

# Lecture 3 18.06 (2020) [Syllabus link](#)

## Solving $Ax=b$

### Outline

1. Many views on the same problem
2. Breakdown
3. Special Matrices, Complexity
  - a. Diagonal
  - b. Identity
  - c. Upper Triangular
  - d. General Matrices = Gaussian Elimination

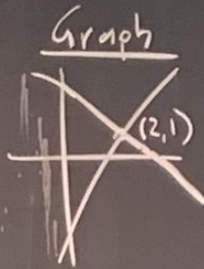
# 2x2 systems: High School view / Linear Transformation View

2x2 systems

High School Algebra

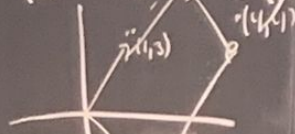
$$x + 2y = 4$$

$$3x - 2y = 4$$



Linear Combo

$$x \begin{pmatrix} 1 \\ 3 \end{pmatrix} + y \begin{pmatrix} 2 \\ -2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



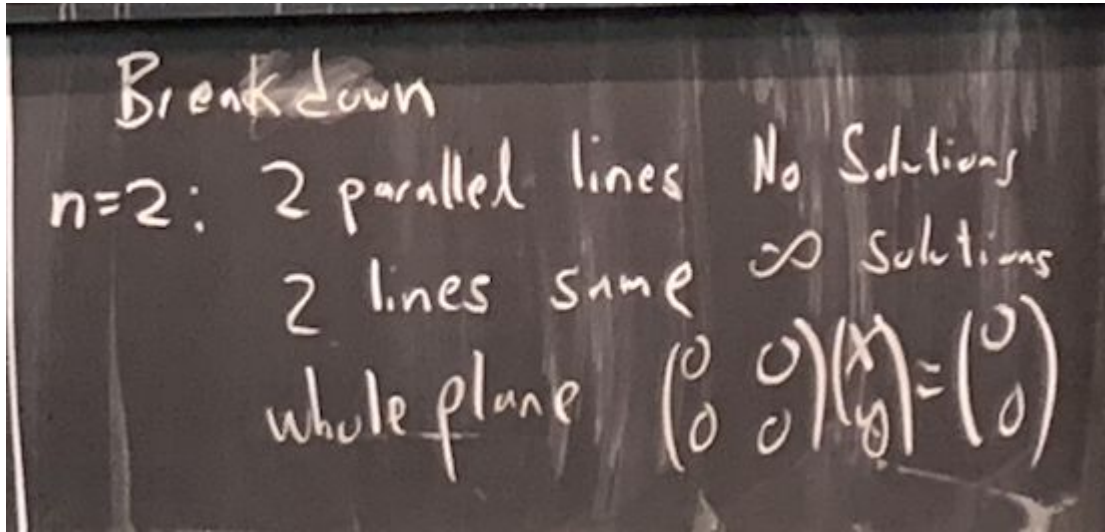
Matrix Algebra

$$\begin{pmatrix} 1 & 2 \\ 3 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$

Lin Trans View

After transforming  
"orig ear" is at (4,4)  
where did it start?

## 2x2 examples of breakdown



## 3x3 systems

Example

$$\begin{pmatrix} 1 & 4 & 1 \\ 4 & 3 & 1 \\ 5 & 2 & 6 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$$

Generically the picture would be three planes intersecting at a point.

Some breakdown possibilities: no solutions or infinitely many (think about what else there could be)

parallel planes — no solutions!

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \text{ Solutions: } \mathbb{R}^3$$

$n=4$  and higher

$n=4$  and higher

Hyperplane ← "Picture" of a linear equation in high dim

$\mathbb{R}^2$  - Plane

$\mathbb{R}^3$  - 3 dim space

In general

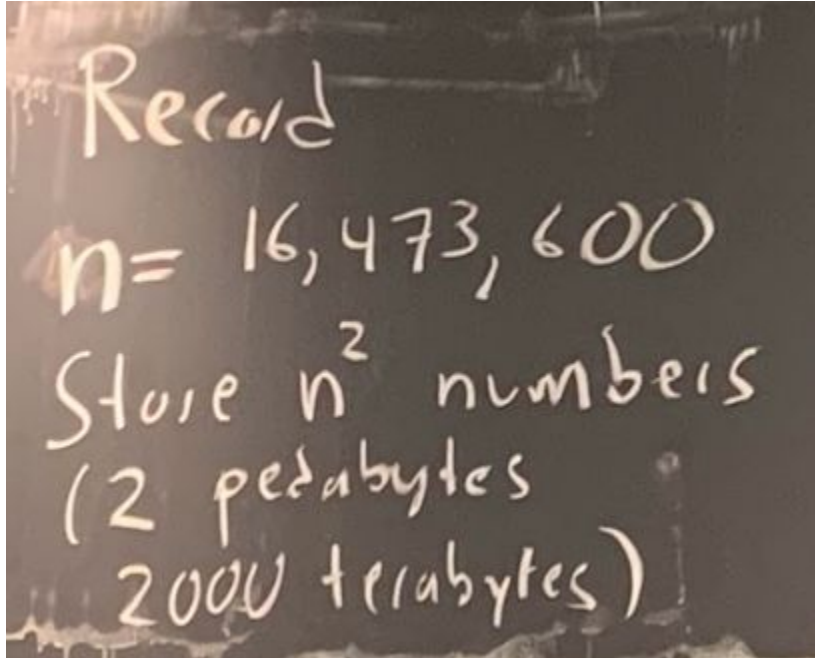
$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & & & a_{nn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$\text{Find } x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Solve

$$Ax = b \text{ for } x$$

# The record in 2020 for (general) systems solving



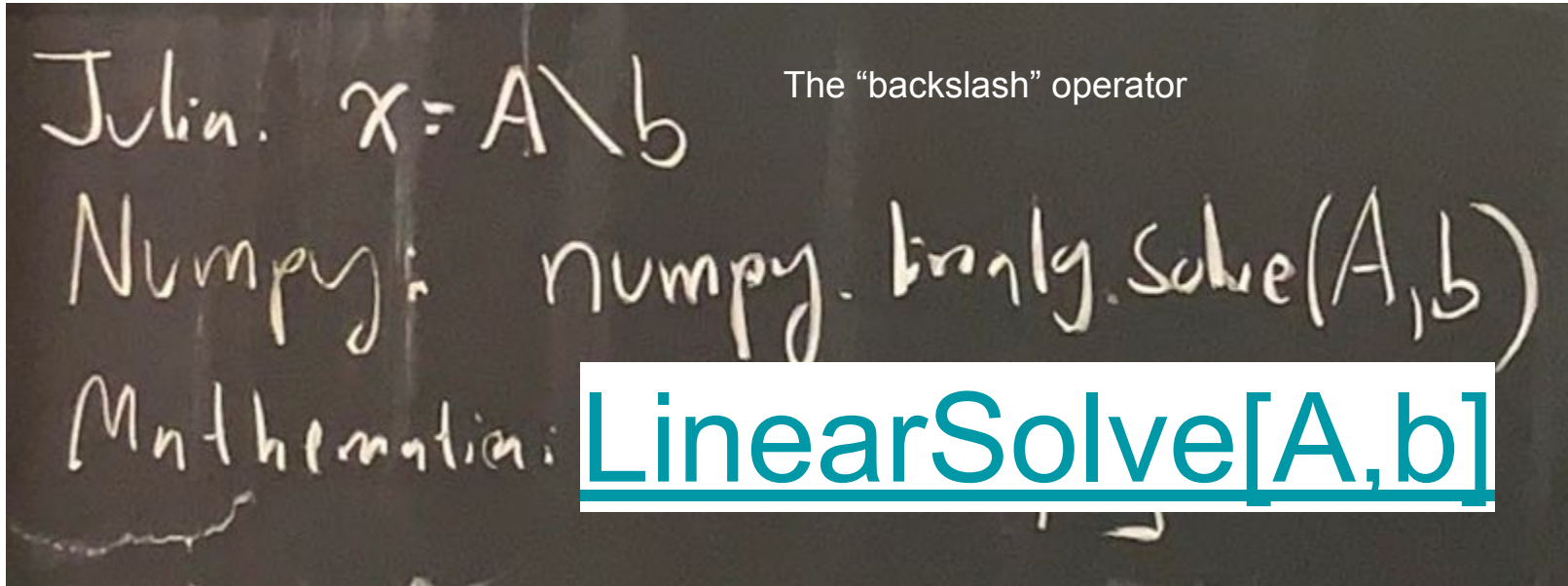
( Storing the matrix might require the memory  
In thousands of your laptops)

Another view : find the linear combination

$$\text{Find } x_1, \dots, x_n$$
$$x_1 \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} \\ \vdots \\ a_{n2} \end{pmatrix} + \dots + x_n \begin{pmatrix} a_{1n} \\ \vdots \\ a_{nn} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$



# Solving in a few packages (requires an A and a b)



Numpy: <https://numpy.org/doc/1.18/reference/generated/numpy.linalg.solve.html>

All packages use LAPACK under the hood for general systems. So solving is not truly different in any of these packages.



# Some Special Matrices, Storage and Complexity

3. Special Matrices

3a) Diagonal

$$D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$$

$n=3$

$$D = \begin{pmatrix} 1 & & \\ & 2 & \\ & & 3 \end{pmatrix} \leftarrow \text{Storage needed } n$$
$$DX = \begin{pmatrix} 4 \\ 8 \\ 12 \end{pmatrix} \quad X = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}$$

$DX=b$

$$d = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \leftarrow \text{diagonal entries}$$
$$x = b/d \quad \leftarrow n \text{ divisions}$$

$N_{\text{div}} = 0$

Complexity - How many operations

Complexity - How many operations
"Identity"
$I = I_n = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$
Storage: $n$
Complexity: $Ix=b, 0$

Diagonals need to store  $n$  numbers  
And solving  $Ax=b$  requires only  $n$  Divisions. We say Diagonals have  $n$  parameters.

The identity has only one parameter (the number " $n$ ")  
And requires no operations to solve  $Ix=b$ .

# Upper Triangular

Often denoted U for “upper”

Requires  $n(n+1)/2$  storage locations (or parameters) which we approximate for  $n$  largish by  $n^2/2$

Solution method called “back substitution” as we can work backwards with only one unknown at a time that we can solve for as long as the diagonal is not zero.

The image shows handwritten notes on a chalkboard. On the left, under the heading "Triangular", a system of equations is written: 
$$\begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 \\ -8 \end{pmatrix}$$
 Below this, the equations are solved: 
$$2y = -8$$
$$y = -4$$
 In the center, a vertical line separates this from the right side. On the right, under the heading "Upper Triangular", a matrix  $U$  is shown: 
$$U = \begin{pmatrix} u_{11} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & u_{nn} \end{pmatrix}$$
 To the right of the matrix, the summation formula is written: 
$$1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

# Back Substitution

Step 0:  $u_{nn}x_n = b_n$  Solution:  $x_n = b_n / u_{nn}$

Step 1:  $u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1}$  Solution:  $x_{n-1} = (b_{n-1} - u_{n-1,n}x_n) / u_{n-1,n-1}$

...

Step k: Solve  $\sum_{i=k,\dots,n} u_{ki} x_i = b_k$  for  $x_k$  given  $x_{k+1}$  through  $x_n$  are now known

The solution:  $x_k = (b_k - \sum_{i=k+1,\dots,n} u_{ki} x_i) / u_{kk}$

...

# Complexity of upper triangular solve

Step  $k$  has  $k$  “ $\pm$ ”s,  $k$  multiplies, and one divided

Approximate count is  $n^2$  operations (work it out!)

Using  $(1+2+\dots+n-1) = n(n-1)/2 \approx n^2/2$  for large  $n$

# What is a parameter?

A parameter is defined as a number that is needed to specify an object.

So we have seen diagonals have  $n$  parameters, upper triangular  $n(n+1)/2$ , and unit lower triangular  $n(n-1)/2$ .

A vector that sums to 0 only needs  $n-1$  parameters because if you know  $n-1$  you can deduce the last one.

The identity needs only the number  $n$ . So does the  $n \times n$  zero matrix.

# When do we need to include the “n” as a parameter?

When talking about parameters, normally we begin with some set in context.

- Example 1. The set of all vectors in  $\mathbb{R}^n$ . Parameters,  $v_1, \dots, v_n$ . The  $n$  is “built” into the set.
- Example 2. The set of all vectors. Here  $n$  can be inferred from the parameters. On a computer if you type `[1,5,6,9]`, the computer can infer you have a vector in  $\mathbb{R}^4$ .
- Example 3: The set of all  $n \times n$  upper triangular matrices. Again, “built-in”
- Example 4: the set of all upper triangular matrices. The  $n$  can be inferred.
- Example 5: The set of vectors orthogonal to `[1,1,...,1]`. If you specify  $n-1$  numbers, then  $n$  can be inferred as one more than the list size.
- Example 6: The set of all identity matrices.  $n$  is required as a parameter.