Andrew Shao

# Linear Algebra I

# Unit 2: Linear Systems

## Systems of Linear Equations:

Linear Equations can be written in the form of: $a_1x_1 + ... + a_ix_i = b$

A system of $m$ Linear Equations has pattern:

$a_{11}x_1 + a_{12}x_2 + ... + x_{1i}x_i = b_1$

...

$a_{m1}x_1 + ... + x_{mi}x_i = b_m$

Note that $a_{nj}$ represents the nth equation at the jth index. This notation is common in Linear Algebra.

A solution is represented by the vector $\begin{bmatrix} s_1 \\ \dots \\ s_i \end{bmatrix}$, where s_1 represents the value of x_1 that makes all equations true. Note that this can be multiple values, and the solution set is the set of all vectors that satisfy the system.

*Definition:* If a system of linear equations has at least one solution(at least one point that lies on all hyperplanes), it is a consistent system. Otherwise, it is considered to be inconsistent.

Systems can either have one solution, no solutions, or infinitely many solutions.

## Reducing Matrices to Solve Linear Equations:

As you may know from basic Algebra, you can solve equations using substitution and elimination. But, it is super slow and messy to do all of this by hand to solve systems every single time.

We can instead represent a system of linear equations as a coefficient matrix.

Ex. $a_{11}x_1 + ... + a_{1i}x_i = b_1$

$a_{m1}x_1 + ... + a_{mi}x_i = b_m$ , As:

$$\begin{bmatrix} a_1 1 & a_1 2 & \dots & a_1 i \\ A_2 1 & a_2 2 & \dots & a_2 i \\ \dots & & & \\ a_m 1 & a_m 2 & \dots & a_m i \end{bmatrix}$$

## Elementary Row Operations:

You can perform elementary row operations on a given matrix to help reduce it and solve a linear equation. There are 3 operations you can perform:

1) Swapping 2 rows
2) Subtracting one row from a multiple of another(NOTE: Not subtracting a multiple of one row from a multiple of another)

3) Multiplying a constant factor to a row.

Matrix A is considered to be *similar* or row equivalent to Matrix B if there exists a set of row operations that allow Matrix A to be transformed into B. These two matrices will have the same solution set.

The process of reducing the matrix to simplest form is called Gauss-Jordan Elimination and it can be done like this:

Note that the vertical bars represent an augmented matrix.

Ex. Reduce $\begin{bmatrix} 1 & 1 & 1|1 \\ 0 & 2 & -6|2 \\ 3 & 6 & -5|4 \end{bmatrix}$

We Start with Subtracting Row 3 by 3 * Row 1(Represented by R_3 - 3R_1)

$$\begin{bmatrix} 1 & 1 & 1|1 \\ 0 & 2 & -6|2 \\ 0 & 3 & -8|1 \end{bmatrix}$$

We then divide row 2 by 2(Represented by R_2 * ½)

$$\begin{bmatrix} 1 & 1 & 1|1 \\ 0 & 1 & -3|1 \\ 0 & 3 & -8|1 \end{bmatrix}$$

R_1 - R_2, R_3 - 3R_2

$$\begin{bmatrix} 1 & 0 & 4|0 \\ 0 & 1 & -3|1 \\ 0 & 0 & 1|-2 \end{bmatrix}$$

R_1 - 4R_3, R_2 + 3R_3

$$\begin{bmatrix} 1 & 0 & 0|8 \\ 0 & 1 & 0|-5 \\ 0 & 0 & 1|-2 \end{bmatrix}$$

Thus, x_1 = 8, x_2 = -5, x_3 = -2

This process is called Gauss Jordan Elimination, and it helps to speed up solving linear equations such as these.

## Reduced Row Echelon Form(RREF):

Has 4 Main Properties:

1) Each non-zero entry has all other entries above it as zero
2) The first non-zero entry is a one, and it's called the leading one
3) All leading ones are right of all leading ones above it
4) The leading one is the only non-zero entry in each column.

The ultimate goal of Gauss Jordan Elimination is to reduce the matrix into a Reduced Row Echelon Form, which will reduce it down to a much simpler system. Then, you either have the values of each variable or you have a small system that is easy to substitute into.

*Free Variables*: Where the column in the RREF isn't a leading one.
The convention is to use variables that aren't free and rewrite them in terms of free variables. Any system with free variables and is consistent will have *infinitely many* solutions.

## An Algorithm for Gauss-Jordan Elimination:

Reducing Matrices can be really fast with intuition, but if you want a direct algorithm to perform matrix reduction, here it is. Also Note: As implemented on my GitHub(Linear Algebra Calculator), this algorithm is really useful when programming a matrix reduction, as computers have no such intuition.

*Reduce-Matrix*:
1) Sort Rows based on Leading zeros, ignoring rows already used.
2) Find the top, and divide all values in the row by the leading coefficient, to reduce the row to a leading 1
   a) Note: This is why this algorithm is slow, as you have a high probability of running into fractions and annoying real numbers
3) If any row with this leading 1 has anything but zero, subtract the two rows by a factor of whatever is non-zero, to remove this entry
4) Add this used row to the already reduced rows, meaning the sorting algorithm will no longer consider it to reduce it to a leading coefficient
5) Check for fully 0 rows, and if there is a fully 0 row, remove it from the matrix or add it to already reduced rows.
6) Repeat steps 1-5 until all rows are exhausted.

This method always works, but is quite a bit more annoying to do by hand and is much slower.

## Homogeneous Systems:

*Definition*: A homogenous system is when the right hand side of the equation contains only zeros, otherwise represented as an augmented matrix as so: $[x \mid 0]$. Since there is always the trivial solution in a homogeneous system, we know that every homogeneous system is consistent.

This can be used to see if a set is linearly independent or not, by reducing such a matrix and checking for *free variables*. Reducing the homogeneous system can also tell you the *solution space* of a given system(which is the geometric representation of the solution(ex. Line, plane, hyperplane))

## Rank:

*Definition:* The *rank* of a matrix is the number of leading zeros that can be found inside of the *RREF* of the matrix. Please take into account that the rank must be computed using the RREF of the matrix. This is equivalent to the number of variables subtract the number of free variables.

Augmented Matrices also have a rank, which can include the augmented portion if there is no leading one in the left side.

Properties of Rank:

- If rank(A) < rank([A | b]), then the system is inconsistent.
- Rank(A) = # of equations iff the system is consistent for all augmented b.

# Next Unit: Matrices.