EECS 16A Designing Information Devices and Systems I
Spring 2018 Homework 2

This homework is due February 5, 2018, at 23:59.
Self-grades are due February 8, 2018, at 23:59.

Submission Format
Your homework submission should consist of **two** files.

- `hw2.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a PDF.

  If you do not attach a PDF "printout" of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible. Assign the IPython printout to the correct problem(s) on Gradescope.

- `hw2.ipynb`: A single IPython notebook with all of your code in it.

Submit each file to its respective assignment on Gradescope.

1. (PRACTICE) Powers Of Nilpotent Matrices

Do this problem if you would like more mechanical practice with matrix multiplication.
The following matrices are examples of a special type of matrix called a nilpotent matrix. What happens to each of these matrices when you multiply it by itself four times? Multiply them to find out. Why do you think these are called "nilpotent" matrices? (Of course, there is nothing magical about 4×4 matrices. You can have nilpotent square matrices of any dimension greater than 1.)

(a) Calculate $\mathbf{A}^4$ by hand. Make sure you show what $\mathbf{A}^2$ and $\mathbf{A}^3$ are along the way.

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

(b) Calculate $\mathbf{B}^4$ by hand. Make sure you show what $\mathbf{B}^2$ and $\mathbf{B}^3$ are along the way.

$$\mathbf{B} = \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix}$$

2. Elementary Matrices

This week, we learned about an important technique for solving systems of linear equations called Gaussian elimination. It turns out that each row operation in Gaussian elimination can be performed by multiplying the augmented matrix on the left by a specific matrix called an *elementary matrix*. For example, suppose we want to row reduce the following augmented matrix:

$$
\mathbf{A} = \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 15 \\ 0 & 1 & 0 & 3 & -7 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right] \tag{1}
$$

What matrix do you get when you subtract the 4th row from the 2nd row of **A** (putting the result in row 2)? (You don't have to include this in your solution.) Now, try multiplying the original **A** on the left by

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

(You don't have to include this in your solutions either.) Notice that you get the same thing.

$$
\mathbf{EA} = \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 15 \\ 0 & 0 & 0 & 1 & -2 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right]
$$

E is a special type of matrix called an *elementary matrix*. This means that we can obtain the matrix **E** from the identity matrix by applying an elementary row operation – in this case, subtracting the 4th row from the 2nd row.

In general, any elementary row operation can be performed by left multiplying by an appropriate elementary matrix.

In other words, you can perform a row operation on a matrix **A** by first performing that row operation on the identity matrix to get an elementary matrix (see below), and then left multiplying **A** by the elementary matrix (like we did above).

$$
\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_2 - R_4 \mapsto R_2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{E}
$$

(a) Write down the elementary matrices required to perform the following row operations on a 4×5 augmented matrix.

- Switching rows 1 and 3
- Multiplying row 3 by -5
- Adding $3 \times$row 2 to row 4 (putting the result in row 4) and subtracting row 2 from row 1 (putting the result in row 1)

Hint: For the last one, note that if you want to perform two row operations on the matrix ***A****, you can perform them both on the identity matrix and then left multiply* ***A*** *by the resulting matrix.*

(b) Now, compute a matrix **E** (by hand) that fully row reduces the augmented matrix **A** given in Equation 1 – that is, find **E** such that **EA** is in reduced row echelon form. Show that this is true by multiplying out **EA**. When an augmented matrix is in reduced row echelon form, it will have the form

$$\mathbf{EA} = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & b_1 \\ 0 & 1 & 0 & 0 & b_2 \\ 0 & 0 & 1 & 0 & b_3 \\ 0 & 0 & 0 & 1 & b_4 \end{array}\right]$$

Once you have found the required elementary matrices, plug them into the iPython notebook to get the matrix **E**. Verify by hand that multiplying **E** and **A** gives you the identity matrix augmented with constants (as **EA** shown above).

Hint: As before, note that you can either ***apply a set of row operations to the same identity matrix*** *or* ***apply them to separate identity matrices and then multiply the matrices together****. Make sure, though, that you apply the row operations and multiply the matrices in the correct order.*

3. Small angle optics

Physical rules governing optics are non-linear in general. For example, refraction of light is governed by the following law, called Snell's law,

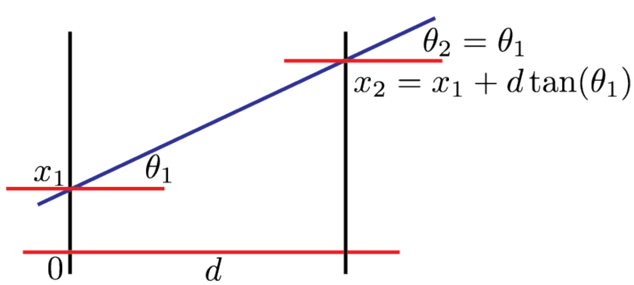
$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2),$$

where n_1 and n_2 are the refractive indices of the input and output media respectively, and θ_1 and θ_2 are the angle the light makes with respect to the normal of the surface, for incoming and outgoing light respectively. As can be seen, the input and output angles are a non-linear function of each other.

However, for small enough angles, we have the following relations, which we call the small angle approximation

$$\sin(\theta) \approx \theta,$$
$$\cos(\theta) \approx 1,$$
$$\tan(\theta) \approx \theta.$$

To see how the small angle approximation is used, say we have a light ray at position x_1 and traveling at angle θ_1 with respect to a chosen axis, which we call the optical axis. After the light travels length d with respect to the optical axis, the position and angle become:



$$x_2 = x_1 + d\tan(\theta_1),$$
$$\theta_2 = \theta_1.$$

Using the small angle approximation we have

$$x_2 = x_1 + d\theta_1,$$
$$\theta_2 = \theta_1.$$

Now we can represent these equations as a matrix operation:

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}.$$

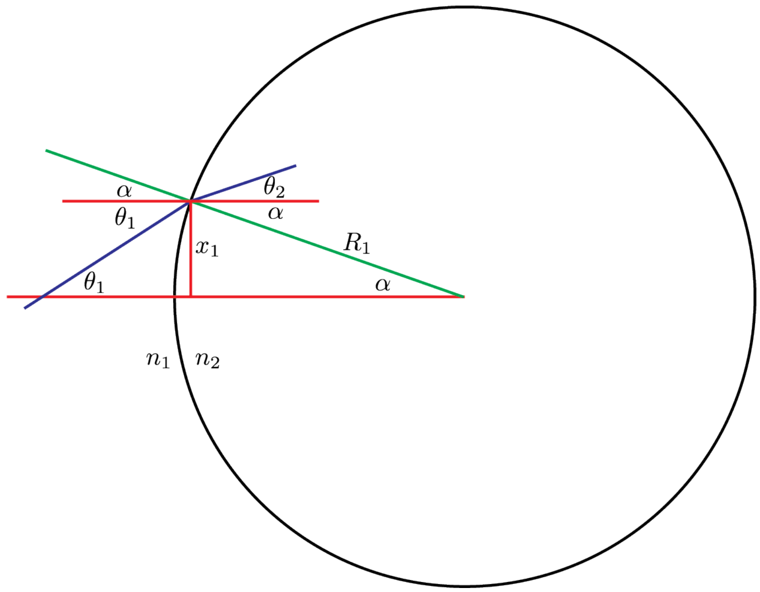


Figure 1: Front surface of the thin lens.

Another common optical event is refraction of light at a spherical surface. (Lenses are frequently have a spherical profile since this is easier to manufacture.) For the refraction at the front surface of a spherical lens, shown in Figure 1 we have

$$x_2 = x_1$$
$$n_1 \sin(\theta_1 + \alpha) = n_2 \sin(\theta_2 + \alpha).$$

Using the small angle approximations, the equation becomes

$$n_1(\theta_1 + \alpha) = n_2(\theta_2 + \alpha)$$

Furthermore we can approximate $\alpha = \sin(\alpha) = x/R_1$

$$\theta_2 = \frac{n_1 - n_2}{n_2 R_1} x_1 + \frac{n_1}{n_2} \theta_1.$$

We can represent this equation, along with the $x_1 = x_2$ equation, in matrix form:

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{n_1 - n_2}{n_2 R_1} & \frac{n_1}{n_2} \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix} = A_1 \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}.$$

(a) Using similar steps of the above derivation derive the equation for the back curved surface of a spherical lens, shown in Figure 2. That is, find the matrix A_2 giving

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = A_2 \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}.$$

In particular looking at Figure 2, we see that

$$n_2 \sin(\theta_i) = n_1 \sin(\theta_o). \tag{2}$$

Write θ_i as a function of θ_1 and α, and θ_o as a function of θ_2 and α. Substitute inside equation 2 and use the small angle approximations. Then use $\alpha = \sin(\alpha) = x_1/R_2$.

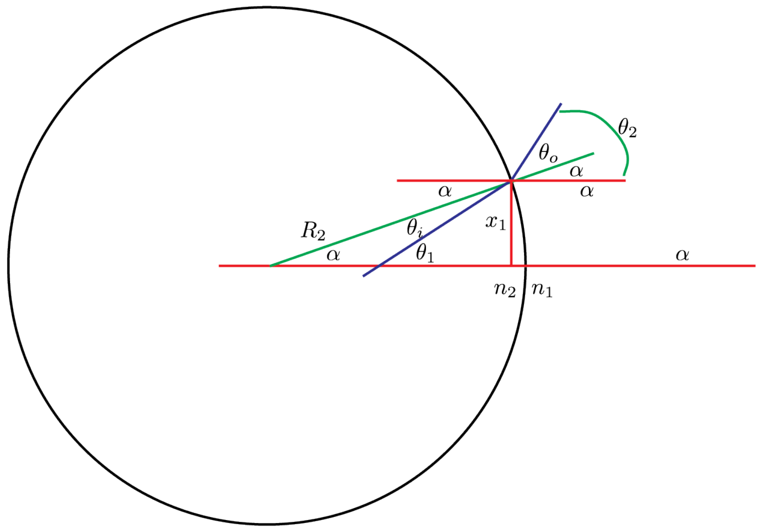


Figure 2: Back surface of the thin lens.

(b) Multiplying the two matrices as A_2A_1, find the transfer function of the thin lens as a whole.

(c) We want to find the focal length of the thin lens. Focal length is the point behind the lens where any ray incident at zero degrees focuses at $x_2 = 0$ independent of x_1. To do that, find d such that x_2 given by the following equation

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} A_2 A_1 \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$$

is exactly 0 irrespective of x_1.

Compare your results to wikipedia article

https://en.wikipedia.org/wiki/Thin_lens

Are you happy with your results?

4. Show It

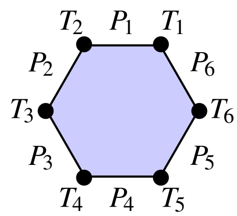
Let n be a positive integer. Let $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k\}$ be a set of k linearly dependent vectors in $\mathbb{R}^n$. Show that for any $n \times n$ matrix $\mathbf{A}$, the set $\{\mathbf{A}\vec{v}_1, \mathbf{A}\vec{v}_2, \ldots, \mathbf{A}\vec{v}_k\}$ is a set of linearly dependent vectors. Make sure that you prove this rigorously for all possible matrices $\mathbf{A}$.

5. Figuring Out The Tips

A number of people gather around a round table for a dinner. Between every adjacent pair of people, there is a plate for tips. When everyone has finished eating, each person places half their tip in the plate to their left and half in the plate to their right. In the end, of the tips in each plate, some of it is contributed by the person to its right, and the rest is contributed by the person to its left. Suppose you can only see the plates of tips after everyone has left. Can you deduce everyone's individual tip amounts?

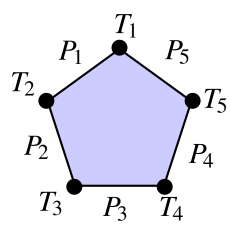
Note: For this question, if we assume that tips are positive, we need to introduce additional constraints enforcing that, and we wouldn't get a linear system of equations. Therefore, we are going to ignore this constraint and assume that negative tips are acceptable.

(a) Suppose 6 people sit around a table and there are 6 plates of tips at the end.



If we know the amounts in every plate of tips (P_1 to P_6), can we determine the individual tips of all 6 people (T_1 to T_6)? If yes, explain why. If not, give two different assignments of T_1 to T_6 that will result in the same P_1 to P_6.

(b) The same question as above, but what if we have 5 people sitting around a table?



(c) If n is the total number of people sitting around a table, for which n can you figure out everyone's tip? You do not have to rigorously prove your answer.

6. Image Stitching

Often, when people take pictures of a large object, they are constrained by the field of vision of the camera. This means that they have two options how they can capture the entire object:

- Stand as far as away as they need to to include the entire object in the camera's field of view (clearly, we do not want to do this as it reduces the amount of detail in the image)
- (This is more exciting) Take several pictures of different parts of the object, and stitch them together, like a jigsaw puzzle.

We are going to explore the second option in this problem. Daniel, who is a professional photographer, wants to construct an image by using "image stitching". Unfortunately, Daniel took some of the pictures from different angles as well as from different positions and distances from the object. While processing these pictures, Daniel lost information about the positions and orientations from which the pictures were taken. Luckily, you and your friend Marcela, with your wealth of newly acquired knowledge about vectors and rotation matrices, can help him!

You and Marcela are designing an iPhone app that stitches photographs together into one larger image. Marcela has already written an algorithm that finds common points in overlapping images and it's your job to figure out how to stitch the images together. You recently learned about vectors and rotation matrices in EE16A, and you have an idea about how to do this.

Your idea is that you should be able to find a single rotation matrix, $\mathbf{R}$, which is a function of some angle, θ, and a translation vector, $\vec{T}$, that transforms every common point in one image to that same point in the other image. Once you find the the angle, θ, and the translation vector, $\vec{T}$, you will be able to transform one image so that it lines up with the other image.

Suppose $\vec{p}$ is a point in one image and $\vec{q}$ is the corresponding point (i.e., they represent the same thing in the scene) in the other image. You write down the following relationship between $\vec{p}$ and $\vec{q}$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}(\theta)} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

This looks good, but then you realize that one of the pictures might be farther away than the other. You realize that you need to add a scaling factor, $\lambda > 0$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \lambda \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{3}$$

(For example, if $\lambda > 1$, then the image containing q is closer (appears larger) than the image containing p. If $0 < \lambda < 1$, then the image containing q appears smaller.)

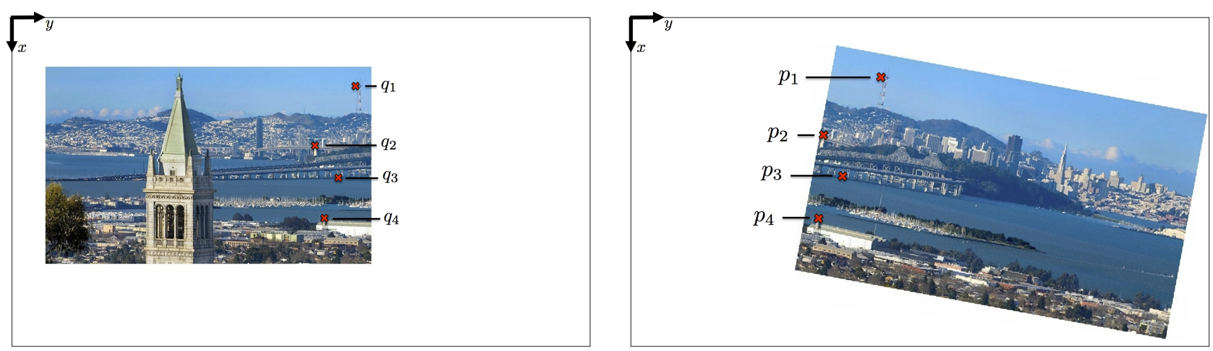


Figure 3: Two images to be stitched together with pairs of matching points labeled.

You are now confident that if you can find θ, $\vec{T}$, and λ, you will be able to reorient and scale one of the images, so that it lines up with the other image.

Before you get too excited, however, you realize that you have a problem. Equation 3 is not a linear equation with respect to θ, $\vec{T}$, and λ. You're worried that you don't have a good technique for solving nonlinear systems of equations. You decide to talk to Marcela and the two of you come up with a brilliant solution.

You decide to "relax" the problem, so that you're solving for a general matrix $\mathbf{R}$ rather than a perfect scaled rotation matrix. The new equation you come up with is:

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{4}$$

This equation is linear, so you can solve for $R_{xx}, R_{xy}, R_{yx}, R_{yy}, T_x$, and T_y. Also you realize that if $\vec{p}$ and $\vec{q}$ actually do differ by a rotation of θ degrees and a scaling of λ, you can expect that the general matrix $\mathbf{R}$ that you find will turn out to be a scaled rotation matrix with $R_{xx} = \lambda\cos(\theta)$, $R_{xy} = -\lambda\sin(\theta)$, $R_{yx} = \lambda\sin(\theta)$, and $R_{yy} = \lambda\cos(\theta)$.

(a) Multiply Equation 4 out into two scalar linear equations. What are the known values and what are the unknowns in each equation? How many unknowns are there? How many equations do you need to solve for all the unknowns? How many pairs of common points $\vec{p}$ and $\vec{q}$ will you need in order to write down a system of equations that you can use to solve for the unknowns?

(b) Write out a system of linear equations that you can use to solve for the values of $\mathbf{R}$ and $\vec{T}$.

(c) In the IPython notebook `prob2.ipynb`, you will have a chance to test out your solution. Plug in the values that you are given for p_x, p_y, q_x, and q_y for each pair of points into your system of equations to solve for the parameters $\mathbf{R}$ and $\vec{T}$. You will be prompted to enter your results, and the notebook will then apply your transformation to the second image and show you if your stitching algorithm works.

(d) We will now explore when this algorithm fails. For example, the three pairs of points must all be distinct points. Show that if $\vec{p}_1, \vec{p}_2, \vec{p}_3$ are *collinear*, the system of equations (4) is underdetermined. Does this make sense geometrically?
(Think about the kinds of transformations possible by a general affine transformation. An affine transformation is one that preserves points. For example, in the rotation of a line, the angle of the line might change, but the length will not. All linear transformations are affine. **Definition of Affine.**)
Use the following fact: $\vec{p}_1, \vec{p}_2, \vec{p}_3$ are collinear iff $(\vec{p}_2 - \vec{p}_1) = k(\vec{p}_3 - \vec{p}_1)$ for some $k \in \mathbb{R}$.

(e) **(PRACTICE)** Show that if the three points are not collinear, the system is fully determined.

(f) **(PRACTICE)** Marcela comments that perhaps the system (with three collinear points) is only underdetermined because we "relaxed" our model too much by allowing for general affine transformations, instead of just isotropic-scale/rotation/translation. Can you come up with a different representation of Equation 3, that will allow for recovering the transform from only *two* pairs of distinct points?
(*Hint: Let* $a = \lambda\cos(\theta)$ *and* $b = \lambda\sin(\theta)$. *In other words, enforce* $R_{xx} = R_{yy}$ *and* $R_{xy} = -R_{yx}$).

7. Homework Process and Study Group

Who else did you work with on this homework? List names and student ID's. (In case of homework party, you can also just describe the group.) How did you work on this homework?