score differential for game k.

Example 3.3. *Suppose I have 3 teams, A, B, and C. Suppose Team A beat Team B by 7, Team C beat Team A by 1, Team C beat team B by 1, and Team B beat Team C by 10, we would get the following system of equations:*

$$\begin{aligned} r_A - r_B &= 7 \\ r_C - r_A &= 1 \\ r_C - r_B &= 1 \\ r_B - r_C &= 10 \end{aligned}$$

Which will give us the following matrix system, $Xr = b$:

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{8}$$

Note that each row of our system of equations representing a particular game from the data set. Our columns represent the different teams being ranked. This usually means in practice that our system has more rows than columns.

(a) **We have a problem.** This system is inconsistent. That is, there are __________ solutions. What method can we use to "approximate" a solution?

(b) Use least squares to solve the following inconsistent solution.

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix}$$

(c) We now have a new problem: This system has infinitely many solutions. To fix this, we replace the last row of our new matrix with all 1's and the last entry in the vector with 0. We won't go into the technical details for why we do this, but this adjustment forces the ratings to sum to 0.
Do this and solve the new system.

(d) The Colley and Massey Rating Methods are good, but what are some other things we may want to consider when making our ranking?

Example 3.4. ***Easy Adjustment for blowouts:*** *Suppose you want to balance out any blow-outs from the games. This is quite easy with the Massey Method. Suppose you want to consider any game in which one team beats the other team by 15 points or more a blow out and count them all the same. We would just adjust our Massey System by capping the size of the win.*

(a) Now use the Massey Method **with a 10 max point differential** for the following system:
During the Early Part of the Season: Team B beats C by 8 points. Team D beats E by 5 points.
During the Middle Part of the Season:
Team A beats D by 10 points. Team E beats C by 5 points.
During the Later Part of the Season:
Team A beats E by 4. Team E beats B by 20 points.

Draw a directed graph of this the season. **Note that this question asks you to cap the point differential.**

(b) Write the system of equations for this season using the Massey method.

(c) Solve the system (using least squares) to find the ratings for each team, and use those ratings to rank them.

3.2.4 Adding Weights to Massey Systems

We can also weight the Massey rating system to give us a more personalized rating. The key now is that we count total weighted games **and** weighted scores.

Example 3.5. *Suppose Team A beat Team B by 7 mid season, Team C beat Team A by 1 early in the season, Team C beat Team B by 1 mid-season, and Team B beat Team C by 10 late in the season. Suppose we want to weight early games by* 0.5*, mid games by* 1*, and late games by* 5*.*

Solution: Using a normal, unweighted Massey system with a 10 point cap with the columns representing teams A, B, and C respectively and the rows representing games 1, 2, 3, and 4, we would have the following system:

$$Xr = b \Rightarrow \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{9}$$

When we use the Massey system, we can actually use something called a *Weight Matrix*,a diagonal matrix in which each diagonal entry (i, i) is equal to the weight for that given game/ We multiply this matrix to both sides of Equation 9 (the equation above) by our weight matrix: $WXr = Wb$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{10}$$

Which gives us

$$\begin{bmatrix} 1 & -1 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & -1 & 1 \\ 0 & 5 & -5 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 7 \\ 0.5 \\ 1 \\ 50 \end{bmatrix} \tag{11}$$

Next we set up the least squares solution:

$$\begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -1 & 0 & -1 & 5 \\ 0 & 0.5 & 1 & -5 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & -1 & 1 \\ 0 & 5 & -5 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -1 & 0 & -1 & 5 \\ 0 & 0.5 & 1 & -5 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 0.5 \\ 1 \\ 50 \end{bmatrix} \tag{12}$$

This gives us

$$\begin{bmatrix} 1.25 & -1 & -0.25 \\ -1 & 27 & -26 \\ -0.25 & -26 & 26.25 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6.75 \\ 242 \\ -248.75 \end{bmatrix} \tag{13}$$

After replacing the last row or our matrix with 1's and 0, we get:

$$\begin{bmatrix} 1.25 & -1 & -0.25 \\ -1 & 27 & -26 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6.75 \\ 242 \\ 0 \end{bmatrix} \tag{14}$$

We then solve this new weighted system following the same methods as an unweighted Massey model. Solving this gives us: $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 5.489 \\ 1.977 \\ -7.466 \end{bmatrix}$ Which gives us the ranking: Team A, Team B, and Team C.

(a) **Bonus:** Use a weighted Massey Method which ranks games won (loss) early in the season as 0.5 wins (losses), games won in the mid-season ranked as 1 win (loss) , and games won late in the season as 1.5 wins (losses) to rank the system from Problems 15 - 17, except **without** the max point differential.

(b) If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: `http://bit.ly/linearalgebralabs2020`

End of Lab. There are extra examples of Massey and Colley Systems are on the next page.

3.2.5 Extra Examples of Massey and Colley

Example 3.6. *Suppose Team A beats Team B by 5, Team C beat Team A by 1, Team C beat Team B by 1, and Team B beat Team C by 10. Rank these teams using the unweighted Massey Method.*

Solution: First we set up the following system of equations:

$$\begin{aligned} r_A - r_B &= 7 \\ r_C - r_A &= 1 \\ r_C - r_B &= 1 \\ r_B - r_C &= 10 \end{aligned}$$

Which will give us the following matrix system, $Xr = b$:

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{15}$$

Note that each row of our system of equations representing a particular game from the data set. Our columns represent the different teams being ranked. Unfortunately, this system, $Xr = b$ will usually be inconsistent (rarely does a team beat a team twice by the same score and note in this example that is true). Therefore the Massey Method uses least squares to solve its system. To find a least squares solution we multiply both sides of our original equation $Xr = b$, by the transpose of X. The transpose matrix, denoted X^T, is a matrix whose rows are the columns of the original matrix X. Thus we now have the system $X^T X r = X^T b$:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -2 \\ -1 & -2 & 3 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ -8 \end{bmatrix} \tag{17}$$

Remember sometimes there are multiple least squares solutions. As is the case for this example. To fix this, Massey needed to force his system to have one unique solution. He did this by replacing one of the rows with an equation which is not in the span of the other rows of the system (that is, not a linear combination of the other rows). He could replace any row, but for consistency, Massey replaces the last row of the system with $\sum_{i=1}^{m} r_i = 0$. This ultimately means that the sum of the ratings of the teams should add up to 0. After this adjustment, we can now solve the system.

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -2 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 0 \end{bmatrix} \tag{18}$$

Solving this system gives us

$$\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix}$$

We order the teams ranking by the value of the ratings with the largest rating corresponding with the top ranked team. Thus in the example above, we would rank team A first, Team B second, and Team C third. Note that if we ranked the teams by win percentage, we would have the ranking Team C, Team A, and Team B.

Example 3.7. *Now suppose we want to set up the same weighted system as above using the Massey Method. Suppose Team A beat Team B by 5 mid season, Team C beat Team A by 1 early in the season, Team C beat team B by 1 mid-season, and Team B beat Team C by 10 late in the season. Suppose we want to weight early games by* 0.5*, mid games by* 1*, and late games by* 3*. Find a ranking for the teams using a weighted Massey System.*

Solution: Notice when we incorporate weights into the Massey system, it is a little easier than with the Colley Method since we can use a weight matrix. This weighted matrix is a diagonal matrix in which each diagonal entry (i, i) is equal to the weight for that given game. Thus we can start with the same original matrix system for Massey, but multiply both sides of the equation by our weight matrix, $WXr = Wb$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 1 \\ 1 \\ 10 \end{bmatrix} \tag{19}$$

Which gives us

$$\begin{bmatrix} 1 & -1 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & -1 & 1 \\ 0 & 3 & -3 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 7 \\ 0.5 \\ 1 \\ 30 \end{bmatrix} \tag{20}$$

Next we set up the least squares solution:

$$\begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -1 & 0 & -1 & 3 \\ 0 & 0.5 & 1 & -3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & -1 & 1 \\ 0 & 3 & -3 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -1 & 0 & -1 & 3 \\ 0 & 0.5 & 1 & -3 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 0.5 \\ 1 \\ 30 \end{bmatrix} \tag{21}$$

This gives us

$$\begin{bmatrix} 1.25 & -1 & -0.25 \\ -1 & 11 & -10 \\ -.25 & -10 & 10.25 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6.75 \\ 82 \\ -88.75 \end{bmatrix} \tag{22}$$

After replacing the last row or our matrix with 1's and 0, we get:

$$\begin{bmatrix} 1.25 & -1 & -0.25 \\ -1 & 11 & -10 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 6.75 \\ 82 \\ 0 \end{bmatrix} \tag{23}$$

Solving this gives us: $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 5.314 \\ 1.627 \\ -6.941 \end{bmatrix}$ Which gives us the ranking: Team A, Team B, Team C.

Example 3.8. *Suppose we have teams A, B, and C where Team C beats A by 17 points during the first part of the season. During the middle of the season, Team B beats C by 10 points and Team A beats B by 10 points. And suppose near the end of the season, Team A beats B by 5 points. Find the rating using a weighted Massey Method in which we use the weights 0 for an early game, 0.5 for a mid-season game and 1 for a game played late in the season and we use a 10 point cap.*

Solution: Using a normal, unweighted Massey system with a 10 point cap with the columns representing teams A, B, and C respectively and the rows representing games 1, 2, 3, and 4, we would have the following system:

$$\begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 5 \end{bmatrix} \tag{24}$$

When we use the Massey system, we can actually use something called a *Weight Matrix* and multiply it to both sides of Equation 24 (the equation above). Thus we get:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \\ 10 \\ 5 \end{bmatrix} \tag{25}$$

We now get the weighted Massey system $\tilde{M}\mathbf{r} = \tilde{b}$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0 \\ 1 & -1 & 0 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 0 \\ 5 \\ 5 \\ 5 \end{bmatrix} \tag{26}$$

Next we find the Least Squares Solution to this system :$\tilde{M}^T\tilde{M}\mathbf{r} = \tilde{M}^T\tilde{b}$:

$$\begin{bmatrix} 0 & 0 & 0.5 & 1 \\ 0 & 0.5 & -0.5 & -1 \\ 0 & -0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0 \\ 1 & -1 & 0 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 0 & 0 & 0.5 & 1 \\ 0 & 0.5 & -0.5 & -1 \\ 0 & -0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \\ 5 \\ 5 \end{bmatrix} \tag{27}$$

Completing the matrix multiplication gives us:

$$\begin{bmatrix} 1.25 & -1.25 & 0 \\ -1.25 & 1.5 & -0.25 \\ 0 & -0.25 & 0.25 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 7.5 \\ -5 \\ -2.5 \end{bmatrix} \tag{28}$$

Finally we replace the last row of $\tilde{M}^T\tilde{M}$ with 1's and the last row of $\tilde{M}^T\tilde{b}$ with 0 to get:

$$\begin{bmatrix} 1.25 & -1.25 & 0 \\ -1.25 & 1.5 & -0.25 \\ 1 & 1 & 1 \end{bmatrix} \mathbf{r} = \begin{bmatrix} 7.5 \\ -5 \\ 0 \end{bmatrix} \tag{29}$$

Solving this gives us $\mathbf{r} = \begin{bmatrix} 7.5 \\ 1.3 \\ -8.6 \end{bmatrix}$ which gives us the ranking: A, B, C.

Example 3.9. *Suppose Team A beats Team B by 5, Team C beat Team A by 1, Team C beat team B by 1, and Team B beat Team C by 10. Use the unweighted Colley Method to rank the following teams.*

Solution: Using the Colley Method, we would set up the following system of equations:

$$\begin{aligned} (2+2)r_A &= 1 + \frac{1-1}{2} + r_B + r_C \\ (2+3)r_B &= 1 + \frac{1-2}{2} + r_A + 2r_C \\ (2+3)r_C &= 1 + \frac{2-1}{2} + r_A + 2r_B \end{aligned}$$

Rearranging the system gives us:

$$\begin{aligned} 4r_A - r_B - r_C &= 1 \\ -r_A + 5r_B - 2r_C &= \frac{1}{2} \\ -r_A - 2r_B + 5r_C &= \frac{3}{2} \end{aligned}$$

Solving this nice symmetric system gives us the following matrix system, $Cr = b$:

$$\begin{bmatrix} 4 & -1 & -1 \\ -1 & 5 & -2 \\ -1 & -2 & 5 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{3}{2} \end{bmatrix} \tag{30}$$

Solving this system gives us $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.42857 \\ 0.57143 \end{bmatrix}$. Putting the ratings in order from largest to smallest, we get the following ranking: Team C first, Team A second, and Team B third.

Example 3.10. *Suppose we have the same system as in Example 3.9: Team A beat Team B by 5, Team C beat Team A by 1, Team C beat team B by 1, and Team B beat Team C by 10. But now suppose we wanted to weight the time in the season in which these teams played. Suppose Team A beat Team B by 5 mid season, Team C beat Team A by 1 early in the season, Team C beat team B by 1 mid-season, and Team B beat Team C by 10 late in the season. Suppose we want to weight early games by* 0.5*, mid games by* 1*, and late games by* 3*. Find the ranking for these teams using this weighted Colley Method.*

Solution: We first set up the weighted Colley system:

$$\begin{aligned} (2+1.5)r_A &= 1 + \frac{1-.5}{2} + r_B + 0.5r_C \\ (2+5)r_B &= 1 + \frac{3-2}{2} + r_A + 4r_C \\ (2+4.5)r_C &= 1 + \frac{1.5-3}{2} + 0.5r_A + 4r_B \end{aligned}$$

(Note: Team A has played a total of 1 early game and 1 mid game which means a total of 1.5 weighted games $(1 \cdot 0.5 + 1 \cdot 1)$ and has a total of 1 weighted wins and 0.5 weighted losses. Team B has played two mid games and 1 late game which means Team B has played a total of 5 weighted games $(1 \cdot 3 + 2 \cdot 1)$ and has 3 weighted wins and 2 weighted losses. etc.) Rearranging the system gives us:

$$\begin{aligned} 3.5r_A - r_B - 0.5r_C &= \frac{5}{4} \\ -r_A + 7r_B - 4r_C &= \frac{3}{2} \\ -0.5r_A - 4r_B + 6.5r_C &= \frac{1}{4} \end{aligned}$$

Solving this system gives us the following matrix system, $Cr = b$:

$$\begin{bmatrix} 3.5 & -1 & -0.5 \\ -1 & 7 & -4 \\ -0.5 & -4 & 6.5 \end{bmatrix} \cdot \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} \frac{5}{4} \\ \frac{3}{2} \\ \frac{1}{4} \end{bmatrix} \tag{31}$$

Solving this system gives us $\begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix} = \begin{bmatrix} 0.56593 \\ 0.52747 \\ 0.40659 \end{bmatrix}$. Which gives us the ranking Team A, Team B, then Team C. Note that this is different from what we got when we did the basic unweighted Colley Method in Example 3.9.

4 Lab 4: Radiography and Tomography Part 2

4.1 Pre-Lab 4

Read the following pages entitled "Transmission Radiography and Tomography: A Simplified Overview" as a prerequisite to Lab 2. You may not understand everything from this handout, but I would like you to look for the linear algebra in the methods and try to get a general overview of the Radiography Process. **You do not need to fully understand anything else besides the linear algebra concepts that come up.** I have included the following question for you to answer to help guide you in your reading (below) along with the goals and summary of terminology of what you should get out of the reading after the reading section.

4.1.1 Goals from the Reading Assignment:

- To understand a unified and simplified physical model of taking an x-ray radiograph of an object.
- To recognize and appreciate that, sometimes, complex phenomena can be reasonably modeled with straightforward mathematics.
- To recognize and appreciate that mathematical assumptions should be backed by physical reality in order to preserve model integrity. This is justification!
- To be exposed to the notation used in the Lab 2.
- To understand the matrix equation representation of the radiographic process.

4.1.2 Questions from the Reading Assignment:

(a) Given $T_{kj} = 0.42$, what does this value mean?

(b) What is accomplished in the matrix multiply when multiplying an object vector by the k^{th} row of T?[2]

[2] More questions on back!

(c) Explain how you change the radiographic operator when changing from one view to many.

(d) Why do we use pixel values β_k where β_k is defined as $\beta_k \equiv -\alpha \ln(\frac{p_k}{p_0})$ instead of the expected photon count p_k?

Transmission Radiography and Tomography

A Simplified Overview

This material provides a brief overview of radiographic principles for the model in Lab #2 of the Radiography and Tomography Linear Algebra Modules. The goal is to develop the basic discrete radiographic operator for axial tomography of the human body. To accomplish this goal, it is **not** necessary to completely understand the details of the physics and engineering. We wish to arrive at a mathematical formulation descriptive of the radiographic process and establish a standard scenario description with notation.

What is Radiography?

Transmission radiography and tomography are familiar and common processes in today's world, especially in medicine and non-destructive testing in industry. Some examples include

- Single-view X-ray radiography is used routinely to view inside the human body; for example, bone fracture assessment, mammography, and angiographic procedures.
- Multiple-view X-ray radiography is realized in computerized axial tomography (CAT) scans used to provide 3D images of body tissues.
- Neutron and X-ray imaging is used in industry to quantify manufactured part assemblies or defects which cannot be visually inspected.

Transmission Radiography is the process of measuring and recording changes in a high-energy particle beam (X-rays, protons, neutrons, etc.) resulting from passage through an object of interest.

Tomography is the process of inferring properties of an unknown object by interpreting radiographs of the object.

X-rays, just like visible light, are photons or electromagnetic radiation, but at much higher energies and outside of the range of our vision. Because of the wavelength of typical X-rays (on the order of a nanometer), they readily interact with objects of similar size such as individual molecules or atoms. This property makes them particularly useful in transmission imaging. Figure 1 is a cartoon of a typical x-ray radiographic experiment or procedure. An x-ray beam is produced with known energy and geometric characteristics. The beam is aimed at a region of interest. The photons interact with matter in the region of interest, changing the intensity, energy and geometry of the beam. A detector measures the pattern

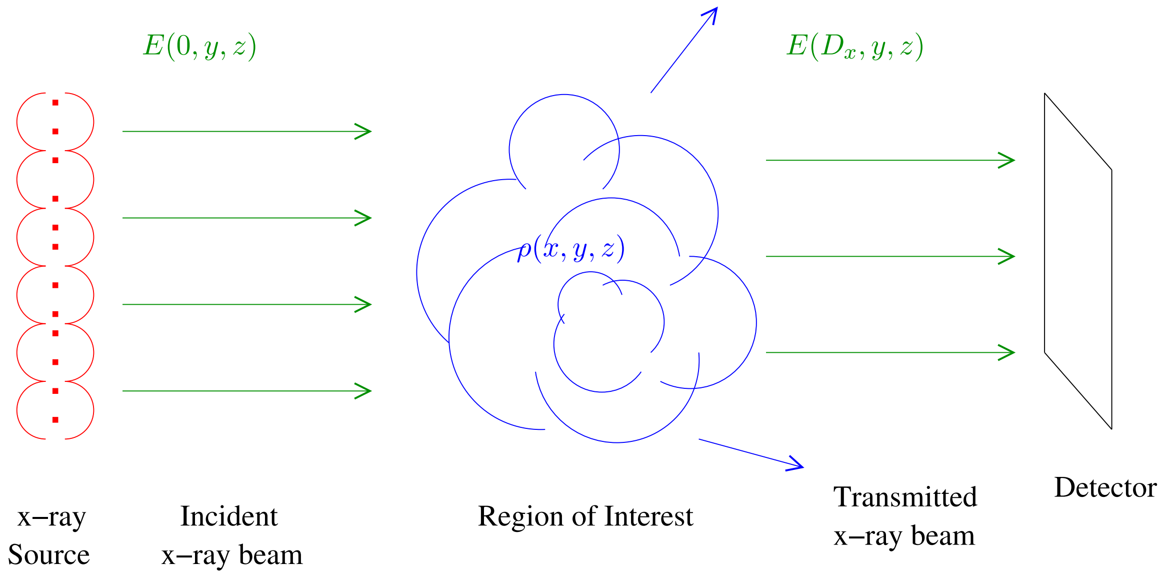


Figure 1: Typical radiographic experiment.

(and possibly the distribution) of incident energy. The detection data, when compared to the incident beam characteristics, contains the known signature of the region of interest. We consider the mathematics and some of the physics involved in each step with the goal of modeling a radiographic transformation appropriate for mixed soft and hard tissue axial tomography of the human body.

The Incident X-ray Beam

We begin with an x-ray beam in which the x-ray photons all travel parallel to each other in the beam direction, which we take to be the positive x-direction. Additionally we assume that the beam is of short time duration, the photons being clustered in a short pulse instead of being continuously produced. A beam with these geometric characteristics is usually not directly achievable through typical x-ray sources (see supplementary material for some discussion on x-ray sources). However, such a beam can be approximated readily through so-called collimation techniques which physically limit the incident x-rays to a subset that compose a planar (neither convergent nor divergent) beam.

While not entirely necessary for the present formulation, we consider a monochromatic x-ray source. This means that every x-ray photon produced by the source has exactly the same "color." The term "monochromatic" comes from the visible light analog in which, for example, a laser pointer may produce photons of only one color, red. The energy of a single photon is proportional to its frequency ν, or inversely proportional to its wavelength λ. The frequency, or wavelength, determine the color, in exact analogy with visible light. In particular, the energy of a single photon is $h\nu$ where the constant of proportionality h is known as Planck's constant.

The intensity (or brightness), $E(x,y,z)$, of a beam is proportional to the photon density. The intensity of the beam just as it enters the region of interest at $x = 0$ is assumed to be

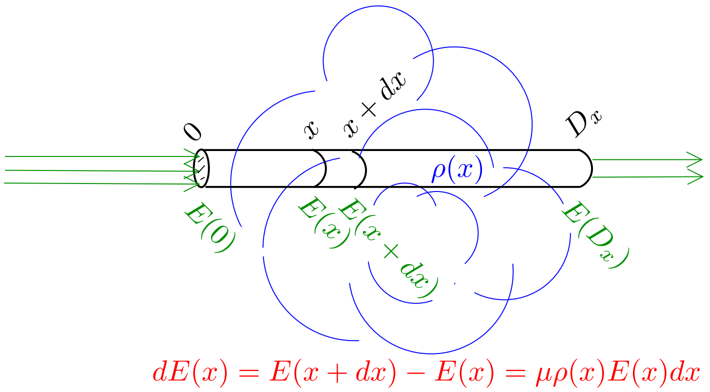


Figure 2: X-ray beam attenuation computation for a beam path of fixed y and z.

the same as the itensity at the source. We write both as $E(0, y, z)$. It is assumed that this quantity is well known or independently measureable.

X-Ray Beam Attenuation

As the beam traverses the region of interest, from $x = 0$ to $x = D_x$ (see Figure 2), the intensity changes as a result of interactions with matter. In particular, a transmitted (resultant) intensity $E(D_x, y, z)$ exits the far side of the region of interest. We will see that under reasonable assumptions this transmitted intensity is also planar but is reduced in magnitude. This process of intensity reduction is called attenuation. It is our goal in this section to model the attenuation process.

X-ray attenuation is a complex process that is a combination of several physical mechanisms (outlined in the supplementary material) describing both scattering and absorption of x-rays. We will consider only Compton scattering which is the dominant process for typical medical x-ray radiography. In this case, attenuation is almost entirely a function of photon energy and material mass density. As we are considering monochromatic (monoenergetic) photons, attenuation is modeled as a function of mass density only.

Consider the beam of initial intensity $E(0, y, z)$ passing through the region of interest, at fixed y and z. The relative intensity change in an infinitesimal distance from x to $x + dx$ is proportional to the mass density $\rho(x, y, z)$ and is given by

$$dE(x, y, z) = E(x + dx, y, z) - E(x, y, z) = -\mu\rho(x, y, z)E(x, y, z)dx$$

where μ is a factor that is nearly constant for many materials of interest. We also assume that any scattered photons exit the region of interest without further interaction. This is the so-called single-scatter approximation which dictates that the intensity remains planar for all x.

Integrating over the path from $x = 0$ where the initial beam intensity is $E(0, y, z)$ to $x = D_x$ where the beam intensity is $E(D_x, y, z)$ yields

$$\frac{dE(x, y, z)}{E(x, y, z)} = -\mu\rho(x, y, z)dx$$

$$\int_{E(0,y,z)}^{E(D_x,y,z)} \frac{dE(x,y,z)}{E(x,y,z)} = -\mu \int_0^{D_x} \rho(x,y,z)dx$$

$$\ln E(D_x,y,z) - \ln E(0,y,z) = -\mu \int_0^{D_x} \rho(x,y,z)dx$$

$$E(D_x,y,z) = E(0,y,z)e^{-\mu \int_0^{D_x} \rho(x,y,z)dx}.$$

This expression shows us how the initial intensity is reduced, because photons have been scattered out of the beam. The relative reduction depends on the density (or mass) distribution in the region of interest.

Radiographic Energy Detection

The transmitted intensity $E(D_x,y,z)$ continues to travel on to a detector (e.g. film) which records the total detected energy in each of m detector bins. The detected energy in any bin is the intensity integrated over the bin cross-sectional area. Let p_k be the number of x-ray photons collected at detector bin k. p_k is then the collected intensity integrated over the bin area and divided by the photon energy.

$$p_k = \frac{1}{h\nu} \iint_{(bin\ \ k)} E(0,y,z) \left(e^{-\mu \int_0^{D_x} \rho(x,y,z)dx}\right) dydz.$$

Let the bin cross sectional area, σ, be small enough so that both the contributions of the density and intensity to the bin area integration are approximately a function of x only. Then

$$p_k = \frac{\sigma E(0,y_k,z_k)}{h\nu} e^{-\mu \int_0^{D_x} \rho(x,y_k,z_k)dx},$$

where y_k and z_k locate the center of bin k. Let p_k^0 be the number of x-ray photons initially aimed at bin k, $p_k^0 = \sigma E(0,x,y)/h\nu$. Due to attenuation, $p_k \leq p_k^0$ for each bin.

$$p_k = p_k^0 e^{-\mu \int_0^{D_x} \rho(x,y_k,z_k)dx}.$$

Equivalently, we can write (multiply the exponent argument by σ/σ):

$$p_k = p_k^0 e^{-\frac{\mu}{\sigma} \int_0^{D_x} \sigma\rho(x,y_k,z_k)dx}.$$

The remaining integral is the total mass in the region of interest that the x-ray beam passes through to get to bin k. We will call this mass s_k. Now we have

$$p_k = p_k^0 e^{-s_k/\alpha},$$

where $\alpha = \sigma/\mu$. This expression tells us that the number of photons in the part of the beam directed at bin k is reduced by a factor that is exponential in the total mass encountered by the photons.

Finally, we note that the detector bins correspond precisely to pixels in a radiographic image.

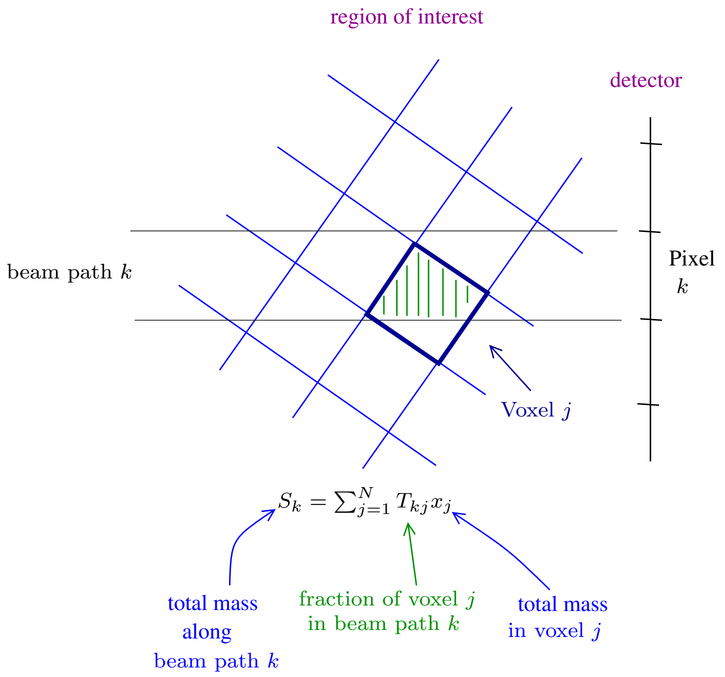

region of interest
detector
beam path k
Pixel k
Voxel j
$S_k = \sum_{j=1}^N T_{kj} x_j$
total mass along beam path k
fraction of voxel j in beam path k
total mass in voxel j

Figure 3: Object space and radiograph space discretization.

The Radiographic Transformation Operator

We consider a region of interest subdivided into N cubic voxels (three-dimensional pixels). Let x_j be the mass in object voxel j and T_{kj} the fraction of voxel j in beam path k (see Figure 3). (Note that this is a different context, in particular, x_j is *not* related to the direction that the x-ray beams are traveling!) Then the mass along beam path k is

$$s_k = \sum_{j=1}^{N} T_{kj} x_j,$$

and the expected photon count at radiograph pixel k, p_k, is given by

$$p_k = p_k^0 e^{-\frac{1}{\alpha}\sum_{j=1}^{N} T_{kj} x_j},$$

or equivalently,

$$b_k \equiv \left(-\alpha \ln \frac{p_k}{p_k^0} \right) = \sum_{j=1}^{N} T_{kj} x_j.$$

The new quantities b_k represent a variable change that allows us to formulate the matrix expression for the radiographic transformation

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

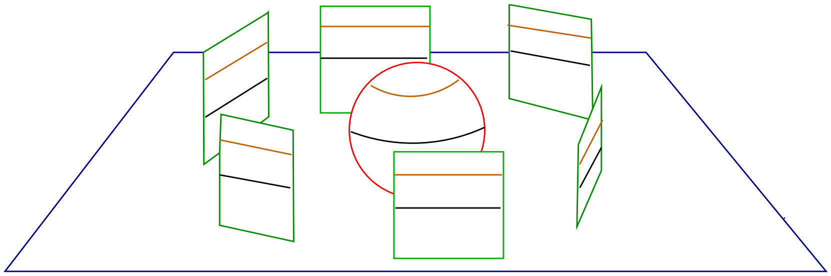


Figure 4: Example axial radiography scenario with six equally spaced views. Horizontal slices of the object project to horizontal rows in the radiographs.

This expression tells us that given a voxelized object mass distribution image $\mathbf{x} \in \mathbb{R}^N$, the expected radiographic data (mass projection) is image $\mathbf{b} \in \mathbb{R}^m$, with the two connected through radiographic transformation $T \in \mathcal{M}_{m \times N}(\mathbb{R})$. The mass projection $\mathbf{b}$ and actual photon counts $\mathbf{p}$ and $\mathbf{p}^0$ are related as given above. It is important to note that b_k is defined only for $p_k > 0$. Thus, this formulation is only valid for radiographic scenarios in which every radiograph detector pixel records at least one hit. This is always the case for medical applications which require high constrast and high signal-to-noise ratio data.

Multiple Views and Axial Tomography

Thus far, we have a model that can be used to compute a single radiograph of the region of interest. In many applications it is beneficial to obtain several or many different views of this region. In some industrial applications, the region of interest can be rotated within the radiographic apparatus, with radiographs obtained at various rotation angles. In medical applications the radiographic apparatus is rotated about the region of interest (including the subject!). In this latter case, the voxelization remains fixed and the coordinate system rotates. For each of a view angles the new m pixel locations require calculation of new mass projections T_{kj}. The full multiple-view operator contains mass projections onto all $M = a \cdot m$ pixel locations. Thus, for multiple-view radiography: $\mathbf{x}$ is still a vector in $\mathbb{R}^N$, but $\mathbf{b}$ is a vector in $\mathbb{R}^M$ and T is a matrix operator in $\mathcal{M}_{M \times N}(\mathbb{R})$.

Finally, we make the distinction between the general scenario and axial tomography (CAT scans). In principle, we could obtain radiographs of the region of interest from any direction (above, below, left, right, front, back, etc.). However, in axial tomography the physical limitations of the apparatus and subject placement dictate that views from some directions are not practical. The simplest scenario is to obtain multiple views by rotating the apparatus about a fixed direction perpendicular to the beam direction. This is why CAT machines have a donut or tube shaped appearance within which the apparatus is rotated. The central table allows the subject to rest along the rotation axis and the beam can pass through the subject along trajectories.

This axial setup also simplifies the projection operator. If we consider the ℓ^{th} slice of the region of interest, described by an $n \times n$ arrray of N voxels, the mass projections of this slice will only occur in the ℓ^{th} row of pixels in each radiographic view see Figure 4.

As a result, 3D reconsructions can be obtained by a series of independent 2D reconstructed slices. For example, the brown slice of the spherical object (represented in $\mathbb{R}^N$) is related to the collection of brown rows of the radiographs (represented in $\mathbb{R}^M$) through the projection operator $T \in \mathcal{M}_{M\times N}(\mathbb{R})$. The black slice and black rows are related through the *same* projection operator.

Model Summary

The list below gathers the various mathematical quantities of interest.

- N is the number of object voxels.
- M is the number of radiograph pixels.
- $\mathbf{x} \in \mathbb{R}^N$ is the material mass in each object voxel.
- $\mathbf{b} \in \mathbb{R}^M$ is the mass projection onto each radiograph pixel.
- $\mathbf{p} \in \mathbb{R}^M$ is the photon count recorded at each radiograph pixel.
- $\mathbf{p}^0 \in \mathbb{R}^M$ is the incident photon count possible for each radiograph pixel.
- $T \in \mathcal{M}_{N\times M}(\mathbb{R})$ is voxel volume projection operator. T_{ij} is the fractional volume of voxel j which projects orthogonally onto pixel i.
- $\mathbf{b} = -\alpha \ln\left(\frac{\mathbf{p}}{\mathbf{p}^0}\right)$.
- $\mathbf{b} = T\mathbf{x}$ is the (mass projection) radiographic transformation.

The description of images (objects and radiographs) as vectors in $\mathbb{R}^N$ and $\mathbb{R}^M$ is computationally useful and more familiar than a vector spaces of images. One should keep in mind that this is a particular representation for images which is useful as a tool but is not geometrically descriptive. The price we pay for this convenience is that we no longer have the geometry of the radiographic setup (pixelization and voxelization) encoded in the representation.

A vector in $\mathbb{R}^3$, say $(1, 2, 5)$, is a point in a three-dimensional space with coordinates described relative to three orthogonal axes. We can actually locate this point and plot it. An image represented in $\mathbb{R}^3$, say $(1, 2, 5)$, is *not* a point in this space. Without further information about the vector space of which it is a member, we cannot draw this image. The use of $\mathbb{R}^3$ allows us to perform scalar multiplication and vector addition on images *because these operations are equivalently defined on* $\mathbb{R}^3$.

Model Assumptions

The radiography model we have constructed is based on a number of approximations and assumptions which we list here. This list is not comprehensive, but it does gather the most important concepts.

- **Monochromaticity.** Laboratory sources generate x-rays with a wide range of energies as a continuous spectrum. We say that such x-ray beams are polychromatic. One method of approximating a monochromatic beam is to precondition the beam by having the x-rays pass through a uniform material prior to reaching the region of interest. This process preferentially attenuates the lower energy photons, leaving only the highest energy photons. This process is known as beam-hardening. The result is a polychromatic beam with a narrower range of energies. We can consider the beam to be approximately monochromatic, especially if the attenuation coefficient(s) of the material, μ, is not a strong function of photon energy.

- **Geometric Beam Characteristics.** Laboratory sources do not naturally generate planar x-ray beams. It is more characteristic to have an approximate point source with an intensity pattern that is strongly directionally dependent. Approximate planar beams with relatively uniform intensity $E(0, y, x)$ can be achieved by selective beam shielding and separation of source and region of interest. In practice, it is common to use the known point source or line source characteristics instead of assuming a planar beam. The model described here is unchanged except for the computation of T itself.

- **Secondary Radiation.** Our model uses a single-scatter approximation in which if a photon undergoes a Compton scatter, it is removed from the analysis. In fact, x-rays can experience multiple scatter events as they traverse the region of interest or other incidental matter (such as the supporting machinery). The problematic photons are those that scatter one or more times *and* reach the detector. This important secondary effect is often approximated by more advanced models.

- **Energy-Dependent Attenuation.** The attenuation coefficient μ, which we have taked to be constant, is not only somewhat material dependent but is also beam energy dependent. If the beam is truly monochromatic this is not a problem. However, for a polychromatic beam the transmitted total energy will depend on the *distribution* of mass along a path, not just the total mass.

- **Other Attenuation Mechanisms.** We have included only Compton scattering in the model. Four other mechanisms (outlined in the supplementary material) contribute to the attenuation. While Compton scattering is the dominant contributor, photoelectric scattering will have some effect. It becomes important at the lower energies of interest and for materials of relatively high atomic number – such as calcium which is concentrated in bone. The major effect of ignoring photoelectric scattering is quantitative mass uncertainty.

- There are a number of **Detector-Related Effects** which affect radiograph accuracy. Energy detection efficiency can be a function of beam intensity, photon energy and even pixel location. Detectors are subject to point-spread effects in which even an infinitesimally narrow beam results in a finitely narrow detection spot. These types of effects are usually well-understood, documented and can be corrected for in the data. Detection is also prone to noise from secondary radiation, background radiation, or simply manufacturing variances.

Additional Resources

There is a variety of online source material that expands on any of the material presented here. Here are a few starting points.

`https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/cc_rad_index.htm`

`http://web.stanford.edu/group/glam/xlab/MatSci162_172/LectureNotes/01_Properties%20&%20Safety.pdf`

`http://radiologymasterclass.co.uk/tutorials/physics/x-ray_physics_production.html`

Several physical processes contribute to absorption and scattering of individual photons as they pass through matter. Collectively, these processes alter the geometry and intensity of a beam of such photons. What follows is a brief description of each.

- The **Photoelectric Effect** is the absorption of an x-ray photon by an atom accompanied by the ejection of an outer-shell electron. This ionized atom then re-absorbs an electron and emits an x-ray of energy characteristic of the atom. The daughter x-ray is a low-energy photon which is quickly re-absorbed and is effectively removed from the x-ray beam. Photoelectric absorption is the dominant process for photon energies below about 100keV and when interacting with materials of high atomic number.

- **Rayleigh Scattering** is the process of a photon interacting with an atom without energy loss. The process is similar to the collision of two billiard balls. Rayleigh scattering is never the dominant mechanism, at any energy.

- **Compton Scattering** occurs when an x-ray photon interacts with an electron imparting some energy to the electron. Both electron and photon are emitted and the photon undrgoes a directional change or scatter. Compton Scattering is the dominant process for soft tissue at photon energies between about 100keV through about 8MeV.

- **Pair Production** is the process in which a photon is absorbed producing a positron-electron pair. The positron quickly decays into two 510keV x-ray photons. Pari production is only significant for photon energies of several MeV or more.

- **Photodisintegration** can occur for photons of very high energy. Photodisintegration is the process of absorption of the photon by an atomic nucleus and the subsequent ejection of a nuclear particle.

4.1.3 Summary of Terminology from the Reading Assignment:

It may be useful to have this out as a reference during Lab 2. As this lab follows the lecture on the physics of radiography, much of this material is review from your reading, but is provided here for clarity and continuity. It is important to keep the notation consistent. There are several details to keep in mind:

Pixel versus Voxel. "Voxel" is the standard term for a general multi-dimensional pixel – a hyperpixel. In this Lab, for consistency of language, we use "voxel" whenever we are considering the area of interest (or object) and we use "pixel" whenever we are considering the radiograph.

Square Voxel Array. The total number of voxels is $N = n^2$. This is a simplification. There is no theoretical or application limitation that requires the use of a square grid of square voxels. In general the area of interest can be partitioned in *any* finite fashion. We have chosen an $n \times n$ grid for computational and visual simplicity.

Voxel and Pixel Size. Voxels are of unit size. This is a choice that makes the computations and discussion simpler. In quantitative applications, one must be careful to use correct units. Pixel size is given in terms of the voxel size by specifying the scale factor $ScaleFac$. This units simplification does not alter the quality or details of computational results – quantitative application results differ only by a multiplicative constant. One can think of this as working in unfamiliar units.

Limitations on n and m. Both n and m must be non-negative even integers. This is *only* because this makes the code tomomap.m easier to compose and compute. If the student attempts to use an odd integer for n or for m the code will stop and report an error message.

Radiographs Arrangement. Each radiograph has the same number of pixels, m. This is another simplification that makes computations simpler. In practice, radiographic angles are imposed by either rotating the object or rotating the source-detector combination, and taking individual radiographs at separate times. There is nothing special about measuring radiographic angles in degrees east of south. This is just the way the module code tomomap.m was written. Notice that for each radiograph, the center point of the object grid projects perpendicularly to the center point of the radiograph.

Visual Setup. Keep in mind that in applications the detectors are much more distant from the area of interest as would be indicated in the figures. All radiographs are typically taken equidistant from the object.

4.2 Lab 4: Radiography and Tomography: Part 2

An Abbreviated View of Radiography

Transmission Radiography is the process of measuring and recording changes in a high-energy particle beam (X-rays, protons, neutrons, etc.) resulting from passage through an object of interest.

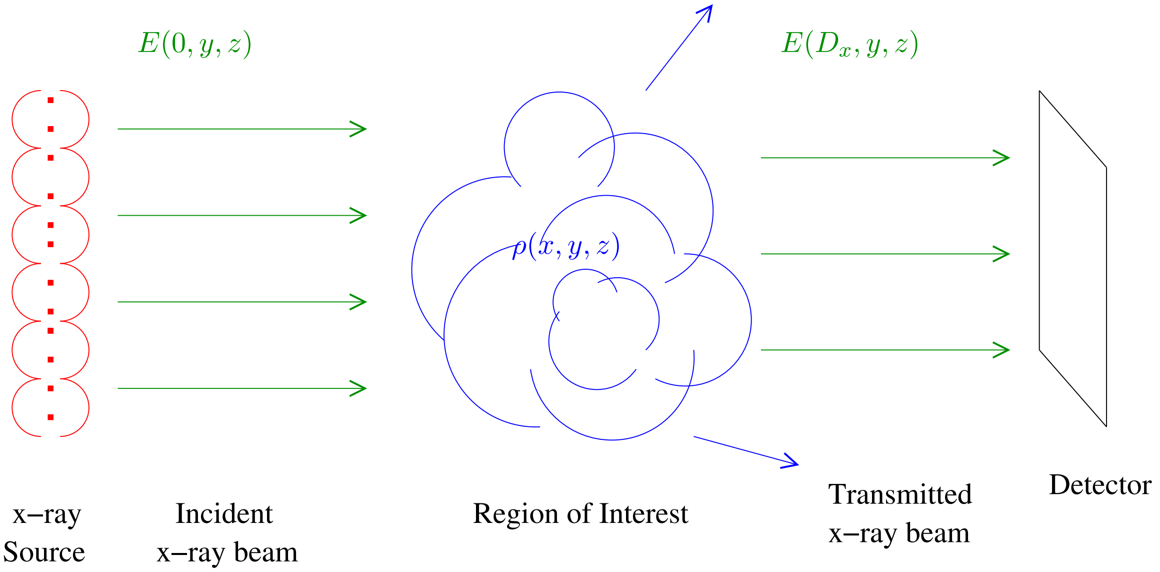


Figure 2: Typical radiographic experiment. The function E measures intensity of the x-ray beam.

Radiographic Scenarios and Notation

A single-view radiographic setup consists of an area of interest where the object will be placed, and a single screen onto which the radiograph will be recorded. A multiple-view radiographic setup consists of a single area of interest experimentally designed so that radiographs of this area can be recorded for different locations about the object.

The geometry of radiographic scenarios is illustrated in figures 1 and 2. The notation is as follows.

- Slice of region of interest: n by n array of voxels.
- Total number of voxels in each slice is $N = n^2$.
- Each voxel has a width and height of 1 unit.
- For each radiographic view we record m pixels of data.
- The width of each pixel is $ScaleFac$. If $ScaleFac = 1$ then pixel width is the same as voxel width.

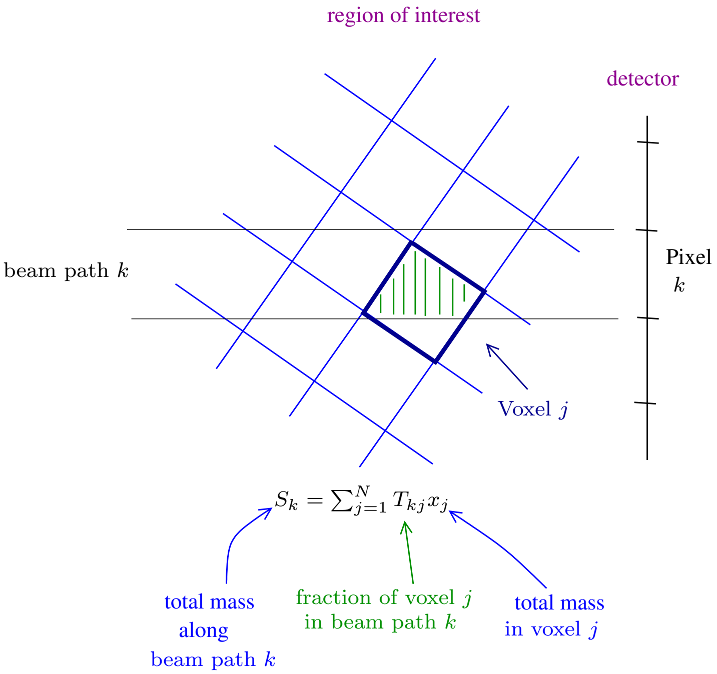

region of interest
detector
beam path k
Pixel k
Voxel j
$S_k = \sum_{j=1}^{N} T_{kj} x_j$
total mass along beam path k
fraction of voxel j in beam path k
total mass in voxel j

Figure 3: Object space and radiograph space discretization.

- Number of radiographic angles (views): a.
- Total number of pixels in the radiograph: $M = am$
- Angle of the i^{th} view (measured in degrees east of south): θ_i
- Object mass at voxel j is x_j
- Recorded radiograph value at pixel k is b_k

In this Lab we will be constructing matrix representations of the radiographic transformation operators for example scenarios. Recall that an object can be represented by an N-dimensional vector x and a set of radiographs can be represented by an M-dimensional vector b. What will be the size of the corresponding matrix operator that maps object vector x to radiograph vector b? Recall the definition of the matrix operator from the radiography discussion. Notice that particular values for x and b are *not necessary* for computing the matrix of the radiographic transformation!

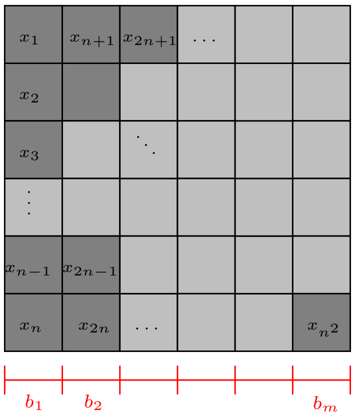


Figure 1: The geometry of a single view radiographic transformation.

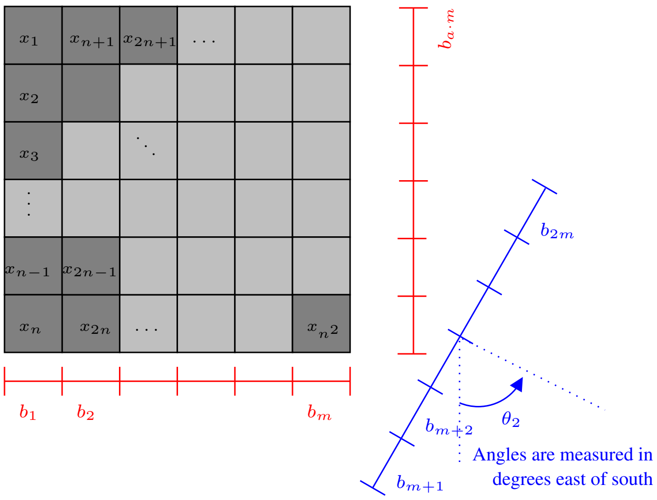


Figure 2: The Geometry of a multiple view radiographic transformation showing view 1, view 2, and view a.

A First Example

Let's look at a specific scenario. For the setup pictured below, we have:

- Total number of voxels: $N = 4$ $(n = 2)$.
- Total number of pixels: $M = m = 2$
- $ScaleFac = \sqrt{2}$
- Number of views: $a = 1$
- Angle of the single view: $\theta_1 = 45°$

Recalling that T_{kj} is the fraction of voxel j which projects perpendicularly onto pixel k, the matrix associated with this radiographic setup is $T = \begin{bmatrix} {}^1/_2 & 1 & 0 & {}^1/_2 \\ {}^1/_2 & 0 & 1 & {}^1/_2 \end{bmatrix}$. Be sure and check this to see if you agree! Hence, for *any* input vector x, the radiographic output is $b = Tx$. Find the output when the object is the vector $x = \begin{bmatrix} 10 \\ 0 \\ 5 \\ 10 \end{bmatrix}$.

.

In Class Example:

We will first do an example together in class. Suppose you have the setup where

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 2$
- $ScaleFac = \sqrt{2}$
- Number of views: $a = 2$
- Angle of the views: $\theta_1 = 45°, \theta_2 = 135°$

(a) Sketch this setup.

(b) Calculate the matrix associated with the setup.

(c) Repeat step (b) using the code `tomomap`. You will need to use the function `tomomap.m` found in blackboard under Lab 2 code.

.

Radiography and Tomography Lab 2

Now it's your turn! **Whatever you do not finish, you will need to finish outside of class. The due date for this lab is 1 week after we have started it, unless the instructor chances the due date. If you are working with a partner, please only submit <u>ONE</u> lab with both of your names!** There are 2 parts to this lab. Complete Parts I and II.

Note: Some of the following exercises will ask you to use Matlab or Octave to compute radiographic transformations. You will need to use the function `tomomap.m` found in blackboard under Lab 2 code.

Part I: Finding the matrix associated with a radiographic transformation, and using the matrix to determine radiographic outputs.

Your instructor may assign one or more of the following scenarios.

(a) Suppose you have the setup where

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 2$
- $ScaleFac = 1$
- Number of views: $a = 2$
- Angle of the views: $\theta_1 = 0°, \theta_2 = 90°$

i. Sketch this setup.

ii. Calculate the matrix associated with the setup.

iii. Find the radiographs of the following objects. Remember you will need to convert these objects into a vector in order to multiply it by T. Refer back to our First Example for help!

x=

4	3
6	1

y=

0	1
1	0

(b) Suppose you have the setup where

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 4$
- $ScaleFac = \sqrt{2}/2$
- Number of views: $a = 1$
- Angle of the views: $\theta_1 = 45°$

i. Sketch this setup.

ii. Calculate the matrix associated with the setup.

iii. Repeat step (b) using the code tomomap. How you can use this is first by opening the mfile tomomap.m in MatLab or drag it into Octave. The code to do this is T=full(tomomap(n,m,th,ScaleFac)) so for example if $n = 4$, $m = 4$, and the angles are 0, 20, and 90, with a scale factor of 1, you would write
T=full(tomomap(4,4,[0 22.5 90],1))
Note to write $\sqrt{2}$, write sqrt(2).

(c) **Extra Credit Example:** (this one requires some geometric maneuvering)
Suppose you have the setup where

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 2$
- $ScaleFac = 1$
- Number of views: $a = 2$
- Angle of the views: $\theta_1 = 0°, \theta_2 = 45°$

i. Sketch this setup.

ii. Calculate the matrix associated with the setup.

iii. Repeat step (b) using the code tomomap. How you can use this is first by opening the mfile tomomap.m in MatLab or drag it into Octave. The code to do this is T=full(tomomap(n,m,th,ScaleFac)) so for example if $n = 4$, $m = 4$, and the angles are 0, 22.5, and 90, with a scale factor of 1, you would write
T=full(tomomap(4,4,[0 22.5 90],1))

Part II: Radiographs of Linear Combinations of Objects

(d) Consider the objects from exercise 1 above: **x** (left) and **y** (right).

x=

4	3
6	1

y=

0	1
1	0

For both of the assigned transformations, T in questions 1 and 2 from Part I above, answer the following questions.

(a) What is the radiograph of the object $3x$?

(b) What is the radiograph of the object $0.5y$?

(c) What is the radiograph of the object $3x + 0.5y$?

(d) What do you notice? Can you generalize your observations?

(e) If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: `http://bit.ly/linearalgebralabs2020`

5 Lab 5: Translation and Rotation Matrices

5.1 Pre-Lab 5

In preparation for this lab, please watch this video https://youtu.be/ysOEnGg_rz0 and then answer the following questions. The video starts with Translation and Rotation Matrices, and discusses homogeneous coordinate representations around the 14:55 time stamp.

(a) Find a 2 by 2 matrix that reflects the point (x, y) about the x axis.

(b) Find a 2 by 2 matrix that scales the point (x, y) by 0.5 in the x direction and 3 in the y direction.

(c) Find a 3 by 3 matrix that scales the point (x, y, z) by 2 in the x direction and 4.5 in the direction, but does not adjust the scaling of the y direction.

(a) Find 3 different homogeneous coordinate representations for the Cartesian point $(1, -2)$.

5.2 Lab 5 In-Class Portion

(a) Rewrite your matrices found in questions 17a and 17b from the Pre-Lab if we were using the homogeneous coordinate point $(x, y, 1)$. (You should have 3 by 3 matrix.)

(b) At the end of the video, we discussed how we can rotate the point $(x, y, 1)$ about the point (a, b) by θ degrees counterclockwise by using this matrix multiplication:

$$M = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$$

Simplify this matrix multiplication to 1 matrix (instead of 3) and then use it to rotate the point $(4, 2)$ 90° about the point $(1, 1)$.

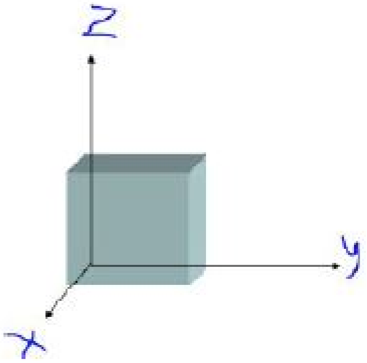
Translation and Rotation Matrices for 3D

Wireframe models are often digitally create an object in a computer since they can easily be rotated, translated, and scaled using matrix multiplication. These wireframe models are used for 3D printing, movie/video game animations, simulations to determine aerodynamics, and more. To create a wireframe we need to find the coordinate of each point and determine which vertices are connected with lines (which creates a system of vectors). We can store all the coordinates of our vertices in the rows of a matrix. So for example, if we have 300 vertices, our matrix will have 3 rows and 300 columns. If we represent our points in homogeneous coordinates, we would have 4 rows and 300 columns. We can generalize our results from Translation and Rotation Matrices in 2D to 3D to find matrices to rotate and translate these wireframes.[3]

[3] For Some great CAD Notes: See Dr Shriram Hegde's course site: http://web.iitd.ac.in/~hegde/ Images from this assignment are from his course notes.

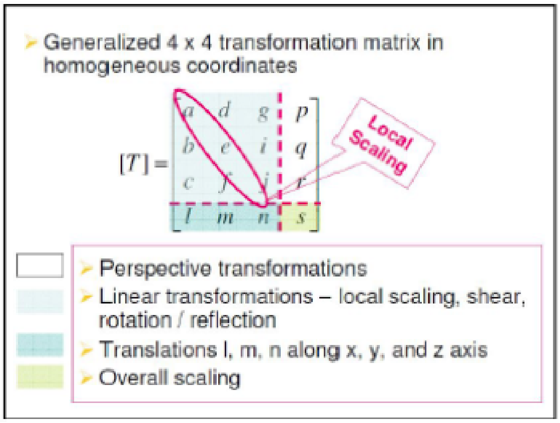
First we need to represent a wireframe using homogeneous coordinates. Recall, homogeneous point (x, y, z, h) corresponds with the point $(\frac{x}{h}, \frac{y}{h}, \frac{z}{h})$ in 3-space.

We will be using this cube as a working example:
Suppose our cube has vertices at:
$(0, 0, 1), (2, 0, 1), (2, 3, 1),$
$(0, 3, 1), (0, 0, 0), (2, 0, 0),$
$(2, 3, 0), \&(0, 3, 0).$



We can write these in homogenous coordinates by adding a dimension and in this case, we will just use 1.
So our homogeneous coordinates are:
$(0, 0, 1, 1), (2, 0, 1, 1), (2, 3, 1, 1), (0, 3, 1, 1),$
$(0, 0, 0, 1), (2, 0, 0, 1), (2, 3, 0, 1), \&(0, 3, 0, 1)$

So we can represent this cube with the matrix: $M = \begin{bmatrix} 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 3 & 0 & 0 & 3 & 3 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$



- Generalized 4 x 4 transformation matrix in homogeneous coordinates

$$[T] = \begin{bmatrix} a & d & g & p \\ b & e & i & q \\ c & f & j & r \\ l & m & n & s \end{bmatrix}$$

□	Perspective transformations
🟦	Linear transformations – local scaling, shear, rotation / reflection
🟦	Translations l, m, n along x, y, and z axis
🟩	Overall scaling

To the left is a general transformation matrix which shows how different parts of the matrix can change a wireframe object. We will now give some examples of some of transformations.

Uniform Scaling:
Want to scale our cube by s?

Use $T_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}$.

Note that if $s < 1$ our cube becomes uniformly larger. If $s > 1$ our cube shrinks uniformly.

(a) Multiplying our cube by $T_2 \cdot M = \begin{bmatrix} 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 3 & 0 & 0 & 3 & 3 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$.

Note our $s = 2$ so T_2 uniformly shrinks our cube. Why is this? [Hint: convert from homogeneous coordinates to (x, y, z)]

Non-Uniform/local Scaling:
Want to scale our cube by different values for each direction?

Use $T_s* = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Where a scales the x coordinate, e scales the y coordinate, and j scales the z coordinate.

(b) Show that for $a = 1/2, e = 1/3$, and $j = 1$ scales our cube down to the unit cube by multiplying $T_s * \cdot M = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 3 & 0 & 0 & 3 & 3 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} =$

Translation Matrices:
To translate your wireframe by p in the x direction, q in the y direction, and r in the z direction use: $T_t = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$

If you are interested in learning more -like how the shear a wireframe or move your wireframe about an arbitrary axis, etc. we encourage you to go to Dr Shriram Hegde's homepage `http://web.iitd.ac.in/~hegde/` Go to Courses → Computer Aided Design → Course Notes → Lecture 6.

(a) Find the translation matrix to move your cube by 1 in the x direction, -1 in the y direction, and 2 in the z direction.

Rotation Matrices:
To Rotate an image θ degrees about the following axis (using homogeneous coordinates):

$$\text{x-axis: } R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{y-axis: } R_x(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{z-axis: } R_Z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) What is the rotation matrix that will rotate our cube 90 degrees about the y axis?

Note: Remember, just like with 2D objects, we can multiply matrices together to get multiple movements of our wireframes.

Below is Octave code for rotating a 2D-point 90 degrees counterclockwise:

```
% Create rotation matrix
theta = 90; % to rotate 90 counterclockwise
R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
% Rotate your point
point = [3; 5]; % This is an arbitrarily selected point
rotpoint = R*point;
```

(a) With this as a guide, write code that will rotate the point (1, 7) by 270 degrees counterclockwise. You can fill in the blanks below as to what you type.

```
% Create rotation matrix
theta = ___; % to rotate ___ counterclockwise
R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
% Rotate your point
point = [___; ___]; % arbitrarily selected
rotpoint = R*point;
```

(b) For the previous question, what did you get as an output for rotpoint?

(c) Write code below that would rotate a point 270 degrees **clockwise**.

(d) If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: http://bit.ly/linearalgebralabs2020

5.3 Bonus Questions

Follow along with the tutorial at https://www.mathworks.com/help/symbolic/examples/rotation-matrix-and-transformation-matrix.html in order to answer the following questions.

(a) At the beginning of the tutorial, you are shown how to plot a certain parametric surface. Write code that will rotate that parametric surface about the y-axis by 60 degrees **counterclockwise**. Attach that code **and** the image that you obtain to this lab. If you have, write "See Attached" below.

(b) Near the middle to end of the tutorial, you are shown how to scale the parametric surface as well as rotate it. Write code that will first scale that parametric surface by 0.5 in the x-direction and by 2 in the y-direction, then will rotate the scaled surface about the x-, y-, and z-axis by 30 degrees **counterclockwise**, in order z, then y, then x. Attach that code **and** the image that you obtain to this lab. If you have, write "See Attached" below.

6 Lab 6: Radiography and Tomography Lab Part 3

6.1 Pre-Lab 6

This lab will motivate some of the computational techniques we will use to determine properties of transformations in addition to putting these properties into a real world context.[4] Please review the misconsceptions below and the In-Class example and then answer the questions on the next page. Do not worry if you get stuck, but try to answer each question.

In Part 2 of the Radiography and Tomography Lab, you found several radiographic transformation operators. The object image consisted of four (or in one case, sixteen) voxels and was represented as a vector in $\mathbb{R}^4$. The radiographic image was represented as a vector in $\mathbb{R}^M$, one entry for each radiographic pixel. The price we pay for this representation is that we no longer have the geometry of the radiographic setup encoded in the representation. The use of representations in $\mathbb{R}^n$ is a computational tool and not geometrically descriptive of vector spaces of images. We want to reiterate that this is *only* a representation and that these images are **not** vectors in $\mathbb{R}^M$. Because of these new image representations, each transformation could be constructed as a matrix operator in $\mathcal{M}_{M\times 4}(\mathbb{R})$.

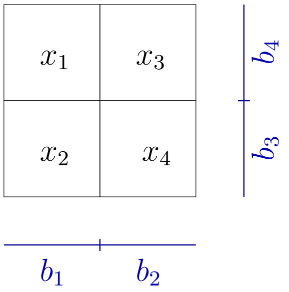
Common misconceptions at this point in the lab:

- **Misconception**: Each radiographic view corresponds to a different transformation.
  Truth: Multiple radiographic views are combined into a single radiographic image and is the product of a single transformation acting on our object. The scenario details (number and placement of views, voxelization of the object, etc.) and the domain and codomain descriptions $\mathcal{I}_{n\times n}(\mathbb{R})$ and $\mathcal{I}_{m\times 1}(\mathbb{R})$ should make this clear.
- **Misconception**: Objects cannot have negative valued entries.
  Truth: Objects are vectors in $\mathcal{I}_{2\times 2}(\mathbb{R})$, with entries in the field of Real numbers. Our particular application suggests that realistic objects must have non-negative entries. However, not all *meaningful* objects need be non-negative. For example, two non-negative objects, u and v, can have a difference object, $d = u - v$, which has negative entries. And d is a very important object!
- **Misconception**: Radiograph pixel widths can be individually chosen.
  Truth: In our scenarios, each pixel has the same width as every other pixel. Object voxels have unit width and radiograph pixels have width ScaleFac.
- **Misconception**: Objects are matrices.
  Truth: The space of radiographic objects $\mathcal{I}_{n\times n}(\mathbb{R})$ is isomorphic to $\mathcal{M}_{n\times n}(\mathbb{R})$.

[4] Radiography/Tomography ©2016 T. Asaki, C. Camfield, H. Moon, M. Snipes

In Class Example:

Consider the radiographic setup described below.



- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 2$
- $ScaleFac = 1$
- Number of views: $a = 2$
- Angle of the views: $\theta_1 = 0°, \theta_2 = 90°$

The transformation matrix for this map is $T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$.

6.1.1 Pre-Lab 6 Questions:

Answer the following questions about this radiographic setup above, justifying your conclusions. Answer the questions on this paper and I will go around and check your answers before class.

(a) Are any nonzero objects invisible to this operator? If so, give an example. We say that an object is nonzero if not all entries are zero. We say that an object is invisible if it produces the zero radiograph.

(b) Describe the set of all invisible objects. This could involve an equation that the entries would have to satisfy or a few specific objects that could be used to construct all other such objects. Be creative.

(c) a) Is it possible for two different objects to produce the same radiograph? If so, give an example.

b) Choose any two objects that produce the same radiograph and subtract them. What is special about the resulting object?

(d) Are there radiographs (in the appropriate dimension for the problem) that cannot be produced as the radiograph of *any* object? If so, give an example.

(e) Describe the set of radiographs that can be produced from all possible objects. This may require similar creativity.

6.1.2 In-Class Section of the Pre-Lab:

(f) Show that the set of all invisible objects is a *subspace* of the vector space of all objects.

(g) Find a set of objects that *spans* the space of all invisible objects. Is your set linearly independent? If not, find a linearly independent set of objects that spans the space of invisible objects. The linearly independent set you have found is a *basis* for the space of invisible objects.

(h) The set of all possible radiographs is a subset of a vector space. What is this vector space? Show that the set of all possible radiographs is a subspace of this vector space.

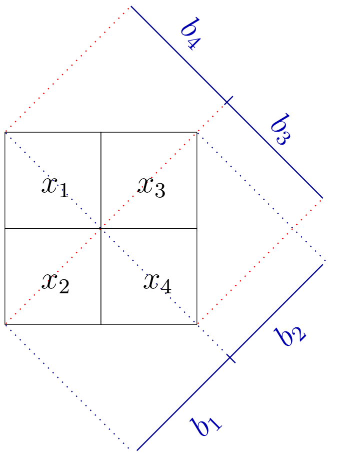
(i) Find a basis for the set of all possible radiographs. That is, find a linearly independent set of radiographs that spans the space of possible radiographs.

Summary:

6.2 Lab 6 Questions

Now it's your turn! For each of the two setups (B and C) below, Answer the following questions. In looking at more examples you will get a better sense of the underlying principles at work. We will use your observations to develop deeper linear algebra theories. Whatever you do not finish in class, you will need to finish outside of class. The due date for this lab is 1 week after we have started it, unless the instructor chances the due date. If you are working with a partner, please only submit <u>ONE</u> lab with both of your names!

Setup B

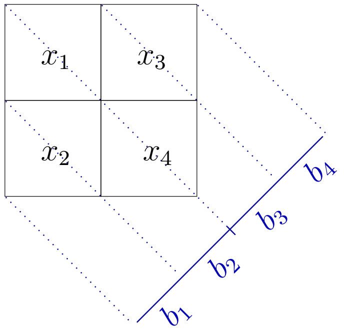


The Transformation for this map is

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

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 4$
- $ScaleFac = \sqrt{2}/2$
- Number of views: $a = 1$
- Angle of the views: $\theta_1 = 45°$

Setup C



The Transformation for this map is $T =$

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

- Height and width of image in voxels: $n = 2$ (Total voxels $N = 4$)
- Pixels per view in radiograph: $m = 2$
- $ScaleFac = 2$
- Number of views: $a = 2$
- Angle of the views: $\theta_1 = 90°$, $\theta_1 = 315°$

Questions for Each Setup

Answer the following questions for Set Up B and Setup C. First answer questions for Set up B then answer all questions for Setup C (to help with grading). Then answer the general questions below. Thanks!

(a) Describe the set of all invisible objects by finding a basis for this space. Are any nonzero objects invisible to this operator? If so, give an example. (See your Pre-Lab for the definitions: nonzero object and invisible object.)

(b) Is it possible for two different objects to produce the same radiograph? If so, give an example and subtract them. What is special about the resulting object?

(c) Describe the set of radiographs that **can** be produced from all possible objects.

(d) Are there radiographs (in the appropriate dimension for the problem) that cannot be produced as the radiograph of *any* object? If so, give an example.

(e) Find a basis for the space of all possible radiographs. That is, find a linearly independent set of radiographs that spans the space of possible radiographs.

(f) Do you think this transformation is injective? Why or why not?

(g) Do you think this transformation is onto? Why or why not?

Question: In this lab, you found bases for two different vector spaces (invisible objects, and possible radiographs). What reasons might you be interested in these bases, in the context of radiography?

Survey Feedback Question : If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: `http://bit.ly/linearalgebralabs2020`

Bonus Question: Can you find a radiographic setup for square objects with 4 voxels so that there are no invisible objects? What properties must the associated linear transformation have?

7 Lab 7 - Applications of Markov Chains and Singular Value Decomposition

7.1 Pre-Lab Reading: Applications of SVD

Singular values appear in many linear algebra applications like clustering (eigenvectors help with this too!), statistics, signal processing, and other applications that involve big data. If we had an Applied Linear Algebra II course (oh boy!) we would go into more examples of these applications, but for now we will just discuss one major application and discuss a more general use of SVD which will give some insight regarding its importance for applications which use big data.

As we have seen in our radiography/tomography labs, Matrices can be used to store and transfer digital images. SVDs of matrices can help use complete these processes efficiently. That is, SVDs can help reduce redundant information inside an image. They do this by takinge advantage of the fact that pixels near one another in a picture often have similar greyscale levels. Thus it may be possible to represent the image using less storage space while still keeping the essential parts of the picture. Now SVD isn't always the typical method used in data compression algorithms. There are other methods and versions of SVD which do an even better job. Nevertheless, Singular Value Decomposition still gives us an idea about how image compression can be done and why it sometimes leads to blurrier pictures than the original. SVD also is used in Principle Component Analysis, a technique to organize analyze multivariate data (and can used for face recognition algorithms and merging of pictures of faces). Unfortunately, we do not have time to cover this interesting topic.

General Idea: If $A = U\Sigma V^T$, we can use the *outer product expansion* to express $A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + ... + \sigma_n u_1 n v_n^T$ where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n > 0$.

The terms with the largest singular values contain most of the "data" or information about the image whereas smaller singular values don't contribute as much. So the idea is, we can sometimes discard many of these smaller low singular value terms, and still get a good approximation of our original image. Thus we can have $A \approx \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + ... + \sigma_k u_1 k v_k^T$ for $k < n$.

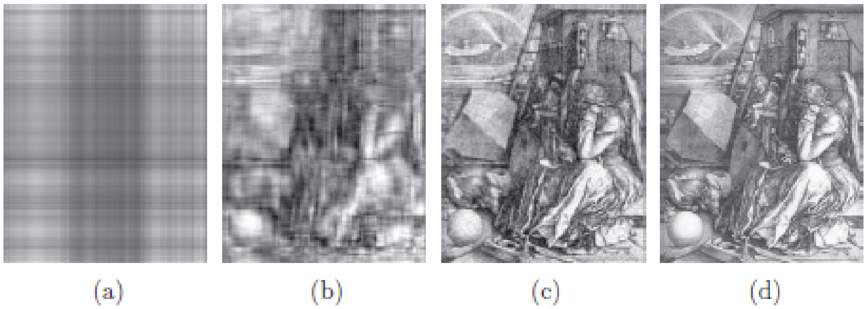
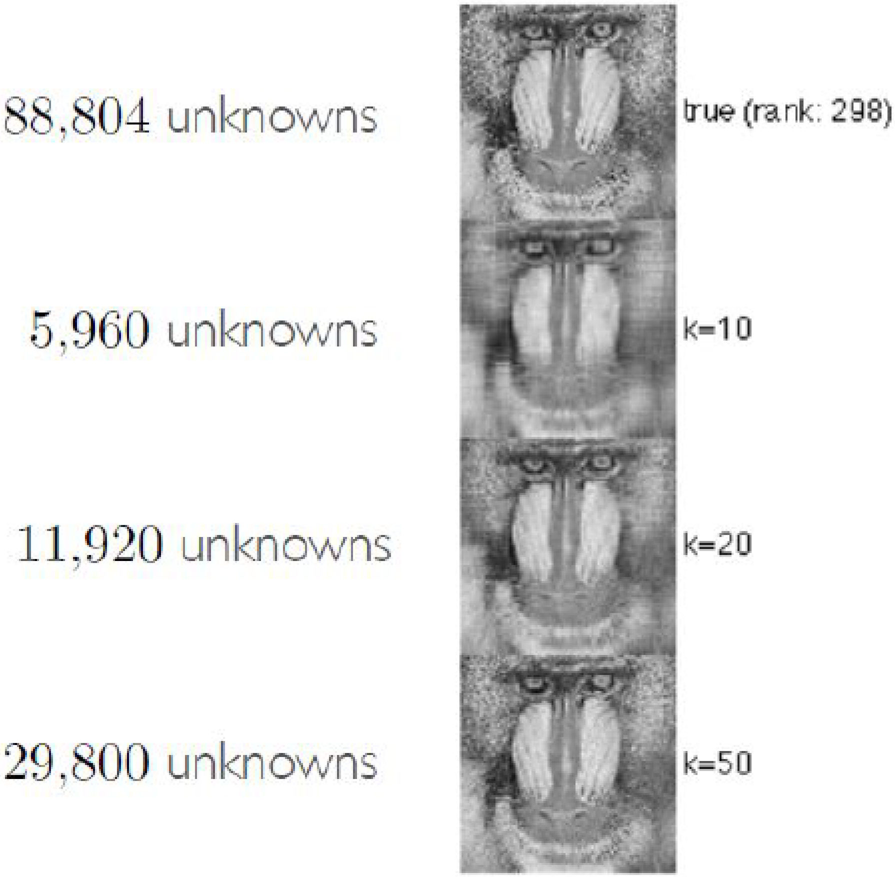


Figure 9.2. The closest rank 1 (a) and rank 10 (b) approximations to the Dürer print. In (c) and (d), we see the closest rank 50 approximation and the original print. Can you tell which is which? [5]

Example: If an image has m by n pixels, the storage requires $m \cdot n$ numbers. But if we approximate A with k singular values, $A \approx \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + ... + \sigma_k u_k v_1 k^T$, this requires only $(m + n)$ numbers! Savings Galore!



[5] Images from our text *When Life is Linear*

7.2 Example 1: Image Compression

We will not have you explore data compression and choosing singular values in order to approximate their image. Go to Blackboard and download the Lab 7 matlab file (SVD Compress) and the picture provided. Upload these files to Octave. (You can use your own picture, but Octave works best with true greyscale pictures. MatLab may be able to handle larger images.)

(a) Change the values for k. and see how that changes the picture. Use the Snip tool to attach pictures of the images which you get using 1 singular value, 20, 50, and 100 to your lab report.

7.3 Example 2: Approximating Matrix Rank

Most numerical calculations used in solving $Ax = b$ are as reliable as they can be when we use the Singular Value Decomposition for A. We have mentioned before that orthogonal matrices are nice and stable. That means that they do not affect lengths of vectors or angles between vectors, which decreases the likelihood of error. It also means that small changes to the data will not cause large changes to the solutions to the system. What this means is that the U and V^T vectors in the SVD of A, don't have the errors. These errors are stored in Σ. If the singular values of A are super, super small, it is often really difficult to avoid roundoff error. We can approximate or anticipate error by calculating something called the *condition number* of a matrix. This condition number affects the sensitivity of a solution to $Ax = b$ to errors/changes in the entries of A -that is it affects the stability of A. This condition number can be calculated many different ways, but the most commonly used calculation is the ratio of the largest and smallest singular value of A.

Definition 7.1. *If A is an invertible $n \times n$ matrix, then the ratio of $\frac{\sigma_1}{\sigma_n}$ of the largest and smallest singular values of A gives us the* ***condition number*** *of A.*

Numerical Note: The larger the condition number, the closer the matrix is to being singular (that is, non invertible). For example, condition number of I_n is 1 and a non-invertible matrix has an infinite condition number. If a matrix is really closed to be being singular, we say it is *ill-conditioned.* Ill-conditioned matrices can become singular if some of the entries are changed very slightly -which means they are in trouble if we have round-off error!

(a) Suppose A is 4 by 4 matrix with singular values $\sigma_1 = 2, \sigma_2 = 1, \sigma_3 = 10^{-1}$ and $\sigma_4 = 10^{-3}$. What is the condition number of A?

(b) We can use the condition number of a matrix to estimate the accuracy of a solution to $Ax = b$. Supposed the condition number of A is 10^k ($k > 0$) and A and b are accurate to r significant digits, then the calculated solution to $Ax = b$ should be accurate to at least $r - k$ significant digits. Suppose we the condition number of A is $1,000$ and A and b are accurate to 12 digits. How accurate will the solution to $Ax = b$ be?

Definition 7.2. *Round-off error also affects a computer's ability to determine the rank of a matrix. The* ***numerical rank*** *of a matrix is the perceived rank of a matrix by a computer.*

Definition 7.3. *The "****Machine epsilon****" (ϵ) of a computer gives an upper bound on the relative error due to rounding in floating point arithmetic.*
Recall, relative error is $|\frac{actual - approx}{approx}|$.

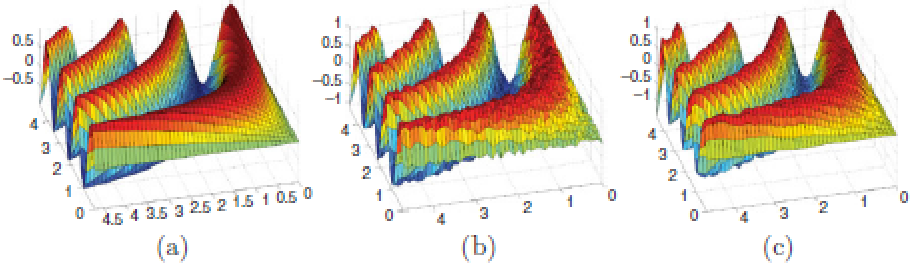
Knowing a computer's Machine epsilon and a matrix's SVD can help us determine the numerical rank of a matrix. Here's How:

Determining the Numerical Rank of a Matrix: Suppose a computer has a Machine Epsilon, ϵ and suppose matrix A has singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$. Let $b = \sigma_1 \cdot \epsilon \cdot n$ where n is the dimension of A and let k be the largest integer such that $\sigma_k \geq b$. Then k is the numerical rank of A.

(a) Suppose A is 4 by 4 matrix with singular values $\sigma_1 = 2, \sigma_2 = 1, \sigma_3 = 10^{-1}$ and $\sigma_4 = 10^{-7}$ and we have a Machine Epsilon, $\epsilon = 10^{-5}$. What is the numerical rank of A? What is the rank of A?

7.4 Example 3: Noise Data

SVD can help decrease noise in data: When an image has noise (errors or random values), sometimes a lower rank approximation can improve the quality of the data. While you may not get the perfect completely-non-noise data back, reducing the noise can help you get a better underlying picture of what is happening with the data. So sometimes, your goal is to reduce the noise.



(a) (b) (c)

Figure 9.3. A 3D graph (a), with noise (b), that is reduced (c) using the SVD.

In some circumstances you know there is noise, but you don't know how much noise. So how do we know which rank k approximation we should use to approximate our matrix? One method to help is to plot the singular values of our matrix. If we look at our figure below (from our text *When Life is Linear*), notice for what value we have a steep drop-off. This drop-off tell us which rank k approximation we should use.

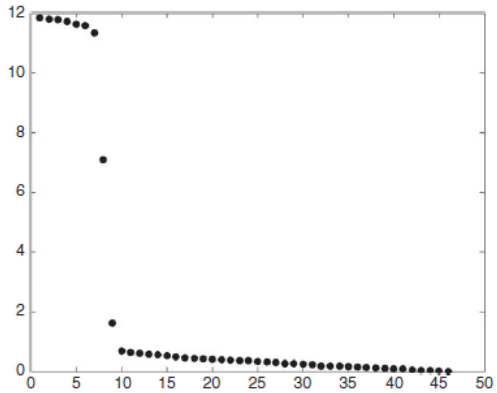


Figure 9.4. Singular values of the matrix containing the data graphed in Figure 9.3 (b).

(a) From the chart above, how many singular values would you choose to use?

7.5 Markov Chain Application: Board Games

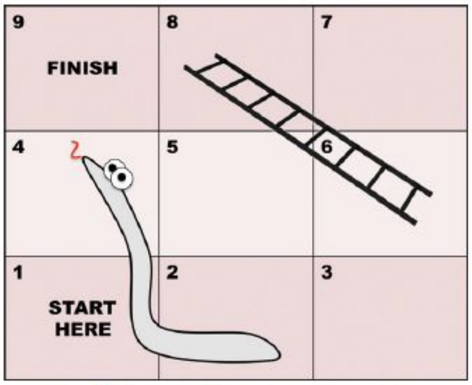


Figure 10.2. A small game of Snakes and Ladders.

Consider the board game above from your textbook, *When Life is Linear*. Here are the rules: You start on square 1 and you win when you reach square 9. On each turn, you roll a die. Rolling a 1 or 2 means you do no move and stay on your current square. Rolling 3 or 4 you move ahead one square If you roll a 5 or 6, you move ahead 2 squares. You must climb the ladder if you are in square 6 and slide down the snake if you land on square 4. Answer the following questions about this board game.

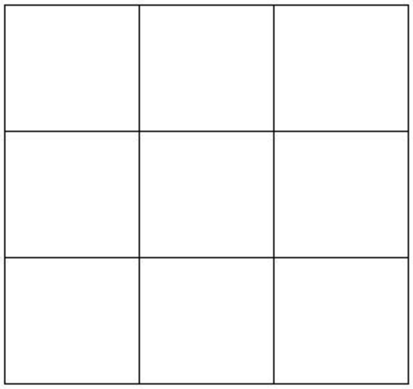
(a) Find the transition matrix for this Board Game.
Hint: Note that the (1,1) entry of your matrix would be $\frac{1}{3}$ since the probability you stay at start would be 1/3 because you would stay if you roll a 1 or 2. the (5,1) entry would be 0 though because there is no roll that would take you from start to the 5th square.

(b) Use Markov Chains to determine the minimum moves it could take to win the game.

Hint: Use $x^0 = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$ *as the initial state (being at start) and you want to find the first* x^k *state such that the 9th row of the vector is nonzero.*

(c) How many moves is it necessary for it to be 50% likely that someone has won?

Turn over for one more question!



(d) Create your own version of the game Snakes and Ladders above, making sure you have at least one snake (a space that brings you to a lower numbered space) and one ladder (a space that brings you to a higher numbered space). Then find the transition matrix for your version of the game. (Use the space below for your answer/work or another sheet of paper!)

(e) If you are willing to provide feedback for this lab, please fill out the survey for our study at the following link: `http://bit.ly/linearalgebralabs2020`

8 Extra Lab 8: Radiography and Tomography Part 4:

Reconstruction Without an Inverse[6]

8.1 Pre-Lab 8

Read the following pages and answer the Pre-Lab questions.

In this lab, we will consider the cases when the matrix representation of the radiographic transformation does not have an inverse. (Note: we say that $T : V \to W$ has an **inverse** $T^{-1} : W \to V$ if $T^{-1}T = I_V$, the identity mapping from V to V, and $TT^{-1} = I_W$ the identity mapping from W to W. We say that $T : V \to W$ has a **one-sided inverse**[7], $P : W \to V$ if $P \circ T = I_V$.)

For example, let $T\colon \mathbb{R}^2 \to \mathbb{R}^3$ be given by the matrix

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix}.$$

This has a one-sided inverse matrix P given by

$$P = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \end{pmatrix}.$$

P is the one-sided inverse because $PAx = x$. That is $PA = I_2$ as shown below.

$$PA = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

[6] Radiography/Tomography ©2016 T. Asaki, C. Camfield, H. Moon, M. Snipes

[7] More specifically, this is a *left inverse* of T.

Begin by considering the following theorem.

Theorem 8.1. *Let V and W be finite-dimensional vector spaces and $T : V \to W$ a linear transformation. If rank$(T) = \dim(V)$ then $T^T T$ is invertible*[8]*.*

Idea Behind Proof: Hint: What type of matrix is $T^T T$... and use a result from our notes

A more formal way to prove this is as follows:

Proof. Suppose V and W are finite-dimensional vector spaces and suppose $T : V \to W$ is a linear transformation such that $\text{rank}(T) = \dim(V)$. Thus $T : V \to W$ has full rank since $\text{rank}(T) = \dim(V) = n$. This means that ran$(T)$ is a subspace of W with the same dimension as V. So, we know that if we define the restriction of T^T to ran(T) to be $\tilde{T^T} : \text{ran}(T) \to V$, then $\tilde{T^T}$, having full rank maps onto V. That is, $\text{ran}(\tilde{T^T}) = V$. So, $T^T T : V \to V$ has only a trivial null space and therefore $T^T T$ has linear independent columns and thus by the Invertible Matrix Theorem has an inverse.

□

Here is an alternate proof:

Proof. We can show that $\text{null}(T^T T) = \{0\}$, then because $T^T T$ is square we have that $T^T T$ is invertible. Let $x \in \text{null}(T^T T)$. We have

$$0 = x^T 0 = x^T T^T T x = (Tx)^T (Tx) = \|Tx\|^2$$

which implies $Tx = 0$, that is, $\text{null}(T^T T) = \text{null}(T)$. Now, since $\text{rank}(T) = \dim(V)$, $\text{null}(T) = \{0\}$ and $\text{null}(T^T T) = \{0\}$. Therefore $T^T T$ has linear independent columns and thus by the Invertible Matrix Theorem has an inverse. □

[8] Note that since V and W are finite-dimensional spaces, we can fix bases for each, and then in those coordinates, the transformation T can be represented by a matrix. T^T is the transpose of this matrix

8.1.1 Pre-Lab 8 Questions: Inverting Radiographic Transformations

In this section, we will consider the following example: We are given a radiograph with 24 pixels that was created by applying some radiographic transformation, T to an object with 16 voxels.

(a) (Review Question) Give a scenario for a radiographic transformation T that fits the above example. Don't calculate a matrix T, rather give the following:

- Size of the object: ____ × ____.
- Number of pixels per view:
- Number of views:

(b) Suppose we know the radiograph b and we want to find the object x, where $Tx = b$.

i. What properties must the *transformation* T have so that the one-sided inverse of T exists?

ii. When $N \leq M$ (as in the example above), what matrix properties must the *matrix representation* of T have so that it has a one-sided inverse?

(c) For ease of notation, we use the same notation for the matrix and the transformation, that is, we call the matrix representation of T, T. Suppose, $N \leq M$ and a one-sided inverse, P of T exists. This means that for all objects x with radiographs $Tx = b$, we have $x = Pb$. If T were invertible, then $P = T^{-1}$. But, in the example above, we know that T is not invertible (it is not even a square matrix!). Using the following steps, find the one-sided inverse of T.

i. One strategy is to compose T with a linear transformation $A : \mathbb{R}^M \to \mathbb{R}^N$ so that the product is invertible. (Can you see why this would be a good idea?) If $Tx = b$ and A is any linear transformation (whose domain is $\mathbb{R}^M$), then $ATx = Ab$. Since T is one-to-one, we know that in order for the composition $A \circ T = AT$ to be invertible, the only vector in $\text{ran}(T)$ that can be in $\text{null}(A)$ is the zero vector. (Think about this!) What other properties must A have so that AT is invertible?

ii. Provide a matrix A that has the properties you listed in 5(c)i and such that the composition AT is invertible.

iii. Solve for x in the matrix equation $ATx = Ab$ using the A you found, and provide a representation of the one-sided inverse P.

(d) Putting this all together now, state a necessary and sufficient condition for T to have a one-sided inverse.

8.2 Lab 8: Application to a Small Example

Consider the following radiographic example.

- Total number of voxels: $N = 16$ $(n = 4)$.
- Total number of pixels: $M = 24$
- $ScaleFac = 1$
- Number of views: $a = 6$
- View angles: $\theta_1 = 0°$, $\theta_2 = 20°$,$\theta_3 = 40°$,$\theta_4 = 60°$,$\theta_5 = 80°$,$\theta_6 = 100°$.

(a) Use tomomap.m to compute T and verify that the one-sided inverse of T must exist. Note that function tomomap returns a transformation matrix in sparse format. To use and view as a full matrix array use the command T=full(T); after constructing T.

(b) Compute the one-sided inverse P. Use P to find the object that created the following radiograph vector (by copying and pasting into Octave or Matlab from the electronic copy of this document in our class folder on the P: drive):

```
b=[0.00000
   32.00000
   32.00000
    0.00000
    1.97552
   30.02448
   30.02448
    1.97552
    2.71552
   29.28448
   29.28448
    2.71552
    2.47520
   29.52480
   29.52480
    2.47520
    1.17456
   30.82544
   30.82544
    1.17456
    1.17456
   30.82544
   30.82544
    1.17456]
```

Use the following code (with the modifications indicated!):

```
P=    ;  %insert your formula for P here!!
x=    ;  %insert what x is in terms of other quantities here!!
figure;
imagesc(reshape(x,a,b));  %change a and b to the dimensions of your object!
colormap(gray);
caxis([0 16]);
colorbar
```

Here are some notes about the commands in the code above:

- reshape(x,a,b) takes x and reshapes it into a $a \times b$ matrix.
- imagesc is a more general array plotting command than imshow. While imshow does a good job at plotting arrays with square pixel size and nice grayscale, it will plot small arrays too small to be seen!
- colormap(gray) tells imagesc to use a grayscale plot.
- caxis declares the grayscale limits, in just the same way that imshow does it. The difference is that it is a separate command with imagesc.
- colorbar forces the plot to contain a colorbar reference.

Application to Brain Reconstruction

Now, let's reconstruct some brain images from radiographic data. This section will guide you in this process.

(a) Collect the necessary provided files and place them in your working Octave/Matlab directory.

  i. Data File: Lab5Radiographs.mat

  ii. Plotting Script: ShowSlices.m

  iii. Octave/Matlab Code: tomomap.m

(b) Choose whether to create a new script file (".m" file) for all of your commands or to work at the Octave/Matlab prompt.

(c) Load the provided radiographic data with the following command.

```
load Lab5Radiographs.mat
```

This line loads an array variable named B which is a 12960x362 array. Each column is a radiograph corresponding to one horizontal slice of the human head. Each radiograph has 12960 total pixels spread across many views. The first 181 columns are noiseless radiographs and the last 181 columns are corresponding radiographs with a small amount of noise added.

(d) Use the familiar function tomomap.m to construct the transformation operator T corresponding to the following scenario (which is the scenario under which the radiographic data was obtained): $n = 108$, $m = 108$, $ScaleFac = 1$, and 120 view angles: the first at $1°$, the last at $179°$, and the rest equally spaced in between (hint: use the linspace command).

(e) Some Octave/Matlab functions do not work with sparse arrays (such as your T). So, simply just make T a full array with this command:

```
T=full(T);
```

(f) It is tempting to compute the one-sided inverse P as found in 5(c)iii of the first part. However, such a large matrix takes time to compute and much memory for storage. Instead we can use a more efficient solver provided by Octave/Matlab. If we seek a solution to $Lx = b$, for invertible matrix L, we find the unique solution by finding L^{-1} and then multiplying it by b. Octave/Matlab does both operations together in an efficient way (by not actually computing L^{-1}) when we use the command x=L\b.

Let's try this with the first radiograph in the matrix B. That is, we will reconstruct the slice of the brain that produced the radiograph that is represented in the 50th column of B. We want to solve the equation you found in 5(c)iii (recall the matrix you found to be A). Using the A you found in 5(c)iii in the following commands

```
b=B(:,50);
x=(A*T)\(A*b);
```

you will find your first brain slice reconstruction. Now to view your reconstruction, use the following commands

```
figure;
x=reshape(x,108,108);
imshow(x,[0,255]);
```

(The reshape command is necessary above because the result x is a $(108 \cdot 108) \times 1$ vector, but we want the object to be 108×108 image.)

(g) Notice also that x and b could be matrices, say X and B. In this case, each column of X is the unique solution (reconstruction) for the corresponding column of B. Use these ideas to reconstruct *all* 362 brain slices using a single Octave/Matlab command.

Use the variable name X. Make sure that X is an 11664x362 array. Now, the first 181 columns are reconstructions from noiseless data and the last 181 columns are reconstructions from noisy data.

(h) Run the script file ShowSlices.m which takes the variable X and plots example slice reconsrructions. Open ShowSlices.m in an editor and observe the line

```
slices=[50 90 130];
```

You can choose to plot any three slices by changing the slice numbers. In the figure, the left column of images are reconstructions from noiseless data, the right column of images are reconstructions from the corresponding noisy data. IMPORTANT: Once you have computed X you need only run ShowSlices.m to view different slices; running the other commands described above is time consumming and unnecessary.

Print out a picture of your favorite set of 3 slices to turn in with your work from this lab.

Noisy Data and Reconstructions

Congratulations! You have just performed your first brain scan tomography. Using your new tool, answer the following questions.

(a) Are the reconstructions from the noiseless radiographs exact discrete representations of the brain? Explain, using linear algebra concepts.

(b) Find a method for determining the relative amount of noise in the noisy radiographs. It might be useful to know that the noisy radiograph corresponding to the radiograph represented in the 50^{th} column of B is in the $(181 + 50)^{th}$ column of B. (Note: There are many ways you can do this, be creative.) Is there a lot or a little noise added to the radiographs?

(c) Are the reconstructions from the noisy radiographs exact representations of the brain?

(d) Compare the degree of "noisiness" in the noisy radiographs to the degree of "noisiness" in the corresponding reconstructions. Draw some conclusion about what this comparison tells us about the practicality of this method for brain scan tomography.

9 Extra Lab 9: Heat Diffusion in Linear Algebra Lab

This lab talks about Heat Diffusion and 1-D Heat Warping and their relationships with Linear Algebra[9]

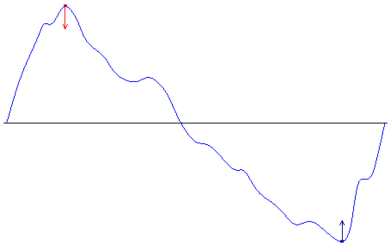
9.1 Pre-Lab 9: Motivating Application for Eigenvalues and Eigenvectors

9.1.1 Review of Heat States in a (Vector) State Space

Recall, we in our notes on Vector Space, we introduced the application of diffusion welding. A manufacturing company uses the process of diffusion welding to adjoin several smaller rods into a single longer rod. The diffusion welding process leaves the final rod heated to various temperatures along the rod with the ends of the rod having the same temperature. Every $a$cm along the rod, a machine records the temperature difference from the temperature at the ends to get heat states. We can represent the heat states of the rod as a vector space.

Heat States in One Dimension
Now consider the function $f : [a; b] \to \mathbb{R}$ that measures, along a rod of length $l = b - a$, the difference in temperature from the temperature at the ends of the rod ($f(a) = f(b) = 0$). The quantity $f(x)$ is the temperature of the rod at position x for any x in the interval $[a, b]$.

Notice that the temperature distribution evolves in time.
1. Places where the temperature is higher than the surrounding area will **cool**.
2. Places where the temperature is cooler than the surrounding area will **warm**.
3. As the heat diffuses, more extreme temperature differences disappear first.
4. The long term behavior is that temperatures smooth out approaching $f(x) = 0$ (where 0 represents room temperature).



We want to explore this application further and formulate our problem in a finite dimensional vector space. We will use our knowledge of linear algebra to compute the heat state at any later time as the heat diffuses.

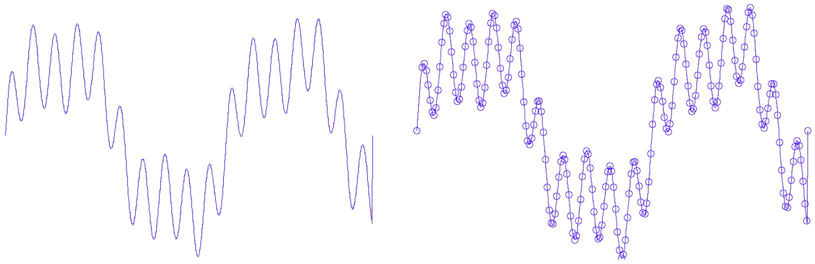
[9] ©2015-2018 T. Asaki, H. Moon, M. Snipes

9.1.2 Heat States in a (Vector) State Space

In the previous part of the discussion we modeled the heat along a bar by a continuous function f, which we call the heat state of the bar. We will *discretize* such a heat state f by sampling the temperature at m locations along the bar. If we space the m sampling locations equally, then for $\Delta x = \frac{L}{m+1} = \frac{b-a}{m+1}$, we can choose the sampling locations to be $a + \Delta x, a + 2\Delta x, \ldots, a + m\Delta x$. (We will assume that the heat state is known (and fixed) at the endpoints so we don't need to sample there!) Then, the discretized heat state is the following vector u in $\mathbb{R}^m$.

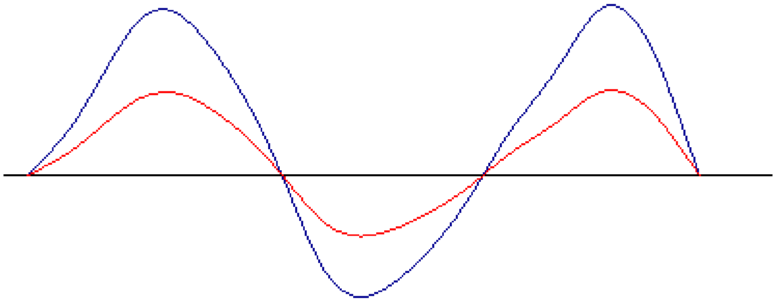
$$[u_1, u_2, ..., u_m] = [f(a + \Delta x), f(a + 2\Delta x), \ldots, f(a + m\Delta x)].$$

Notice that $f(a) = f(b) = 0$, where $b = a + (m+1)\Delta x$. Also, if $u_j = f(x)$ for some $x \in [a, b]$ then $u_{j+1} = f(x + \Delta x)$ and $u_{j-1} = f(x - \Delta x)$. The figure below shows a continuous heat state on the left and a heat state with sampling points marked on the right (although the initial and final points on the heat state are marked in the figure, they are not part of the vector u). A note about notation: in these notes we do not use special notation for vectors (such as bold typeface or arrows).

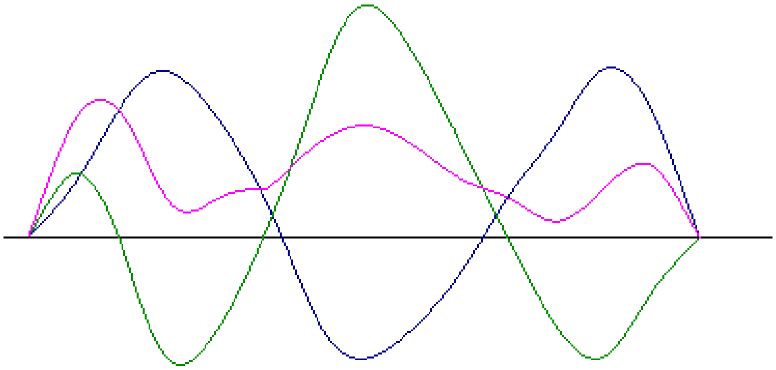


To complete our vector space development (we need the set of heat states to form a vector space so that we can use Linear Algebra in this setting), we must define heat state addition and scalar multiplication.

(a) We define scalar multiplication in the usual componentwise fashion. Scalar multiplication results in a change in **amplitude** only.



(b) We define vector addition in the usual componentwise fashion. Addition can result in changes in both amplitude and period.



9.1.3 The Heat Equation

This section may be of interest to students who have taken differential equations or are used to solving system of equations. We will briefly discuss the motivation for eigenvectors and values at the end before doing our Heat Equation Lab. The heat equation, or diffusion equation, (with fixed unit diffusion coefficient) is

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2}.$$

(a) A heat state vector f is a function of both position and time, though we often do not explicitly write $f(x,t)$ for clarity sake.

(b) The rate of change of temperature at any location x is proportional to the second derivative of the temperature with respect to position.

(c) Temperature is changing fastest at sharp peaks or dips in temperature where the curvature is large.

(d) Regions where temperature is changing slowly are where $|\partial^2 f(x)/\partial x^2|$ is small. This occurs under two different conditions:

   i. Where the temperature profile has nearly **constant** slope (no concavity).

   ii. Near **inflection** points of the temperature profile.

(e) The sign of the second derivative ensures that the heat state function does approach the zero function.

9.1.4 The Discrete Heat Equation, in x

Because derivatives are linear, intuitively we believe that it should be possible to get a good discrete approximation. So, we discretize the derivatives, using the definition of derivative

$$\frac{\partial f}{\partial x} = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} = \lim_{h\to 0} \frac{f(x)-f(x-h)}{h}.$$

Notice that in a discrete setting, it is not possible for $h \to 0$. The smallest h can be is Δx. Let's use this to make the matrix representation for the second derivative operator, which we call D_2. That is, $D_2 u$ approximates $\partial^2 f/\partial x^2$.

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial f}{\partial x} = \quad \tfrac{\partial}{\partial x}\left(\lim_{h\to 0}\tfrac{f(x+h)-f(x)}{h}\right) \approx \quad \frac{\partial}{\partial x}\left(\frac{f(x+\Delta x)-f(x)}{\Delta x}\right)$$

$$= \lim_{h\to 0}\left[\frac{\frac{f(x+h)-f(x)}{\Delta x} - \frac{f(x+\Delta x-h)-f(x-h)}{\Delta x}}{h}\right] \approx \tfrac{f(x+\Delta x)-f(x)-f(x+\Delta x-\Delta x)+f(x-\Delta x)}{(\Delta x)^2}$$

$$\approx \tfrac{1}{(\Delta x)^2}\left[f(x+\Delta x)-2f(x)+f(x-\Delta x)\right]$$

Notice that we have used both a forward and backward difference definition of the derivative in order to make our approximation symmetric. This helps us keep the later linear algebra manageable. In our discretization, we have

$$\frac{\partial^2 u_j}{\partial x^2} = \frac{1}{(\Delta x)^2}(u_{j+1} - 2u_j + u_{j-1}).$$

which is $\frac{1}{(\Delta x)^2}$(next heat state $-2\cdot$(current heat state)+previous heat state). Which makes sense the current heat state should depend on the heat states surrounding it.

Example 9.1. *Write the matrix representation for the second derivative operator, D_2 with respect to the standard basis using $D_2(u_j) = \frac{1}{(\Delta x)^2}(u_{j+1} - 2u_j + u_{j-1})$ for $\{u_1, u_2, ..., u_6\}$*[10]

Answer: Since $D_2(u_1) = [\frac{1}{(\Delta x)^2}(u_2 - 2u_1 + 0)]_U = \frac{1}{(\Delta x)^2} \cdot \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$D_2(u_2) = [\frac{1}{(\Delta x)^2}(u_3 - 2u_2 + u_1)]_U = \frac{1}{(\Delta x)^2} \cdot \begin{bmatrix} 1 \\ -2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

[10] Note: This means that if we want to take a discrete second derivative, we do the matrix multiply $D_2 u$.

$$D_2(u_3) = [\tfrac{1}{(\Delta x)^2}(u_4 - 2u_3 + u_2)]_U = \tfrac{1}{(\Delta x)^2} \cdot \begin{bmatrix} 0 \\ 1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Etc.

$$D_2 = \frac{1}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}.$$

9.1.5 The Discrete Heat Equation, in t

The heat state u also changes over time. So we can write $u = u(t)$, with a time dependence. That means we have from above $\frac{\partial^2 u(t)}{\partial x^2} = D_2 u(t)$.

Now, we use a similar argument to discretize $\frac{\partial u}{\partial t}$.

$$\frac{\partial u}{\partial t} \approx \frac{u(t + \Delta t) - u(t)}{\Delta t}.$$

We want to find the diffusion operator $E : \mathbb{R}^m \to \mathbb{R}^m$, for a fixed Δt, so that $u(t + \Delta t) = Eu(t)$ (That is, $u(t + \Delta t)$ is obtained via a matrix multiply). We put the above discretizations into the heat equation to obtain:

$$\frac{u(t + \Delta t) - u(t)}{\Delta t} = \frac{1}{(\Delta x)^2} D_2 u(t).$$

So,

$$u(t + \Delta t) = u(t) + \frac{\Delta t}{(\Delta x)^2} D_2 u(t) = \left(I + \frac{\Delta t}{(\Delta x)^2} D_2\right) u(t).$$

Let E be defined as the matrix $I + \frac{\Delta t}{(\Delta x)^2} D_2$. Then we have $u(t + \Delta t) = Eu(t)$. That is the next state is given by E times the current state.

Example 9.2. *Compose the matrix representation of E. Note $E = I_n + \frac{\Delta t}{(\Delta x)^2} D_2$ and let $\delta = \frac{\Delta t}{(\Delta x)^2}$ (which is a constant)*

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \\ & & \vdots & & \ddots \end{bmatrix} + \delta * \begin{bmatrix} -2 & 1 & 0 & 0 & \\ 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & 1 & \\ & & \vdots & & \ddots \end{bmatrix} = \begin{bmatrix} 1-2\delta & \delta & 0 & 0 & \\ \delta & 1-2\delta & \delta & 0 & \dots \\ 0 & \delta & 1-2\delta & \delta & \\ & & \vdots & & \ddots \end{bmatrix}$$

Technical Detail: We need $0 < \delta \equiv \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{4}$ for computational stability, though the proof is beyond the scope of this module. Δx is fixed so we need to take small enough time steps Δt to get this inequality. Hence as we let $\Delta x \to 0$, we are also implicitly forcing $\Delta t \to 0$.

So $u(t + \Delta t) = E \cdot u(t) = \begin{bmatrix} 1-2\delta & \delta & 0 & 0 & \\ \delta & 1-2\delta & \delta & 0 & \dots \\ 0 & \delta & 1-2\delta & \delta & \\ & & \vdots & & \ddots \end{bmatrix} \cdot u(t)$

Consider the meaning of the row and column representations of E.

(a) How can we interpret the values in column k of E?
Answer: The kth column shows how the heat at time t distributes to heat at time $t + \Delta t$ at location k in the heat state. In particular, fraction $1 - 2\delta$ of the heat at location k remains at location k and fraction δ of the heat moves to each of the two nearest neighbor locations $k + 1$ and $k - 1$.

(b) How can we interpret the values in row k of E?
Answer: The kth row shows where the heat at time $t + \Delta t$ came from. In particular, fraction $1 - 2\delta$ of the heat at location k was already at location k and fraction δ came from each of the location's two nearest neighbors.

(c) How can we interpret the fact that all but the first and last columns (and rows) sum to unity (1)?
Answer: Based on the above discussion, we see that this guarantees that no heat is lost from the system *except at the end points*. Heat is redistributed (diffused) not lost.

9.1.6 Heat State Evolution

Now, we consider the evolution over more than one time step.

$$\begin{aligned} u(t + \Delta t) &= Eu(t) \\ u(t + 2\Delta t) &= Eu(t + \Delta t) \ = \ E(Eu(t)) \ = \ E^2 u(t) \end{aligned}$$

This means k time steps later,

$$u(t + k\Delta t) = E^k u(t)$$

Example 9.3. *We found the matrix representation of the heat diffusion operator to be*

$$E = \begin{bmatrix} 1-2\delta & \delta & 0 & 0 & \\ \delta & 1-2\delta & \delta & 0 & \dots \\ 0 & \delta & 1-2\delta & \delta & \\ & & \vdots & & \ddots \end{bmatrix}.$$

We see that if we are want to find the kth heat state, $u(k\Delta t)$ in the heat diffusion, we need to use

$$u(k\Delta t) = E^k u(0),$$

where $u(0)$ is the initial heat state. Let's explore what happens when we do these calculations.

(a) What is the equation for the heat state 2 steps in the future? 1000 time steps in the future?

(b) Note if we find E^2 or $(E \cdot E)$, we get

$$E^2 = \begin{bmatrix} (1-2\delta)^2+\delta^2 & 2\delta(1-2\delta) & \delta^2 & 0 & \dots \\ 2\delta(1-2\delta) & (1-2\delta)^2+\delta^2 & 2\delta(1-2\delta) & \delta^2 & 0 \\ \delta^2 & 2\delta(1-2\delta) & (1-2\delta)^2+\delta^2 & 2\delta(1-2\delta) & \dots \\ 0 & \delta^2 & 2\delta(1-2\delta) & (1-2\delta)^2+\delta^2 & \\ \vdots & \vdots & \vdots & \vdots & \dots \end{bmatrix}$$

(c) Does it look like raising E to higher and higher powers is an easy way to find the heat state at some time far in the future (for example, 1000 time steps away)?

(d) Does it look like multiplying over and over by the matrix E is an easy way to find the heat state at some time far in the future?

(e) If you had to perform this kind of operation, what characteristics would you prefer a diffusion operator to have?

Now let's explore morphing images or heat states.

9.1.7 In-Class PreLab Demonstrations Heat Diffusion

Recall the properties of heat diffusion:

(a) Places where the temperature is higher than the surrounding area will cool.

(b) Places where the temperature is cooler than the surrounding area will warm.

(c) As the heat diffuses, sharpest temperature differences (higher frequency wiggles) disappear first.

(d) The long-term behavior is that temperatures smooth out approaching 0 heat everywhere.

Demo 1: We will now show an example diffusion of a heat state (using `HeatEqnClassDemos.m`). In Matlab or Octave, type HeatEqnClassDemos(1)

Note: There is a 10 second pause that allows the class to see the original heat signature. We can use this time to prepare to check the conjectures made above. This demo completes 500 iterations. We should watch for more sharply curving (actually called high frequency) areas to diffuse away faster than low frequency areas. Another way to say this is that small scale features disappear sooner than larger scale features.

Demo 2: Now let's look at a demo which is a better match for the scenario of diffusion welding: HeatEqnClassDemos(2)

Question: What happens if a heat source (or several) are placed in a room: how will the heat diffuse throughout the room?

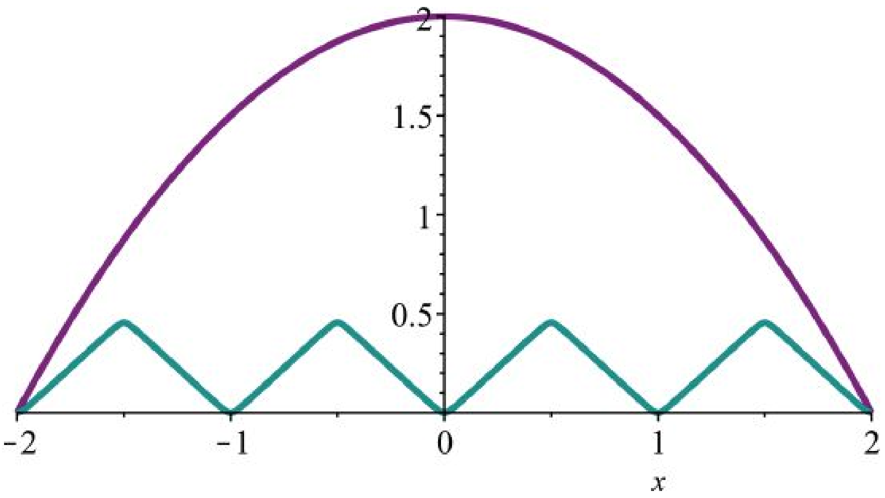
The idea of diffusion can be used to morph one image into another. The video `USA.mp4` uses diffusion to transition from image to image.

The ideas behind the movie creation are explored in the 1-dimensional setting in the following Lab.

9.2 Lab 9: Morphing 1-D heat signatures

Suppose that we start with two heat signatures, $\mathbf{v}$ and $\mathbf{w}$, and we want to heat evolve one into the other. One way to do this is to heat evolve *the difference* between the two heat signatures $\mathbf{v} - \mathbf{w}$.

In the following activity we will explore this process with the heat signatures v and w pictured below. (In the figure, v is the upper function in purple and w is the lower function in blue.)



(a) On a new set of axes, sketch the difference $\mathbf{v} - \mathbf{w}$.

(b) Use your intuition about heat diffusion to sketch some (at least 3) future heat signatures that might arise from evolving $\mathbf{v} - \mathbf{w}$ for different lengths of time. (Add these to your sketch from the previous part). We can give names to the functions you drew. Define a family of heat signatures $\mathbf{x}_t$, where $\mathbf{x}_0 = \mathbf{v} - \mathbf{w}$ is the initial heat signature, and $\mathbf{x}_t$ is the heat signature after time t. (The heat signatures in your drawings that occur earlier in the evolution are associated with lower values of t than the ones that occur later.)

(c) Add a sketch of the limiting heat signature, $\mathbf{x}_\infty$, that results from evolving $\mathbf{v} - \mathbf{w}$. We can define the heat signature at time t to be $\mathbf{x}_t$, creating a family of heat signatures for $t \geq 0$ where

(d) Now let's think about how the sketches you've made from the previous part might help you to morph the function $\mathbf{v}$ into the function $\mathbf{w}$. Consider the related family of heat signatures, where for $t \geq 0$, $\mathbf{z}_t := \mathbf{w} + \mathbf{x}_t$. On a new set of axes, sketch the graphs of $\mathbf{z}_t$ that correspond to the $\mathbf{x}_t$ you sketched above. (Also include $\mathbf{x}_0$ and $\mathbf{x}_\infty$.)

(e) Fill in the blanks: the family $\mathbf{z}_t$ for $t \geq 0$ morphs the heat signature ____________________ into the heat signature ____________________.

(f) The family of functions $\mathbf{z}_t$ shows a heat signature morphing into another heat signature. What do you notice about this morphing? In particular, what types of features disappear and appear?

(g) Another method for morphing heat signature $\mathbf{v}$ into heat signature $\mathbf{w}$ is to first diffuse $\mathbf{v}$ and then follow this with the inverse of the diffusion of $\mathbf{w}$. How is this different from what we obtained above by diffusing the signature $\mathbf{v}-\mathbf{w}$? (You may produce sketches of the heat evolutions of $\mathbf{v}$ and $\mathbf{w}$ to support your discussion.)

Morphing Images

We now consider the task of morphing a 2-dimensional black and white image into another image, like you saw in the demo[11].

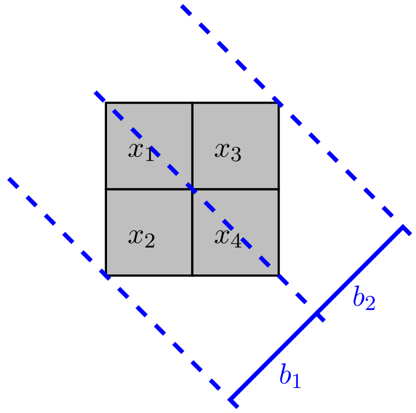
Such an image can be thought of as consisting of pixels in a rectangular array with intensity values assigned to each pixel: white pixels have high intensity (heat) and black pixels have low intensity (heat). As in the one-dimensional case, we will assume the boundary intensities are zero. Boundary pixels are those on the border of the rectangle.

The principles behind heat diffusion in two dimensions are similar to those we have been discussing in one dimension. In particular,

- Places where the temperature is higher than the surrounding area will cool.
- Places where the temperature is cooler than the surrounding area will warm.
- As the heat diffuses, sharpest temperature differences disappear first.
- The long-term behavior is that temperatures smooth out approaching 0 heat everywhere.

Hence if we take an image and heat diffuse it, the final heat state of the image will be the blank (zero) image. We define a morphing between image V and image W as in the one-dimensional case above: by diffusing the difference $V-W$ and considering the associated family X_t as before.

[11] Recall that the demo actually morphed color images; this was accomplished by morphing the red, green, and blue components of the image separately and then combining them.


x_1
x_3
x_2
x_4
b_2
b_1