Week #11: Linear Algebra II

Goals:

- Use MATLAB to solve linear system and interpret the answer the program provides.
- Write the nullspace of A by solving Ax = 0.
- Compute the rank of a matrix.
- Explain the notions of linear independence and bases.

Introduction to Linear Systems

We begin this week with a brief review of solving for the intersection of two lines. Picture the two lines y = x - 1 and y = -x + 4. With a little algebra we can see they intersect at x = $\frac{5}{2}$ and y = $\frac{3}{2}$. If you need to review how to solve for the intersection of two lines, this is an excellent resource:

http://tutorial.math.lamar.edu/Classes/Alg/SystemsTwoVrble.aspx

So far, the lines most of you have worked with are two dimensional. Even when you have something of the form 2x + 6y - 10 = 0, this can be rearranged into

$$y = \frac{1}{3}x + \frac{5}{3}$$

which is the standard slope-intercept form of a line.

Now consider the equation:

$$2x - 4y + 8z - 18 = 0$$

This object is not a **line**, since we cannot put it into slope intercept form. Instead we call it **linear**, since every variable has an exponent of 1, and is not multiplied by another variable. With a line, there are two variables, and so we call it a two-dimensional object. For 2x - 4y + 8z - 18 = 0, there are three variables, so this is a three dimensional object, and three dimensional linear objects are called **planes**. Picture a flat piece of paper floating in the air, and it could be described by a linear equation with three variables.

Anything with more than three variables is impossible to picture in our minds, since we aren't equipped to visualize things in higher dimensions (a sad fact). However, we can represent four dimensional linear objects with equations like

$$a + b + c + d = 12$$

We call linear objects that have a dimension higher than three **hyperplanes**.

At this point you may be wondering how objects in four or more dimensions are relevant to the real world in general, or engineering in particular. Consider the problem of allocating airplanes, food, fuel, pilots, cabin staff, maintenance personnel, replacement parts, supplies like lubricant and cleaning materials, drinks (first class and main cabin, they're very different, etc. etc. for an airline at different airports in possibly different countries. This can involve hundreds of linear equations in hundreds of variables. This means a problem whose geometry involves a space with hundreds of dimensions comes up every day. If we represent each of those things as variables in linear equations, we create something called a **system of linear equations**. A system of linear equations is a way to find where higher dimensional linear objects intersect, if they intersect at all.

System of Linear Equations and the Augmented Matrix

Consider the following three equations. It is possible that these three linear equations intersect at a single point.

$$\begin{aligned} 2x - 3y + 4z &= 15 \\ x + y + z &= 1 \\ 4x + y - 2z &= -3 \end{aligned}$$

There are entire treatments about how to solve this system with algebra, but we are going to skip past that and go right to the computational solution. First, we need to turn our linear system into an **augmented matrix**.

Turning a system of linear equations into an augmented matrix is a simple process. The following example should make it clear. Going back to our system introduced on the last page

$$\begin{aligned} 2x - 3y + 4z &= 5 \\ x + y + z &= 1 \\ 4x + y - 2z &= -3 \end{aligned}$$

The first step is ensuring that all of your variables are in the same order for each equation, and they are equal to some constant (0 is fine). If they are not, rearrange your equations so that the variables in each equation are in the same order, and all constants have been combined into one constant on the other side of the equal sign.

Once we have our linear system in this form, simply remove all of the variables, and change the equal sign in all of the equations to a vertical line like so:

$$\left(\begin{array}{ccc|c} 2 & -3 & 4 & 5 \\ 1 & 1 & 1 & 1 \\ 4 & 1 & -2 & -3 \end{array}\right)$$

This is the augmented matrix that represents the linear system.

Practice turning the following linear systems into augmented matrices

$$\begin{aligned} 12x - 9y + z &= 15 \\ 2x + z - y &= 12 \\ 4y + z + x + 2 &= 4 \end{aligned}$$

$$\begin{aligned} x + 2x - 1.3y - z &= 1 + z \\ 2x + y + z &= 1 + y + z \\ -x + 8y + 2z &= 6 + x + y + z \end{aligned}$$

and use MATLAB to find if they intersect, and at what point. First, enter the left side of the augmented matrix as a matrix into MATLAB, and enter the right side of the matrix as a vector in MATLAB. Once you have done this, use the MATLAB command:

`linsolve(A,b)`

assuming A is your matrix and b is your vector.

You should get

$$\begin{bmatrix} -1.5476 \\ -2.3095 \\ 12.7857 \end{bmatrix}$$

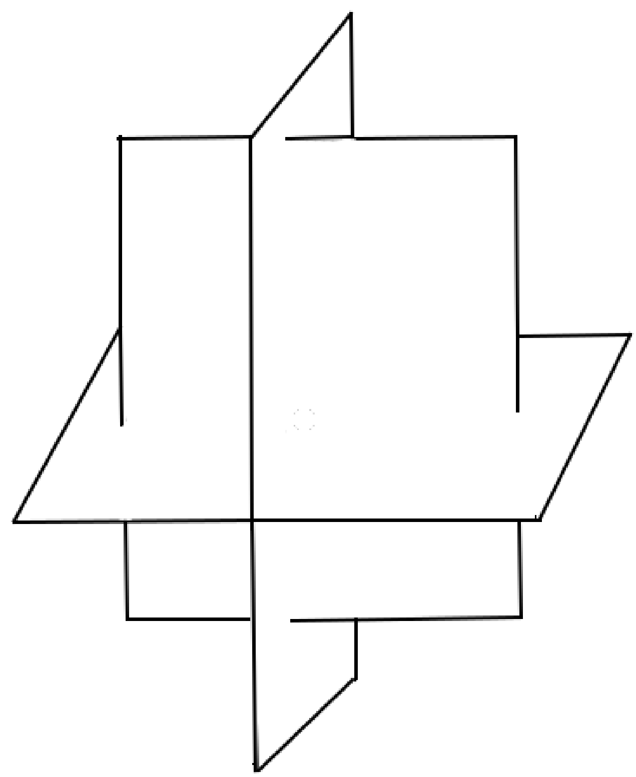
as the solution to the first system and

$$\begin{bmatrix} 0.5 \\ 1.0630 \\ -0.4409 \end{bmatrix}$$

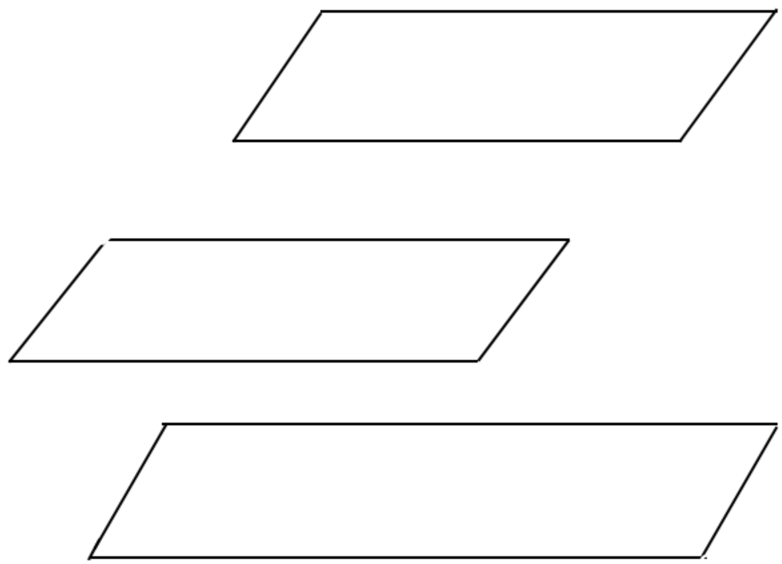
as the solution to the second system. How can we interpret these answers? These are the points where the 3-dimensional planes intersect at a single point. Not all linear systems will intersect at a single point!

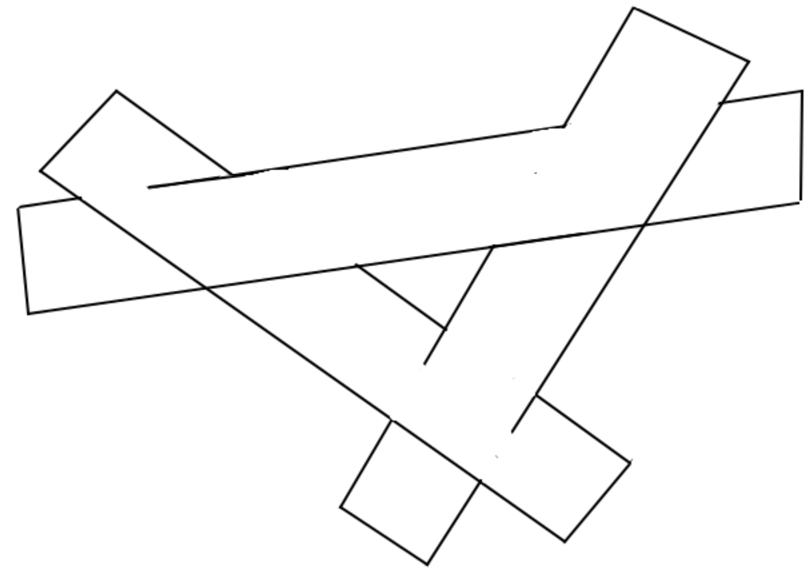
The Possibilities of How Linear Systems can Intersect (or not)

We have already discussed the case where all of the linear objects in a linear system can intersect at one point. This can be pictured in the following manner:



Another possible case is that there is no solution to the linear system. This can be pictured using the following two images:





Notice that in both cases at no point does every plane meet at the same spot.

Consider the linear system

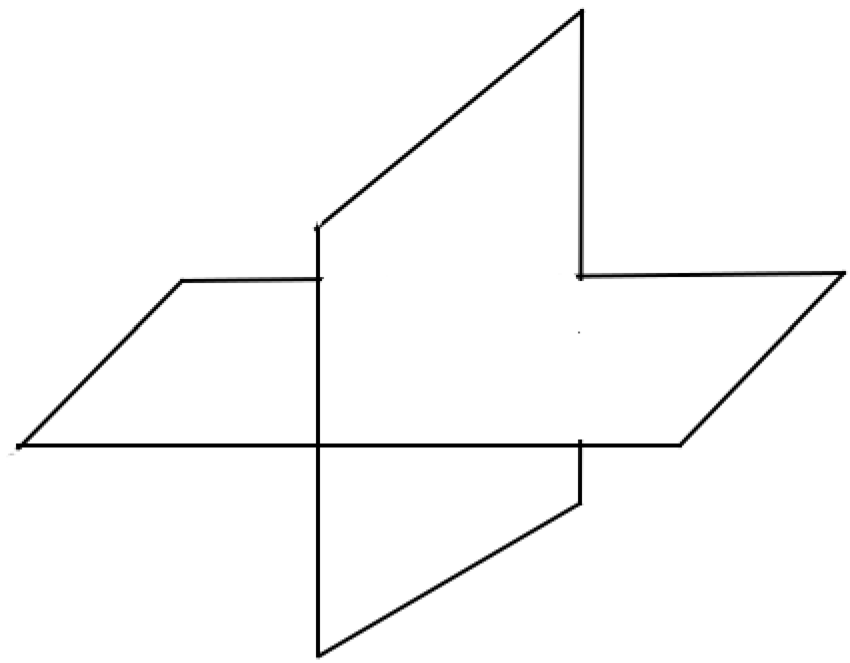
$$\begin{aligned} 2x + 2y + 3z &= 10 \\ 3x - 2y + 2z &= 20 \\ -2x + 3y - 0.5z &= -5 \end{aligned}$$

When trying to solve this system in MATLAB, the following message is returned:

```
Warning: Matrix is close to singular
or badly scaled. Results may be inaccurate.
RCOND = 1.903239e-17.
```

This is the message MATLAB gives when a system of linear equations has no solution. If this system were based on a real life scenario, it would mean that no value of x, y, and z will satisfy the problem.

Now consider the following picture:



In this case, it is possible that two linear objects meet at an **infinite** number of points, and so there are an infinite number of choices that will work.

If we use this system instead:

$$\begin{aligned} 2x + 2y + 3z &= 10 \\ 3x - 2y + 2z &= 10 \\ -2x + 3y - 0.5z &= -5 \end{aligned}$$

and try to solve it in MATLAB, we get the same error message as before:

```
Warning: Matrix is close to singular
or badly scaled. Results may be inaccurate.
RCOND = 1.903239e-17.
```

Notice that MATLAB does not specify the kind of error it gets. The computer generally cannot inform you as to whether your problem has no solution, or infinite solutions. This isn't a very big problem in applications, usually it is enough to know that a system does not have a unique solution.

Use MATLAB to solve the following systems, and be sure to think about how to interpret the answer MATLAB gives:

$$\begin{aligned} 4x + 2y - 13z &= 0 \\ -x - 0.5y + 1.1z &= -1.3 \\ -6.1x - 4.4y - 5.9z &= -5.1 \end{aligned}$$

$$\begin{aligned} 112x + 422y + 369z &= 101088 \\ 553x + 287y + 331z &= 8672 \\ 182x + 631y + 901z &= 38390 \end{aligned}$$

$$\begin{aligned} -x - y - z &= -1 \\ x + y + z &= 1 \\ x - y + z &= 0 \end{aligned}$$

Solution of first system at the point $(5.837, -7.744, 0.605)$

Solution of second system at the point $(-72.298, 538.425, -319.865)$

Third system has no solution.

The Interaction of Matrices and Vectors

Recall that a vector indicates two things, direction and magnitude. We now discuss the concept of matrices multiplied by vectors as a way of describing how to turn, lengthen, and shorten vectors. In fact, a matrix is a way to describe and implement the transformation of a vector.

We describe the multiplication of a matrix $\mathbf{A}$ and a vector $\overrightarrow{x}$ as:

$$\mathbf{A}\overrightarrow{x} = \overrightarrow{y}$$

which produces a new vector $\overrightarrow{y}$.

For example

$$\begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

When the vector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is multiplied by the matrix $\begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix}$

that transforms the vector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ into the vector $\begin{bmatrix} 4 \\ 6 \end{bmatrix}$ which has both a new length and a new direction. The details of how matrix multiplication works is what is covered in a typical linear algebra course, but we will be using MATLAB to do the heavy lifting for us.

Using MATLAB to multiply matrices and vectors

Enter the previous matrix $\mathbf{A} = \begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix}$ and vector $\overrightarrow{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ into MATLAB. The command for matrix multiplication in MATLAB is the * symbol, and we enter the command into MATLAB as:

```
A*b
```

The output is:

```
ans =
          4
          6
```

which is the vector from the previous slide.

If

$$\mathbf{A} = \begin{pmatrix} 6 & -1.6 \\ 2 & -9 \end{pmatrix}$$

and we have vectors

$$\overrightarrow{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \; \overrightarrow{v} = \begin{bmatrix} -10 \\ -4 \end{bmatrix} \; \overrightarrow{w} = \begin{bmatrix} -2 \\ 8 \end{bmatrix}$$

Find

$$\mathbf{A}\overrightarrow{u},\ \mathbf{A}\overrightarrow{v},\ \mathbf{A}\overrightarrow{w}$$

$$\mathbf{A}\overrightarrow{u} = \begin{bmatrix} 4.4 \\ -7 \end{bmatrix},\ \mathbf{A}\overrightarrow{v} = \begin{bmatrix} -53.6 \\ 16 \end{bmatrix},\ \mathbf{A}\overrightarrow{w} = \begin{bmatrix} -24.8 \\ -76 \end{bmatrix}$$

Although there is a good deal of very interesting theory involved in matrix multiplication (called linear operator theory and well worth studying should you get the chance), we will keep our focus on some basic rules that will allow us to use MATLAB most effectively.

The first rule is that **order really matters.** If you were to try to enter the command

```
b*A
```

into MATLAB, the following error is returned:

```
Error using *
Inner matrix dimensions must agree.
```

So when using MATLAB, be sure to have the matrix multiplied by the vector, not the other way around.

The second rule is that the number of elements in the row of a matrix must equal the number of elements in the vector. For example, when trying to multiply $\mathbf{A} = \begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix}$ and vector $\overrightarrow{c} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ in MATLAB, the following error message is returned

```
Error using *
Inner matrix dimensions must agree.
```

This is because there are only two elements in the rows of $\mathbf{A}$, but three elements in the vector $\overrightarrow{c}$.

The third rule is that matrix multiplication of a vector can be stacked. This can be thought of as transforming a vector more than once. Instead of asking MATLAB to do each transformation one at a time, we can enter them all into the command line at once to see what the final result is going to be.

Let $\mathbf{A} = \begin{pmatrix} 2 & -1 \\ 4 & 7 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 8 & 2 \\ -3 & -3 \end{pmatrix}$, and $\overrightarrow{w} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ If I want to know what happens to $\overrightarrow{w}$ if I first multiply it by $\mathbf{B}$ and then by $\mathbf{A}$, that can be represented as

$$\mathbf{AB}\overrightarrow{w}$$

and entered into MATLAB as

```
A*B*w
```

which gives the result

```
ans =
      85
     -19
```

Once again, it is imperative that you remember **order really matters.** If we were to enter the command code

```
B*A*w
```

we would get

```
ans =
      96
    -126
```

Since we are multiplying $\overrightarrow{w}$ by $\mathbf{A}$ first, and multiplying the result of that by $\mathbf{B}$.

If we have matrices

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -1 & 4 & 9 \\ -3 & 2 & -6 \\ 5 & -7 & 8 \end{pmatrix}$$

and the vectors $\overrightarrow{c} = \begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix}$ and $\overrightarrow{d} = \begin{bmatrix} -1 \\ 5 \\ 0.5 \end{bmatrix}$, find

$$\mathbf{AB}\overrightarrow{c},\ \mathbf{AB}\overrightarrow{d},\mathbf{BA}\overrightarrow{c},\ \mathbf{BA}\overrightarrow{d},\mathbf{AA}\overrightarrow{c},\ \mathbf{BB}\overrightarrow{c},\ \mathbf{AA}\overrightarrow{d},\ \mathbf{BB}\overrightarrow{d}$$

$$\mathbf{AB}\overrightarrow{c} = \begin{bmatrix} -15 \\ -42 \\ -69 \end{bmatrix},\ \mathbf{AB}\overrightarrow{d} = \begin{bmatrix} -62.5 \\ -64 \\ -65.5 \end{bmatrix}, \mathbf{BA}\overrightarrow{c} = \begin{bmatrix} 390 \\ -185 \\ -165 \end{bmatrix}$$

$$\mathbf{BA}\overrightarrow{d} = \begin{bmatrix} 423 \\ -208.5 \\ 184.5 \end{bmatrix}, \mathbf{AA}\overrightarrow{c} = \begin{bmatrix} 150 \\ 330 \\ 510 \end{bmatrix},\ \mathbf{BB}\overrightarrow{c} = \begin{bmatrix} -21 \\ 23 \\ -27 \end{bmatrix}$$

$$\mathbf{AA}\overrightarrow{d} = \begin{bmatrix} 171 \\ 387 \\ 603 \end{bmatrix},\ \mathbf{BB}\overrightarrow{d} = \begin{bmatrix} -309.5 \\ 159.5 \\ -230.5 \end{bmatrix}$$

Recall the identity matrix from last week:

$$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{I}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If you multiply a vector with n elements by $\mathbf{I}_n$, the exact same vector is returned as the answer. That is why these matrices are given the special name **identity**.

The Nullspace of a Matrix

We now define the **zero vector** as a vector of n elements, all zero, which we define as $\overrightarrow{0}$. The **nullspace** of a matrix $\mathbf{A}$ is the collection of every vector $\overrightarrow{v}$ such that:

$$\mathbf{A}v = \overrightarrow{0}$$

Finding the nullspace of a matrix is as simple as entering the command

```
null(A)
```

For example, if we want to find the nullspace of

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 4 & 4 \end{pmatrix}$$

after entering the correct command, MATLAB will return

```
ans =
       0.7071
      −0.7071
```

How to interpret this answer? This is the set of all vectors that have position vector $\begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}$. If you were to multiply $\mathbf{A}$ by any vector that is equivalent to this position vector, you would get the zero vector back as your output.

What is the physical interpretation of the nullspace? There are many, but this is the one that is probably most illustrative:
Suppose we are dealing with a rocket and the matrix A is all of the directions we can go based on our thrusters. Now suppose we want to go in a particular direction. If the direction we want to go in is in the nullspace, that is a direction that will completely waste fuel and not change our direction at all.

Find the nullspace of

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 0 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -4 & -1 & 3 \\ -8 & -2 & -2 \\ -8 & -2 & 7 \end{pmatrix} \quad \text{and } \mathbf{C} = \begin{pmatrix} 7 & 12 & 2 & 5 \\ 5 & 5 & 5 & 5 \\ 2 & 7 & -3 & 0 \\ -6 & -2 & 1 & 4 \end{pmatrix}$$

$$\text{null}(\mathbf{A}) = \begin{bmatrix} 0 \\ 0.8321 \\ -0.5547 \end{bmatrix} \quad \text{null}(\mathbf{B}) = \begin{bmatrix} -0.2425 \\ 0.9701 \\ 0 \end{bmatrix} \quad \text{null}(\mathbf{C}) = \begin{bmatrix} -0.4065 \\ 0.3752 \\ 0.6045 \\ -0.5732 \end{bmatrix}$$

The Rank of a Matrix

There will be cases where a matrix does not have any vectors in its nullspace. We now investigate under what conditions that is true. The **rank** of a matrix is defined as the dimension of the vector space spanned by its columns. Interpreting that without a deeper background in linear algebra can be difficult. Finding the rank of a matrix $\mathbf{A}$ with MATLAB is as easy as:

```
rank(A)
```

If we consider the matrix we had earlier

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 4 & 4 \end{pmatrix}$$

and ask for the rank, MATLAB returns

```
ans =
      1
```

Let's consider another matrix

$$\mathbf{B} = \begin{pmatrix} 1 & -1 \\ 2 & 3 \end{pmatrix}$$

If we ask MATLAB for

```
rank(B)
```

we get

```
ans =
      2
```

and

```
null(B)
```

returns

```
ans =
      Empty matrix: 2-by-0
```

The answer of `Empty matrix: 2-by-0` can be interpreted as the zero vector, $\overrightarrow{0}$, which is a vector with n elements, all zero. Which means that if you were to multiply $\mathbf{B}$ by any vector, the only vector that returns 0 as an answer is the zero vector.

Why the difference? What makes matrix $\mathbf{A}$, which has a nullspace that includes more than just the zero vector, different from matrix $\mathbf{B}$? The idea of linear independence of vectors will help us answer that question.

Practice finding the rank and nullspaces of the following matrices.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 0 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -4 & -1 & 3 \\ -8 & -2 & -2 \\ -8 & -2 & 7 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 7 & 12 & 2 & 5 \\ 5 & 5 & 5 & 5 \\ 2 & 7 & -3 & 0 \\ -6 & -2 & 1 & 4 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 3 & -8 & -2 & 1 \\ 5 & 4 & 5 & 4 \\ 1 & -6 & -3 & 0 \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} 4 & 5 & -6 \\ -3 & 5 & 3 \\ 1 & -8 & 2 \\ -6 & -2 & 1 \end{pmatrix}$$

$\text{rank}(\mathbf{A}) = 2$, $\text{rank}(\mathbf{B}) = 2$, $\text{rank}(\mathbf{C}) = 3$, $\text{rank}(\mathbf{D}) = 3$, $\text{rank}(\mathbf{E}) = 3$

Linear Independence

Once again, let's consider the matrices

$$\mathbf{B} = \begin{pmatrix} 1 & 1 \\ 4 & 4 \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} 1 & -1 \\ 2 & 3 \end{pmatrix}$$

Notice that the column vectors of matrix $\mathbf{A}$, $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$ are really the same vector. It will help if we think of the columns as the matrix as directions that we can go in a space. In the case of $\mathbf{A}$, although there are two column vectors, they are really the same vector and so we can only go in one direction. On the other hand, if we look at $\mathbf{B}$, we can see that the column vectors are different, which means we can go in more than one direction. In a two dimensional space, if we can move in two different directions, that means we can reach any point we want to! This captures the notion of **linear independence**.

A set of two or more vectors with dimension n (having n elements) are called **linearly independent** if they **span** the space of all possible vectors of dimension n. Spanning the space implies that we can take any linear combination (multiplying a vector by a number and then adding or subtracting it to another vector) of those vectors to get any vector with n elements. For example, think about the vector $\begin{bmatrix} -2 \\ 2 \end{bmatrix}$. If we consider the column vectors from $\mathbf{B}$, $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} -1 \\ 3 \end{bmatrix}$, we can see that if we combine those two vectors together we get:

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

In fact, we can find some combination of those two column vectors of $\mathbf{B}$ to get *any* vector we can think of. We call this concept spanning the space, and if vectors are linearly independent then they span the space.

The rank of a matrix tells you how many linearly independent column vectors are in that matrix. Let's define the **dimension** of a matrix to be the number of elements in each row. *If the number of linearly independent column vectors (the rank) is equal to the dimension of the matrix, then we say that those column vectors form a **basis**.* Another way to think of a basis is that it is a set of linearly independent vectors whose linear combinations can produce any vector in that space.

A really useful theorem states that for a matrix $\mathbf{A}$:

$$\text{rank}(\mathbf{A}) + \ \text{null}(\mathbf{A}) = \text{dimension}(\mathbf{A})$$

which means that if you are investigating a matrix's properties and find that it does not form a basis for the space, you can identify exactly how many dimensions that matrix cannot cover. If

$$\text{rank}(\mathbf{A}) = \text{dimension}(\mathbf{A})$$

Then the columns of $\mathbf{A}$ form a **basis**.

Find the rank and nullspaces of the following matrices using MATLAB, and determine which ones form a basis for the space. If you are unsure as to what the dimension of a matrix $\mathbf{C}$ is, use the

```
length(C)
```

and MATLAB will return the dimension of the matrix in which you are interested.

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 7 \\ 1 & -9 & 0 \\ 0 & 5 & -1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -6 & 1.1 & 8 & 3.2 \\ -4.5 & 10 & 6.6 & 8.9 \\ 0.5 & 5.8 & -7.1 & 2.3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 2.4 & 9.3 \\ 6.1 & -2.9 \\ 10.1 & 5.4 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 8.6 & 7.2 & 8.8 & -2.6 \\ 2.5 & -0.9 & 12.8 & 3.3 \\ -2.7 & -3.8 & 4.7 & -8.9 \\ 1 & 0 & 0 & -4 \end{pmatrix}$$

$$\text{rank}(A) = 3,\ \text{null}(A) = \text{empty}$$

$$\text{rank}(B) = 3,\ \text{null}(B) = \begin{bmatrix} 0.0740 \\ -0.5577 \\ -0.1897 \\ 0.8047 \end{bmatrix}$$

$$\text{rank}(C) = 2,\ \text{null}(C) = \text{empty}$$

$$\text{rank}(D) = 4,\ \text{null}(D) = \text{empty}$$