

2. Systems of Linear Equations

§2.1 Introduction

No doubt you have encountered linear equations many times during your careers as students. Most likely, the very first equation you had to solve was linear. Yet despite their simplicity, systems of linear equations are of immense importance in mathematics and its applications to areas in the physical sciences, economics, engineering and many, many more. One of the purposes of linear algebra is to undertake a systematic study of linear equations. In this lab, we will use MATLAB to solve systems of linear equations. We will also learn about a very useful application of systems of linear equations to economics and computer science.

§2.2 Systems of Linear Equations

By now we have seen how a system of linear equations can be transformed into a matrix equation, making the system easier to solve. For example, the system

$$\begin{aligned} 2x_1 - x_2 + x_3 &= 8 \\ x_1 + 2x_2 + 3x_3 &= 9 \\ 3x_1 - x_3 &= 3 \end{aligned}$$

can be written the following way:

$$\begin{bmatrix} 2 & -1 & 1 \\ 1 & 2 & 3 \\ 3 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \\ 3 \end{bmatrix}$$

Now, by augmenting the matrix with the vector on the right and using row operations, this equation can easily be solved by hand. However, if our system did not have nice integer entries, solving the system by hand using row reduction could become very difficult. MATLAB provides us with an easier way to get an answer.

A system of this type has the form $Ax = b$, so we can enter these numbers into MATLAB using the following commands:

```
>> A = [2 -1 1; 1 2 3; 3 0 -1]
>> b = [8; 9; 3]
```

(Notice that for the column vector b , we include semicolons after each entry to ensure that the entries are on different rows; the command

```
>> b = [8 9 3]
```

would produce a row vector, which is not the same thing.)

The command

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

will find the solution (if it exists) to our equation $Ax = b$. In this case, MATLAB tells us

```
x =
    2.0000
   -1.0000
    3.0000
```

Remark 2.1 Please take care in entering "A\b" command. It has a backslash "\", NOT a forward slash "/".

Exercise 2.1

(a) Consider the system of equations:

$$\begin{aligned} 2x_1 + x_2 + 5x_3 &= -1 \\ x_1 + 6x_3 &= 2 \\ -6x_1 + 2x_2 + 4x_3 &= 3 \end{aligned}$$

On paper, convert this system of equations into a matrix equation of the form $Cx = d$.

(b) Enter the matrix C and the column vector d into MATLAB, and use the command

```
>> x = C\d
```

to check your solution.

(c) We would expect to get the column vector d in MATLAB if we ran the command $C*x$, right? In other words, $C*x - d$ should be zero. Enter this equation into MATLAB:

```
>> C*x - d
```

What do you get?

Remark 2.2 The discrepancy in part (c) of the above exercise is simply due to rounding error. You'll notice that the error is a vector multiplied by a very small number, one on the order of 10^{-15} . But why is there any error at all? After all, solving by row reduction gave very nice numbers, right? The reason lies in the way MATLAB stores numbers. In this calculation MATLAB represents numbers in "floating point form", which means it represents them in scientific notation to 10^{-14} accuracy. Thus when you see 10^{-14} printed in calculations, it is equivalent to zero.

There is a drawback, however, to solving systems of equations using the command "x = C\d". Let us explore that further.

Exercise 2.2 Consider the following system of equations:

$$\begin{aligned} -10x_1 + 4x_2 &= 0 \\ 15x_1 - 6x_2 &= 0 \end{aligned}$$

As you did in the previous exercise, enter the corresponding matrix C and the column vector d into MATLAB. Then type in

```
>> x = C\d
```

Note the strange output. Include it in your write up. Now go ahead and solve this system by hand. How many free variables do you need to write down the solution? Based on your answer, can you explain why you got the error message when trying to use "x = C\d" command?

To deal with the case of inconsistent systems or systems with infinitely many solutions, it may sometimes be better to use MATLAB to simply row-reduce your matrix and then read off the solutions. We happen to be very lucky since MATLAB has a command that performs Gaussian elimination for you.

Consider the following homogeneous system of equations:

$$\begin{aligned} 6x_1 + 2x_2 + 3x_3 &= 0 \\ 4x_1 + x_2 - 2x_3 &= 0 \\ 2x_1 + x_2 + 5x_3 &= 0 \end{aligned}$$

Enter the corresponding matrix C and the column vector d into MATLAB. Now we want to perform row reduction on the augmented matrix $[C \mid d]$. The command that performs row reduction in MATLAB is called "rref" (it stands for "reduced row echelon form"). Then type in

```
>> rref([C d])
ans =
    1.0000     0    -3.5000     0
         0     1.0000    12.0000     0
         0         0         0         0
```

Recall from class that we can then use the row reduced form to get that

$$\begin{aligned} x_1 &= 3.5x_3 \\ x_2 &= -12x_3 \end{aligned}$$

x_3 here is the free variable, and it can be chosen to be any value. Once chosen, the two equations above determine the values of x_1 and x_2 . For instance, if we choose $x_3 = 2$, then $x_1 = 7$ and $x_2 = 24$. In this way we can write down all the solutions (of which there are infinitely many) to the system above.

Exercise 2.3

(a) Consider the following homogeneous system of equations:



$$\begin{aligned} x_1 - 3x_2 + 2x_3 &= 0 \\ -2x_1 + 6x_2 - 4x_3 &= 0 \\ 4x_1 - 12x_2 + 8x_3 &= 0 \end{aligned}$$

By using the rref command, write down the general solution to this system of equations. How many free variables are required?

Next we turn to some applications of linear algebra, which will involve word problems. Often in a linear algebra course such problems are assigned from the textbook and are to be done using pencil and paper. This can be quite tedious. Hopefully with your burgeoning MATLAB expertise you will be able to see that the mathematical models we will explore in this course work even for systems with hundreds of variables.

§2.3 An Application to Economics: Leontief Models

[Wassily Leontief](#) (1906-1999) was a Russian-born, American economist who, aside from developing highly sophisticated economic theories, also enjoyed trout fishing, ballet and fine wines. He won the 1973 Nobel Prize in economics for his work in creating mathematical models to describe various economic phenomena. In the remainder of this lab we will look at a very simple special case of his work called a closed exchange model. Here is the premise:

Suppose in a far away land of Eigenbazistan, in a small country town called Matrixville, there lived a Farmer, a Tailor, a Carpenter, a Coal Miner and Slacker Bob. The Farmer produced food; the Tailor, clothes; the Carpenter, housing; the Coal Miner supplied energy; and Slacker Bob made High Quality 100 Proof Moonshine, half of which he drank himself. Let us make the following assumptions:

- Everyone buys from and sells to the central pool (i.e. there is no outside supply and demand)
- Everything produced is consumed

For these reasons this is called a *closed* exchange model. Next we must specify what fraction of each of the goods is consumed by each person in our town. Here is a table containing this information:

	Food	Clothes	Housing	Energy	High Quality 100 Proof Moonshine
Farmer	0.25	0.15	0.25	0.18	0.20
Tailor	0.15	0.28	0.18	0.17	0.05
Carpenter	0.22	0.19	0.22	0.22	0.10
Coal Miner	0.20	0.15	0.20	0.28	0.15
Slacker Bob	0.18	0.23	0.15	0.15	0.50

So for example, the Carpenter consumes 22% of all food, 19% of all clothes, 22% of all housing, 22% of all energy and 10% of all High Quality 100 Proof Moonshine.



Exercise 2.4 Note that the columns in this table all add up to 1. Explain why this happens.

Now, let $p_F, p_T, p_C, p_{CM}, p_{SB}$ denote the incomes of the Farmer, Tailor, Carpenter, Coal Miner and Slacker Bob, respectively. Note that each of these quantities not only denotes the incomes of each of our esteemed citizens, but also the cost of the corresponding goods. So for example, p_F is the Farmer's income as well as the cost of all the food. So if the Farmer produces \$100 worth of food, then his income will also be \$100 since all of this food is bought out and the profits go to the Farmer.

The idea is, of course, to be able to figure out how should we price the goods in order for the citizens of Matrixville to survive; i.e. we must find p_F, p_T, p_C, p_{CM} , and p_{SB} subject to the following conditions:

$$\begin{aligned} 0.25p_F + 0.15p_T + 0.25p_C + 0.18p_{CM} + 0.20p_{SB} &= p_F \\ 0.15p_F + 0.28p_T + 0.18p_C + 0.17p_{CM} + 0.05p_{SB} &= p_T \\ 0.22p_F + 0.19p_T + 0.22p_C + 0.22p_{CM} + 0.10p_{SB} &= p_C \\ 0.20p_F + 0.15p_T + 0.20p_C + 0.28p_{CM} + 0.15p_{SB} &= p_{CM} \\ 0.18p_F + 0.23p_T + 0.15p_C + 0.15p_{CM} + 0.50p_{SB} &= p_{SB} \end{aligned}$$



Exercise 2.5 Explain where this system of equations came from and what it means. (Think about what the left hand side and the right hand side of each equation mean.)

Let us denote the column vector $(p_F, p_T, p_C, p_{CM}, p_{SB})^T$ by \mathbf{p} , and let \mathbf{C} be the coefficient matrix of the above system. We can now rewrite that system as

$$\mathbf{C}\mathbf{p} = \mathbf{p}$$

or equivalently

$$\mathbf{C}\mathbf{p} - \mathbf{p} = \mathbf{C}\mathbf{p} - \mathbf{I}\mathbf{p} = (\mathbf{C} - \mathbf{I})\mathbf{p} = \mathbf{0}$$

where \mathbf{I} is the 5×5 matrix with 1's on the diagonal and 0's everywhere else. The property of the matrix \mathbf{I} we are using is that $\mathbf{I}\mathbf{v} = \mathbf{v}$ for any vector \mathbf{v} .

Exercise 2.6

	<p>(a) Enter the matrices \mathbf{C} and \mathbf{I} into MATLAB.</p> <pre>>> C = [0.25 0.15 0.25 0.18 0.20; 0.15 0.28 0.18 0.17 0.05; 0.22 0.19 0.22 0.22 0.10; 0.20 0.15 0.20 0.28 0.15; 0.18 0.23 0.15 0.15 0.50]; >> I = eye(5)</pre> <p>Note that the command "eye(n)" creates an $n \times n$ matrix with 1's on the diagonal and 0's elsewhere.</p>
	<p>(b) Use MATLAB to row reduce the augmented matrix $[\mathbf{C} - \mathbf{I} \mid \mathbf{d}]$ and write down the general solution to $(\mathbf{C} - \mathbf{I})\mathbf{p} = \mathbf{0}$.</p>
	<p>(c) What are the highest and the lowest priced commodities in Matrixville? List the inhabitants of this charming town in order of increasing income. If our friend Bob makes \$40,000 per year, what are the incomes for the rest of the inhabitants? With all the moonshine that Bob drinks, do you think he will live long enough to enjoy his good life?</p>

§2.4 How Google Search Works

The beginning of this story is actually a theorem of Perron and Frobenius from the early 20th century, but we will discuss this later in our tale. Instead we start with a cute word problem illustrating the theorem, which has amused linear algebra students at least as far back as the 1970's. The problem is: In a particular group of people, who is the most popular?

One possible approach is to ask everyone in the group to list their friends in that group. Say there are 4 people, named Jane, Charlie, Mary and Fred, who we affectionately nickname J, C, M and F. The friends list is:

- Jane lists C and M
- Charlie lists J, M and F
- Mary lists J, C and F
- Fred lists J and M

(This experiment is dangerous. You should not try it in your own dormitory.) Our data naturally amounts to a 4×4 matrix of 0's and 1's:

	Jane	Charlie	Mary	Fred
J	0	1	1	1
C	1	0	1	0
M	1	1	0	1
F	0	1	1	0
Total	2	3	3	2

Some people tend to list every one they ever met and others only list closest friends, so to compensate for this we normalize the matrix by dividing each list by the number of people in it:

$$\begin{matrix} & J & C & M & F \\ \begin{matrix} J \\ C \\ M \\ F \end{matrix} & \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix} \end{matrix}$$

We shall call this the **linking matrix** and denote it by L .

Now that we have collected data, we wish to determine a nonnegative number associated to each person called their **popularity**; combined they give

the **popularity vector** $\mathbf{r} = \begin{pmatrix} r_J \\ r_C \\ r_M \\ r_F \end{pmatrix}$ for the group. Its basic property is

A person's popularity is the weighted sum of the popularity of people who reference that person.

We see what the weights are and what this means by the example,

$$r_{\text{Jane}} = \frac{1}{3}r_{\text{Charlie}} + \frac{1}{3}r_{\text{Mary}} + \frac{1}{2}r_{\text{Fred}}$$

which seems reasonable when one considers that Charlie's contribution to Jane's popularity is $\frac{1}{3}r_{\text{Charlie}}$, because Jane constitutes one third of Charlie's friends and r_{Charlie} is Charlie's popularity. In matrix vector form these equations are exactly

$$Lr = r \quad (\text{Equation A.1})$$

which is the same as

$$(L - I)r = 0$$

where I is the 4×4 identity matrix. Solving this equation we can determine the relative "popularity" of Jane, Charlie, Mary and Fred. We ran MATLAB using the `rref` command as in exercise 2.6 and set (arbitrarily, of course) $r_J = 1$ to get $r_J = 1.5$, $r_C = 1.3125$, $r_M = 1.6875$, and $r_F = 1$. Thus the order of popularity, from highest to lowest, is: Mary, Jane, Charlie, and finally, Fred.

Some graduate students at Stanford got the idea that this sort of method could be used to rank a group of web pages. You type in some key words (which we will denote by the set K) and the software identifies vast numbers of webpages W_K (most of which are garbage as far as you are concerned) containing the words in K . The big challenge is automatically finding a few good (popular) webpages. What the Google founders did was have the software look at each webpage w in W_K and determine which other web pages in W_K the page w links to, thus associating to each webpage w a vector L_w of 0's and 1's, just as we did above with the list of students. Next, it builds a matrix L whose columns are the normalized vectors L_w and proceeds just as we did in the popularity problem. Google computes r very quickly and lists for you, the user, the web pages in order of high to low popularity.

Sounds like a good plan, but if you are going to base a company on this and put up lots of money to back it, there are few things you would definitely want to know:

- (a) Does the equation $Lr = r$ above always have a solution (what do you know from the course so far makes you think it seldom will?)
- (b) Will a solution have entries that are nonnegative?
- (c) Is the solution unique? If not, we will have conflicting rankings.

Fortunately, the founders of Google knew

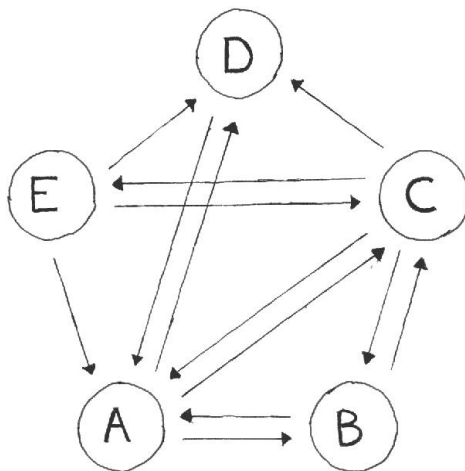
Theorem (Perron-Frobenius) For any matrix L having all entries nonnegative and each column summing to 1, the equation $Lr = r$ has a nonnegative solution r .

Thus the Perron-Frobenius theorem tells us that there always is a solution to the popularity problem. While we do not go into it, the theory also tells us that uniqueness is usually true, and explains the circumstances when it is not.

In the late 1990s the leading internet portal was Yahoo, which employed warehouses full of people to look at web pages and grade them. By using this simple math model (with a few bells and whistles added) Google quickly displaced Yahoo. The ranking you just saw is often called the Page Rank after Larry Page, a founder of Google, who is in a few years became one of the 10 richest Americans. All of this is indeed a core example at the heart of a rapidly growing area of analysis of networks.

There are several morals to draw. Theorems are efficient ways to remember key ideas. What a mathematician calls a proof may well translate to industry as quality control. Lastly, if you learn linear algebra well, then you will probably get rich.

Now it's time for a little practice with the PageRank algorithm. Suppose that we have five websites, namely the sites of your good friends Alan, Beth, Charlie, Dana, and Eleanor, which we'll simply refer to as A, B, C, D, and E. Let's also suppose that the links between the various sites are given by the graph below:



The arrow pointing from C to D means that Charlie's site links to Dana's, etc. For small collections of objects, graphs like this are an effective means of showing the links between sites.

Exercise 2.7

- (a) Create a linking matrix L which contains the information of which site links to which just as we did in the popularity example. Remember to normalize, and be sure that your input is exact (for example, make sure you enter $1/3$ instead of $.333$ - this is important for part b, since our columns must sum to 1). Include all input and output from Matlab.
- (b) Use the `rref` command to find all solutions x to the matrix equation $(L - I)x = 0$. Include all input and output from Matlab (If you get something like "Empty matrix: 5-by-0", be sure to double check your answer to part a!).
- (c) Whose website has the highest PageRank? Explain your answer, especially in light of any negative numbers which may have arisen in part (b). List the remaining websites in order of decreasing PageRank.

In practice L is often huge. So a question to mull over is how can we solve $Lr = r$ for problems of huge size? Working in our favor is the fact that that most entries of L are 0. Such an L is called a sparse matrix, and a major branch of mathematics research is exploiting this to find a solution. What do we do? A hint: Gaussian Elimination (row reduction) will not work. Later in these assignments we shall get a glimpse of something that does.

§2.5 Feedback

We're almost done, but before we finish we'd like to get some (optional) feedback from you so that we can improve this lab. If you have any comments, write them (by hand if you like) on a **separate page** at the end of your assignment, so that we can tear it off before we give back your assignment.

Feedback (Optional)



Do you have any suggestions to help improve this lab? This includes errors or potential errors, typos, unclear statements or questions, or anything else that comes to mind. The more specific you can be, the easier it will be for us to improve the lab.

§2.6 Conclusion

Systems of linear equations, though quite uncomplicated theoretically, show up in many interesting applications. Leontief's closed exchange model and Google's Page Rank algorithm are only two of many such applications. Because systems of linear equations that arise in applications tend to be quite big and messy and not suitable to working with by hand, computer programs, such as MATLAB, turn out to be an indispensable tool for people who must work with such systems in their work.

Last Modified:

06/27/2014