

Lecture 4

[Better Explained Blog on LinAlg:](#)

Linear algebra gives you mini-spreadsheets for your math equations.

(most elementary view of linear algebra)

[GE Demo](#)

Vocabulary: Pivot, Multiplier. The multipliers go in L. With ones on the diagonal,
 $LU = A$.

Transpose (aka adjoint); Flip along main diagonal. Rows & Columns Interchange

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Main Diagonal

If A is $m \times n$, A^T is $n \times m$

$$A_{ij} = (A^T)_{ji}$$

Julia: Uses apostrophe, and uses “lazy evaluation.” (Does not store the transpose, works with it when needed.)

```
A = [1 2; 3 4; 5 6]
B = A'
```

```
2×3 LinearAlgebra.Adjoint{Int64,Array{Int64,2}}:
```

```
1  3  5
2  4  6
```

Data "Bricks" aka Tensors or Multidimensional Arrays

Generalization to higher dimensional arrays
Tensors, data "brick"

$$A \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$$

A is $n_1 \times n_2 \times \dots \times n_d$

$A_{i_1 i_2 \dots i_d}$ is the element with index i_1, i_2, \dots, i_d

Generalization of transpose is a dimension permutation
There are $d!$ permutations

$p = p_1 p_2 \dots p_d$ to be a permutation of $1, 2, \dots, d$

Julia: $B = \text{permutedims}(A, p)$

B will be $n_{p_1} \times n_{p_2} \times \dots \times n_{p_d}$

$$A_{i_1 i_2 \dots i_d} = B_{i_{p_1} i_{p_2} \dots i_{p_d}}$$

e.g. A is $2 \times 4 \times 6 \times 8$
 $p = [2, 3, 4, 1]$

B is $4 \times 6 \times 8 \times 2$

Transpose generalizes to a
"dimension permutation"



Inverses

Only applies to $n \times n$ square

A^{-1} is pronounced “A Inverse”

$$A^{-1}A = I$$

If $A^{-1}A = I$ then $AA^{-1} = I$ and vice versa.

If A^{-1} does not exist, we say that A is “singular”.

If A^{-1} does exist we say that A is “nonsingular” or “invertible”

$$(AB)^{-1} = B^{-1}A^{-1}$$

$Ax = b$ ($n \times n$ system of linear equations)

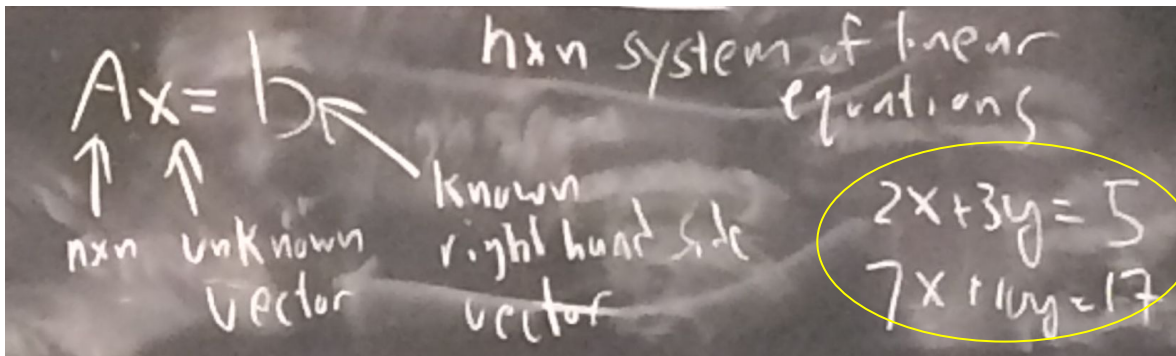
Given $n \times n$ A , and right hand side b , if A is invertible, the unique solution is

$$x = A^{-1}b$$

In software it is often considered disadvantageous for complexity and numerical reasons to ever compute A^{-1} explicitly. A common way to write the solution is

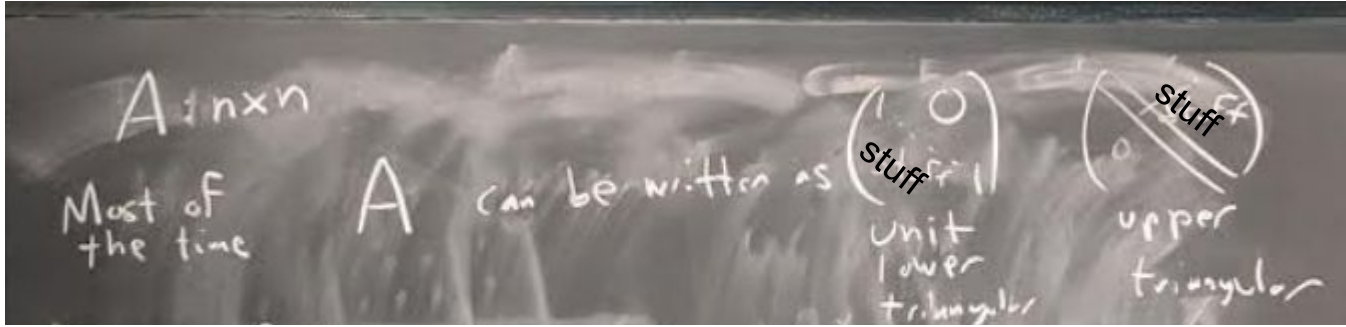
$$x = A \backslash b$$

emphasizing the leftward looking division.



← Generalizes $n=2$ like you may have seen in 7th grade.

$$A = (\text{unit lower triangular}) \times (\text{upper triangular})$$



Exists if the top left entry is not 0, and the top left corner 2×2 is invertible, and the top left 3×3 is invertible, ..., up to the top left $n-1 \times n-1$.

L has $1 + 2 + \dots + n-1 = n(n-1)/2$ parameters, U has $n(n+1)/2$. Together one gets n^2 parameters, corresponding to that of A.

Inner and Outer Products, and matmul tricks

n vectors are sometimes thought of as $n \times 1$ matrices. This causes little trouble in a linear algebra context and tons of trouble in software with vectors and matrices.

The Dot Product is also known as the inner product

Dot Product: $x \cdot y = \sum x_i y_i$
 $x^T y$ think of x as $n \times 1$ and y as $n \times 1$
"T" is inner

$xy^T = \begin{bmatrix} x_1 y_1 & \dots & x_1 y_n \\ \vdots & \ddots & \vdots \\ x_m y_1 & \dots & x_m y_n \end{bmatrix}$ outer product
"T" is outer

$(A^T B)_{ij}$ = dot product of col i of A w/ col j of B
 $(AB^T)_{ij}$ = " " " " row " " " " " "

$(AB)^T = B^T A^T$ $(ABC)^T = C^T B^T A^T$...

(Matmul is a commonly used abbreviation for Matrix Multiply)