Lecture Slides for 18.06 Linear Algebra Spring 2019

Alan Edelman

Syllabus link (serves as a TOC for these slides)

(MIT 18.06 students, comments, fixes, request for further clarifications, welcome. Please keep mathematical)

Lecture 1. A Modern (Personal View) of Linear Algebra

Lecture 1. A Modern (Personal View) of Linear Algebra

The many many applications of linear algebra to science, engineering, machine learning, statistics, ….

The abstractions that let you understand the very fabric of mathematics.

Lecture 1. A Modern (Personal View) of Linear Algebra

Increasingly used in practical Classes worldwide, machine Learning, big data: (See Julia.)

The many many applications of linear algebra to science, engineering, machine learning, statistics, ….

The abstractions that let you understand the very fabric of mathematics.

Lecture 1. A Modern (Personal View) of Linear Algebra

MIT Robotics

Robot Locomotion

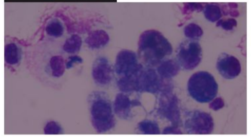
PSR Energy

Energy Analytics and Optimization

NY FEDERAL RESERVE Central Banking

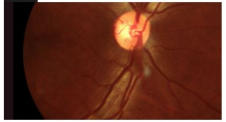
New York Federal Reserve Bank

PATH BIOANALYTICS Precision Medicine



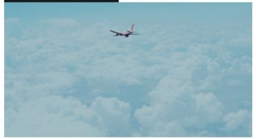
Precision Medicine

IBM Medical Diagnosis



Deep Learning for Medical Diagnosis

LINCOLN LABS / FAA Aviation



Safer Skies

The abstractions that let you understand the very fabric of mathematics.

Lecture 1. A Modern (Personal View) of Linear Algebra

MIT Robotics

Robot Locomotion

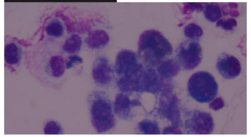
PSR Energy

Energy Analytics and Optimization

NY FEDERAL RESERVE Central Banking

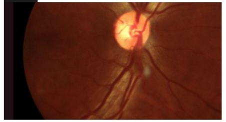
New York Federal Reserve Bank

PATH BIOANALYTICS Precision Medicine



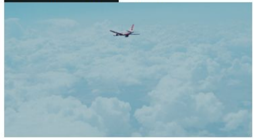
Precision Medicine

IBM Medical Diagnosis



Deep Learning for Medical Diagnosis

LINCOLN LABS / FAA Aviation



Safer Skies

Vector Spaces, Linear Transformations. The relationship of the continuous to the discrete. Seeing things in high dimension spaces, like line fitting.

Lecture 1. A maybe silly analogy with, say, the number 7

Practicality:
The number 7 is useful. You can do things with it, in your head, by hand, or with a machine:

7 + 7 = 14
7 x 7 = 49
7 to the power 7 = I need a computer...

Abstraction:
The idea that “seven”, 7,7.0, vii, siete, sept, 七, שבע, سبعة, เจ็ด, Семь, सात, সাত, Yedi

Are all somehow the same....
Any 4 year old would tell you that the above are all different. Any 6 year old might say they are not!

What almost never happens on computers:

Matrix Inverses

Free variables and pivot variables

Echelon Forms

Determinants (though these are theoretically of immense value)

Representations of large matrices as tables of numbers

What does happen

Structured matrices (and of course sparse)

The SVD (and eigenvalues, but especially the SVD)

Lecture 1:

Linear Algebra lets you not miss the forest for the trees.

Sometimes a matrix IS just a table of numbers.

Sometimes a matrix should not be thought of as a table of numbers.

Sometimes one should step back and see a matrix as being its own entity.

Sometimes index notation A_{ij} is distracting.

Sometimes index notation A_{ij} is comforting.

… this may not make sense yet, but will over the course of the semester.

How 18.06 may differ this semester

- More emphasis on the conceptual and the practical.
- Hand computations: some, but not more than is needed.
- No pivot variable, free variable, echelon forms.
- Less emphasis on hand computation of inverses.
- Raise the importance of the Singular Value Decomposition (we will not think of it as much as an offshoot of the Eigendecomposition, but being its own creature)
- Matrix calculus: gradients with respect to vectors and matrices (Not in 18.02, not really anywhere in standard math courses)
- Applications to machine learning, statistics, and other areas
- Raise importance of linear transformations
- Less overlap with 18.03

1a. vectors

Examples of vectors:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad \text{or} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{such as} \quad x = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad \text{or} \quad y = \begin{bmatrix} 2.3 \\ 5.6 \\ 7.8 \end{bmatrix}$$

x is a 5 vector

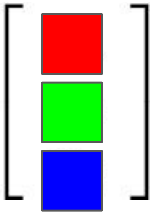
$x \in \mathbb{Z}^5$ ($\mathbb{Z}$ = the integers = $\{0, \pm 1, \pm 2, ...\}$)

$y \in \mathbb{R}^3$ ($\mathbb{R}$ = the reals)

What can be elements of a vector?

Computers use vectors to contain all kinds of objects. Mathematicians usually insist that vector elements come from something (beyond the scope of this class) called a field. The real and the complex numbers are the most typical elements.

Going with the more permissive meaning of vector we can imagine:

$y =$  $\in$ (colors)^3 or nested vectors such as $\begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\ \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\ \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \\ \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix} \end{bmatrix} \in (\mathbb{Z}^3)^4$

1b. Matrices

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \end{bmatrix}$$

Example

$$A= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

A is 3 x 5
$A \in \mathbb{Z}^{3,5}$
Julia: A has type Array(Int64,2)

Int64 means a 64 bit element type and 2 means a 2 dimensional array: a matrix.

1c. Matrix times vector

This you can do in your head:

$$\begin{bmatrix} 12 \\ 32 \\ 52 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

1c. Matrix times vector:

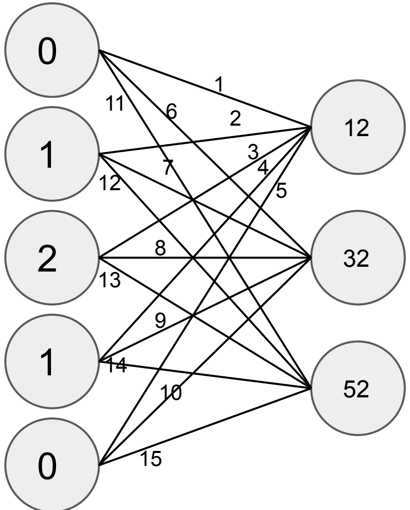
This maybe better done on a machine:

$$\begin{bmatrix} 276 \\ 640 \\ 476 \\ 612 \\ 800 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 4 \\ 1 & 5 & 9 \\ 2 & 6 & 5 \\ 3 & 5 & 8 \\ 9 & 7 & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Neural Network Notation for matrix-vector

Popular right now but perhaps cumbersome

$$\begin{bmatrix} 12 \\ 32 \\ 52 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

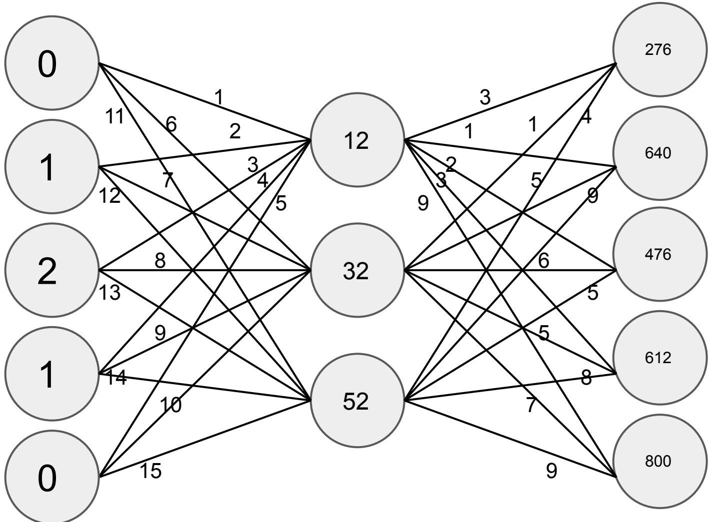


This is called a "linear neural network"
Matrix elements are called weights.

While a bit cumbersome to draw, it does illustrate nicely how every element of the output depends on every element of the input. Sometimes this picture is called a "[complete bipartite graph]"

Matrix times Matrix times vector denoted with a neural net

$$\begin{bmatrix}276\\640\\476\\612\\800\end{bmatrix} = \begin{bmatrix}3&1&4\\1&5&9\\2&6&5\\3&5&8\\9&7&9\end{bmatrix}\begin{bmatrix}1&2&3&4&5\\6&7&8&9&10\\11&12&13&14&15\end{bmatrix}\begin{bmatrix}0\\1\\2\\1\\0\end{bmatrix}$$



Where do the rows of A connect?
Where do the columns of A connect?
Same questions for B?

Lecture 2

Is this a computer scientist’s linear algebra? (of course it isn’t)

Answer: Mathematics is for everyone. Linear Algebra is for everyone.

This class will be as useful for the biologists, the economists, the physicists, the math majors, as well as the computer scientists and electrical engineers.

And maybe even a janitor or two. (for those who saw the movie “Good Will Hunting”)

Summary

Today we will look at various mathematical structures that allow for linear combinations with real numbers. Informally, we will say that any such structure, discrete or continuous, will be called a vector space.

We will see that traditional column vectors of size n, or matrices of size mxn, or functions of a real variable, or even differential operators form a vector space.

This is what we mean by abstraction. On the surface, all of the four spaces above may seem very different. Yet they have something deeply in common. Mathematicians love to find structures that seem different and yet have something in common, and then study the common structure all at once, rather than dealing with the unimportant surface differences.

We feel we are unfolding the very fabric that underlies everything.

Linear Combinations by example

1. Vectors

If v_1 and v_2 are in $\mathbb{R}^n$ and c_1, c_2 are in $\mathbb{R}$
then $c_1v_1 + c_2v_2$ is a <u>linear combination</u> of $v_1 + v_2$.

Examples: (left n=5 with variables, right n=2 with numbers)

$$c_1\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_5 \end{bmatrix} + c_2\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_5 \end{bmatrix} = \begin{bmatrix} c_1x_1 + c_2y_1 \\ c_1x_2 + c_2y_2 \\ \vdots \\ c_1x_5 + c_5y_5 \end{bmatrix}$$

$$2\begin{bmatrix} 5 \\ 6 \end{bmatrix} + 3\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 16 \\ 21 \end{bmatrix}$$

Remark: We'll see in a moment it's reasonable to talk about linear combinations of any number of vectors, not just two. (Notation: $\mathbb{R}$ denotes the real numbers.)

Linear Combinations by example

2. Matrices

Linear Combinations of Matrices

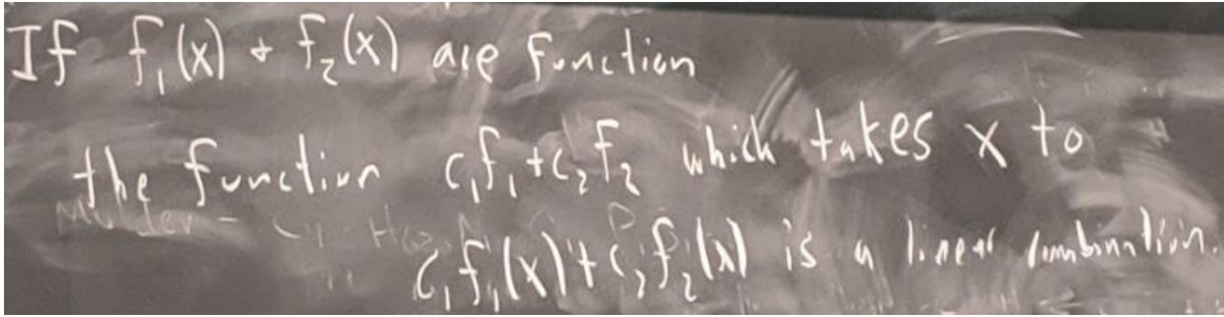
$A \in \mathbb{R}^{m,n}$ & $B \in \mathbb{R}^{m,n}$ $\quad c_1, c_2 \in \mathbb{R}$

$c_1 A + c_2 B$ is a linear combo. of A & B

$$2\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix} + 5\begin{bmatrix}0 & 1\\ 1 & 2\\ -1 & 0\end{bmatrix} = \begin{bmatrix}2 & 9\\ 11 & 18\\ -5 & 12\end{bmatrix}$$

← Silly me. Bottom Left entry should be +5.

Linear Combinations
By example



3. (Real valued) Functions

Consider functions like sin or pow_k defined as the function that takes to sin(x) and x^k respectively.

Can we talk about 10*sin? Sure, it takes x to 10*sin(x). (the times “star” is optional)

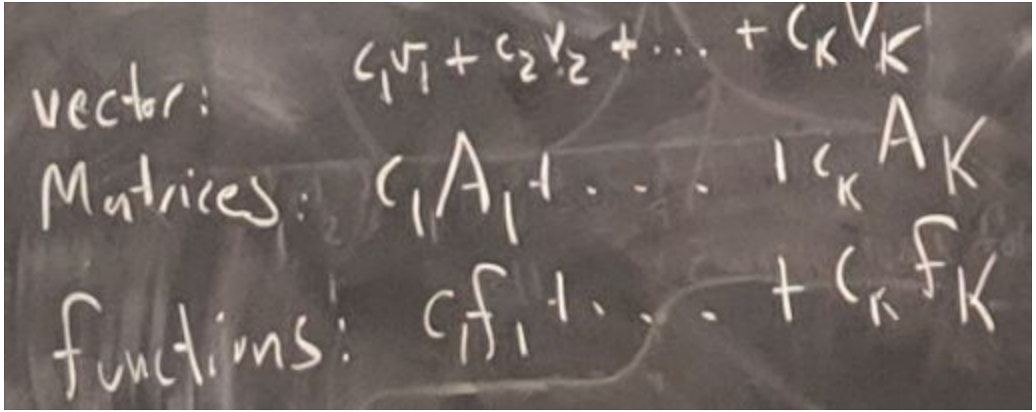
f = (10 * pow_2 + 3 * pow_4) ?

Sure we can, this function takes any x to $10x^2+3x^4$. We can write $f(x) = 10x^2+3x^4$.

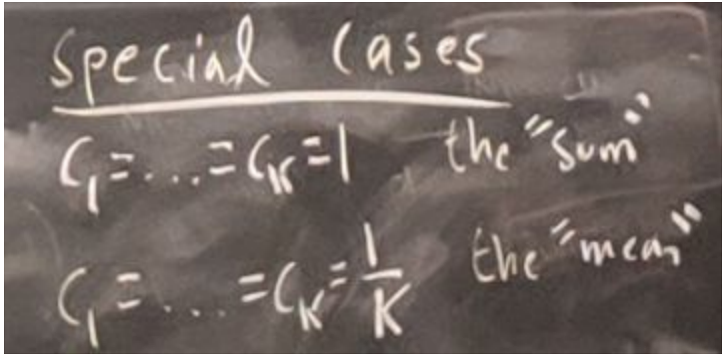
Thus (10 * pow_2 + 3 * pow_4)(2) = 10*4 + 3*16 = 88.

f = 5 exp + 2 log ? Sure $f(x)=5e^x+2\log(x)$.

Linear combinations can be any number of items



← vectors, matrices, and functions: all allow for linear combinations. Mathematicians: let’s focus on what we have in common rather than our differences. (Editorial: All too often humans can take a lesson here.)

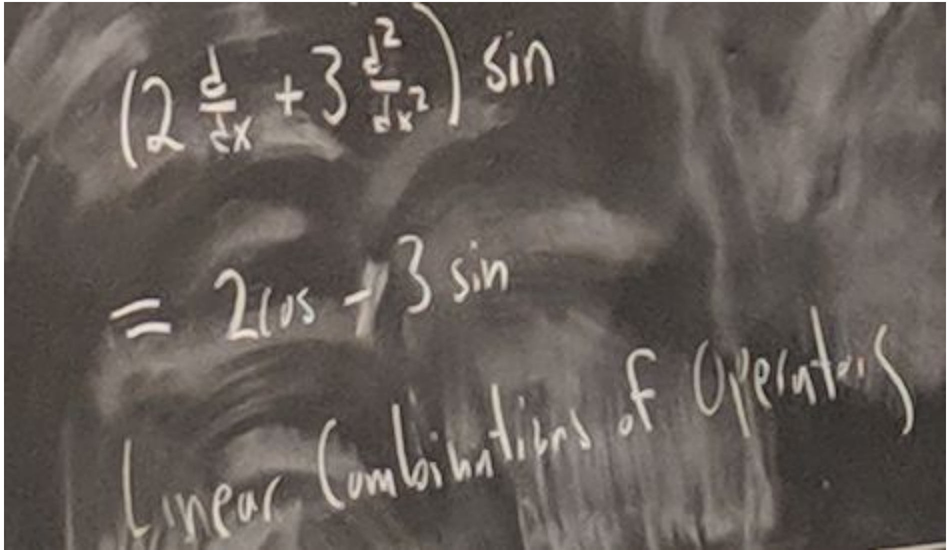


See VMLS pages 17 and 18 for other examples.

Linear Combinations by example

Linear Combinations of Operators such as d/dx or d^2/dx^2.

$(2 * d/dx \;\; + 3 * d^2/dx^2)(f) = 2f' + 3f''$



Example when f = sin

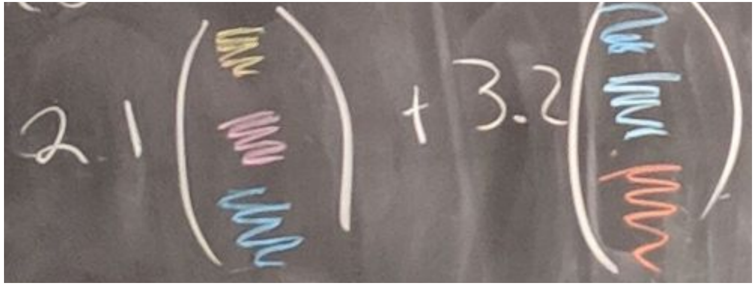
A function is a rule for taking numbers or vectors into new numbers or vectors.

An “operator” might be thought of as a “function function” in that it provides a rule for turning one function into another function.

Vector Spaces

Informal definition of a (real) vector space: Any mathematical set where it is sensible to take real linear combinations, and not go outside the set.

Not $\mathbb{Z}^3$ though, because real linear combinations of integers are not integers.



Also not colors^3 unless some meaning is given to quantities such as the linear combination on the left.

Elementwise Operations

Consider f(x) = sin(x), x^2, or e^x. What should we mean by f(vector) or f(matrix)?? Sometime one sees elementwise application.

For example

$$f\begin{pmatrix}1\\2\\3\end{pmatrix}=\begin{pmatrix}f(1)\\f(2)\\f(3)\end{pmatrix} \quad \text{or} \quad f\begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix}=\begin{pmatrix}f(1)&f(2)&f(3)\\f(4)&f(5)&f(6)\\f(7)&f(8)&f(9)\end{pmatrix}$$

The problem is that squaring a vector leads to confusion, e.g. is it the dot product, or the cross product (which really is a three dimensional thing only) or not really defined. Squaring a matrix is even more confusing -- the matrix multiply or the elementwise square?

Also $e^A = I + A + A^2/2! + A^3/3! + A^4/4! + \ldots.$ Is a thing that is different from the elementwise exponential as every power of A denotes matrix multiply.

Elementwise notational solution

The simple little dot or point (".") which may be thought of as pointwise (synonymous with elementwise) nicely solves this problem.

$$f.\begin{pmatrix}1\\2\\3\end{pmatrix}=\begin{pmatrix}f(1)\\f(2)\\f(3)\end{pmatrix}$$

$$f.\begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix}=\begin{pmatrix}f(1)&f(2)&f(3)\\f(4)&f(5)&f(6)\\f(7)&f(8)&f(9)\end{pmatrix}$$

Elementwise notational solution

f.(vector) or f.(matrix) makes perfect sense elementwise for any function f .

Without the dot

if $f(x)=x^2$

f(vector) should be undefined, f(matrix) should be the matrix multiply square

And if $f(x)=e^x$

f(vector) should be undefined f(matrix) should be the matrix exponential.

Many authors rely on context to disambiguate. Benefits: saves the dots, Downside: can be confusing. Especially if you did want to write a software program. Some find the dots hard to read at first, but useful with practice.

Elementwise notational solution

Can be chained

Notation for a nonlinear neural network: h.(C* h.(B*h.(Ax)))

Example

h(x) is a nonlinear "activation function" such as h(x) = max(x,0). ("activates" when x is positive). (Old fashioned language that seems to be sticking, unfortunately: h(x) is the ReLU, or rectified linear unit function.)

Typical example x is an n_0-vector. A is an n_1 by n_0 matrix.

B is an n_2 by n_1 matrix, and C is an n_3 by n_2 matrix, etc.

Also elementwise

A ./ B is elementwise divide

A .+B is the same as A+B for ordinary vectors and matrices but can generalize

etc

Lecture 3

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 x n, A^T is n x m

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

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

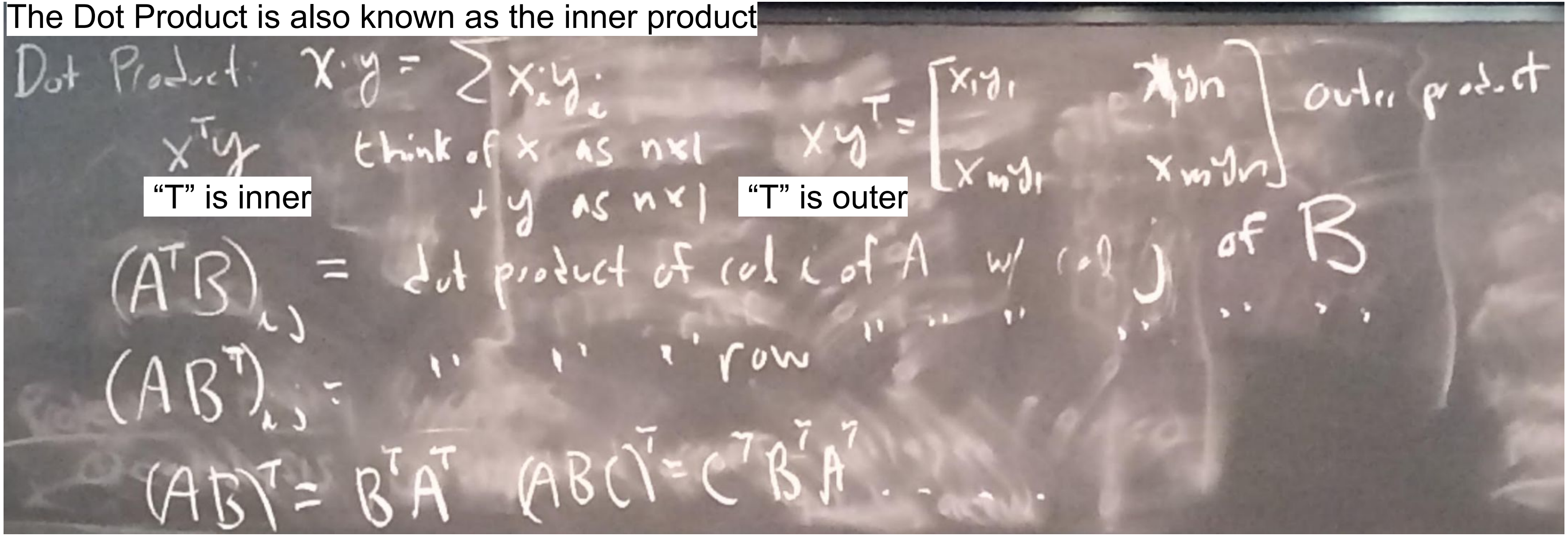
```
A = [1 2;3 4;5 6]
B = A'
```

```
2×3 LinearAlgebra.Adjoint{Int64,Array{Int64,2}}:
 1  3  5
 2  4  6
```

Inner and Outer Products, and matmul tricks

n vectors are sometimes thought of as n x 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

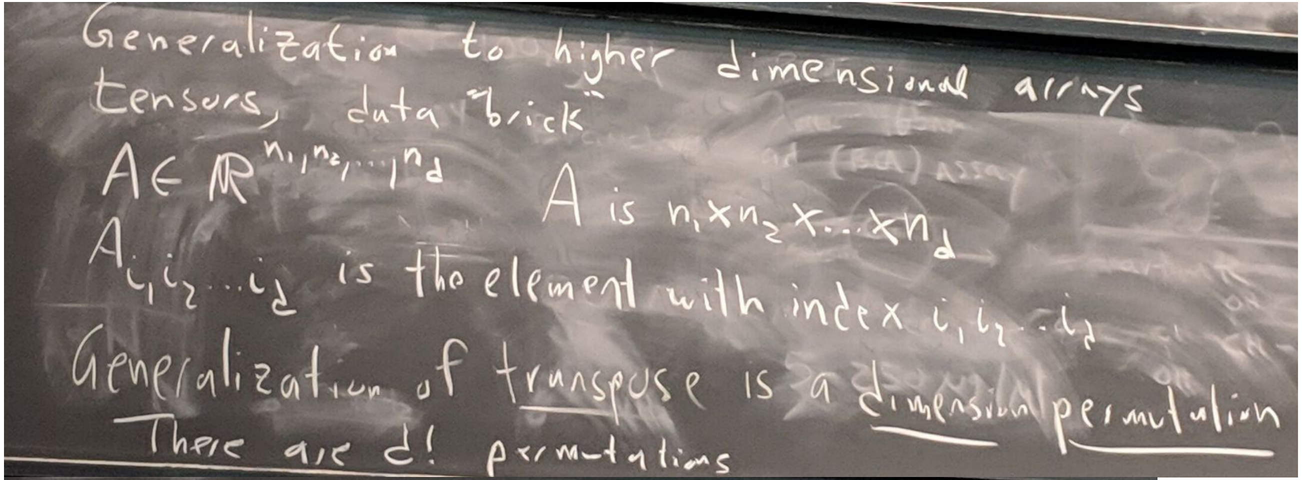


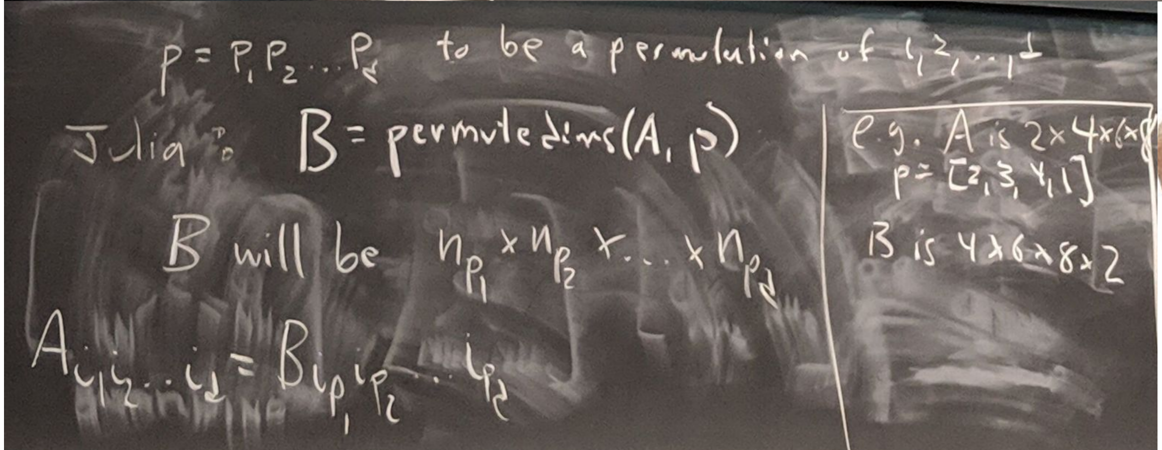
“T” is inner

“T” is outer

(Matmul is a commonly used abbreviation for Matrix Multiply)

Data “Bricks” aka Tensors or Multidimensional Arrays





Transpose generalizes to a “dimension permutation”

Inverses

Only applies to n x 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 x n system of linear equations)

Given n x n A, and right hand side b, if A is invertible, the unique solution is

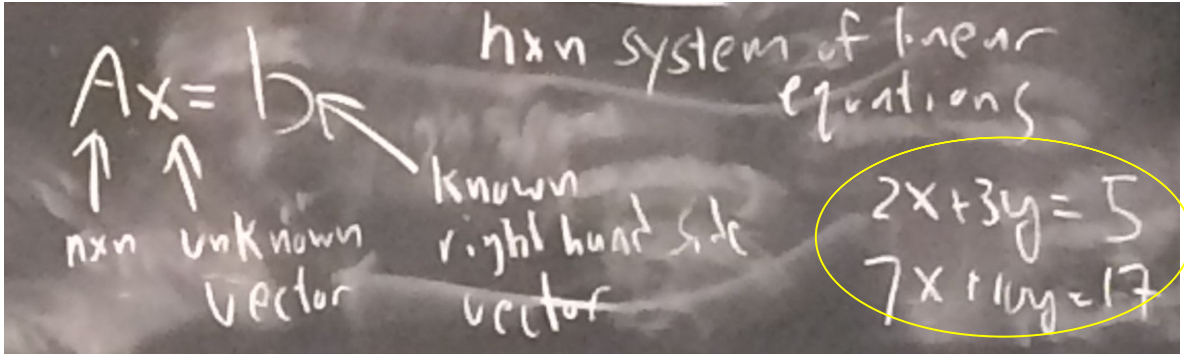
$$x=A^{-1}b$$

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

$$x=A\backslash b$$

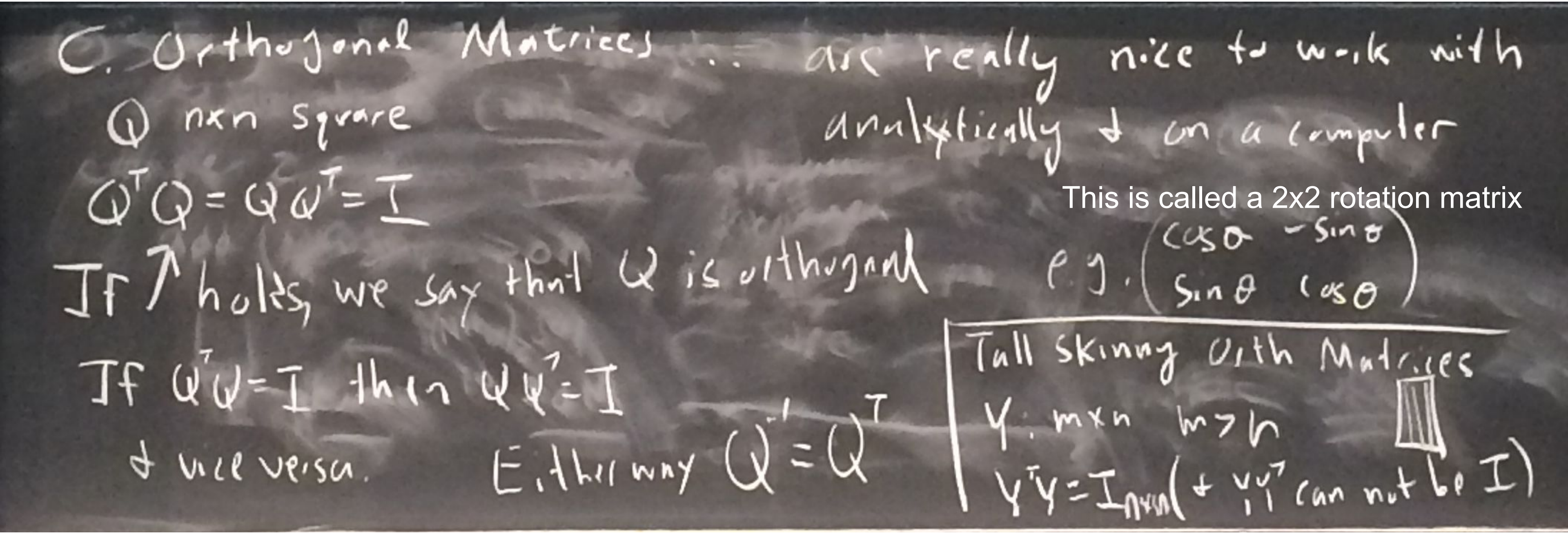
emphasizing the leftward looking division.

Largest general system solved (Nov 2018) is
Nmax = 16,693,248



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

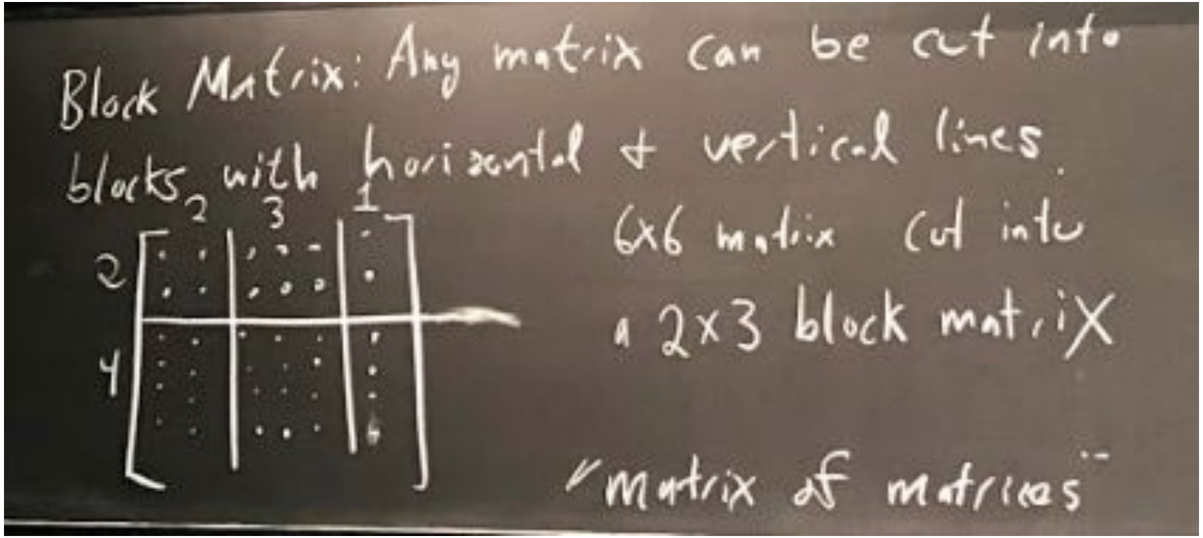
Orthogonal Matrices



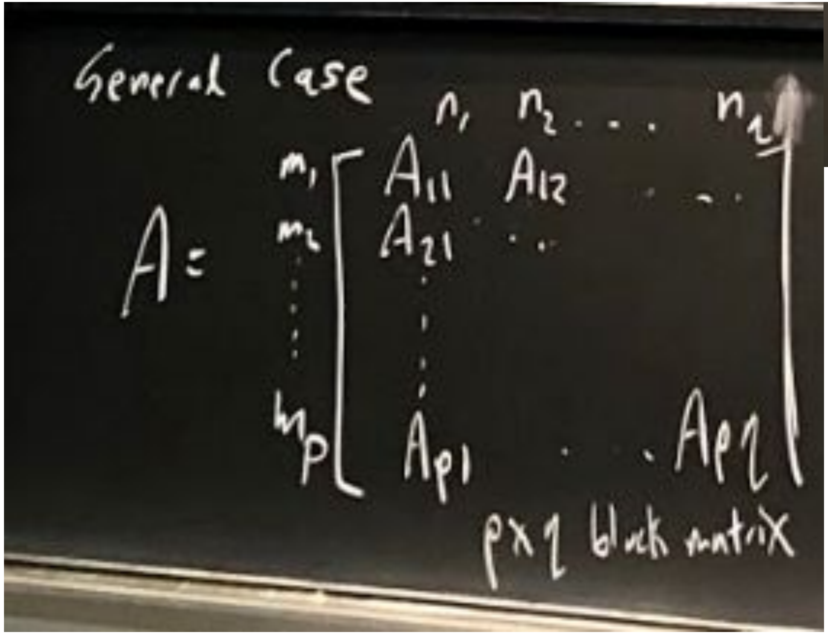
Terminology note: we never say that a matrix is orthonormal, but we do say that the columns of an orthogonal (or tall-skinny) orthogonal matrix are orthonormal.

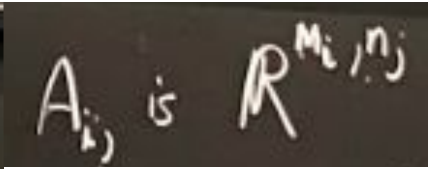
4. Block Matrices

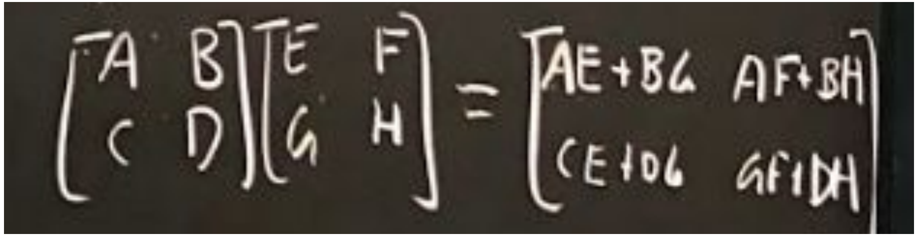
Leads to many new views of matrix multiply. Fuels the matrix multiplies used by scientific applications and machine learning on GPUs (graphics processing units)



General Case of a p x q block matrix



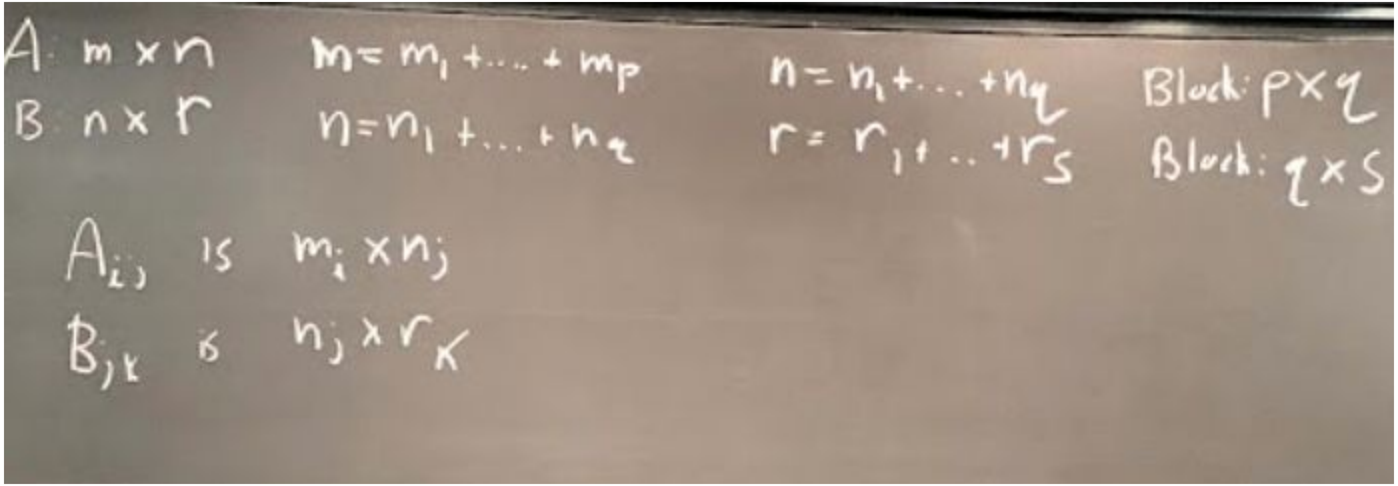




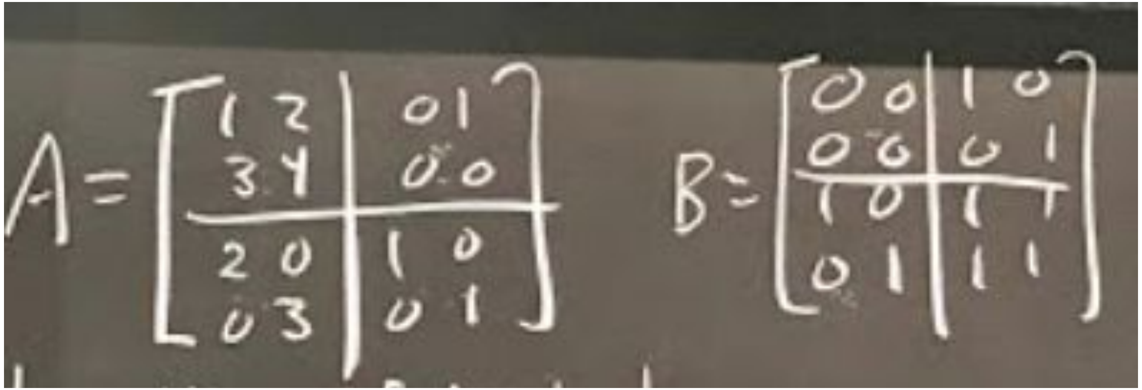
Example Block Matrix Multiply

When compatible , the matrix multiply always works! (Next slide spells out what compatible means)

Compatibility Conditions



Concrete Example

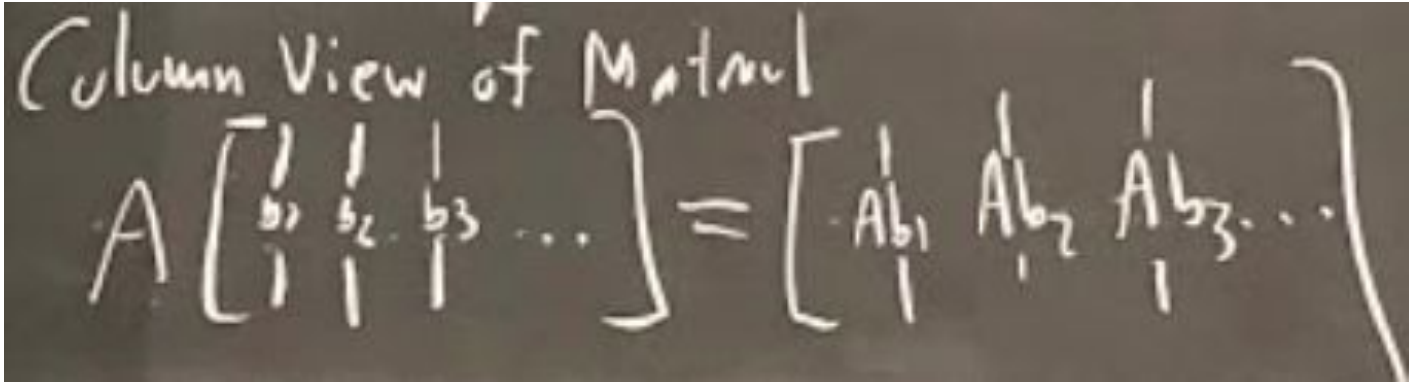


$$AB = \begin{array}{cc|cc} 0 & 1 & 2 & 3 \\ 0 & 0 & 3 & 4 \\ \hline 1 & 0 & 3 & 1 \\ 0 & 1 & 1 & 4 \end{array}$$

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \times \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} + \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \times \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix}$$

Etc. Do check the other 2x2 ! (I blew it in class)

Column view of matmul is a special case

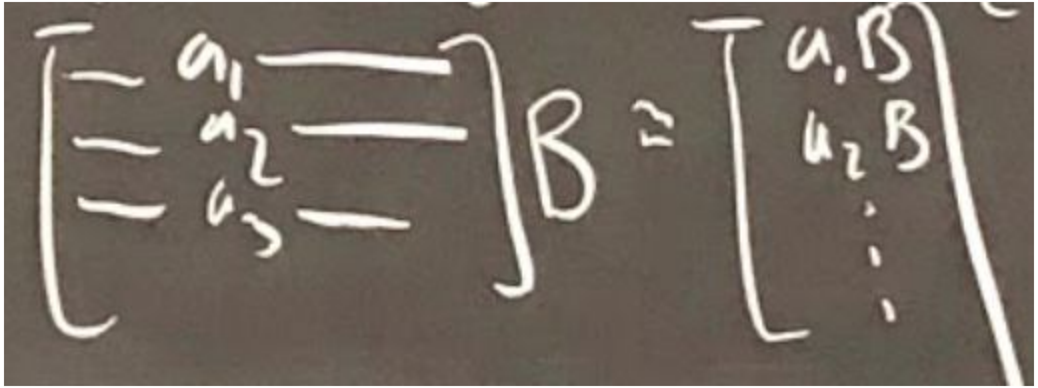


A is block 1 x 1 (A is made up of one m x n matrix)
B is block 1 x r   (B is made up of r columns of size n)

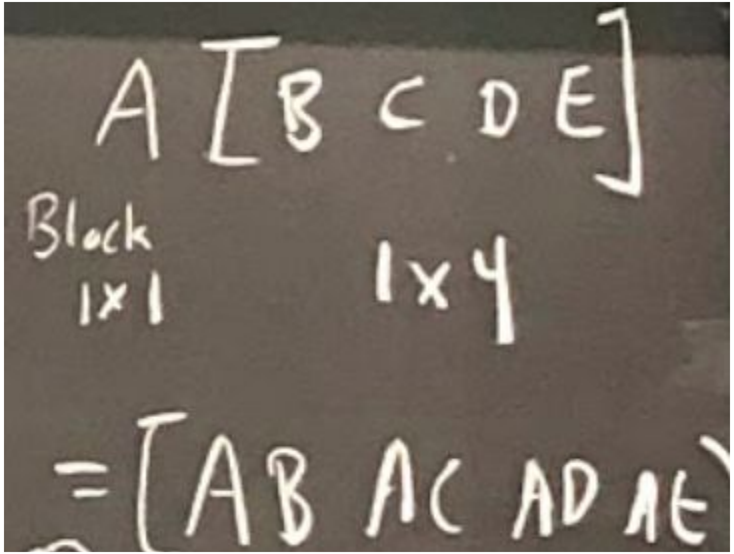
=

A*B is block 1 x r

So is the row view

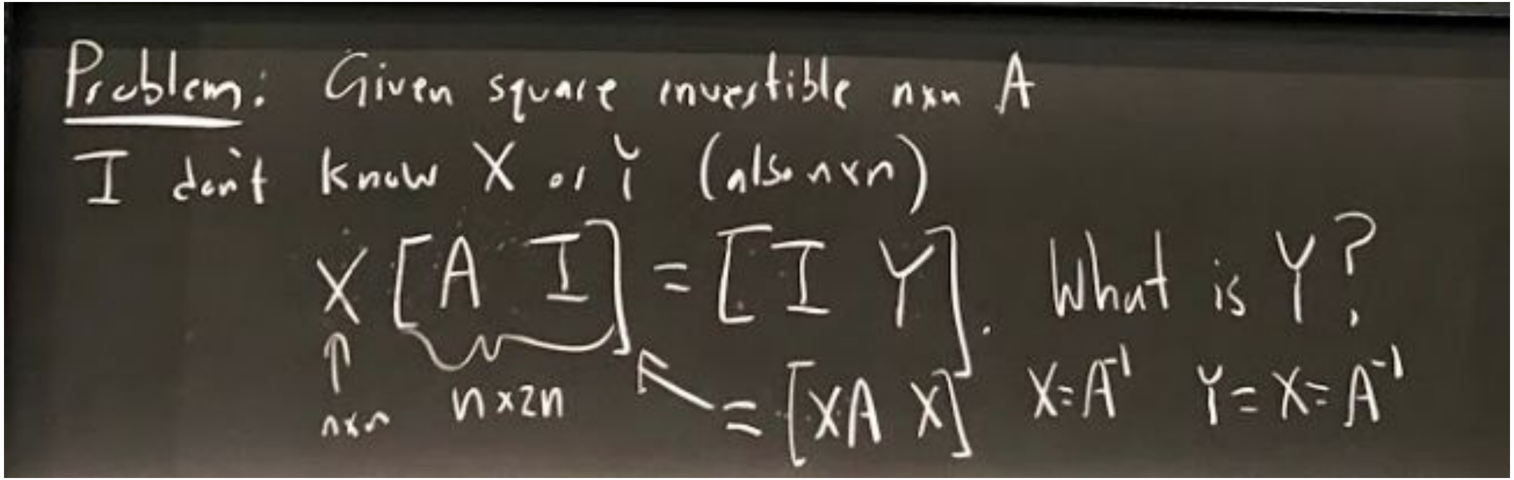


Another Example



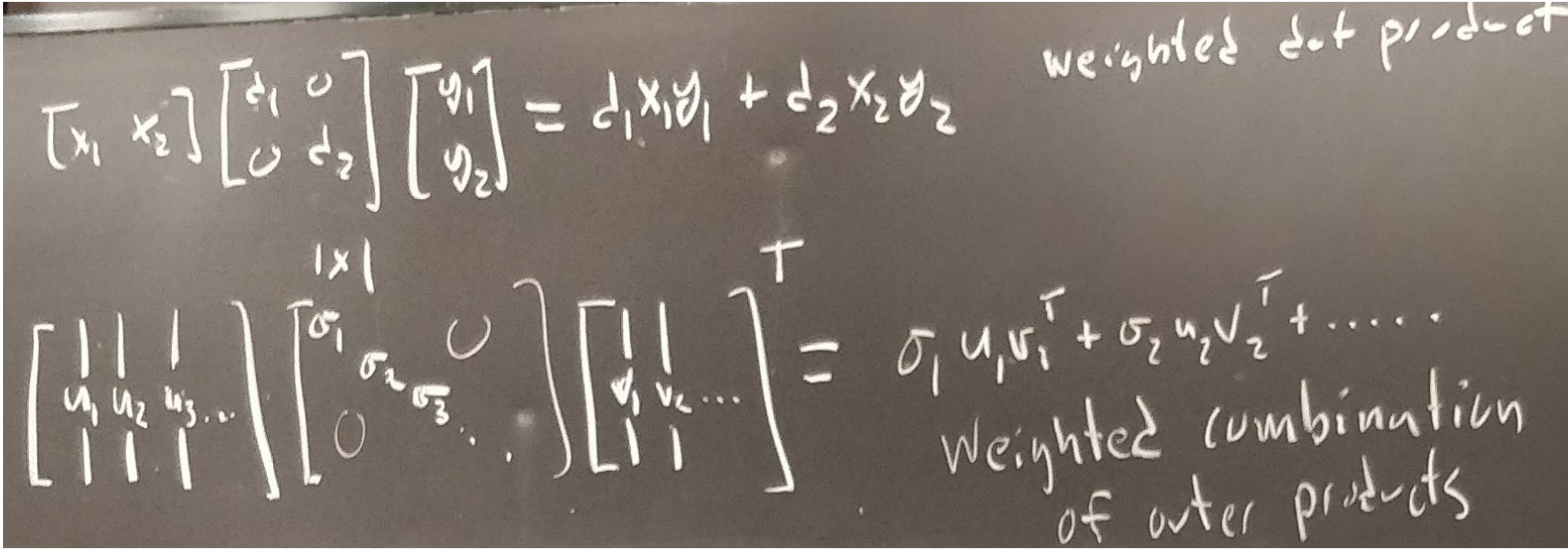
In general, A [B C D E F … G] = [AB AC AD AE AF … AG]

Application



This idea underlies a famous pencil and paper method (not used on computers) called Gauss-Jordan which we are not covering at this time but some of you may have seen.

Weighted dot products and weighted combinations of outer products



A diagonal matrix might be written as Diagonal([d_1,d_2]) or Diagonal([σ_1,σ_2,σ_3,...])

When might you use a weighted dot product?

Suppose you were buying 100 European Widgets at €7 each and 200 British Gadgets at £56 each, and 34 Japanese Mechanisms at ¥2300 each how much is the total cost in dollars?

Answer: x = [100 200 34] y=[€7 £56 ¥2300]' and D=the diagonal matrix of today's exchange rates: D=diagonal([1.13 $/€ , 1.29 $/£, .0090 $/¥])

x * D * y is the dollar cost for the whole shopping cart

When might you use a weighted sum of outer products?

Compression -- an approximation that can be very powerful is to take the outer products corresponding to the largest weights.   It can be more art than science to decide which ones, but nonetheless it can be very effective.

In a few lectures, when we do the svd, you will see an example of image compression using this very approach.

Further Applications

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = AC + BD$$

Block: 1×2 $\quad 2\times 1$

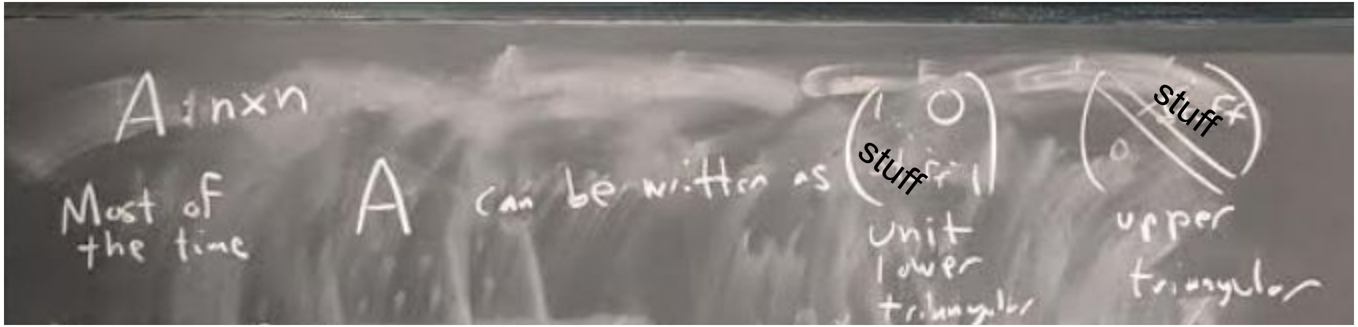
$$\underset{2\times 1}{\begin{bmatrix} A \\ B \end{bmatrix}} \underset{1\times 2}{\begin{bmatrix} C & D \end{bmatrix}} = \begin{bmatrix} AC & AD \\ BC & BD \end{bmatrix}$$

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix} = ACD$$

Factorizations We will See

	Solve Linear Systems	Least Squares	Data Compression and Other Applications and Theory
LU (lower times upper)	✓	❌	❌
QR (orthogonal times upper)	✓	✓	❌
SVD (U Diag(s) V’) Singular value decomposition	✓	✓	✓

A = (unit lower triangular) x (upper triangular)

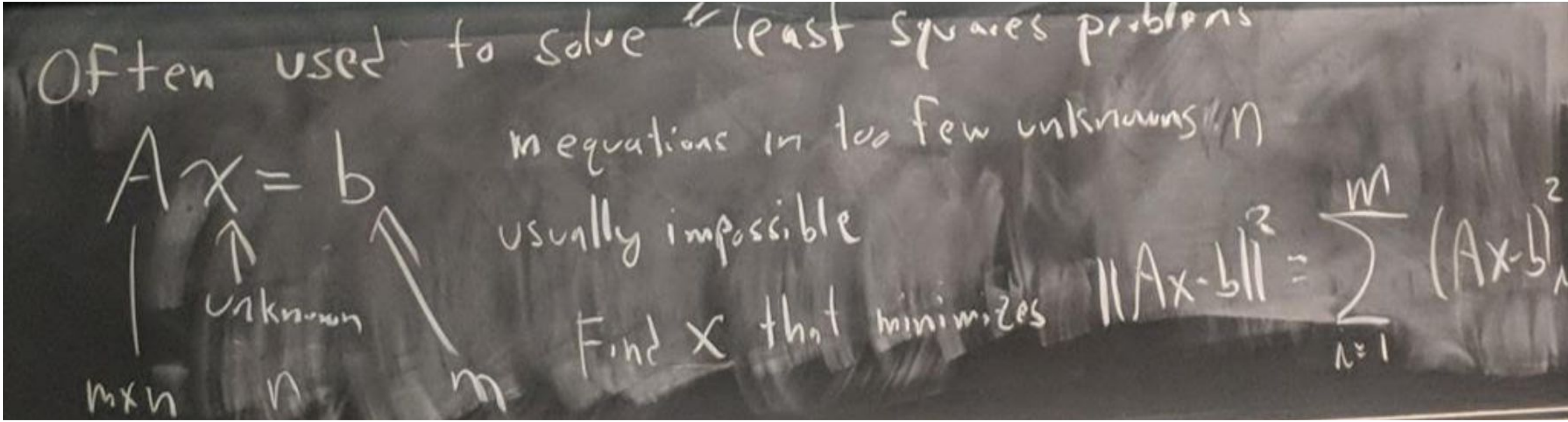


Exists if the top left entry is not 0, and the top left corner 2x2 is invertible, and the top left 3x3 is invertible, …., up to the top left n-1 x n-1.

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

Lecture 5

The QR factorization is useful when you have too many equations in not enough unknowns. Typically these equations can not be solved, so we settle for a least squares solution (one that minimizes the sum of the squares of the elements of Ax-b):



The answer (without intution or proof right now)

$Ax = b$ generally won't have a solution for an x

$QRx = b$ then is an impossible equation generally, but use it anyway!

$Rx = Q^Tb \Rightarrow x = R^{-1}Q^Tb$ (can be written $R\backslash Q^Tb$)

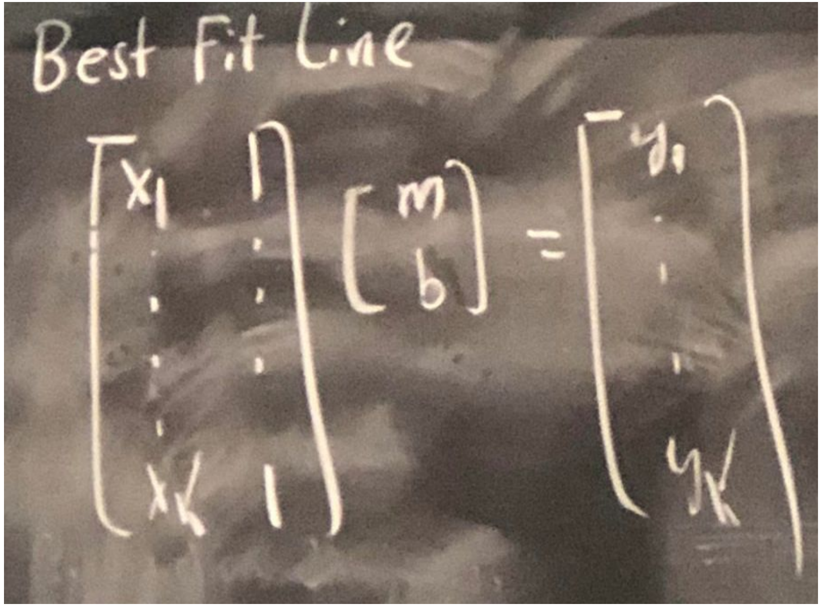
This x will (generally) give a least squares solution (not a solution to $Ax=b$, but rather an x that minimizes $||Ax-b||^2$)

Notation: $||v||^2 = v_1^2 + v_2^2 + \dots$

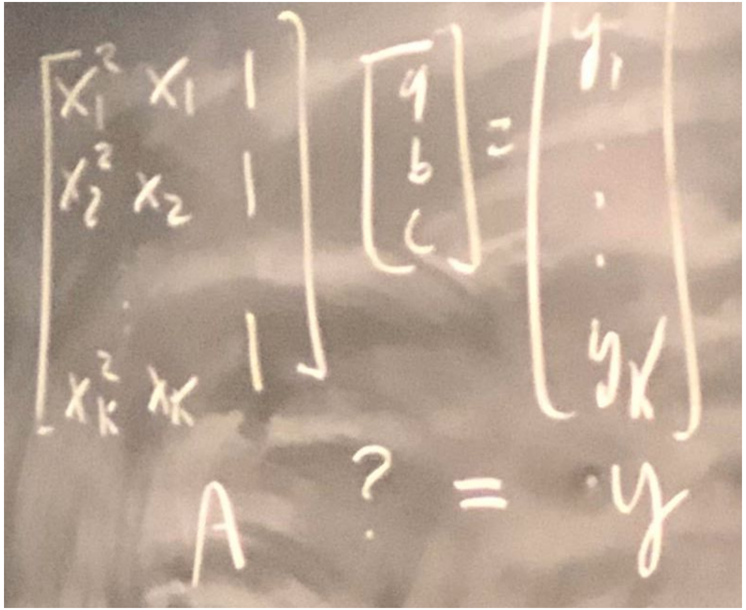
$||v||$ is synonymously pronounced (Euclidean) length(v), norm(v), magnitude(v)

Best fit line

minimizes sum squares of vertical distances to get slope m and intercept b



Fit data with a parabola.



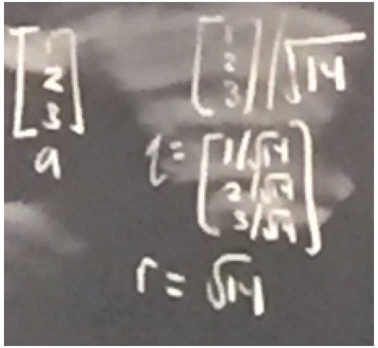
Given $(x_1,y_1),\ldots,(x_k,y_k)$, find coefficients a,b,c which minimizes $\Sigma_i\ (ax_i^2+bx_i+c-y_i)^2$

(May break down if any two x_i coincide)

When n=1 (A is just a single vector a)

a = qr is easy to obtain r = ||a||, q=a/r (the zero vector can be special cased)

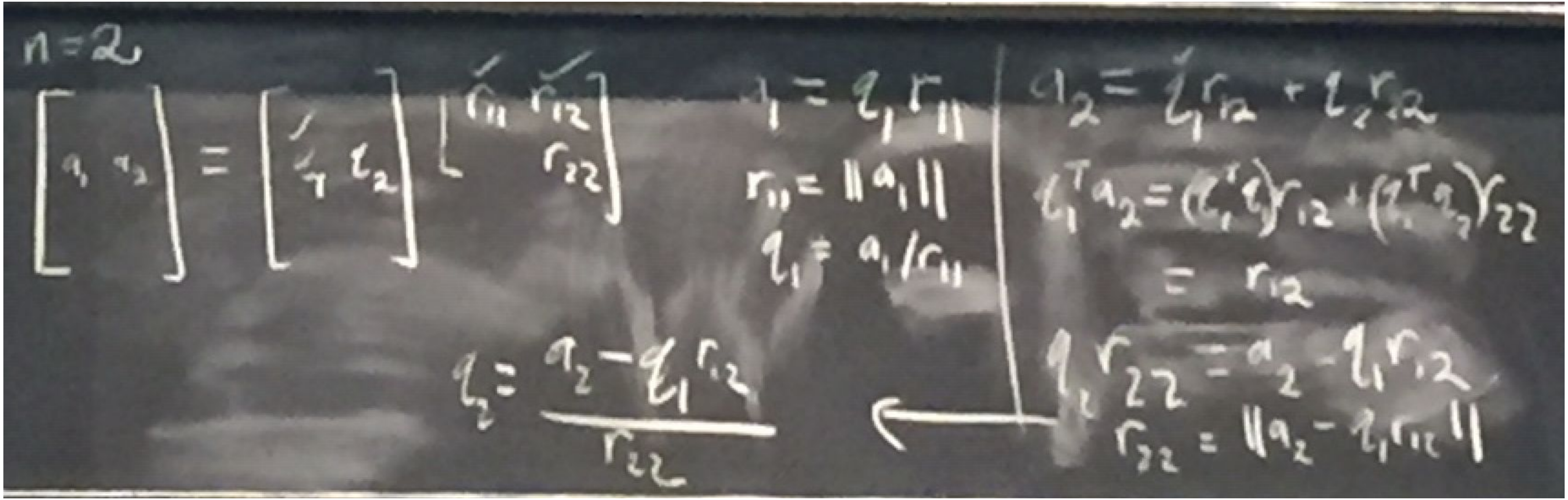
q is a unit vector in the a direction, r is the length



$1^2 + 2^2 + 3^2 = 14$

r = $\sqrt{14}$

When n=2 (A has two columns)



Summary:
$r_{11} = ||a_1||$, $q_1=a_1/r_{11}$ (Why? the left column reduces to the n=1 case on the previous slide)

$r_{12} = q_1^T a_2$, $r_{22}=||a_2-q_1 r_{12}||$, $q_2=(a_2-q_1 r_{12})/r_{22}$

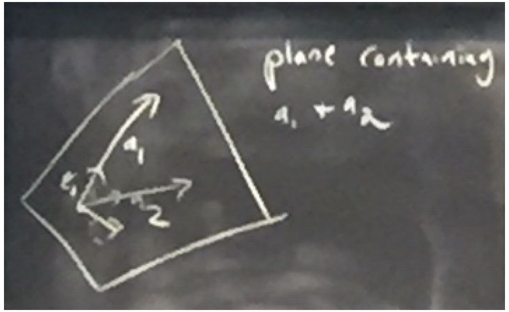
Comment: QR is essentially a classical algorithm known as Gram-Schmidt

Geometrical Intuition

Generically two vectors a_1 and a_2 determine a plane in a high dimensional space.

q_1 is the unit vector in the direction of a_1

q_2 is the perpendicular vector to q_1 in that plane

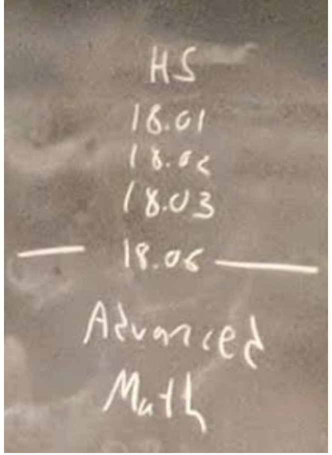


This picture continues a_1,a_2, and a_3 determine a 3d space in n dimensions.

q_3 is the vector perpendicular to q_1 and q_2 in this “3-space”

Mathematicians have no trouble with high dimensional spaces.

Lecture 6



18.06 may not be like previous classes you may have taken. It may be the first advanced math class that you take, concentrating on structure and insight, and just a very little hand computation.

The SVD reveals the deep structures hidden inside a matrix.

The SVD is almost impossible to compute by hand except for a few very special cases where the SVD is almost obvious.

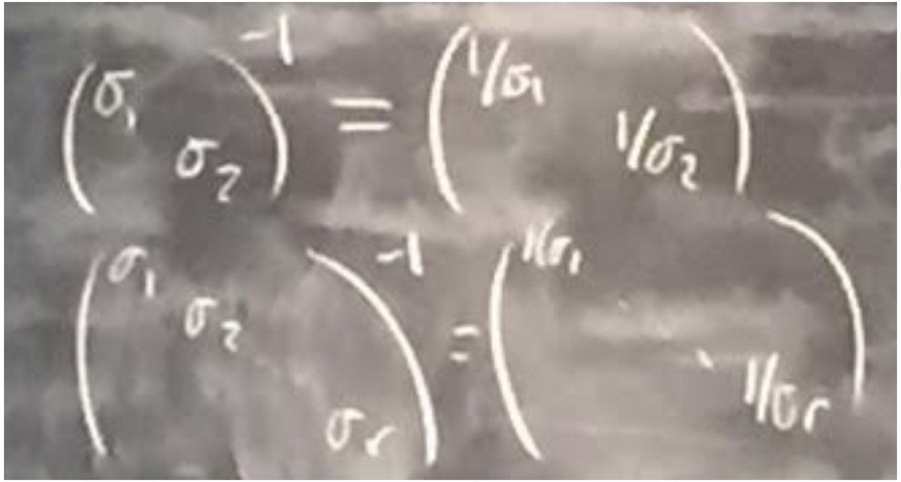
We are not assuming you know what the words eigenvalue or eigenvector mean -- and we are not defining the svd as in strang's book or by what you might find on wikipedia.

The SVD is always written $A=U\Sigma V^T$ but you may see different formats:

		Is Σ generally square? Invertible?	Are U and V necessarily square?	On a computer?
smallest	Rank r format	yes/yes	no	rarely
medium	Partial format	yes/no	The smaller one is	often
largest	Full Format	no/NA	yes	sometimes

There is generally the same information content in all the formats, and we do not advise overly worrying or memorizing the different formats. Just understand them.

Rank r format



A is m x n

rxr Σ is always invertible

The σ_i are called singular values

$A=U\Sigma V^T$,
where

U is mxr tall skinny orthogonal ($U^TU=I_r$)

V is nxr tall skinny orthogonal ($V^TV=I_r$)

Σ = diagonal($\sigma_1,\sigma_2,\ldots,\sigma_r$) where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$

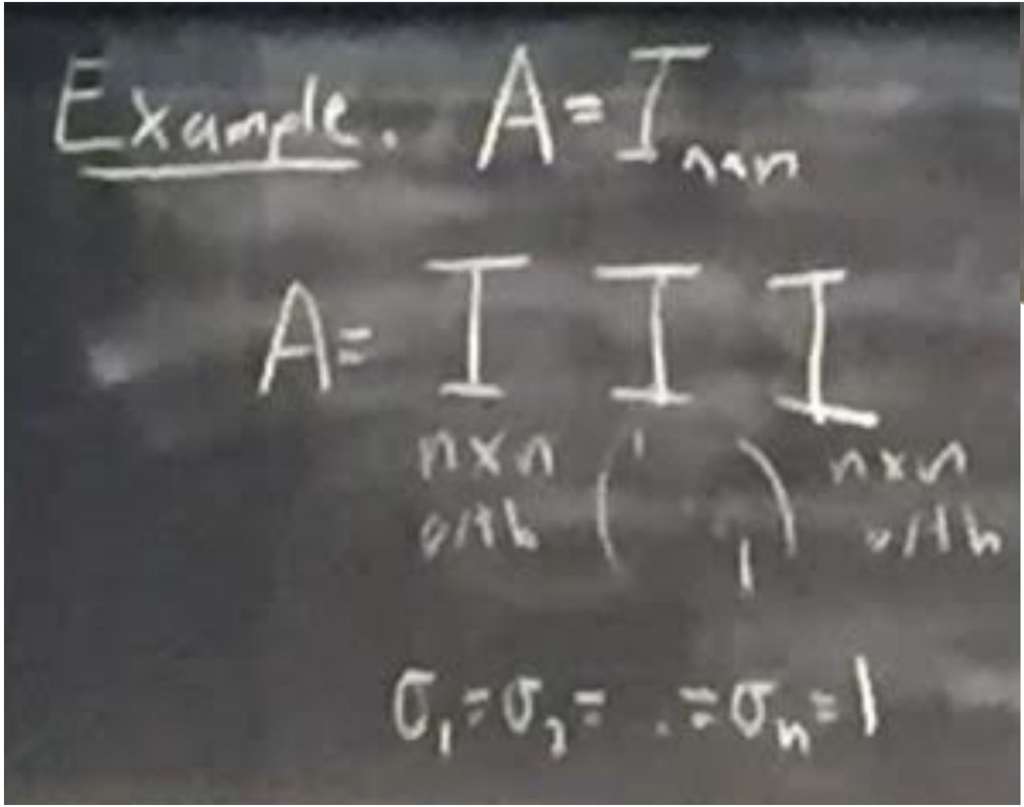
Columns of U&V are called singular vectors

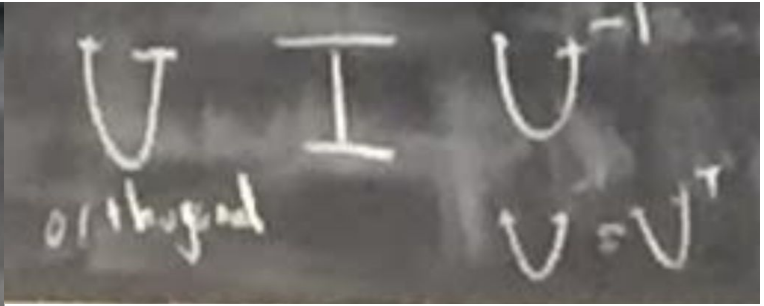
Uniqueness: Given an A there are multiple SVDs Possible, but:

The singular values are uniquely determined by A. If the singular values are distinct, the singular vectors are unique up to sign. If not, there is freedom in the singular vectors.

For example:if $A = U\Sigma V^T$ then $A = (-U)\Sigma(-V)^T$.

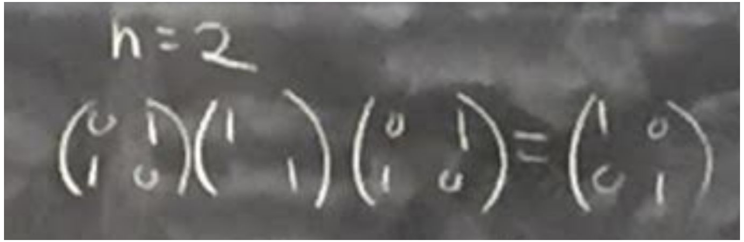
Example of an svd that is easy:





Notice if U is orthogonal ($U^T=U^{-1}$),
Then $I = U\ I\ U^T$ is an SVD.

Simple example when n=2



The rank

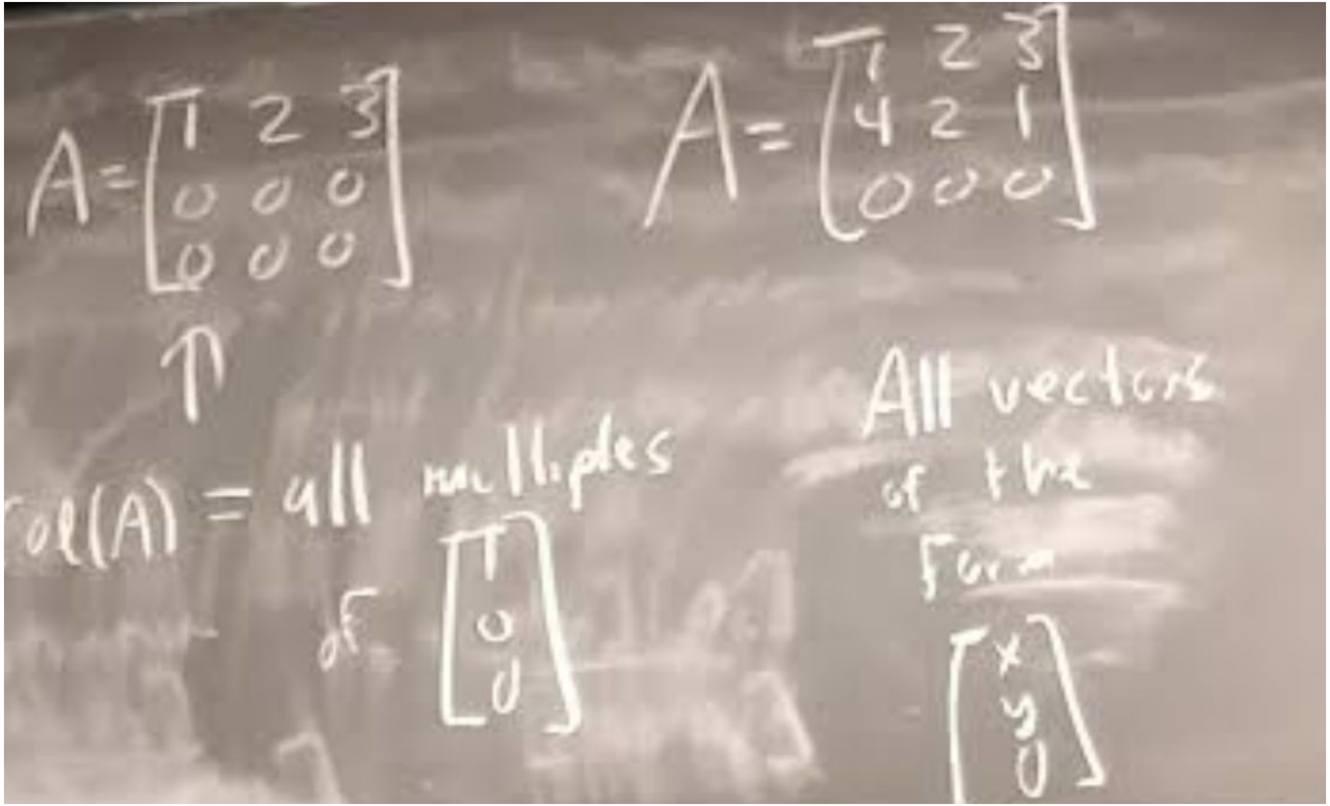
Def: The **rank** of A is r, the number of positive singular values in the svd.

The column space of A

Def: The **column space** of A consists of all linear combinations of columns of A. For every A it is a vector space. Equivalently

$\mathrm{col}(A) = \{Ax: x \in R^n\}$

Example column spaces that are easy to see



The Singular Value Decomposition in this class

- The SVD will show up throughout this semester. Students in the past who tried to google the SVD, hoping it can be mastered with reading one paragraph on wikipedia, found themselves unnecessarily panicking. The fact is we will bring out the properties of the svd in small measured steps in an order that makes sense to us.
- While there will be almost no hand computations of the svd, you will learn to recognize svd’s when they are easy to see.

Lecture 7.

Applying the SVD to image Compression

See the Julia Notebook

The SVD as the sum of r rank 1 matrices:

$$A = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T$$

Rank k approximation (take first k summands above)

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T$$

Four fundamental vector spaces for any matrix A

col(A) = The Column Space of A = $\{Ax : x \in R^n\}$

null(A) = The Nullspace of A = $\{x \in R^n : Ax = 0\}$

row(A) = The Row Space of A = $\{A^T x : x \in R^m\}$

null(A^T) = The Left Nullspace of A = $\{x \in R^m : A^T x = 0\}$

Lecture 8

If $A = U\Sigma V^T$ is the rank-r svd, We will prove that col(A) = col(U) always.

First we observe that if $A = U\Sigma V^T$ then $U=AV\Sigma^{-1}$.

Proof strategy: We will need to show

1. If y=Ax for some x, then y=Uw for some w. Conversely
2. If y=Uw for some w, then y=Ax for some x.

Note: Proving only 1 tells us that col(A) $\subseteq$ col(U), but not that col(A)=col(U).

Proof that col(A) = col(U). (U from the rank-r svd)

1. Suppose that $y=Ax$. Then $y = U\Sigma V^T x = Uw$, where $w=\Sigma V^T x$.
2. Suppose that $y=Uw$. Then $y = AV\Sigma^{-1}w = Ax$, where $x=V\Sigma^{-1}w$ since $U=AV\Sigma^{-1}$.

End of Proof

We note that the rank-r svd of A^T is always $V\Sigma U^T$.
Applying the theorem to A^T immediately tells us that
$\mathrm{row}(A) = \mathrm{col}(A^T) = \mathrm{col}(V)$.

Corollary: $Ax=b$ is solvable exactly when $UU^Tb=b$

Note: Next week we will find out that UU^T is a "projection matrix", specifically the projection matrix that projects onto the column space of A. Thus one way to understand this corollary is that $Ax=b$ is solvable when the projection of b onto the column space is b itself.

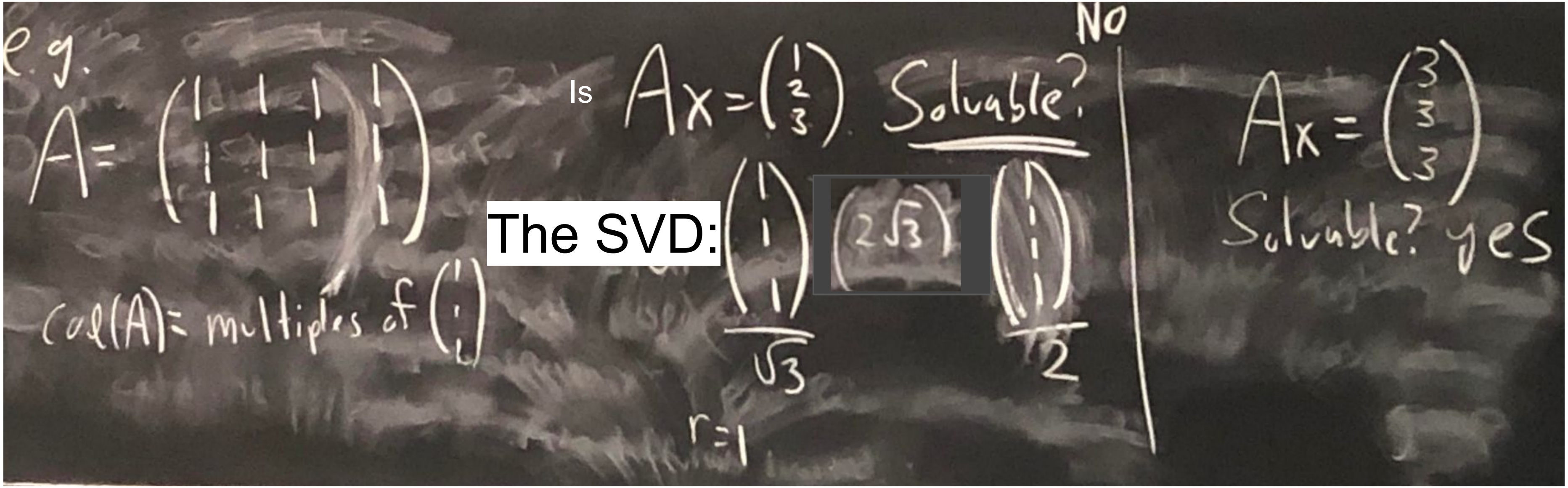
Proof of corollary. As before we will do this in two parts:

1. Suppose $Ax=b$ is solvable, then $b=UU^Tb$
2. Suppose $b=UU^Tb$, then $Ax=b$ is solvable.

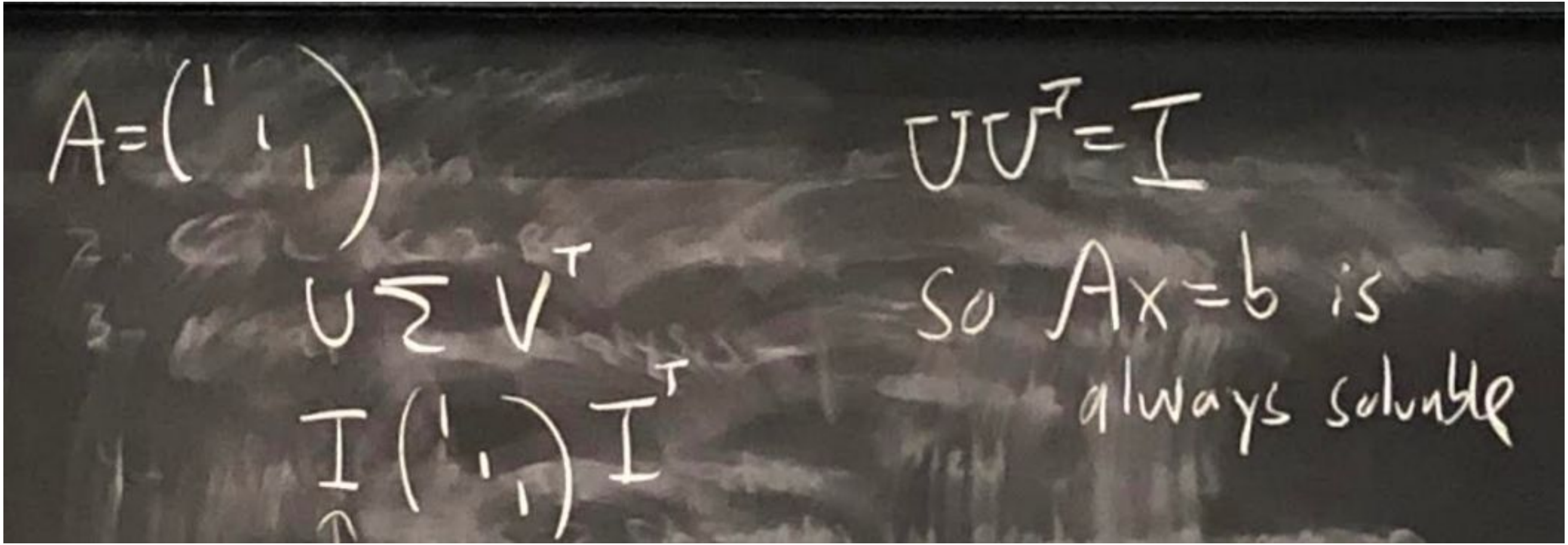
Proof of 1. If $b=Ax=U\Sigma V^Tx$, then $UU^Tb = U(U^TU)\Sigma V^Tx=U\Sigma V^Tx=b$

Proof of 2. If $b=UU^Tb$, then if $x = V\Sigma^{-1}U^Tb$, it follows that $Ax=U\Sigma V^TV\Sigma^{-1}U^Tb=UU^Tb=b$.

Example: (perhaps too easy a case)

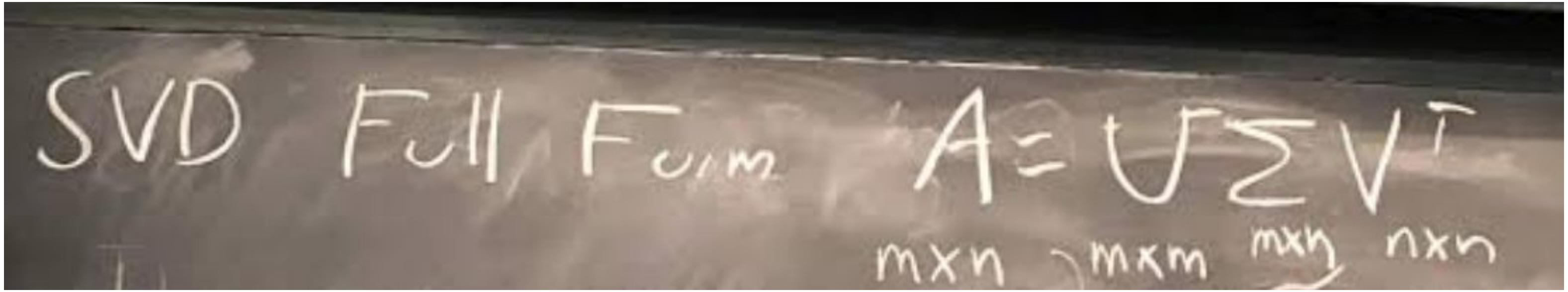


Another easy case



Generally, if A is square invertible, Ax=b is always solvable, so $UU^T b=b$ for all b, making UU^T the identity hence U is square orthogonal.

The Full Form of the SVD



$$\begin{bmatrix} U_1 \mid U_2 \end{bmatrix} \left[\begin{array}{c|c} \Sigma_r & 0 \\ \hline 0 & 0 \end{array}\right] \begin{bmatrix} V_1 \mid V_2 \end{bmatrix}^T = U_1\Sigma_r V_1^T + U_2 0 V_2^T = U_1\Sigma_r V_1^T$$

Notational note: All of the various forms tend to use the same letters U, Σ and V^T and which form is being used must be discerned by context.
On this slide, we are representing the pieces from the rank r svd as U_1, Σ_r, and V_1
The full form completes the rank r form by making U and V square orthogonal, and padding Σ with zeros.

The four fundamental spaces are in the SVD

$\text{col}(A) = \text{col}(U_1)$

$\text{row}(A) = \text{col}(V_1)$

$\text{null}(A) = \text{col}(V_2)$

$\text{null}(A^T) = \text{col}(U_2)$

Proof that null(A) = col(V_2)

Class note: we didn't make the proof this far in lecture, so this will not be on the exam, but the statement may be on the exam

Friday.

Since $V=[V_1|V_2]$ is square orthogonal $VV^T=I=V_1V_1^T + V_2V_2^T$

1. Assume that $x=V_2w$ for some w. Must show $Ax=0$.
2. Assume that $Ax=0$ for some x. Must show $x=V_2w$ for some w.

Proof of 1. On next slide.

Proof of 2. If $U_1\Sigma_rV_1^Tx=0$ then multiplying by $\Sigma_r^{-1} U_1^T$ gives $V_1^Tx=0$.

Since $I=V_1V_1^T + V_2V_2^T$, we have that

$$x = (V_1V_1^T + V_2V_2^T)\, x = V_1(V_1^Tx) + V_2V_2^Tx = V_2(V_2^Tx). \text{ Let } w = V_2^Tx.$$

Proof of 1 from the previous slide

1. Assume that $x=V_2w$ for some w. Must show $Ax=0$.

V is orthgonal so $[V_1|V_2]^T[V_1|V_2] = \begin{pmatrix} I_r & 0 \\ 0 & I_{n-r} \end{pmatrix}$

In particular $[V_1|V_2]^T V_2 = [0|I_{n-r}]^T$

So $$\left[\begin{array}{c|c} \Sigma_r & 0 \\ \hline 0 & 0 \end{array}\right] [\ V_1 \mid V_2\ ]^T\ V_2 = [\Sigma_r * 0 + 0 * I_{n-r}|0]^T = 0$$

So $AV_2x=0$

10. Projection Matrices

Quick summary

If A is a matrix with linearly independent columns (see next lecture), then

$P = A(A^TA)^{-1}A^T$

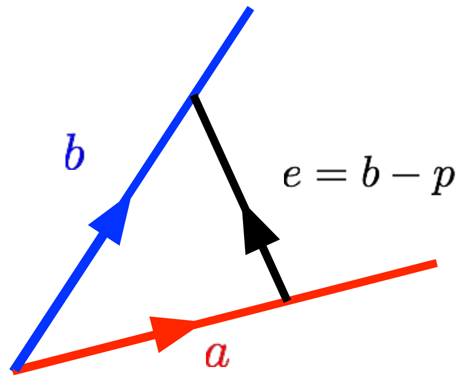
Projects onto the column space of A. Note that P is symmetric and $P^2=P$.

If A=QR, then we can write the simpler form, $P=QQ^T$.

This is often preferred in practice.

Thus if U is the U of the rank-r svd, UU^Tb projects U onto the column space of A.

Projection of a vector onto a vector



Projection of b onto a is a vector p

The projection p is parallel to a:

$$\boxed{p = xa, \quad x \in \mathbb{R}}$$

The error $e = b\text{-}p$ is perpendicular to a:

$$a^T(b - xa) = 0$$

$$\boxed{\implies x = \frac{a^T b}{a^T a}}$$

We can write the projection using a *projection matrix* : P

$$\boxed{\begin{aligned} p &= Pb \\ P &= \frac{aa^T}{a^T a} \end{aligned}}$$

This projection matrix is rank 1. Its column space is spanned by the vector a. It is symmetric and idempotent:

$$\boxed{\begin{aligned} P^T &= P \\ P^2 &= P \end{aligned}}$$

Projection of a vector onto a set of vectors

Take a set of independent vectors $a_1, a_2, ..., a_n \in \mathbb{R}^m$

Want to find the linear combination of the a_i that is the best approximation to $b \in \mathbb{R}^m$

$$p = \hat{x}_1 a_1 + \hat{x}_2 a_2 + ... + \hat{x}_n a_n = A\hat{x}$$

$$A = (a_1|a_2|...|a_n)$$

The error $e = b - p$ should be orthogonal to each of the a_i

$$\implies a_i^T (b - A\hat{x}) = 0$$ for each of the a_i

$$\implies A^T (b - A\hat{x}) = 0$$

$$\boxed{\implies A^T A\hat{x} = A^T b}$$

these are the "normal equations"

The projection is then $p = A\hat{x}$

Projection of a vector onto a set of vectors

We again have a projection matrix

$$p = Pb$$

$$\boxed{P = A(A^TA)^{-1}A^T}$$

This is a rank-*n* matrix. It is also symmetric and idempotent. We can check this, e.g.

$$\begin{aligned} P^T &= \left(A(A^TA)^{-1}A^T\right)^T \\ &= A\left((A^TA)^{-1}\right)^T A^T \\ &= A(A^TA)^{-1}A^T \\ &= P \end{aligned}$$

$$\begin{aligned} P^2 &= \left(A(A^TA)^{-1}A^T\right)\left(A(A^TA)^{-1}A^T\right) \\ &= A(A^TA)^{-1}(A^TA)(A^TA)^{-1}A^T \\ &= A(A^TA)^{-1}A^T \\ &= P \end{aligned}$$

Suppose *A* has a *QR* decomposition, where *R* is invertible . Then *P* simplifies:

$$\begin{aligned} & P = QR((QR)^TQR)^{-1}(QR)^T \\ \implies & P = QR(R^TQ^TQR)^{-1}R^TQ \\ \implies & P = QRR^{-1}(R^T)^{-1}R^TQ \\ \implies & P = QQ^T \end{aligned}$$

12. Linear Independence, Span, Basis, Dimension

Review and closer look at square matrices.

Square matrices A may be invertible or singular.

Invertible:

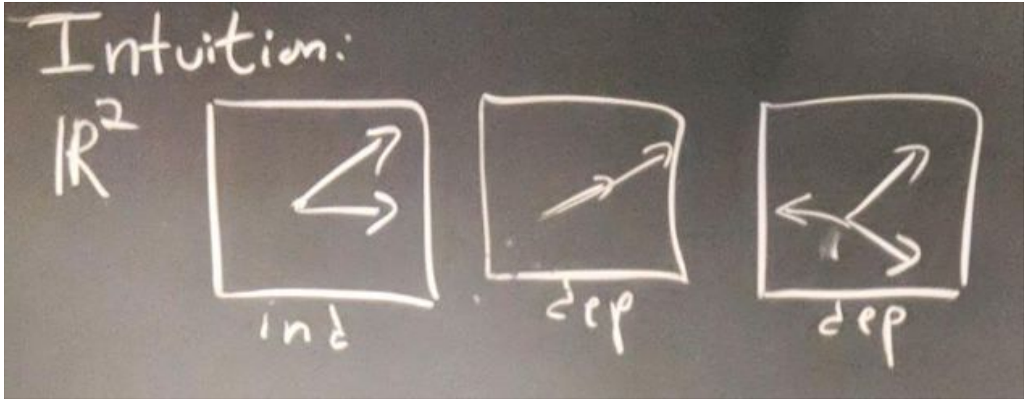
- $Ax=b$ always has a solution, $x=A^{-1}b$ (we will say cols of A span R^n)
- $Ax=0$ only when $x=0$ (we will say cols of A are linearly independent)
- Compact SVD & Full SVD are the same
- Rank $r = n$
- Null space = the zero vector
- Column space = R^n

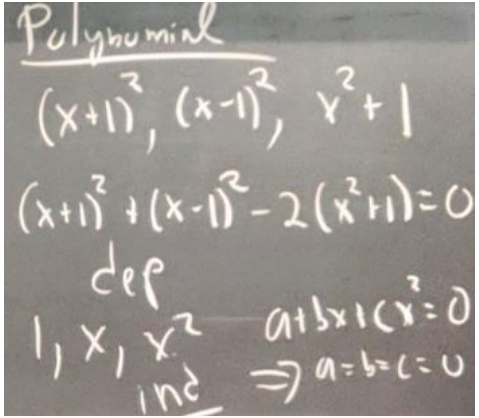
Singular square matrices

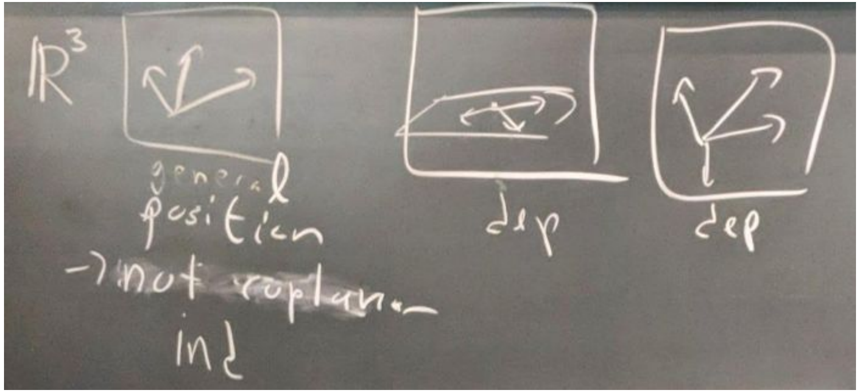
- $Ax=b$ does not always have a solution, only if $b=UU^Tb$, and $UU^T \neq I$, where U is from the compact SVD
- $Ax=0$ has a non-trivial solution, any linear combination of the columns of V_2 from the full SVD
- Compact and full SVD are not the same
- Rank $r < n$
- Null Space non-trivial (we will say the columns are linearly dependent)
- Column space not all of R^n (we will say the columns do not span R^n)

Reminder: singular is synonymous with non-invertible
Non-singular is synonymous with invertible

Pictures of Linear Ind/Dependence to start the story







More than n vectors in R^n will always be dependent.

Definition valid in any vector space (not just column vectors)

Given vectors $a_1, a_2, \dots, a_n$ in a vector space, we say that $a_1, \dots, a_n$ are <u>independent</u> if

$$x_1 a_1 + x_2 a_2 + \dots + x_n a_n = 0 \text{ only when } x_1 = x_2 = \dots = x_n = 0.$$

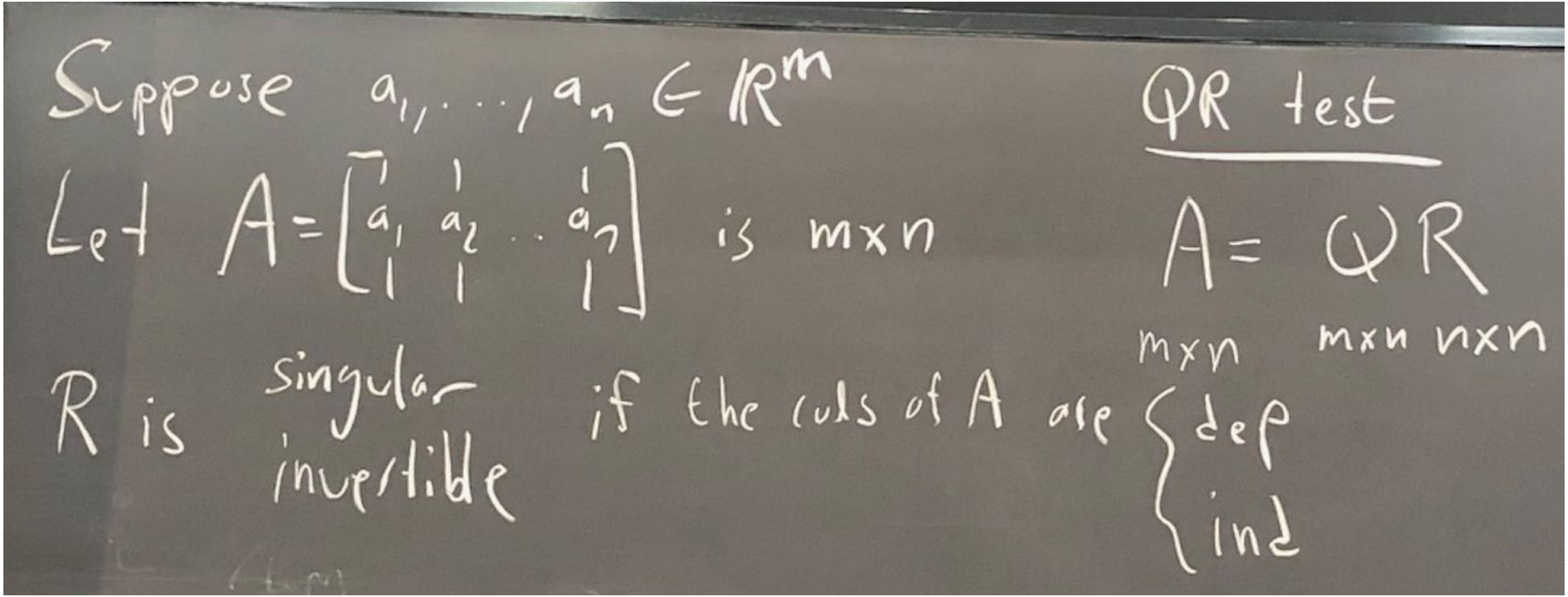
Otherwise, we say $a_1, \dots, a_n$ are dependent.

<u>Examples</u>

$$n=1 \quad a_1 \text{ is } \begin{Bmatrix} \text{ind} \\ \text{dep} \end{Bmatrix} \text{ if } \begin{matrix} a_1 \neq 0 \\ a_1 = 0 \end{matrix}$$

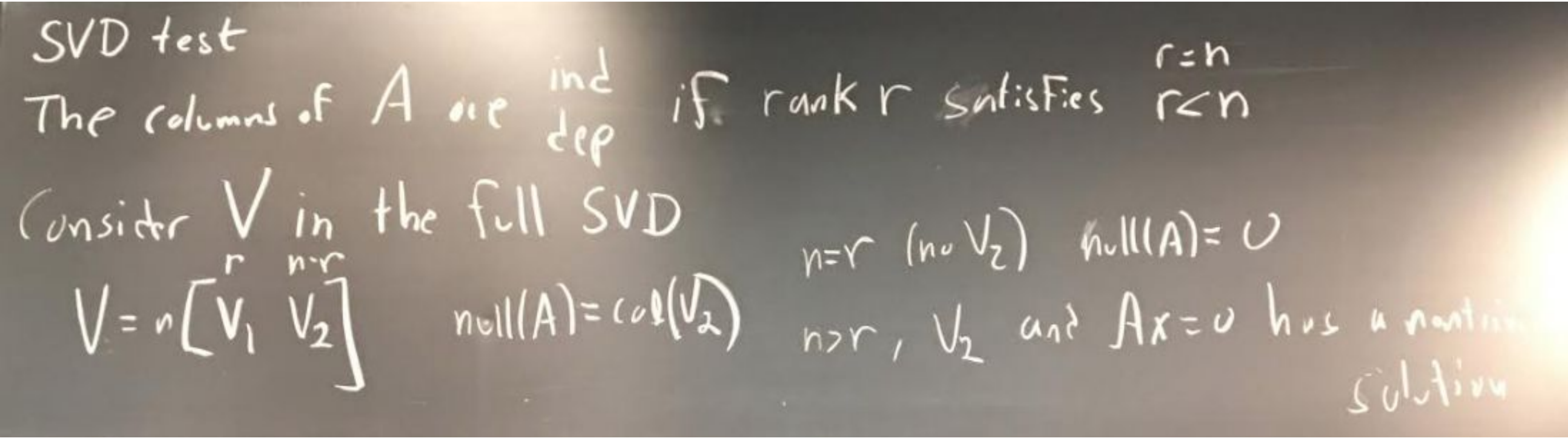
$$n=2 \quad a_1 + a_2 \text{ is } \begin{cases} \text{ind} \\ \text{dep} & \text{if one is a scalar multiple of the other} \end{cases}$$

We can say more if the vectors are column vectors



Proof. If R is singular, then we have a nontrivial x for which Rx=0 so QRx=Ax=0. If R is invertible, Ax=0 only When QRx=0 or $R^{-1}Q^TQRx=x=0$ meaning x can only be 0.

Svd test for linear independence



Aside, Do short wide orthogonal matrices exist? can U be mxn with n>m and $U^TU=I_n$

Answer: No

Suppose $U = [U_1 \mid U_2]$ with U_1 square (mxm) and U_2 (m x (n-m))

Clearly $U_1^TU_2 = 0$ so $U_1U_1^TU_2 = 0 =U_2$

since U_1 is square and $U_1U_1^T=I_m$.

$U_2 = 0$ contradicts that $U_2^TU_2=I_{n-m}$

So no short wide orthogonal matrices exist.

In general, when do matrices have a non-trivial nullspace?

Answer: When $n>r$.

It's possible for tall skinny, square, and short wide matrices to have a non-trivial nullspace. Short wide matrices always have a non-trivial nullspace.

In general, when do matrices have the trivial nullspace?

Answer: When $n=r$.

Short wide matrices can not have the trivial nullspace. Tall skinny or square might or might not.

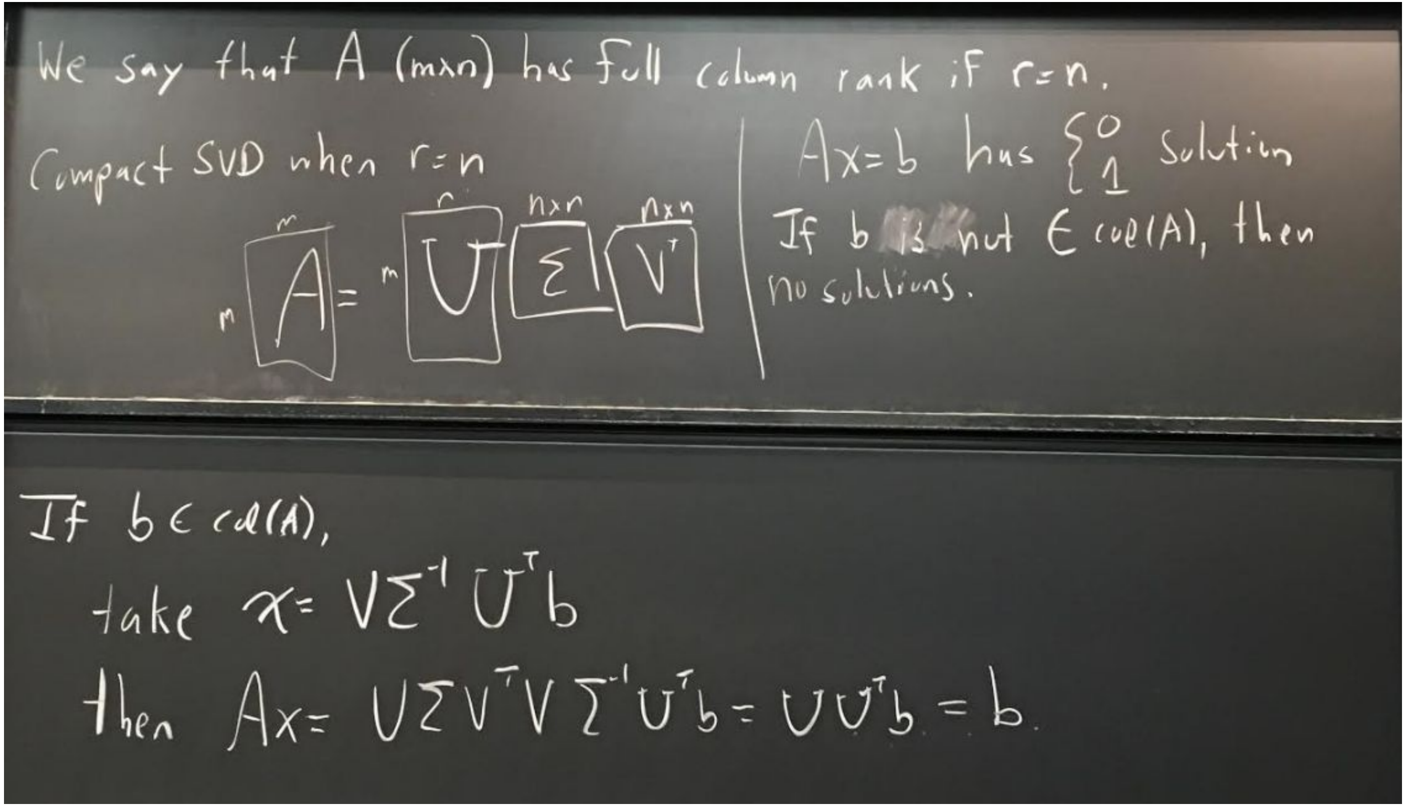
What are the possible ranks r for an m x n matrix?

Answer: r can be 0,1,2, …. up to min(m,n).

Proof: In the compact SVD, U and V may be square or tall skinny orthogonal, but never short wide. Since r is the number of columns in U it is at most m, and also it is the number of columns in V so it is at most n.

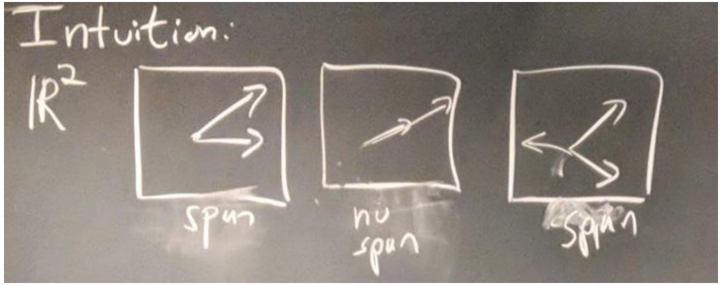
The columns are linearly independent exactly when the rank is n. We say that A has "full column rank." This can happen for a square matrix or a tall skinny matrix. But a short wide matrix can have rank at most $m < n$.

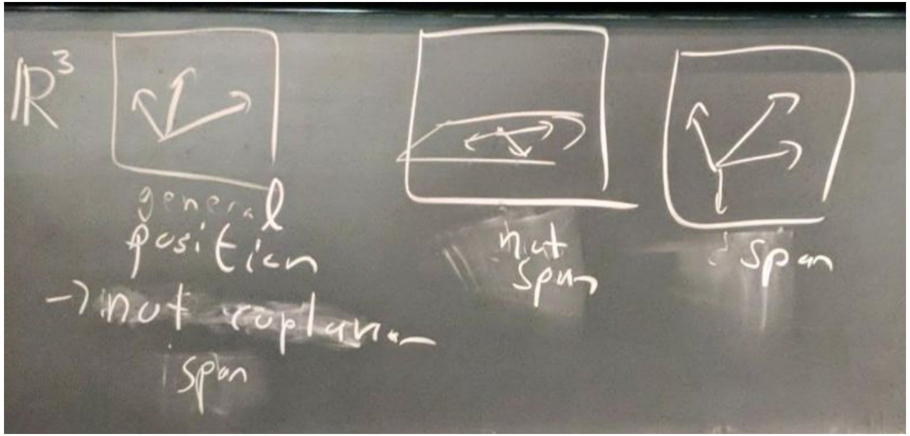
Summary of full column rank (A has ind columns)



Vectors that “span” a vector space

Intuitive pictures:

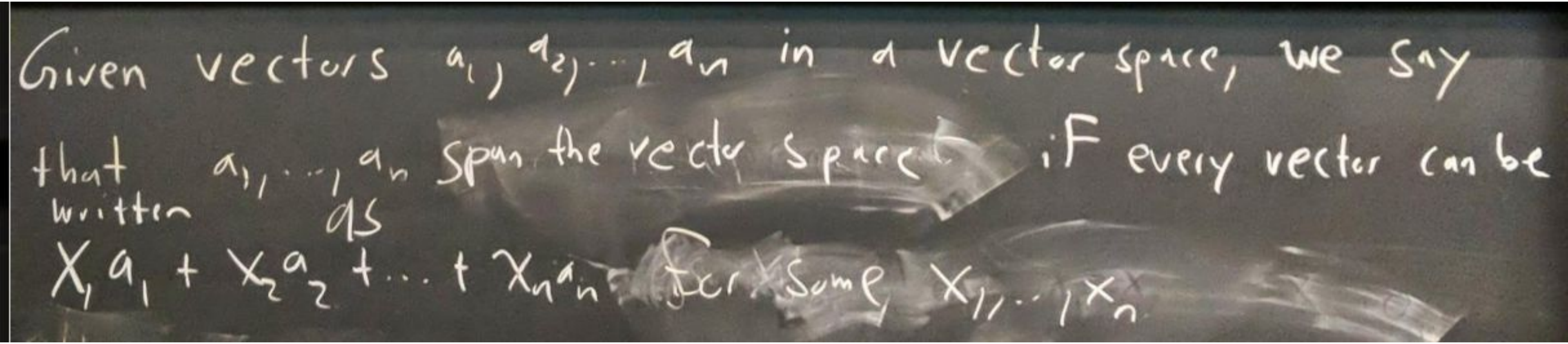




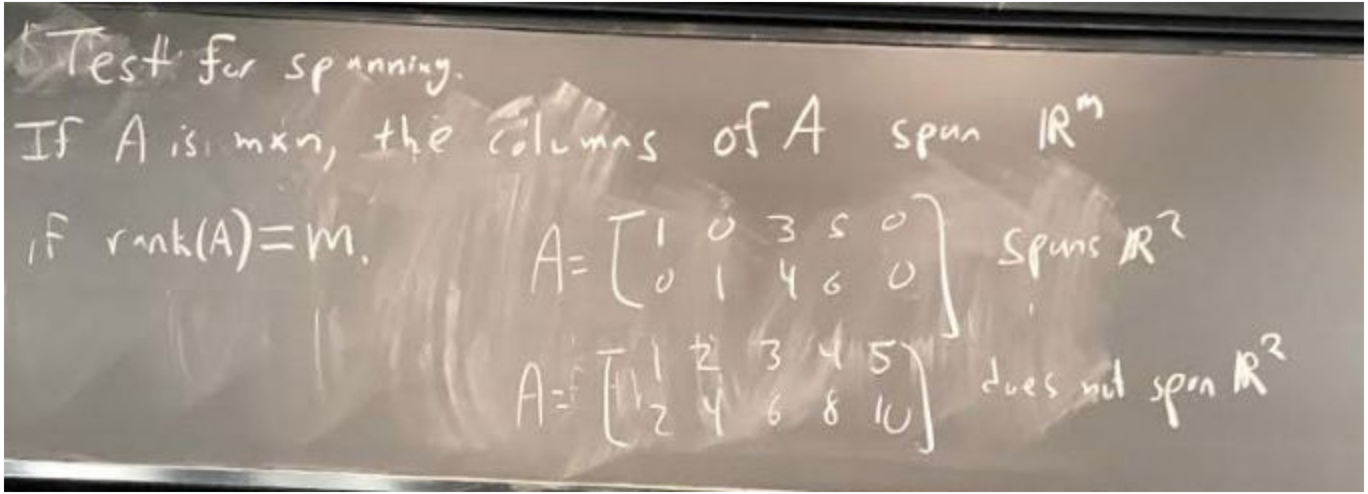
The polynomials
$(x+1)^2, (x-1)^2, x^2+1$ do not span the quadratics

The polynomials
$1, x, x^2$ do span the vector space of quadratics

Definition of span



Test for spanning



Note that only short wide matrices can have columns that span R^m

Ax=b always has a solution if the columns span R^m. There may be infinitely many solution when n>m or one solution when n=m.

Basis

If a set of vectors are linearly independent and span the space, we say that it is a basis.

If we have a matrix A, the columns are a basis for R^m exactly when m=n, and A is nonsingular.

Dimension

We will see in a moment that while a vector space can have many bases, every basis has exactly the same number of elements which we call the dimension of the vector space.

We already saw this for column vectors. Every basis for R^m can be represented as the columns of a square invertible mxm matrix, hence there are exactly m vectors in the basis.

Using matrices to connect vectors in vector spaces that are not column vectors:

$[1\ \ x+1\ \ (x+1)^2\ \ (x+1)^3] = [1\ \ x\ \ x^2\ \ x^3] * \text{(Pascal)}$

$$\text{Pascal (upper triangular)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

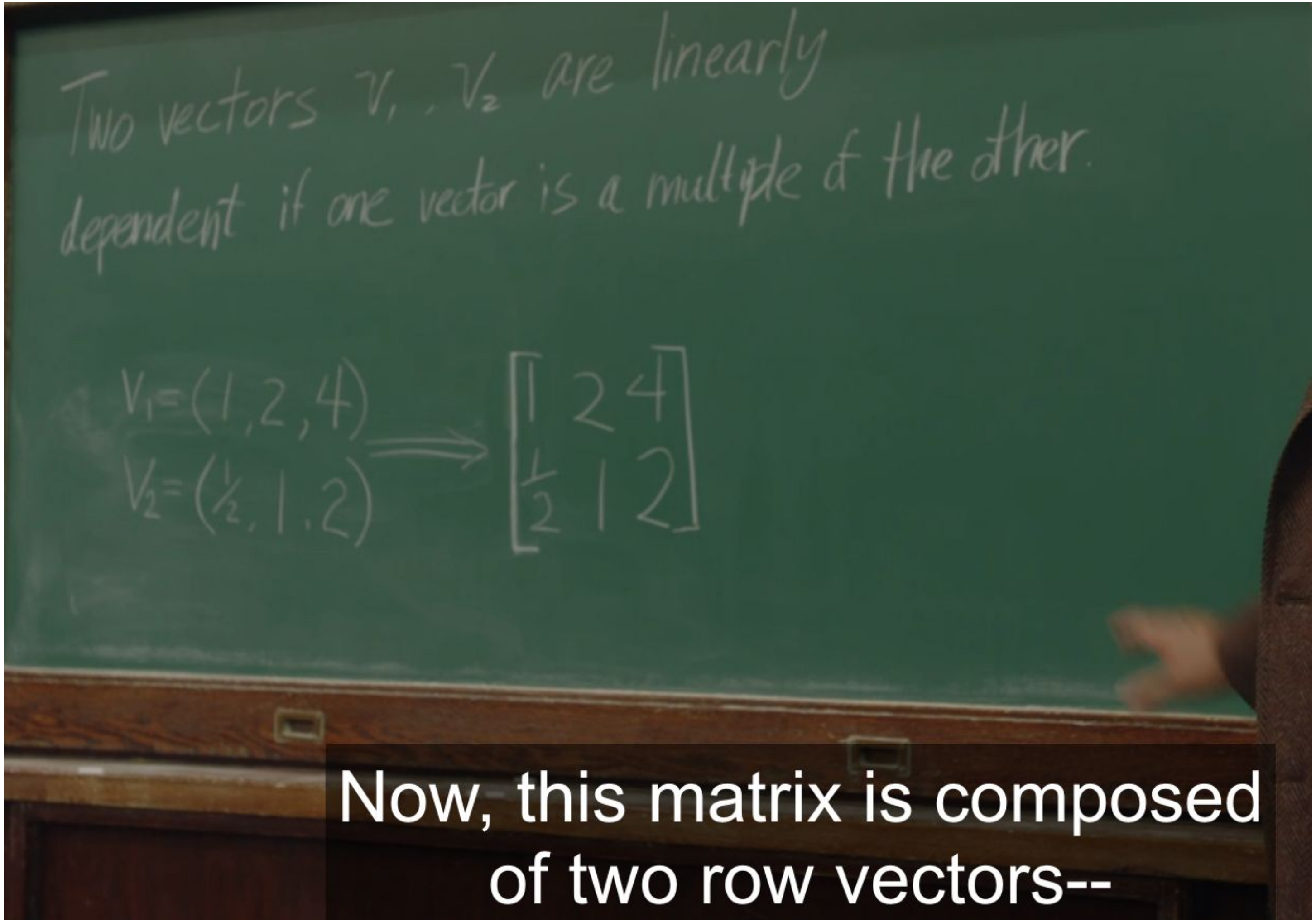
13. The Complete Solution to Ax=b

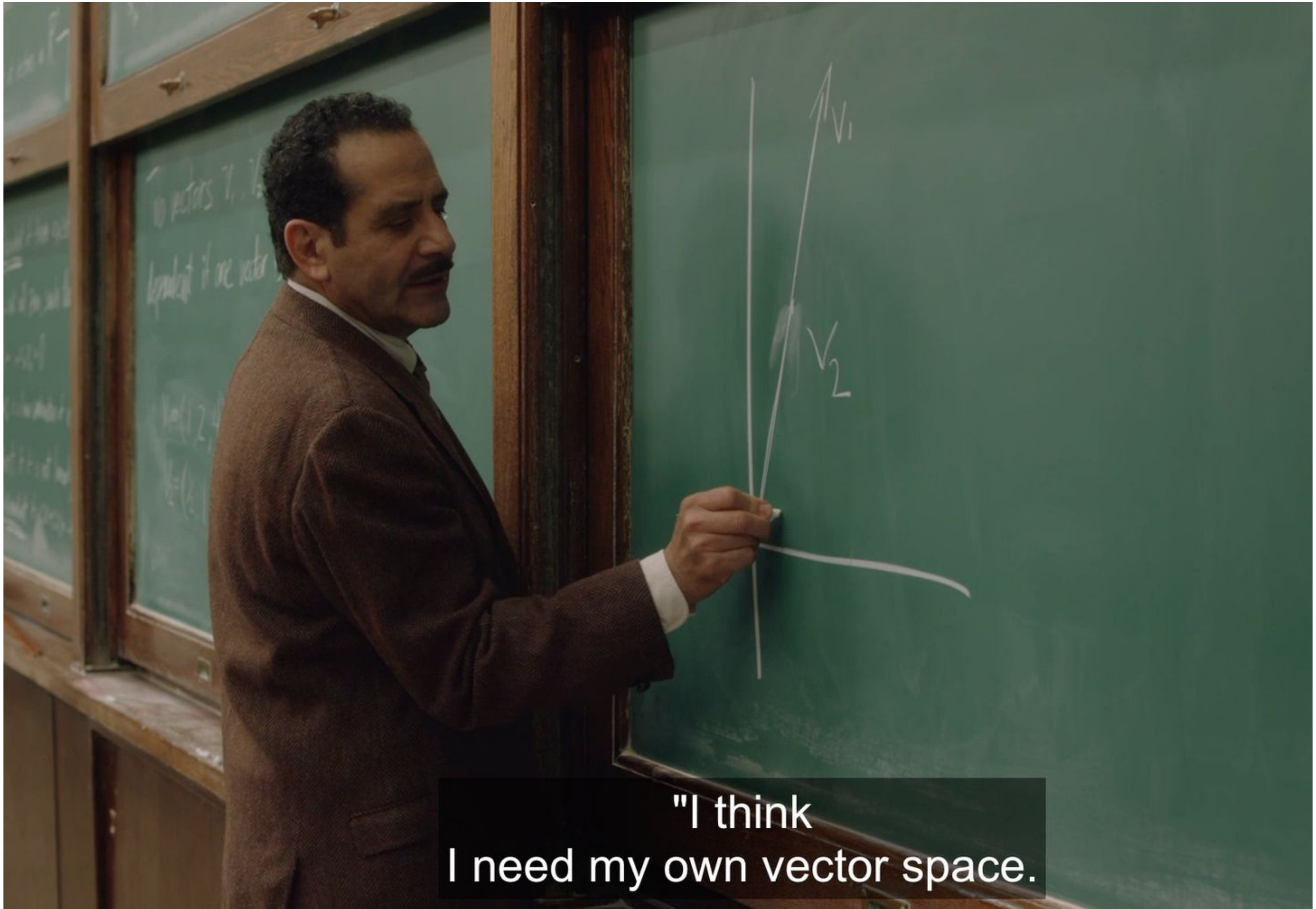
We began with a fun clip from the Marvelous Mrs. Maisel Season 1 Episode 2 13:35 to 16:19 (if you have amazon prime it's fun to watch over and over again).

Mrs. Maisel is an aspiring comedienne in the late 1950s whose father Professor Weisman is a fictitious mathematician at Columbia university. Mrs. Maisel and her husband are separating which is very upsetting to her father who brings his concerns, thinly disguised, into a mathematics lecture.

Weisman's lecture covers the linear dependence of two colinear vectors using the older language of echelon forms (rather than the svd) and the quaint term "nullity" which is the dimension of the nullspace. On the board the very definition of linear dependence that we use in our own class can be seen.

Weisman explains that vectors can't go off and be linearly independent just because they want to. "In this room v_2 is never going to break that vow and decide it doesn't need that vector anymore" and "tell v_1, I think I need my own vector space."


Two vectors v_1, v_2 are linearly
dependent if one vector is a multiple of the other.
$V_1 = (1, 2, 4)$
$V_2 = (\frac{1}{2}, 1, 2)$
$\begin{bmatrix} 1 & 2 & 4 \\ \frac{1}{2} & 1 & 2 \end{bmatrix}$
Now, this matrix is composed
of two row vectors--


V1
V2
"I think
I need my own vector space.

Data View of Rank

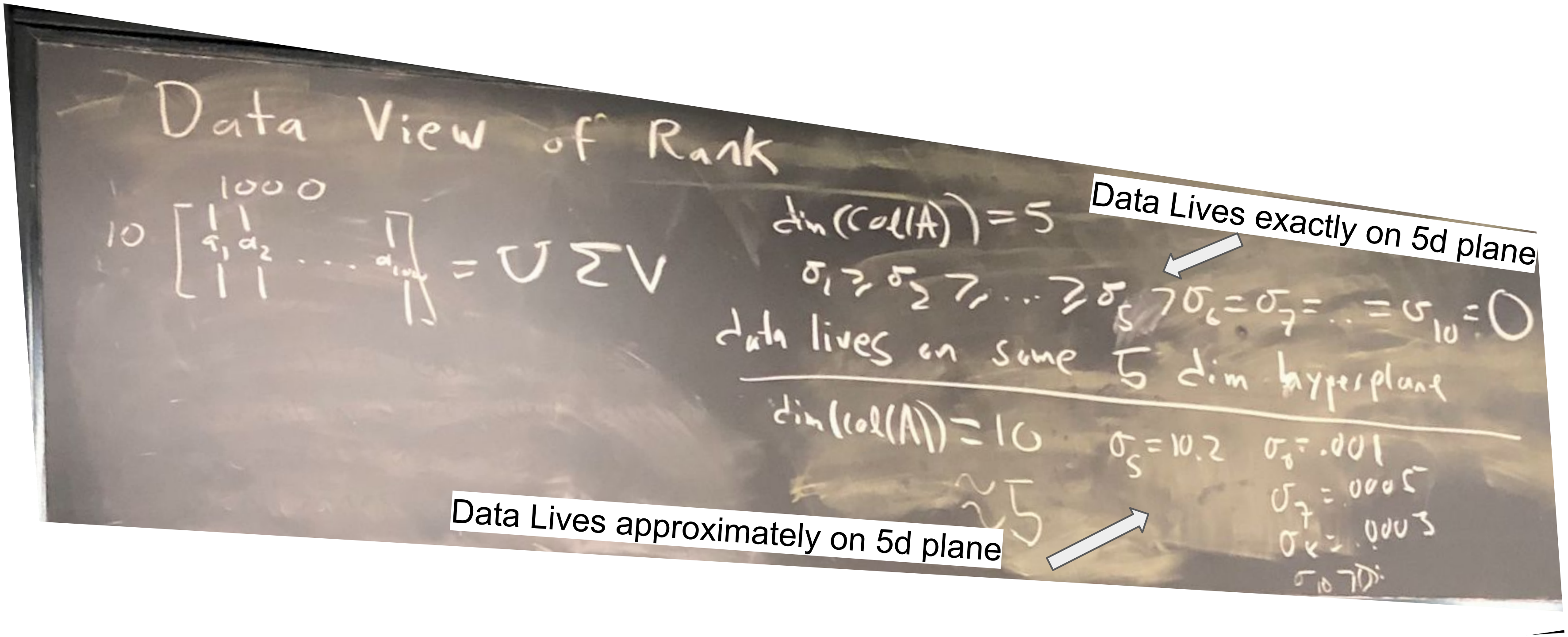
In many situations one might have something like 1000 vectors in R^{10}.

One has to choose in software whether to store the data in a 10x1000 matrix, or as a vector of vectors, i.e., one could store 1000 vectors in R^{10}.

In Julia the first case would be labelled 10×1000 Array{Float64,2} indicating a 10x1000 array whose elements are 64 bit floats, in a 2 dimensional structure.

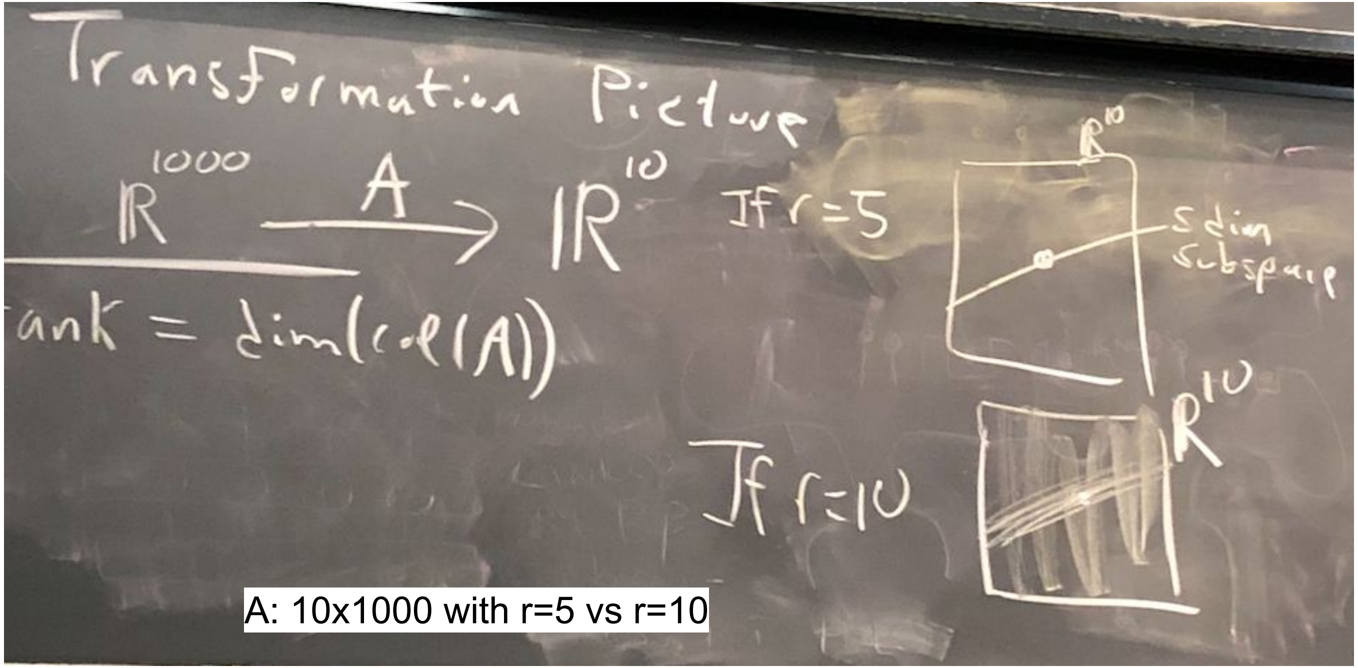
In the second case it would be labelled 1000-element Array{Array{Float64,1},1} indicating a 1000 element 1 dimensional array whose elements are 1 dimensional arrays.

→ Either way, if there are 10 non-zero singular values but, say, five are big compared to the other five, then the data is nearly in a 5d subspace, and this is useful!

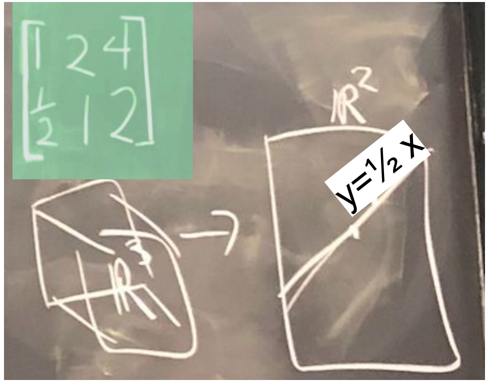

Data View of Rank
1000
$10 \left[a_1 \; a_2 \; \cdots \; a_{1000} \right] = U \Sigma V$
$\dim(\mathrm{col}(A)) = 5$
Data Lives exactly on 5d plane
$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_5 > \sigma_6 = \sigma_7 = \cdots = \sigma_{10} = 0$
data lives on some 5 dim hyperplane
$\dim(\mathrm{col}(A)) = 10$
≈ 5
$\sigma_5 = 10.2$ $\sigma_6 = .001$
$\sigma_7 = .0005$
$\sigma_8 = .0003$
$\sigma_{10} > 0$
Data Lives approximately on 5d plane

Transformation Picture of Rank

We imagine A being applied to all of R^n, and ask the geometry of the complete set of outputs which may be all of R^m, or a hyperplane within R^m. The geometry is always a hyperplane of dim = the rank.



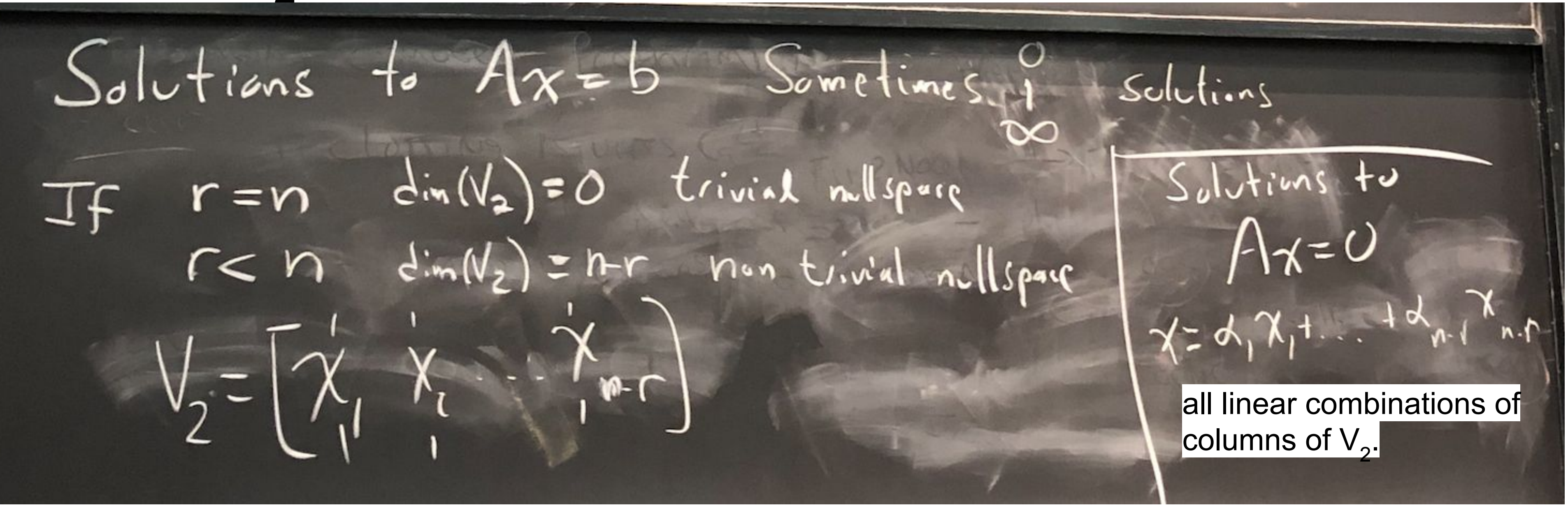
Marvelous Mrs. Maisel matrix
A: 2x3 with r=1:



If $\dim(V_2)=0$, we have a trivial nullspace $\Rightarrow$

$Ax=b$ has 0 or 1 solution.

If $\dim(V_2)\neq 0$, then $Ax=b$ has 0 or ∞ solutions:



all linear combinations of columns of V_2.

Solutions to Ax=b

If $b \notin \mathrm{col}(A)$ there are no solutions

If $b \in \mathrm{col}(A)$ let x_p be any solution at all (call this a "particular" solution)

Suppose $Ax_p=b$ and also $Ax_q=b$ for $x_q \neq x_p$. Then $A(x_q - x_p)=0$ so x_q-x_p is in the nullspace.

The complete solution to Ax=b is then

$x = x_p + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_{n-r} x_{n-r}$, where the α_i are arbitrary real numbers

Questions we might ask about the process of taking x into b=Ax

For the target space R^m. Consider the two alternatives:

1. Every $b \in R^m$ can be reached
2. Some $b \in R^m$ can not be reached

For the source space R^n. There are similarly two alternatives:

3. Two or more vectors can produce the same b

4. $Ax_1 \neq Ax_2$ if $x_1 \neq x_2$

Exploring the cases

Case 1: The following are equivalent conditions on A

- rank A = m
- Columns of A span R^m
- Ax=b always has at least one solution for all b (there may be infinite)

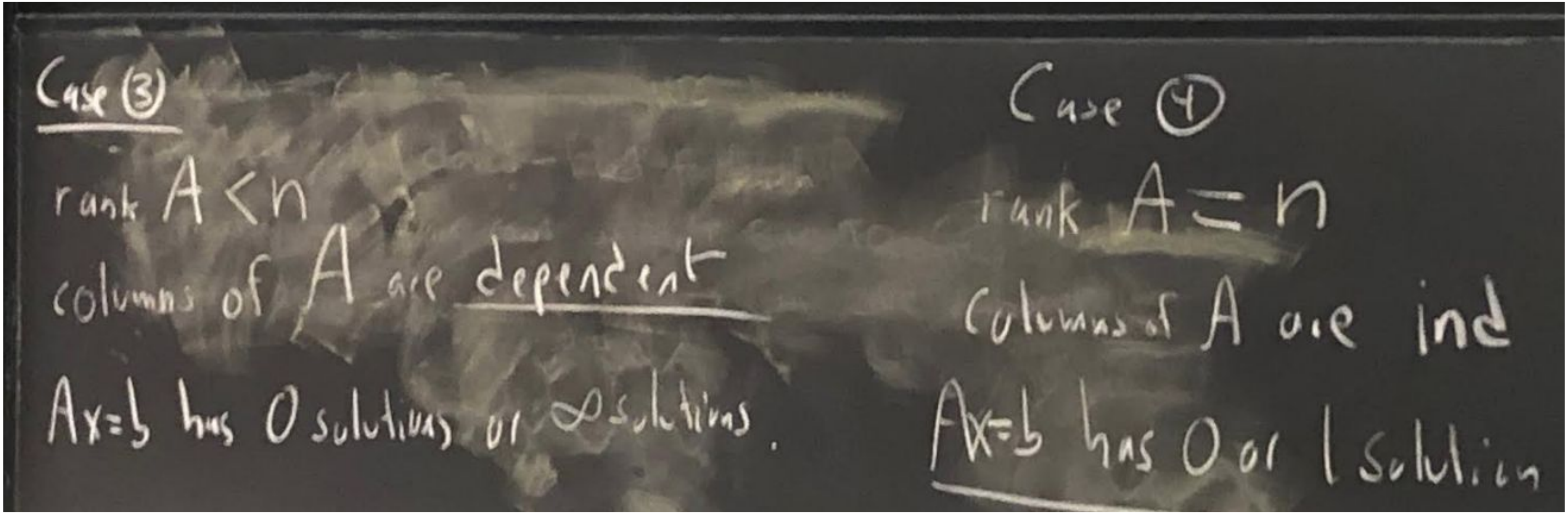
(Note the above case can only happen if $n \geq m$)

Case 2: rank A < m

The following are equivalent conditions and are the exact opposite of Case 1

- Rank A < m
- Columns of A do not span R^m
- Ax=b can have 0 solutions for some values of b

Cases 3 and 4 are also opposite:



For sure m ≥ n in case 4

Questions to test understanding

Come up with matrices that are

Case 1 & 3? (short&wide full row rank)

Case 1 & 4? (Square invertible)

Case 2 & 3? (not full row or full col rank)

Case 2 & 4? (tall&skinny full column rank)

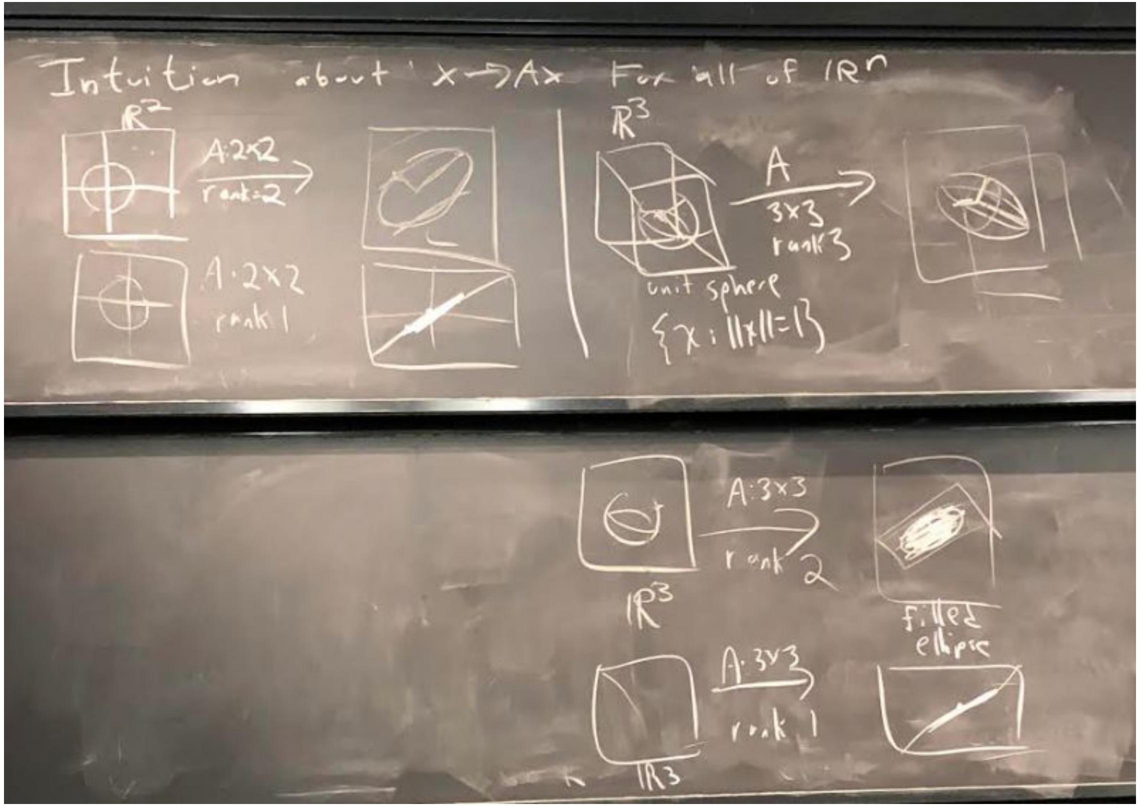
Curiosity -- the 0 matrix of any size has rank r=0

Kind of an empty or degenerate case:

U is mx0, Σ is 0x0, V is nx0

(no positive singular values)

14. Further Intuition of the Geometry of x→ Ax



Rank is the dimension of the image hyperplane.

We will prove later that the surface of the sphere {||x||=1} gets transformed into the surface of a full ellipsoid with semi-axes exactly the singular values. (Not eigenvalues generally, if you know what that is.)

The idea is that if you start with the unit sphere V^T does nothing, Σ scales along the coordinate axes and U rotates the ellipsoid into a non-standard position.

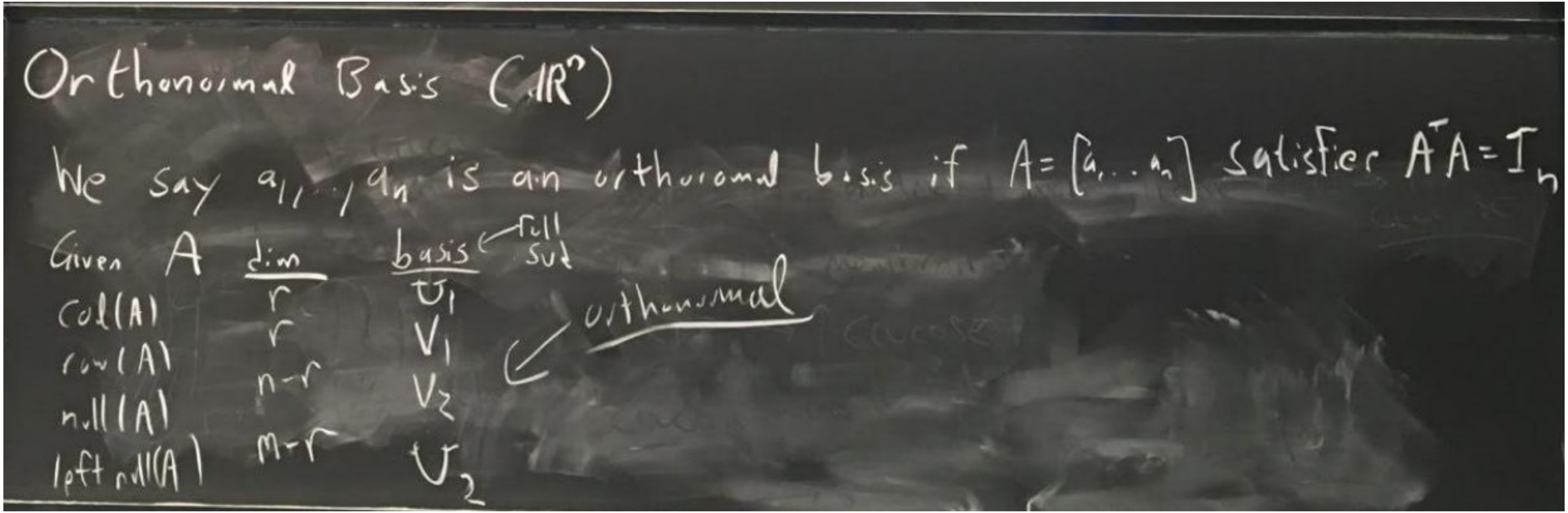
General vector spaces:

Notationally: you are allowed to put elements of any vector space in a row vector and combine them with one or more columns of real numbers:

$$\begin{bmatrix} 1 & x & x^2 & x^3 & x^4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = 1 + 2x + 3x^2 + 4x^3 + 5x^4$$

These are functions of x not numbers

Orthonormal Bases



The four fundamental spaces for a matrix A have orthonormal bases in the columns of U_1,U_2,V_1,V_2 from the full SVD. This is really useful in practice.

Formalizing Dimension

In every day speech, it is common to describe a line as one dimensional, a plane as two dimensional, and space as three dimensional. We are even familiar with the number line as coordinates in 1d, the x,y axes in 2d, and x,y,z in 3d. We recognize coordinates as useful for two reasons:

1) Every point has coordinates and
2) These Coordinates are unique

We now know in the language of linear algebra that as long as two vectors are not colinear in R^2, or three are not coplanar in R^3 we also have coordinates because 1) corresponds to spanning and 2) to linear independence.

But what's really cool is …. (next slide)

The number of vectors in a basis is constant for any vector space.

While there are infinitely many bases usually (any two noncolinear vectors in the plane, for example is a basis), the number of vectors in a basis depends only on the vector space. In other words, it's impossible to have a basis for R^3 which fewer than 3 vectors or more than 3 vectors. Similarly it's impossible to have a basis for the functions $a+bx+cx^2$ with more than three polynomials. (e.g. $1,x,x^2$ is a basis but so is 1, $2x+3$, $4+5x+6x^2$.)

As intuitive as this may seem, it's a real testament to the human intellectual spirit to be able to logically nail this down with mathematics.

Proof: Let $a_1,...,a_m$ and $b_1,...,b_n$ be bases for a vector space.

Since the b span the vector space, a_1 can be written

$a_1 = b_1 P_{11} + b_2 P_{21} + \ldots + b_n P_{n1}$ for some scalars $P_{11}, P_{21}, \ldots, P_{n1}$

(that's what it means to span)

In fact we can do this for every i and thereby get an n x m matrix P such that

$[a_1\ a_2 \ldots a_m] = [b_1\ b_2 \ldots b_n]\ P$

Switching the a's span the vector space, there is also an mxn W such that

$[b_1\ b_2 \ldots b_n] = [a_1\ a_2 \ldots a_m]\ W$

Proof that dimension depends on the vector space (cont.)

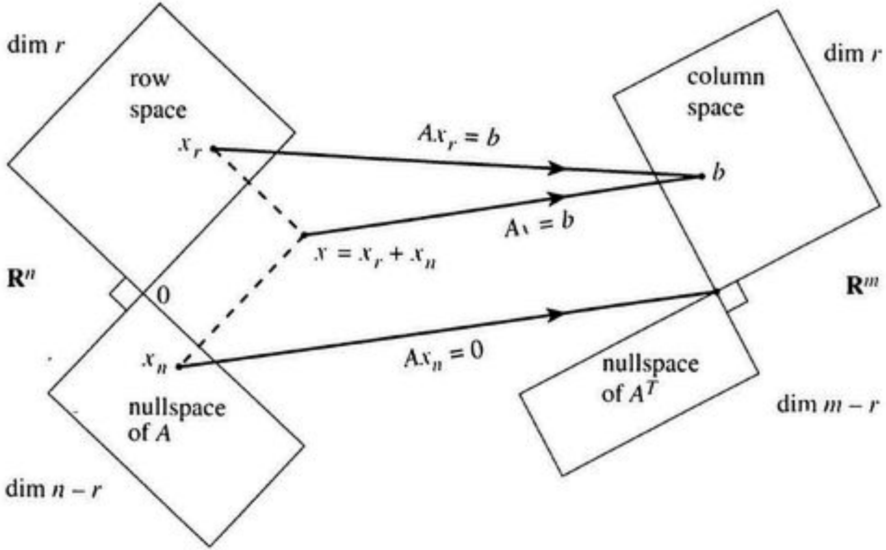
Now $[b_1\ b_2\ \ldots\ b_n]\ P\ W = [a_1\ a_2\ \ldots\ a_m]\ W = [b_1\ b_2\ \ldots\ b_n]$

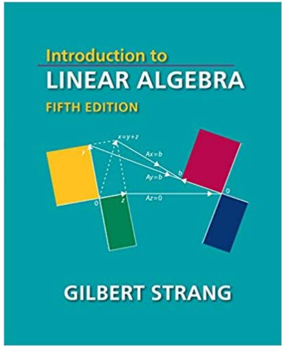
So we have PW=I is one possibility, but is it the only one? Well yes, because we also have the fact that we have not used yet, that the $[b_1\ b_2\ \ldots\ b_n]$ are linearly independent. Similarly WP=I as well.

Thus we know that P and W are square and invertible, so in particular m=n, and the number of elements in the “a” basis (m) is exactly the number in the “b” basis (n).

15. The Fundamental Subspace Picture

In the next two lectures we will try to understand the diagramatic picture that Strang has put on the cover of his book

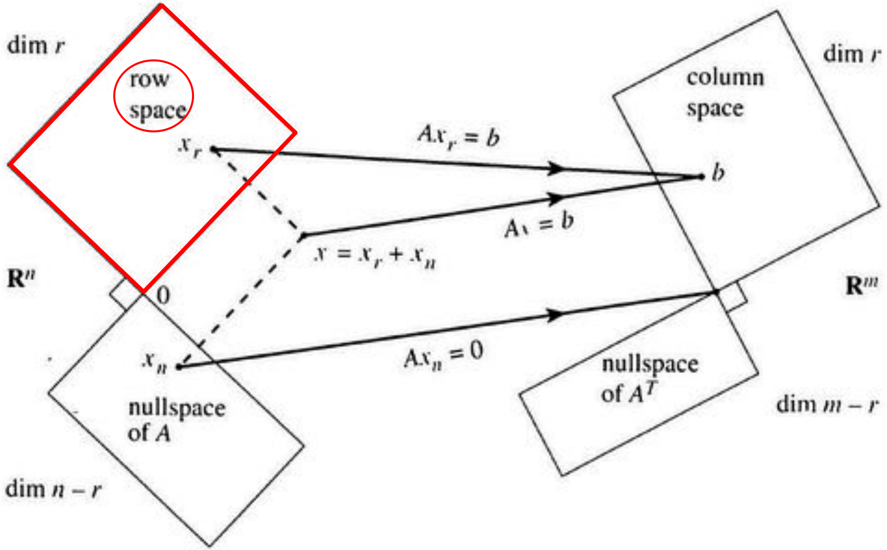




Some questions that come to mind are
1) what are all the boxes, arrows, and labels trying to denote?
2) It all seems complicated? Why do mathematicians find this whole story so fundamental?

Gil Strang's [paper](#) on the topic (Fundamental Theorem of Linear Algebra) is a good read!

The four spaces portrayed as rectangular blocks

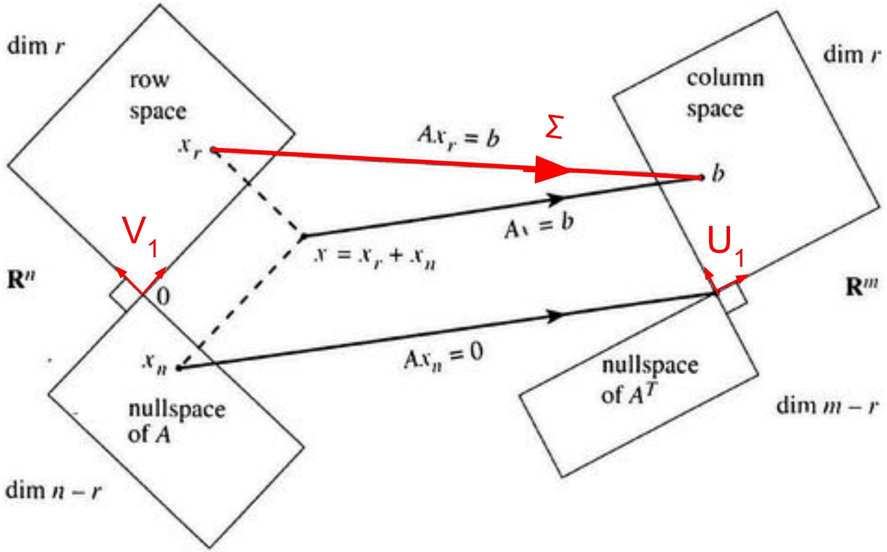


This is a pictorial diagram.
A subspace is never a rectangle: it can be a point, line, plane, or higher dimensional object.

Two subspaces would only fit on a piece of paper in combinations such as {point,point}, {point,line},{line,line},{point,plane}, but not literally as portrayed.

The purpose of this picture, then, is not to accurately portray subspaces, but rather to guide our understanding of the structure of the linear operator that takes every vector x in R^n, to Ax in R^m.

We now have the vocabulary to really understand the red arrow:

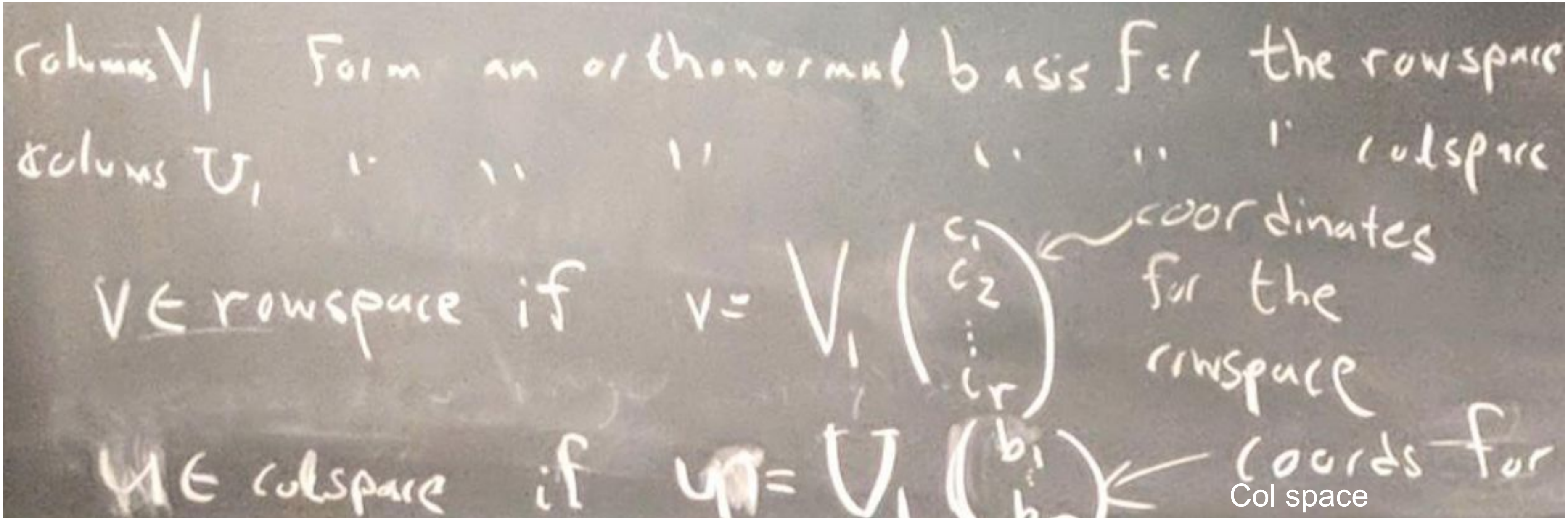


The red arrow says that a key component of the operation x→ Ax is the action from the row space to the column space.

How might we make this action concrete?

We are about to show that in the natural coordinates that we already know, the rxr matrix Σ from the compact svd perfectly describes the red arrow. The coordinates say what linear combination of the columns of V_1 go in, and that of U_1 come out.

A from row space to column space in “natural” coordinates (from the full svd):

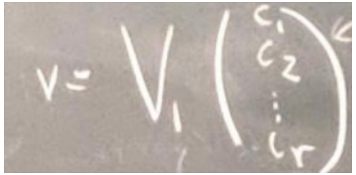


$c_1,\ldots,c_r$ are natural coordinates for the rowspace $v = V_1c = A^T(U_1\Sigma^{-1}c)$ is a linear combination of the rows.
$b_1,\ldots,b_r$ are natural coordinates for the col space $u = U_1b = A\,(V_1\Sigma^{-1}b)$ is a linear combination of the cols.

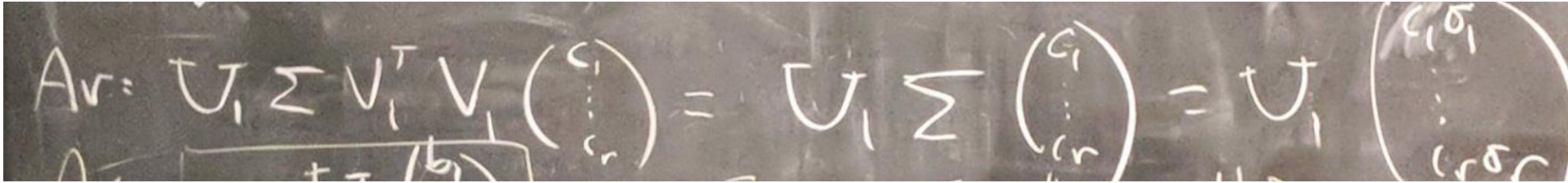
So what makes these coordinates so natural?

They seem kind of complicated. Why not just take the rows directly?

The naturalness is the form of the operation $x \to Ax$ in these coordinates.

If 

then $u = Av = AV_1c = U_1\Sigma V_1^TV_1c =$



Simply put, we have $b = \Sigma c$, which replaces the mxn A by a simpler rxr Σ which is diagonal with positive entries directly taking a rowspace vector to a column space vector through these fancy coordinates.

Restricting to the rowspace (going in) and col space (going out), we excise the non-trivial nullspaces as well.

Summary of previous two slides

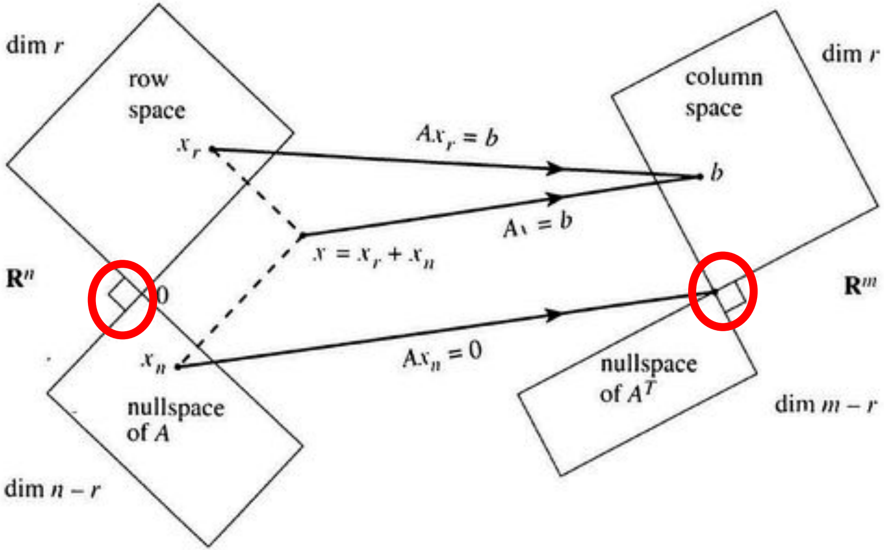
We defined $b,c \in R^r$ as column space and row space coordinates.

We use b and c to represent the vector $u = U_1 b$ and $v = V_1 c$ respectively in the original coordinate system.

Now suppose $u = A v$ in the original coordinate system. Then written in the new coordinate system, $v = A u$ becomes $c = \Sigma b$.

	Original Coordinate System	New System	Relationship
Row space	$v \in$ r dimensional subspace of R^n	$c \in R^r$	$v = V_1 c$; $c = V_1 V_1^T v$
Column space	$u \in$ r dimensional subspace of R^m	$b \in R^r$	$u = U_1 b$; $b = U_1 U_1^T u$
Linear Map	$u = A v$	$c = \Sigma b$.	

The next feature to explain: the right angles:



The concept here is the orthogonality of subspaces.

Orthogonal Subspaces

If V and W are vector subspaces of R^n, we say that V and W are ***orthogonal*** if for every $v \in V$, and $w \in W$, we have $v^T w=0$ (meaning $v \perp w$)

Note: All multiples of v and all multiples of w are orthogonal subspaces if $v \perp w$

Note: The subpsace {0} is orthogonal to every subspace.

Note: A hyperplane through 0 and its normal are orthogonal subspaces.

Note: Two hyperplanes can not be orthogonal in R^3 as they must intersect in at least a line. And a line can not be perpendicular to itself.

Note: The plane consisting of vectors of the form (x,y,0,0) and the plane with vectors (0,0,z,w) form an example of two orth planes in four dimensions.

The Orthogonal Complement

Given a vector subspace V of R^n, $V^\perp$ (pronounced “V perp”) is the set of all w such w is perpendicular to every vector v in V.

It is the largest subspace orthogonal to V -- any subspace orthogonal to V is a subset of $V^\perp$

Note that $(V^\perp)^\perp = V$ always.

In 3d: The complement of a line is a plane, the complement of a plane is a line, the complement of {0} is all of R^3 and vice versa.

The right angles in strang's picture

$\mathrm{Col}(A)^{\perp} = \mathrm{LeftNull}(A)$

$\mathrm{Row}(A)^{\perp} = \mathrm{Null}(A)$

$\mathrm{Null}(A)^{\perp} = \mathrm{Row}(A)$

$\mathrm{LeftNull}(A)^{\perp} = \mathrm{Col}(A)$

16. Least squares many ways

Earlier in the semester we saw the x that minimizes $(Ax-b)^T(Ax-B)$ could be obtained from QR. We mentioned there were some degenerate cases where it won't work. In this lecture we point out clearly that it works when A has full column rank.

We also show how to get the same answer with calculus and with subspaces. Least squares is important enough to derive many ways.

Rank Results

Claim 1: for any mxn A, rank(A) = r = rank($A^T A$)

(note A is mxn but $A^T A$ is nxn)

Take $A=U\Sigma V^T$ as the compact svd

$$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$$

Then $A^T A = V\Sigma(U^T U)\Sigma VT = V\Sigma^2 V^T$

Is a compact svd of $A^T A$ hence it also has rank r.

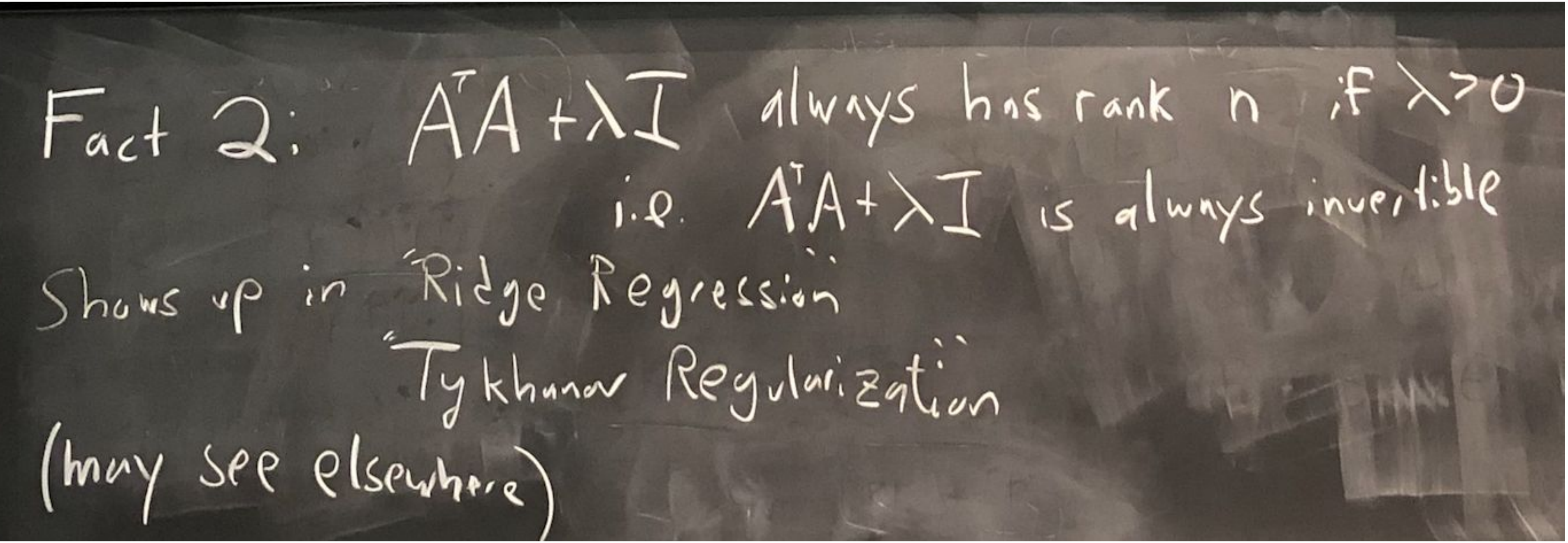
Notice $V\Sigma^2 V^T$ is an SVD of A^TA

$$\Sigma^2 = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_r^2 \end{pmatrix} \quad \sigma_1^2 > 0$$

Corollary:

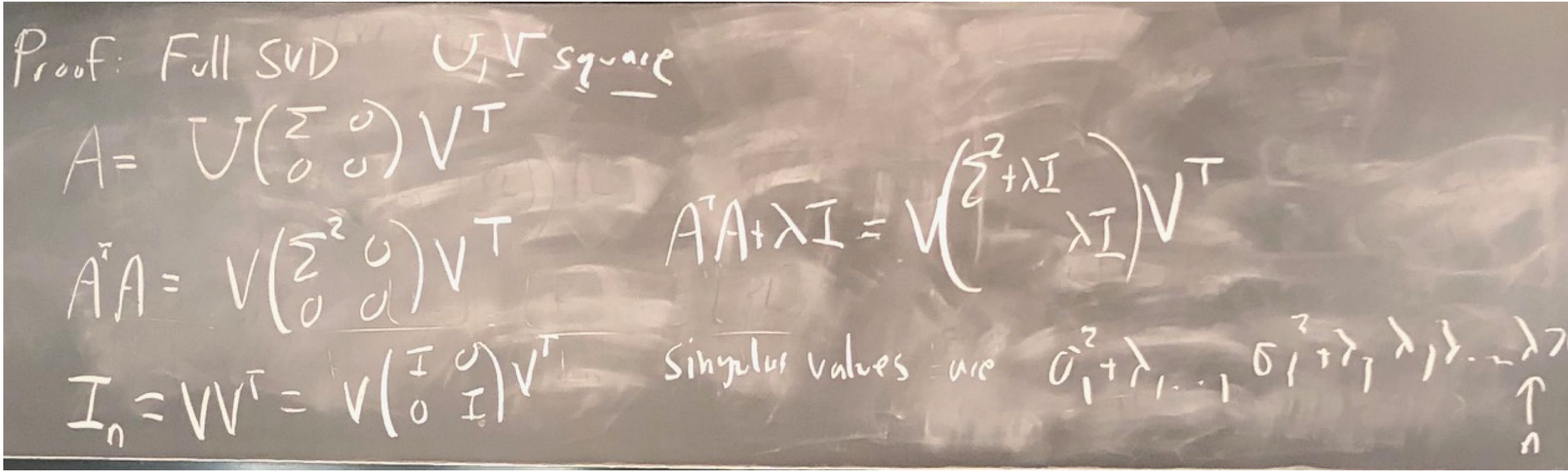
A^TA is invertible iff rank $A = n$, Full col rank, A has ind columns

This is an “if and only if”, because if A^TA is invertible, we can get V and Σ from the svd of A^TA and then $U = \quad \Sigma^{-1}VA$ and we have a compact SVD of A showing it has full rank.

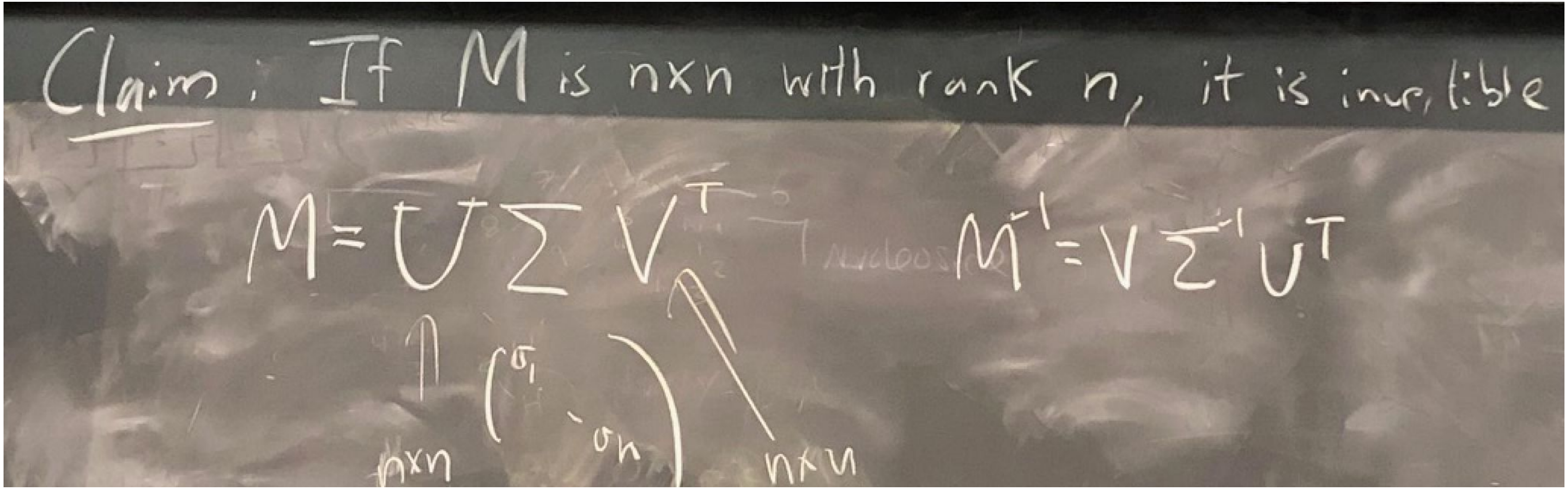


We only mention the terms “ridge regression” and “Tikhonov (Ти́хонов is spelled in English many ways) regularization” as you may see it in other classes.

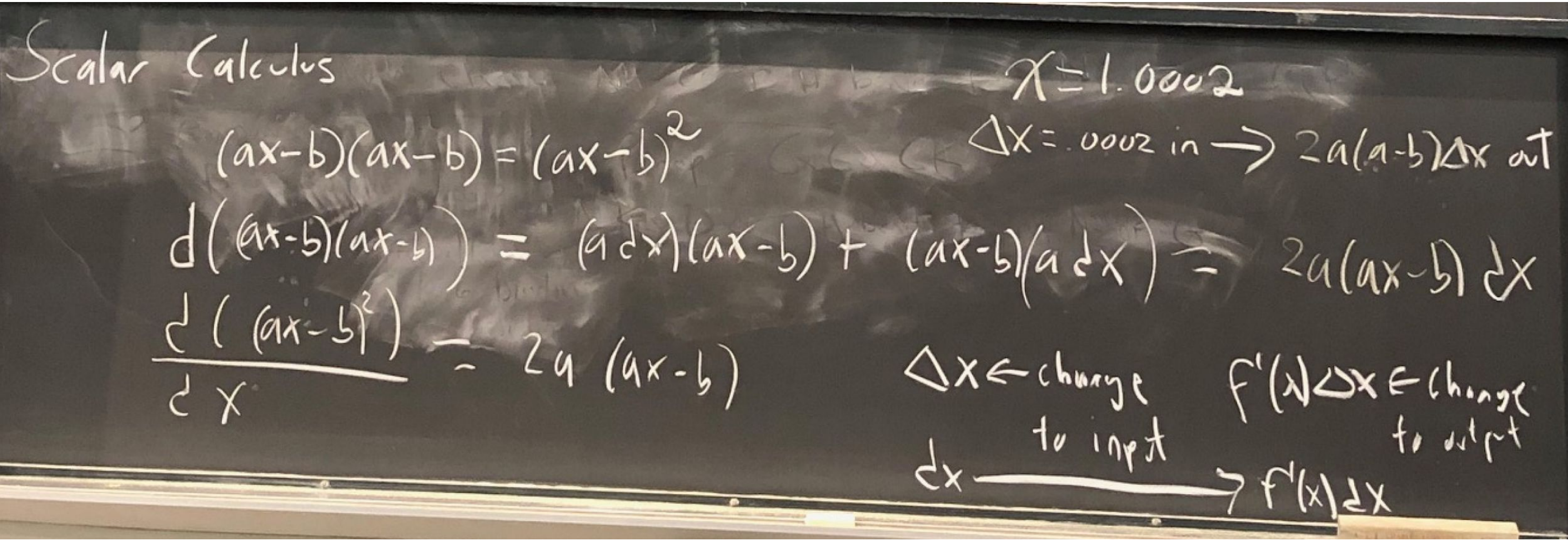
Proof of Fact 1 uses the full SVD where U,V are square
(The trick is to add the λI on the "inside")



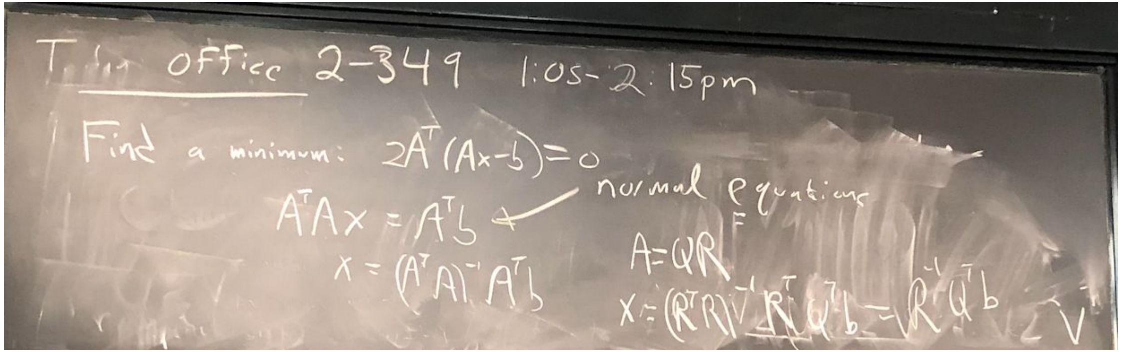
Relevant to Fact 1:
Why are square matrices of full rank invertible?
We can write the inverse explicitly: (The full SVD =The compact SVD)



Scalar calculus takes a little dx input change to f’(x)dx output change. (We will use dx or Δx synonymously)



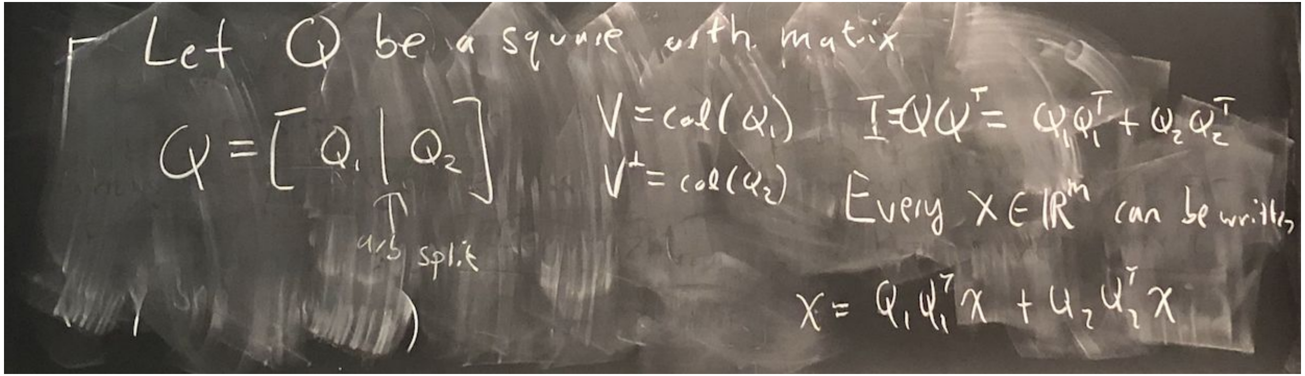
To find the minimum of $(Ax-b)^T(Ax-b)$ we solve the "normal equations" $A^TAx=A^Tb$ to get $x=(A^TA)^{-1}A^Tb$.



This works when A has ind columns.

Resolving into components

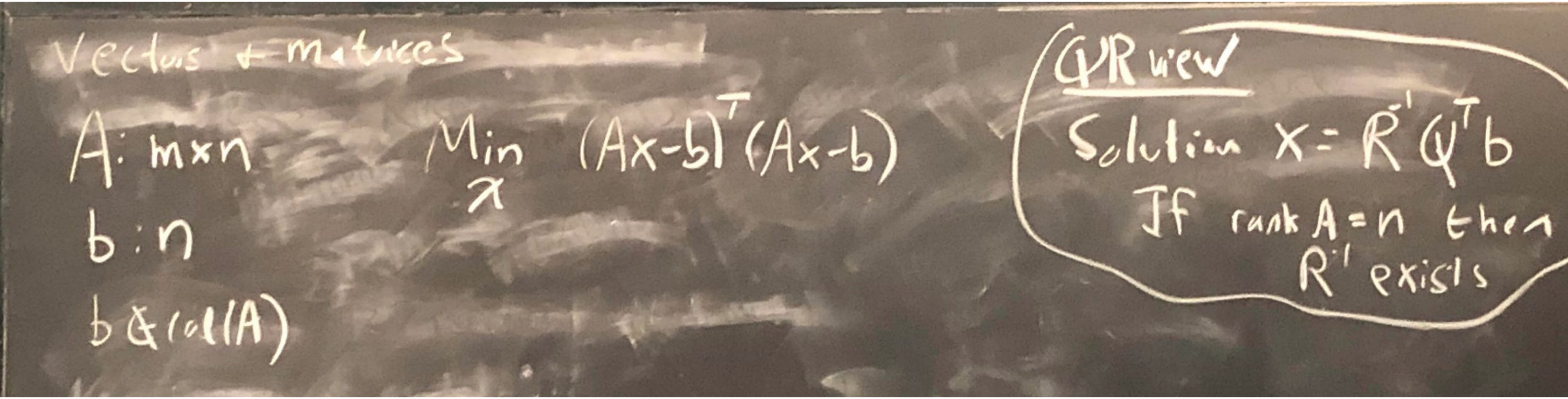
Any vector x can be written as a component in the V direction and the $V^{\perp}$ direction:



Notice that $Q_1Q_1^T$ is the projection matrix onto V and $Q_2Q_2^T$ is the projection matrix onto $V^{\perp}$

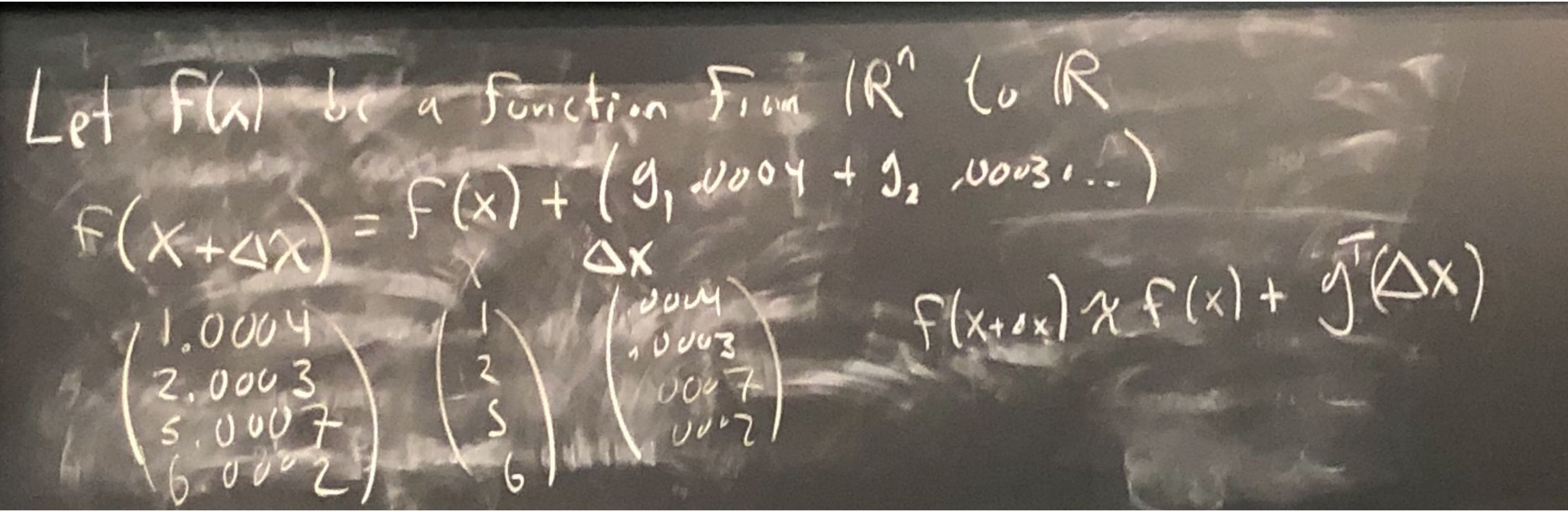
And $I = Q_1Q_1^T + Q_2Q_2^T$

We have already seen the QR view of minimizing $||Ax-b||^2$

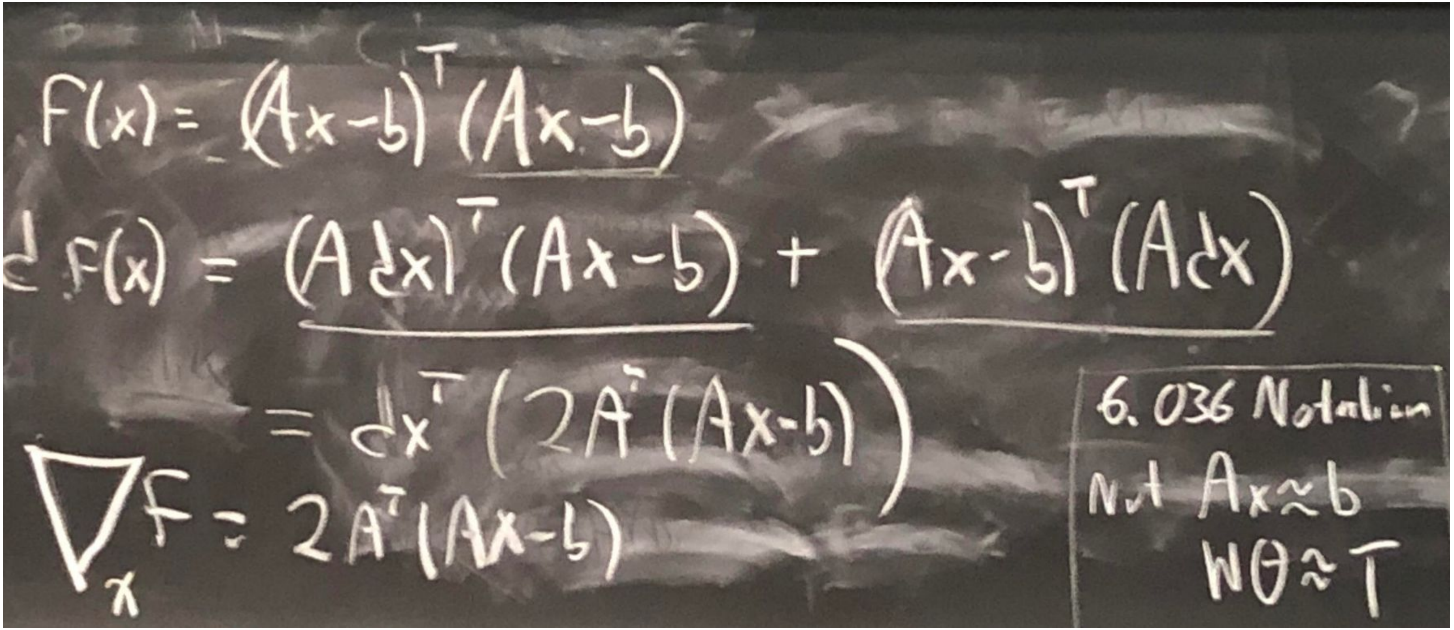


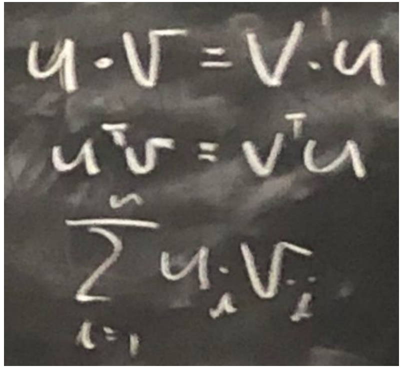
What is a gradient? It's the vector g that works best in the equation below.

It is the linear function that writes Δf as $g^T \Delta x$



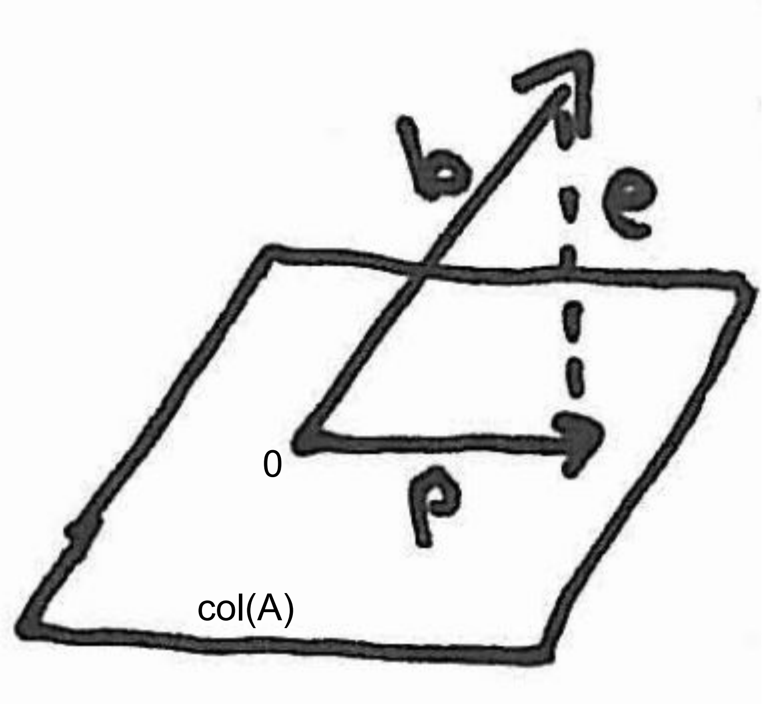
Calculating the gradient without indices: Just use the product rule but don't change the order





Remember that the dot product is commutative,
It is used on the left to get the factor of 2.

We can also minimize $(Ax-b)^T(Ax-b)$ with subspaces

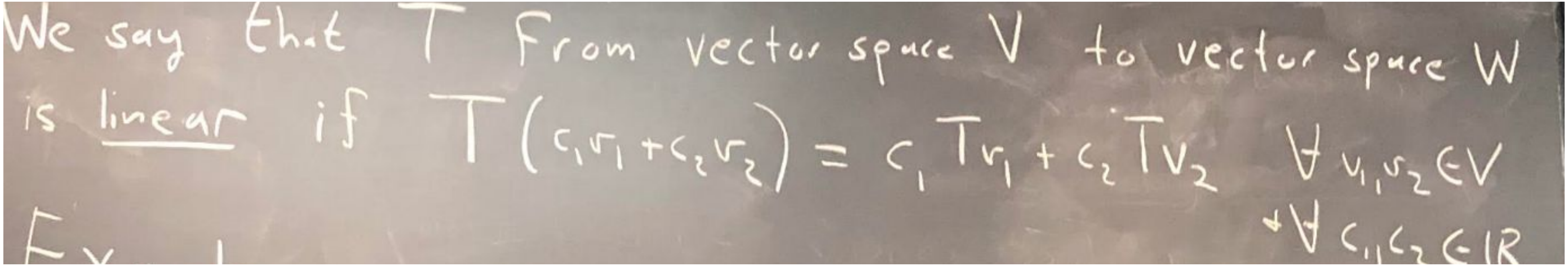


The vector e is orthogonal to every vector in col(A). Thus $e \in \mathrm{col}(A)^{\perp}$.

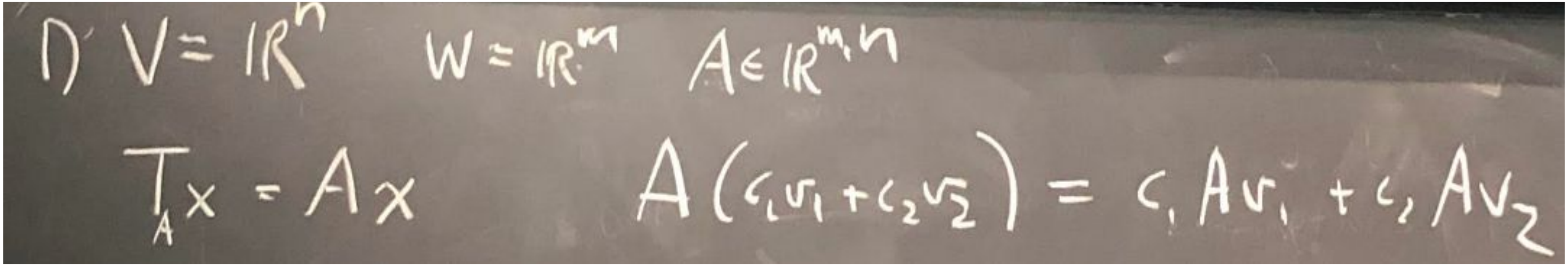
But $\mathrm{col}(A)^{\perp} = \mathrm{null}(A^T)$.

So $A^Te = A^T(b-p) = 0$, but p is Ax for some x, giving $A^TAx = A^Tb$ for the x that minimizes $(Ax-b)^T(Ax-b)$ or $x = (A^TA)^{-1}A^Tb$.

17. Linear Transformations & Matrix Calculus

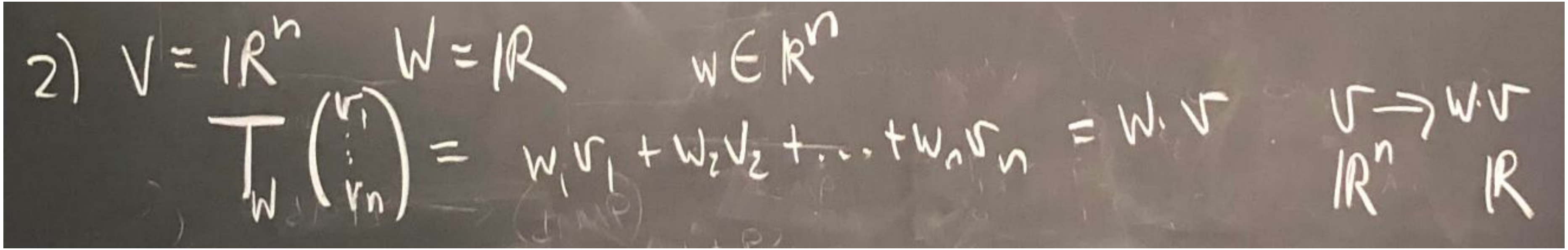


Examples: Fix an mxn matrix A. Then the map Tx=Ax is a linear transformation from R^n to R^m.



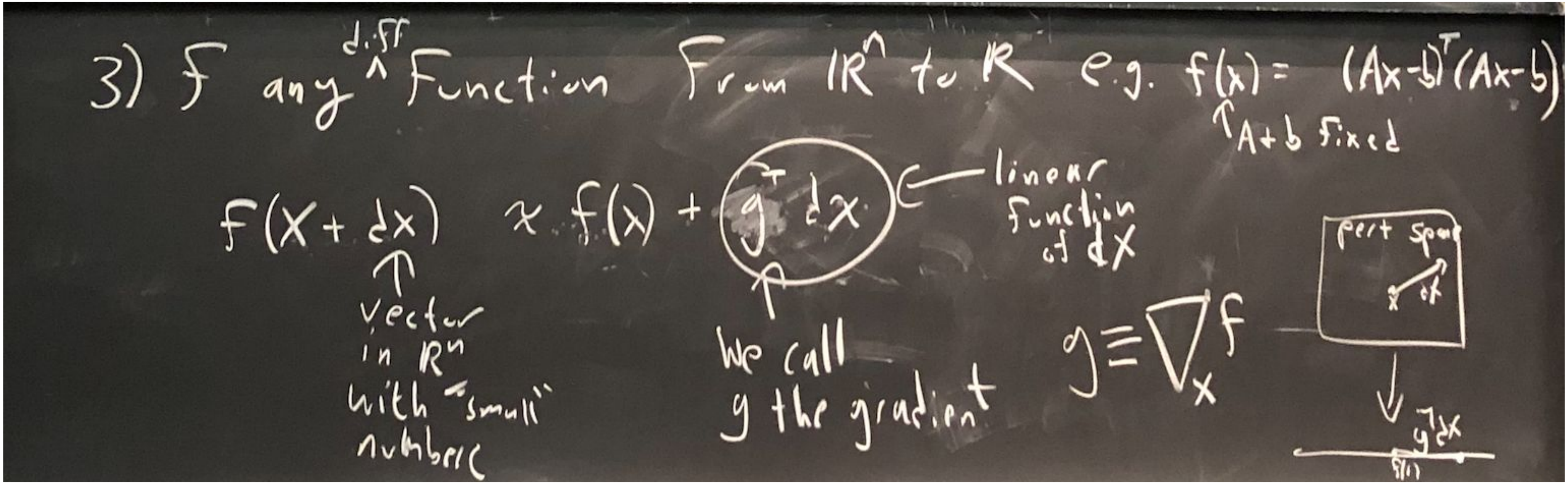
Examples of linear transformations

Every linear transformation from R^n to R takes the form of a dot product: $Tv = w^T v$

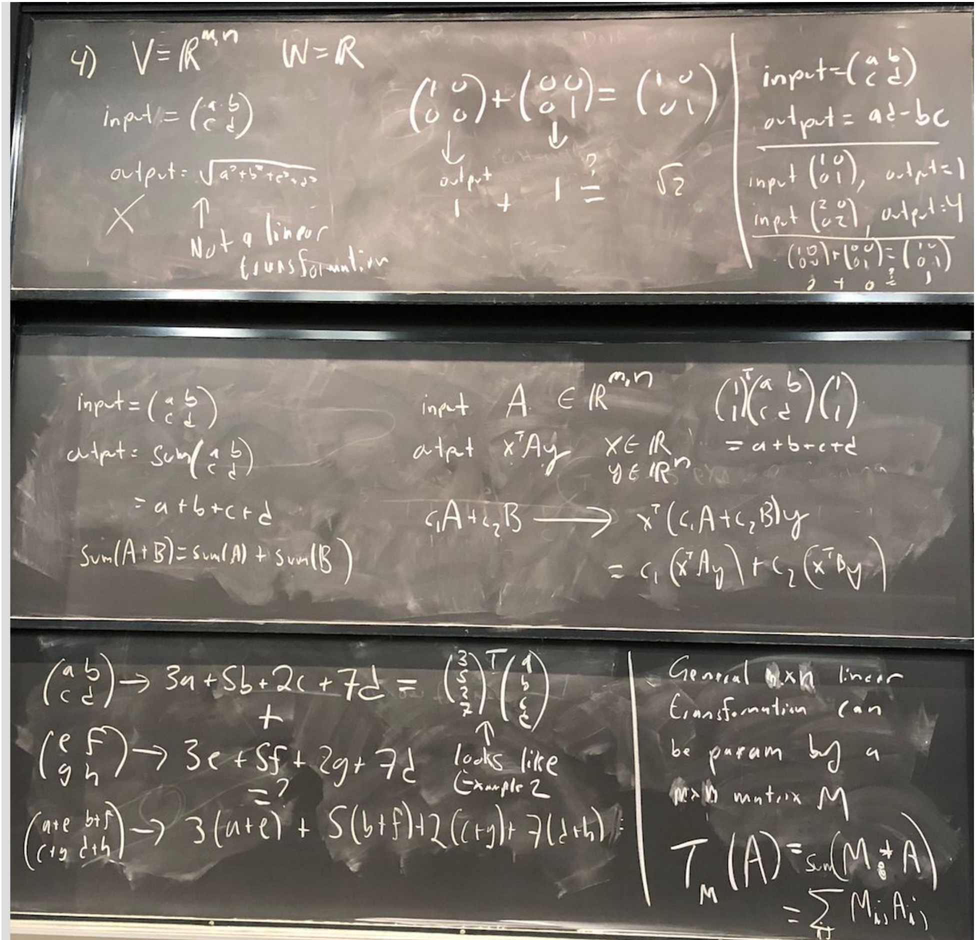


3. Gradient example: the map from dx to df is linear

Hence it is a dot product, and g is the vector that specifies which dot product



Linear Transformations from 2x2 matrices to R



Sqrt of sum of squares is not linear
Neither is a*d - b*c (known as the determinant)

The sum of the entries is linear
$A \rightarrow x^T A y$ is linear given a fixed x and y

For any mxn, $A \rightarrow$ sum(M .* A) is linear

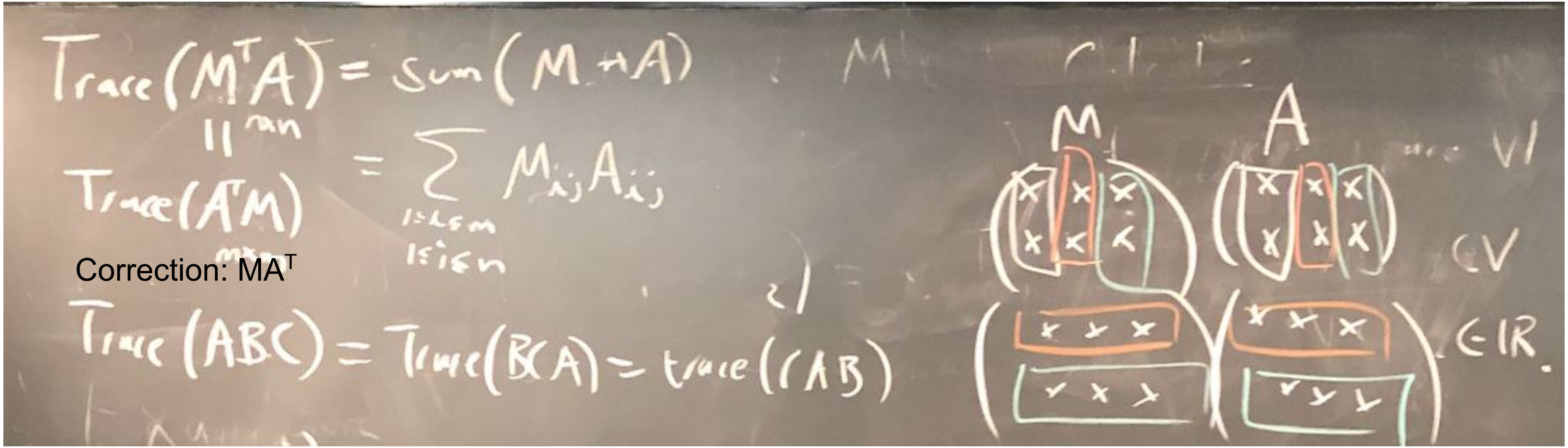
The trace of a matrix

trace(A) or tr(A) for a square matrix is the sum of the diagonal elements of A.

It is a linear transformation from square matrices to R.

A nice property is the cyclic property: tr(ABC) = tr(BCA) = tr(CAB), which holds for any number of matrices, not just three.

We can write linear functions from matrices to scalars as Trace(M^TA) = Trace(MA^T).



Correction: MA^T

This applies to mxn matrices : Note M^TA is nxn and MA^T is mxm but it works out that trace(M^TA) = trace(A^TM).

Suggestion: understand why sum(M.*A) and the trace formulas evaluate to the same thing.

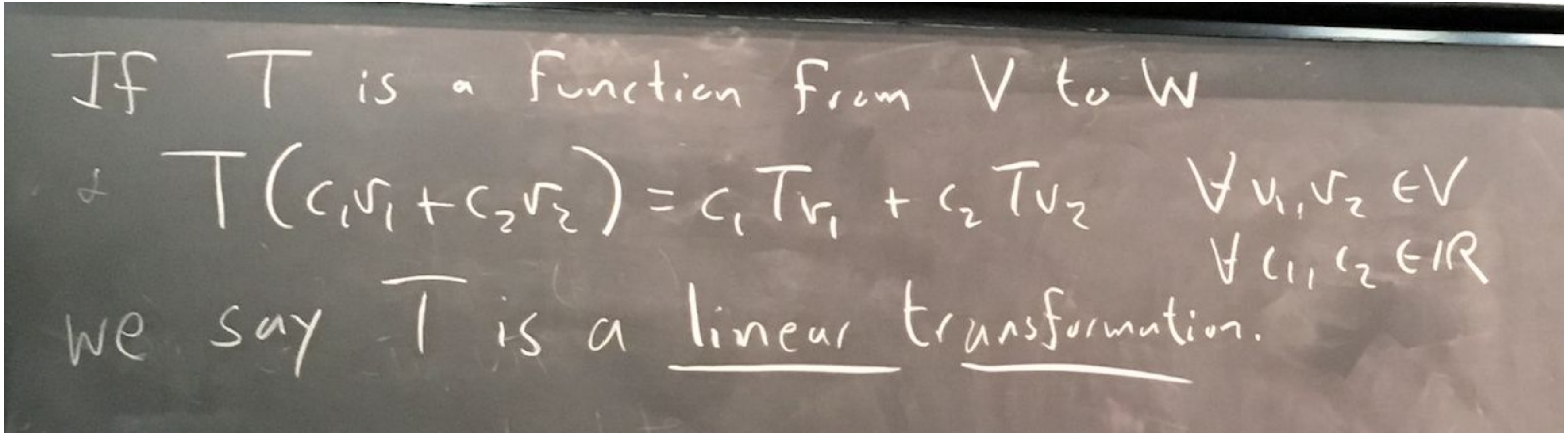
Summary

For any differentiable function, there is a linear transformation that takes dx to df. When x is explicitly in R^n and f is in R, there is a vector called the gradient that encodes this linear transformation. It gives the direction of maximal ascent.

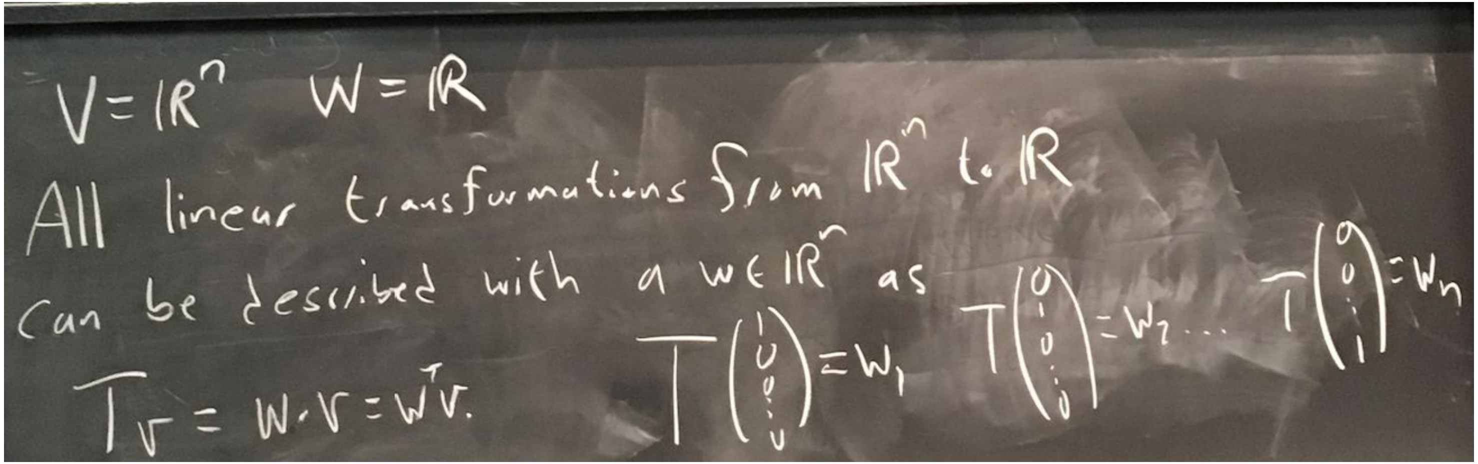
If we are going from matrices to scalars, (notation: dA to df), then there is a matrix that encodes the maximal ascent direction. Nonetheless it's not clear that representing the transformation on a computer with a matrix is the best structure.

18. Linear Transformations and Matrix Calculus continued

Reminder: Def of Linear Transformation



1. Review: From V=R^n to W=R, a linear transformation is given by a vector w through a dot product



Different w's define different linear transformations. There are n parameters needed to define a specific linear transformation from R^n to R.

If you are given a linear transformation from R^n to R as a "black box" you can recover the elements of w by applying the transformation to the columns of the identity.

2. Review: $V=R^{mn}$ to $W=R$

$$\underset{m\times n}{A} \longrightarrow \mathrm{sum}(M.*A) = \mathrm{tr}(\underbrace{A^T M}_{n\times n})$$

$$= \mathrm{tr}(\underbrace{A M^T}_{m\times m})$$

2 by 2 example:

$$A = \begin{pmatrix} x & y \\ z & w \end{pmatrix} \qquad M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$M.*A = \begin{pmatrix} ax & by \\ cz & dw \end{pmatrix} \qquad \mathrm{sum}(M.*A) = ax+by+cz+dw$$

$$A^T M = \begin{pmatrix} ax+cz & ? \\ ? & by+dw \end{pmatrix}$$

$$\mathrm{tr}(A^T M) = ax+cz+by+dw$$

$$A M^T = \begin{pmatrix} ax+by & ? \\ ? & cz+dw \end{pmatrix}$$

$$\mathrm{tr}(AM^T) = ax+by+cz+dw$$

One needs mn parameters to specify the linear transformation, these are the elements of the matrix M.

3. Review V=functions and W=R

Take some interval [a,b]

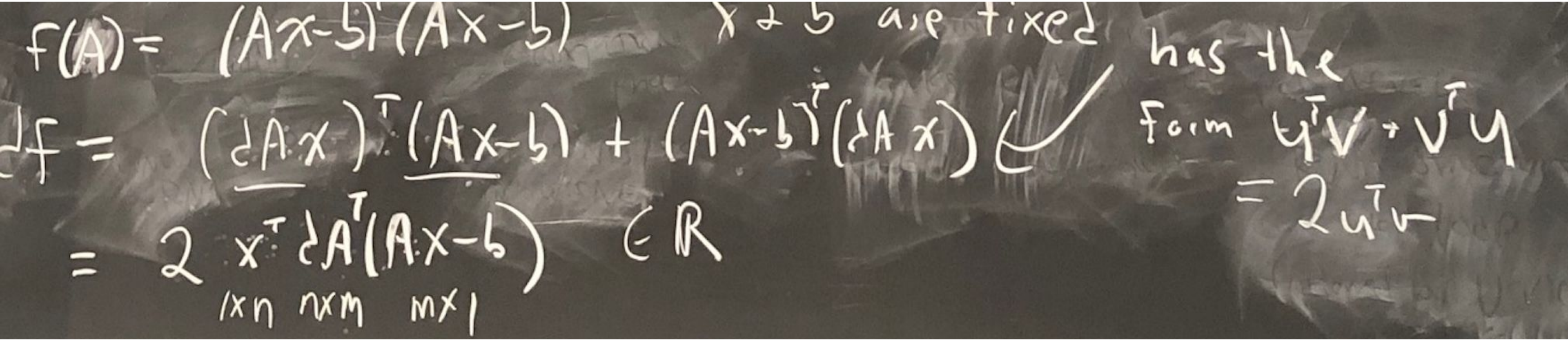
Linear transformations can be given by a function h(t) on [a,b] such that

$$f \to \int_a^b h(t) f(t)\, dt$$

This doesn't cover linear functions like f → f(0) but if you are familiar with the Dirac delta function, then even function evaluation can be put in this framework.

Perturbing $f(A) = (Ax-b)^T(Ax-b)$

$A \rightarrow A+dA$ (fix b and x)



Conclusion $df = f(A+dA)-f(A) = 2\ x^T * dA^T * (Ax-b)$

This is a linear transformation from dA to df, but not in the M format of the earlier slide

One can use a linear algebra trick to put df into the "M" format:

$x^T dA^T (Ax-b) = \mathrm{trace}(x^T dA^T (Ax-b))$ since it is a scalar.

$= \mathrm{trace}(dA^T (Ax-b)x^T)$ by the cyclic property.

Thus we can take $G = (Ax-b)x^T$ (which is an mxn matrix) as the gradient

f(A+dA) = f(A) + sum(G.*dA) to first order.

Linear Transformations from R^{nn} to R^{nn}

Some linear transformation examples are

$A \rightarrow M.*A$, $A \rightarrow MA$, $A \rightarrow AN$, $A \rightarrow MAN$, (Where M and N are fixed nxn matrices)

The general linear transformation requires n^4 parameters (There are n^2 basis matrices with one 1 and the rest 0 and each one is mapped to an nxn matrix)

The examples above are therefore not the most general possibilities

Derivatives of matrix to matrix functions

$d(A^2) = d\,(AA) = dA*A + A*dA$ (this cannot be simplified)

$d(A^3) = d(AAA) = dA*A^2 + A*dA*A + A^2\,dA$ (this cannot be simplified)

What about $d(A^{-1})$?

Let $X= A^{-1}$

$X*A = I$ so $d(X*A)=dI = 0$ (since I is constant as a function of A)

So $dX*A + X*dA =0$ or solving for dX, $d(A^{-1}) = dX = -X*dA*A^{-1} = - A^{-1} * dA * A^{-1}$

Which generalizes the familiar scalar derivative: $d(1/a) = -(1/a^2)\,da$

Demo

We demonstrated the formulas $d(A^2) = dA*A + A*dA$

and $d(A^{-1}) = - A^{-1} * dA * A^{-1}$

Julia Notebook or download pdf

Further remarks about dx, dA, etc.

In scalar calculus the use of Leibniz’s dx notation hardly gets explained well in notations such as dy/dx or df = f’(x)dx or ∫f(x)dx. I think most of us just get used to it and accept it.

When moving to vectors and matrices, perhaps it’s worth reminding ourselves what dx is supposed to have meant.

Indeed dx is simultaneously 1) an artifact of history 2) a symbolic placeholder and 3) an infinitesimal quantity which we might hope to replace with a small number on a computer.

In most high school and college calculus courses, it is the “symbolic placeholder” that we get used to the most, though the infinitesimal shows up in these courses. →

Further remarks about dx, dA, etc. cont

So what does it mean in scalar calculus to say if $f(x)=x^2$, $df = 2x\,dx$?

It means that if dx is actually a very very small number, we say that

$$f(x+dx) - f(x) = 2x\,dx \ + \text{something tiny we don't care about}$$

The (something tiny) in the limit as $dx \to 0$ is 0 when compared to dx.

Wait! $(x+dx)^2 - x^2 = 2x\,dx + (dx)^2$. Yes, that's true but we don't care about second order terms in calculus. I have never seen a high school or college calculus class mention $(dx)^2$

Because calculus is the art of linear approximations. The same carries through to vectors and matrices. Do see the demos. →

Further remarks about dx, dA, etc. cont

If f is a function from vectors x to scalars and $df = g^T dx = dx^T g$

Then the column vector g is the gradient of f.

If f is a function from matrices A to scalars and $df = tr(G^T dA) = tr(dA^T G) = sum(G.*dA)$

Then G is the direction of maximal increase of f(A) and -G is the direction of maximal decrease which can be useful in gradient descent algorithms.

We can call G the gradient

Linear transformations from R^n to R^m

There are mn parameters to nail down a linear transformation from R^n to R^m

All such transformations have the form $y = M * x$. If you specify M you specify the transformation.

The Jacobian matrix

If $x \in R^n$ and $f(x)$ is a differentiable function whose values are in R^m, then the linearization is given by the Jacobian matrix:

$df = J * dx$, where $J_{ij} = \partial f_i / \partial x_j$.

E.g.

$f(x) = h.(Wx-b)$ (comes up a lot in machine learning, where h is a scalar function)

$df = h'.(Wx-b) .* (Wdx)$ using the chain rule

This linear transformation is not explicitly in the J form, and maybe it's better that way, but if one insists one would write $J = \text{Diagonal}(h'.(Wx-b)) * W$

Vec and reshape: trivial linear transformations

Given $A \in R^{m,n}$, $vec(A) \in R^{m*n}$ is the flattening of A by columns

There is also reshape which is easy to understand by example:

(always start with the left column, etc.)

```
julia> A = [1 3 5; 2 4 6]
2×3 Array{Int64,2}:
 1  3  5
 2  4  6

julia> vec(A)
6-element Array{Int64,1}:
 1
 2
 3
 4
 5
 6

julia> reshape(A,3,2)
3×2 Array{Int64,2}:
 1  4
 2  5
 3  6
```

General form of a linear transformation from R^{mn} to R^{pq}

Given a matrix M that is p*q by m*n (so m*n*p*q parameters)

T(A) = reshape(M * vec(A), p, q)

Gives a linear transformation from R^{mn} to R^{pq}

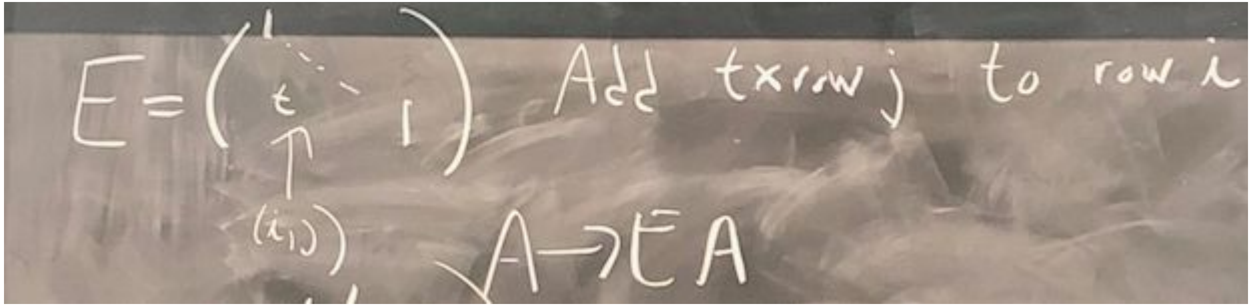
Every linear transformation has this form for some such M.

In practical applications, it is unusual to need this full form.

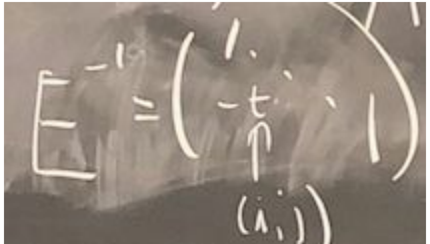
19. One last transformation and determinants

The map that takes an A from R^{mn} multiplies row j by $t \in R$ and adds it to row i is linear.

It can be written as $A \to EA$ where E is the identity with the add-on of t in entry (i,j).

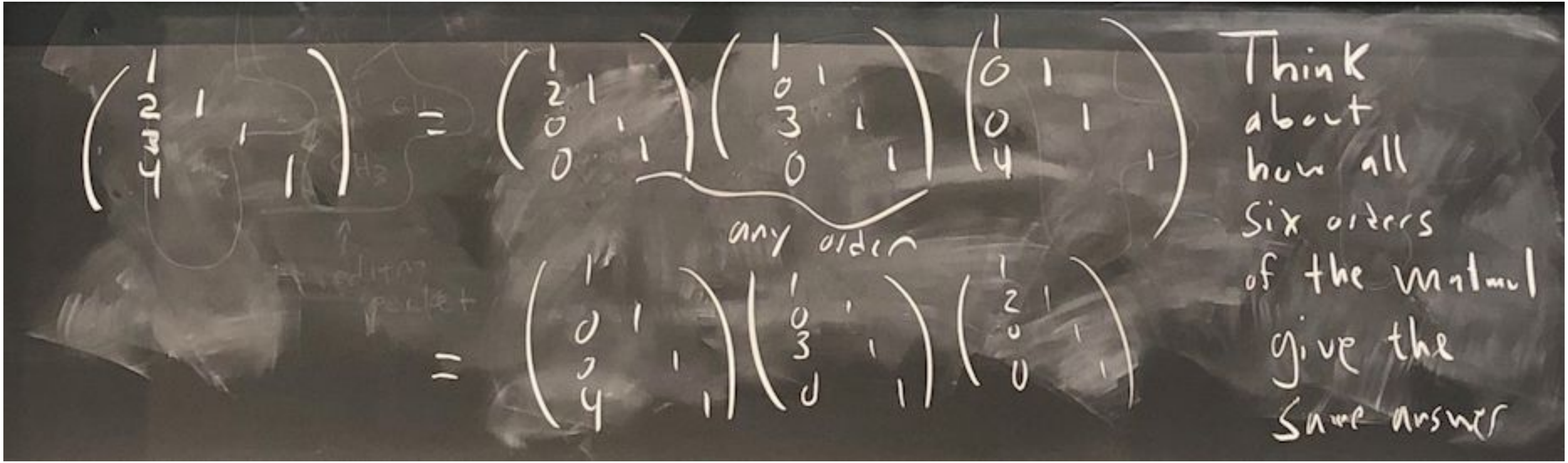


The inverse can be understood in terms of what it does rather than mechanically:



Subtract t*row j from row i.

What if we combine or reorder?



Determinants: three possible ways

1. Formula
2. Algorithm
3. Axiomatic Approach -- Very interesting intellectual method
   With just three axioms, one can show that there is only one function that takes in a matrix and returns a scalar that satisfies the axioms.

Note: Determinants are very very important for theory. Their use on computers is much less. (In floating point, you don't exactly ask if a matrix is singular, but is it singular or nearly singular, which often is want you want to know in the real world but not in theoretical math)

Explicit Formulas for determinants: get unusable quickly

```julia
#using Pkg
#Pkg.add("SymPy")
using SymPy, LinearAlgebra
```

```julia
A(n) = [ symbols("A_$i$j",real=true) for i=1:n, j=1:n]
```

```julia
A(2)
```

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

```julia
det(A(2))
```

$$A_{11}A_{22} - A_{12}A_{21}$$

```julia
A(3)
```

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

```julia
det(A(3))
```

$$A_{11}A_{22}A_{33} - A_{11}A_{23}A_{32} - A_{12}A_{21}A_{33} + A_{12}A_{23}A_{31} + A_{13}A_{21}A_{32} - A_{13}A_{22}A_{31}$$

```julia
A(4)
```

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

```julia
det(A(4))
```

$$\begin{aligned} &A_{11}A_{22}A_{33}A_{44} - A_{11}A_{22}A_{34}A_{43} - A_{11}A_{23}A_{32}A_{44} + A_{11}A_{23}A_{34}A_{42} + A_{11}A_{24}A_{32}A_{43} - A_{11}A_{24}A_{33}A_{42} - A_{12}A_{21}A_{33}A_{44} + A_{12}A_{21}A_{34}A_{43} \\ &+ A_{12}A_{23}A_{31}A_{44} - A_{12}A_{23}A_{34}A_{41} - A_{12}A_{24}A_{31}A_{43} + A_{12}A_{24}A_{33}A_{41} + A_{13}A_{21}A_{32}A_{44} - A_{13}A_{21}A_{34}A_{42} - A_{13}A_{22}A_{31}A_{44} + A_{13}A_{22}A_{34}A_{41} \\ &+ A_{13}A_{24}A_{31}A_{42} - A_{13}A_{24}A_{32}A_{41} - A_{14}A_{21}A_{32}A_{43} + A_{14}A_{21}A_{33}A_{42} + A_{14}A_{22}A_{31}A_{43} - A_{14}A_{22}A_{33}A_{41} - A_{14}A_{23}A_{31}A_{42} + A_{14}A_{23}A_{32}A_{41} \end{aligned}$$

```
A(5)
```

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix}$$

```
det(A(5))
```

$$\begin{gathered}
A_{11}A_{22}A_{33}A_{44}A_{55} - A_{11}A_{22}A_{33}A_{45}A_{54} - A_{11}A_{22}A_{34}A_{43}A_{55} + A_{11}A_{22}A_{34}A_{45}A_{53} + A_{11}A_{22}A_{35}A_{43}A_{54} - A_{11}A_{22}A_{35}A_{44}A_{53} \\
- A_{11}A_{23}A_{32}A_{44}A_{55} + A_{11}A_{23}A_{32}A_{45}A_{54} + A_{11}A_{23}A_{34}A_{42}A_{55} - A_{11}A_{23}A_{34}A_{45}A_{52} - A_{11}A_{23}A_{35}A_{42}A_{54} + A_{11}A_{23}A_{35}A_{44}A_{52} \\
+ A_{11}A_{24}A_{32}A_{43}A_{55} - A_{11}A_{24}A_{32}A_{45}A_{53} - A_{11}A_{24}A_{33}A_{42}A_{55} + A_{11}A_{24}A_{33}A_{45}A_{52} + A_{11}A_{24}A_{35}A_{42}A_{53} - A_{11}A_{24}A_{35}A_{43}A_{52} \\
- A_{11}A_{25}A_{32}A_{43}A_{54} + A_{11}A_{25}A_{32}A_{44}A_{53} + A_{11}A_{25}A_{33}A_{42}A_{54} - A_{11}A_{25}A_{33}A_{44}A_{52} - A_{11}A_{25}A_{34}A_{42}A_{53} + A_{11}A_{25}A_{34}A_{43}A_{52} \\
- A_{12}A_{21}A_{33}A_{44}A_{55} + A_{12}A_{21}A_{33}A_{45}A_{54} + A_{12}A_{21}A_{34}A_{43}A_{55} - A_{12}A_{21}A_{34}A_{45}A_{53} - A_{12}A_{21}A_{35}A_{43}A_{54} + A_{12}A_{21}A_{35}A_{44}A_{53} \\
+ A_{12}A_{23}A_{31}A_{44}A_{55} - A_{12}A_{23}A_{31}A_{45}A_{54} - A_{12}A_{23}A_{34}A_{41}A_{55} + A_{12}A_{23}A_{34}A_{45}A_{51} + A_{12}A_{23}A_{35}A_{41}A_{54} - A_{12}A_{23}A_{35}A_{44}A_{51} \\
- A_{12}A_{24}A_{31}A_{43}A_{55} + A_{12}A_{24}A_{31}A_{45}A_{53} + A_{12}A_{24}A_{33}A_{41}A_{55} - A_{12}A_{24}A_{33}A_{45}A_{51} - A_{12}A_{24}A_{35}A_{41}A_{53} + A_{12}A_{24}A_{35}A_{43}A_{51} \\
+ A_{12}A_{25}A_{31}A_{43}A_{54} - A_{12}A_{25}A_{31}A_{44}A_{53} - A_{12}A_{25}A_{33}A_{41}A_{54} + A_{12}A_{25}A_{33}A_{44}A_{51} + A_{12}A_{25}A_{34}A_{41}A_{53} - A_{12}A_{25}A_{34}A_{43}A_{51} \\
+ A_{13}A_{21}A_{32}A_{44}A_{55} - A_{13}A_{21}A_{32}A_{45}A_{54} - A_{13}A_{21}A_{34}A_{42}A_{55} + A_{13}A_{21}A_{34}A_{45}A_{52} + A_{13}A_{21}A_{35}A_{42}A_{54} - A_{13}A_{21}A_{35}A_{44}A_{52} \\
- A_{13}A_{22}A_{31}A_{44}A_{55} + A_{13}A_{22}A_{31}A_{45}A_{54} + A_{13}A_{22}A_{34}A_{41}A_{55} - A_{13}A_{22}A_{34}A_{45}A_{51} - A_{13}A_{22}A_{35}A_{41}A_{54} + A_{13}A_{22}A_{35}A_{44}A_{51} \\
+ A_{13}A_{24}A_{31}A_{42}A_{55} - A_{13}A_{24}A_{31}A_{45}A_{52} - A_{13}A_{24}A_{32}A_{41}A_{55} + A_{13}A_{24}A_{32}A_{45}A_{51} + A_{13}A_{24}A_{35}A_{41}A_{52} - A_{13}A_{24}A_{35}A_{42}A_{51} \\
- A_{13}A_{25}A_{31}A_{42}A_{54} + A_{13}A_{25}A_{31}A_{44}A_{52} + A_{13}A_{25}A_{32}A_{41}A_{54} - A_{13}A_{25}A_{32}A_{44}A_{51} - A_{13}A_{25}A_{34}A_{41}A_{52} + A_{13}A_{25}A_{34}A_{42}A_{51} \\
- A_{14}A_{21}A_{32}A_{43}A_{55} + A_{14}A_{21}A_{32}A_{45}A_{53} + A_{14}A_{21}A_{33}A_{42}A_{55} - A_{14}A_{21}A_{33}A_{45}A_{52} - A_{14}A_{21}A_{35}A_{42}A_{53} + A_{14}A_{21}A_{35}A_{43}A_{52} \\
+ A_{14}A_{22}A_{31}A_{43}A_{55} - A_{14}A_{22}A_{31}A_{45}A_{53} - A_{14}A_{22}A_{33}A_{41}A_{55} + A_{14}A_{22}A_{33}A_{45}A_{51} + A_{14}A_{22}A_{35}A_{41}A_{53} - A_{14}A_{22}A_{35}A_{43}A_{51} \\
- A_{14}A_{23}A_{31}A_{42}A_{55} + A_{14}A_{23}A_{31}A_{45}A_{52} + A_{14}A_{23}A_{32}A_{41}A_{55} - A_{14}A_{23}A_{32}A_{45}A_{51} - A_{14}A_{23}A_{35}A_{41}A_{52} + A_{14}A_{23}A_{35}A_{42}A_{51} \\
+ A_{14}A_{25}A_{31}A_{42}A_{53} - A_{14}A_{25}A_{31}A_{43}A_{52} - A_{14}A_{25}A_{32}A_{41}A_{53} + A_{14}A_{25}A_{32}A_{43}A_{51} + A_{14}A_{25}A_{33}A_{41}A_{52} - A_{14}A_{25}A_{33}A_{42}A_{51} \\
+ A_{15}A_{21}A_{32}A_{43}A_{54} - A_{15}A_{21}A_{32}A_{44}A_{53} - A_{15}A_{21}A_{33}A_{42}A_{54} + A_{15}A_{21}A_{33}A_{44}A_{52} + A_{15}A_{21}A_{34}A_{42}A_{53} - A_{15}A_{21}A_{34}A_{43}A_{52} \\
- A_{15}A_{22}A_{31}A_{43}A_{54} + A_{15}A_{22}A_{31}A_{44}A_{53} + A_{15}A_{22}A_{33}A_{41}A_{54} - A_{15}A_{22}A_{33}A_{44}A_{51} - A_{15}A_{22}A_{34}A_{41}A_{53} + A_{15}A_{22}A_{34}A_{43}A_{51} \\
+ A_{15}A_{23}A_{31}A_{42}A_{54} - A_{15}A_{23}A_{31}A_{44}A_{52} - A_{15}A_{23}A_{32}A_{41}A_{54} + A_{15}A_{23}A_{32}A_{44}A_{51} + A_{15}A_{23}A_{34}A_{41}A_{52} - A_{15}A_{23}A_{34}A_{42}A_{51} \\
- A_{15}A_{24}A_{31}A_{42}A_{53} + A_{15}A_{24}A_{31}A_{43}A_{52} + A_{15}A_{24}A_{32}A_{41}A_{53} - A_{15}A_{24}A_{32}A_{43}A_{51} - A_{15}A_{24}A_{33}A_{41}A_{52} + A_{15}A_{24}A_{33}A_{42}A_{51}
\end{gathered}$$

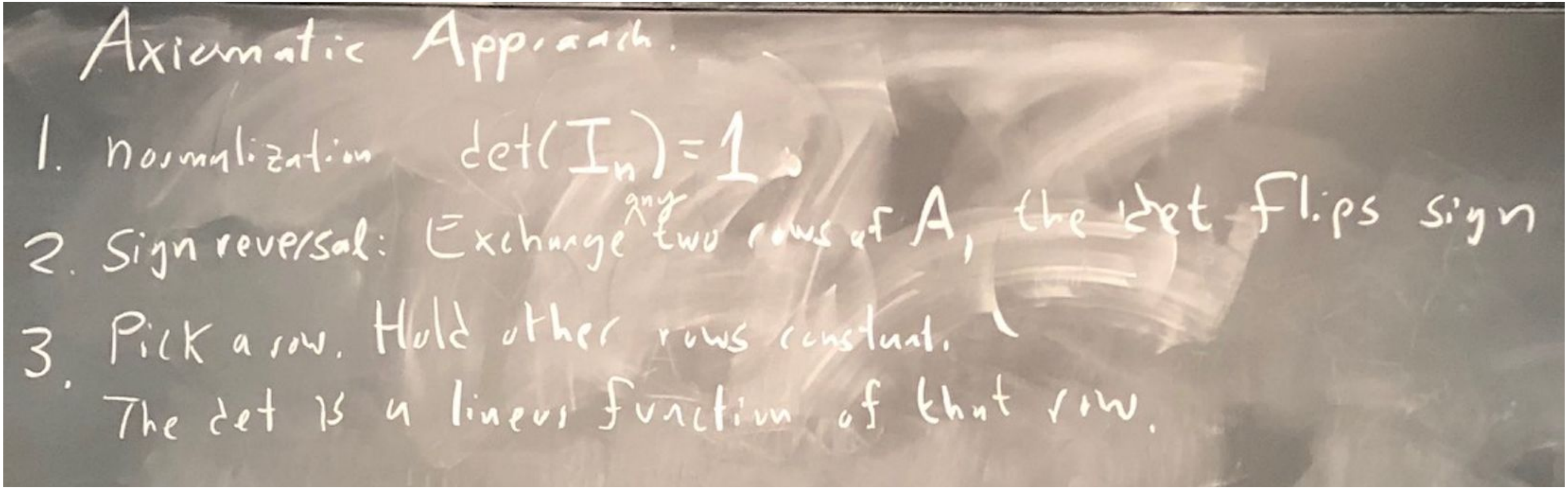
Algorithmic Approach

Do row operations to make the matrix upper triangular and then take the product of the diagonal elements.

If you swap rows, negate the answer for each swap.

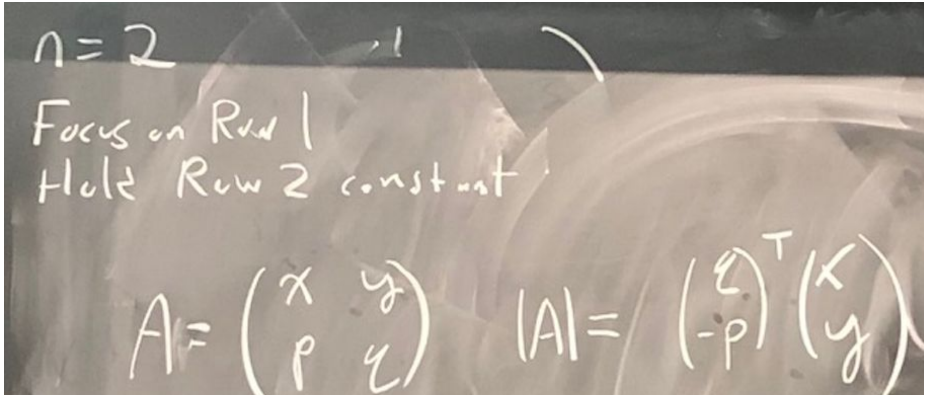
(This is how determinants are computed on real computers if they are computed at all.)

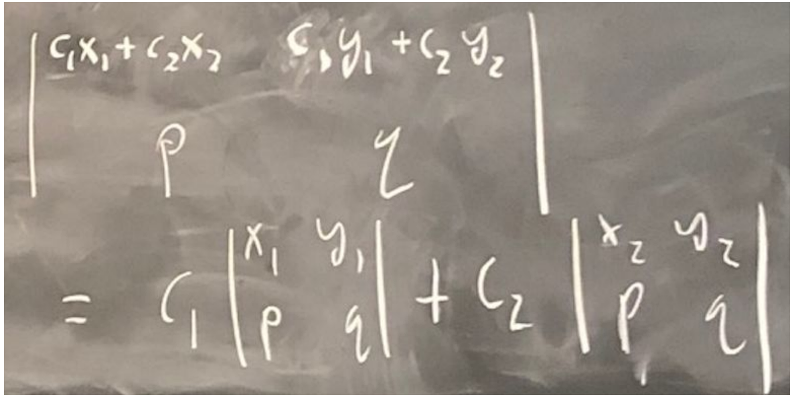
Axiomatric Approach
Three axioms.



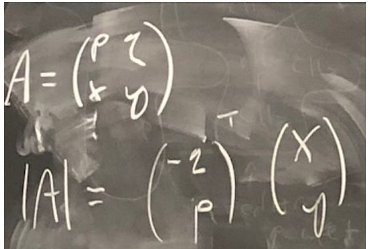
Maybe there isn't any function that satisfies all three, maybe there are infinitely many, but it will turn out there is one.

Axiom 3 example with 2x2 matrices:



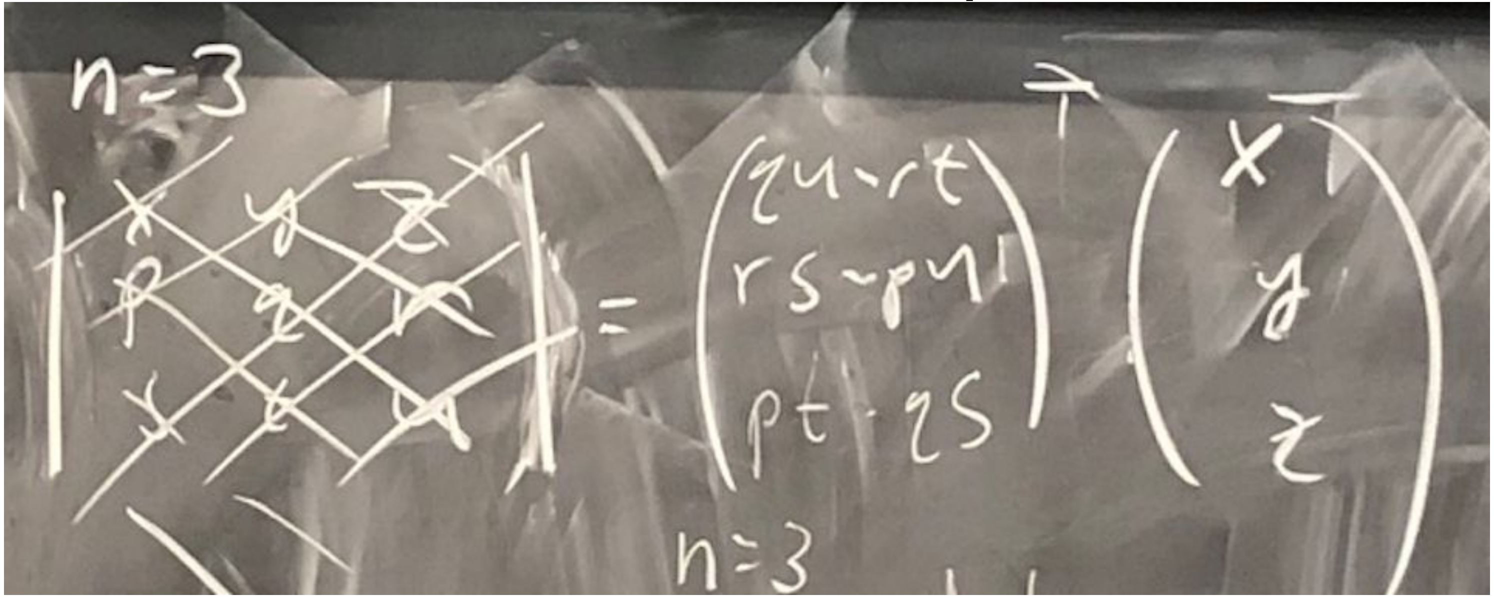


Above shows that 2x2 is a linear transformation of the first row, because we know linear transforms have the form of a dot product. Right shows the second row.



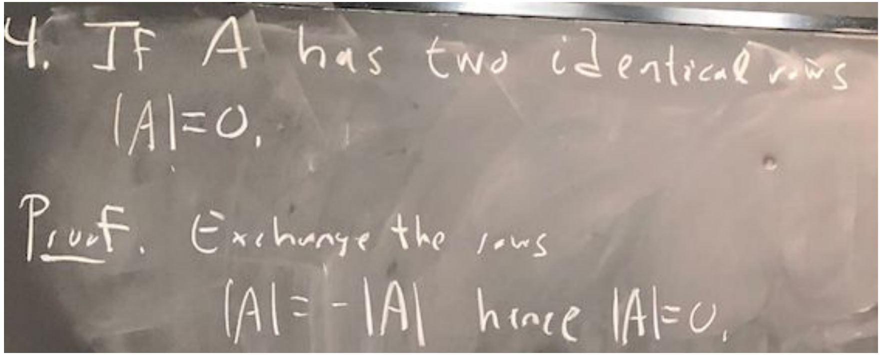
This shows explicitly that the determinant is a linear transformation of the first row.

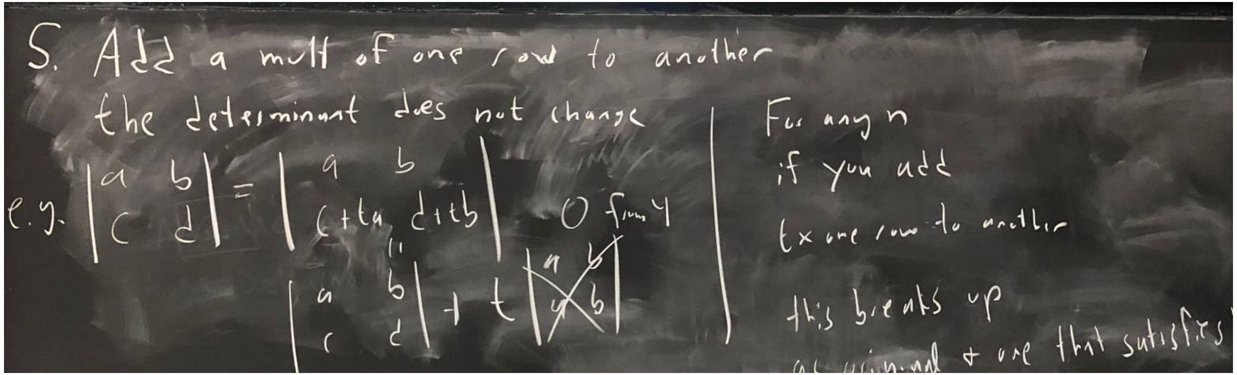
Axiom 3: when n=3 as a dot product



As a side note, the vector in the dot product is the cross product of rows 2 and 3.

Two consequences of our axioms





18.06 Lecture 20

Last Time:

Row Operations/ Elimination

Determinant

Reviewed 2×2 and 3×3 formulas

Sketched general determinant algorithm

Started axiomatic approach to determinant

Today:

Finish axiomatic approach

Properties of determinant

Determinant

- Function: $\mathbb{R}^{n\times n} \longrightarrow \mathbb{R}$

  Input: $n \times n$ matrix

  Output: a single real number

Could write down explicit formula

  Then start proving properties formula satisfies

Instead: write down properties we want the determinant to satisfy

  Show there is a function that satisfies these properties

  Show only one function satisfies these properties

Determinant: Notation

- For a matrix A, denote determinant of A by

  $\det(A)$

  or $|A|$

Example:

matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

determinant $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$

Determinant: Axioms

- Property 1: Identity matrix

    If I is an identity matrix,

    Then $\det(I) = 1$

Examples:

$$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1$$

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = 1$$

Determinant: Axioms

- Property 2: Row swapping

    For any square matrix A

    For any A' obtained from A by swapping any two rows

    $$\det(A') = -\det(A)$$

Example:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \longrightarrow A' = \begin{pmatrix} g & h & i \\ d & e & f \\ a & b & c \end{pmatrix}$$

$$\det(A') = -\det(A)$$

- Property 3: Linearity in each row (does not mean $|A+B| = |A| + |B|$)
    - For any square matrix A and any row i
        - Fix all of A except row i
            - Determinant is a linear function of row i

Example:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$, multiply all entries in row 2 by $\alpha \in \mathbb{R}$

$$\begin{vmatrix} a & b & c \\ \alpha d & \alpha e & \alpha f \\ g & h & i \end{vmatrix} = \alpha \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

- Property 3: Linearity in each row (does not mean $|A+B| = |A| + |B|$)
  - For any square matrix A and any row i
    - Fix all of A except row i
      - Determinant is a linear function of row i

Example:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, A' = \begin{pmatrix} a & b & c \\ d' & e' & f' \\ g & h & i \end{pmatrix} \text{ identical outside row 2}$$

$$\begin{vmatrix} a & b & c \\ d+d' & e+e' & f+f' \\ g & h & i \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} a & b & c \\ d' & e' & f' \\ g & h & i \end{vmatrix}$$

Determinant from the Axioms

- Will show exists exactly one function that satisfies these axioms
  - Axioms force value of $\det(A)$, for any A

Property 1 tells us the determinant of I

Properties 2&3 tell us how determinant changes when matrix changes

Rough idea: For any matrix A, to figure out $\det(A)$

Change A little-by-little until it looks like I

Keep track of how determinant changes along the way

Determinant of a Diagonal Matrix

- (square) Diagonal matrix D

First special case: $D = \text{diag}(d_1, 1,1, \dots, 1)$, $d_1 \in \mathbb{R}$

$$\begin{vmatrix} d_1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_1 \begin{vmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_1$$

Property 3 Property 1

Determinant of a Diagonal Matrix

- (square) Diagonal matrix D

Second special case: $D = \mathrm{diag}(d_1, d_2, 1, \dots, 1)$, $d_1, d_2 \in \mathbb{R}$

$$\begin{vmatrix} d_1 & & & & \\ & d_2 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_2 \begin{vmatrix} d_1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_1 d_2 \begin{vmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_1 d_2$$

Property 3 Property 3 Property 1

Determinant of a Diagonal Matrix

- (square) Diagonal matrix D

General case: $D = \text{diag}(d_1, d_2, d_3, \dots, d_n)$, $d_1, d_2, d_3, \dots, d_n \in \mathbb{R}$

$$\begin{vmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & \ddots & \\ & & & & d_n \end{vmatrix} = \cdots = d_1 d_2 d_3 \cdots d_n \begin{vmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{vmatrix} = d_1 d_2 d_3 \cdots d_n$$

Property 3

Property 1

$\boxed{\det(D) = d_1 d_2 d_3 \cdots d_n}$

Determinant forced by axioms

Determinant: Properties

- If A has two identical rows, $\det(A) = 0$

  Swap identical rows to get “new” matrix $A' = A$

  Property 2: $\det(A') = -\det(A)$

  but $\det(A') = \det(A)$

  So $\det(A) = -\det(A)$

Determinant: Properties

- Adding a multiple of one row of A to a different row of A does not change det

  Example: Add multiple α of first row to second row of 3×3 matrix A

$$
\begin{vmatrix} a & b & c \\ d+\alpha a & e+\alpha b & f+\alpha c \\ g & h & i \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} a & b & c \\ \alpha a & \alpha b & \alpha c \\ g & h & i \end{vmatrix}
$$

$$
= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} + \alpha \underbrace{\begin{vmatrix} a & b & c \\ a & b & c \\ g & h & i \end{vmatrix}}_{0}
$$

$$
= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}
$$

Determinant: Properties

- If A has an entire row of zeros, $\det(A) = 0$

  Add any other row of A to zero row

  By previous, determinant does not change

  But now, two identical rows, so det is 0

$$\begin{vmatrix} a & b & c \\ 0 & 0 & 0 \\ g & h & i \end{vmatrix}$$

$$\begin{vmatrix} a & b & c \\ a & b & c \\ g & h & i \end{vmatrix}$$

Determinant of a Lower Triangular Matrix

- Example: $L = \begin{pmatrix} 5 & 0 & 0 \\ 7 & 2 & 0 \\ -4 & 0 & -1 \end{pmatrix}$

Consider general $n \times n$ lower triangular matrix L

For now, assume all diagonal entries are non-zero

Idea: Transform L to a diagonal matrix

repeatedly add a multiple of a row to another row

det does not change

already know det for diagonal matrix

Determinant of a Lower Triangular Matrix

- Start with arbitrary lower triangular L (with no zeros on diagonal)

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{vmatrix}$$

Eliminate all entries in first col below diagonal

By adding $-\frac{l_{21}}{l_{11}}$ times 1st row to 2nd row

By adding $-\frac{l_{31}}{l_{11}}$ times 1st row to 3rd row

…

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ 0 & l_{22} & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & l_{42} & l_{43} & l_{44} \end{vmatrix}$$

Determinant does not change, only entries in 1st col below diag. change

Determinant of a Lower Triangular Matrix

- Now have lower triangular L with first col clear

Eliminate all entries in second col below diagonal

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ 0 & l_{22} & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & l_{42} & l_{43} & l_{44} \end{vmatrix}$$

By adding $-\frac{l_{32}}{l_{22}}$ times 2nd row to 3rd row

By adding $-\frac{l_{42}}{l_{22}}$ times 2nd row to 4th row

…

Determinant does not change

only entries in 2nd col below diag. change

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ 0 & l_{22} & 0 & 0 \\ 0 & 0 & l_{33} & 0 \\ 0 & 0 & l_{43} & l_{44} \end{vmatrix}$$

Determinant of a Lower Triangular Matrix

- Repeat until matrix is diagonal

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ 0 & l_{22} & 0 & 0 \\ 0 & 0 & l_{33} & 0 \\ 0 & 0 & 0 & l_{44} \end{vmatrix}$$

Det did not change at any point in process

Diagonal entries did not change at any point

$\det(L) = l_{11} l_{22} \cdots l_{nn}$

Determinant of lower triangular matrix is product of diagonal entries

(when all diagonal entries are nonzero)

Determinant of a Lower Triangular Matrix

- What if a diagonal entry is zero?

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & l_{42} & l_{43} & l_{44} \end{vmatrix}$$

Do previous procedure until we arrive at column with zero on diagonal

But then we have zero row

So $\det(L) = 0$

In both cases: $\boxed{\det(L) = l_{11} l_{22} \cdots l_{nn}}$

Determinant of lower triangular matrix is product of diagonal entries

Same holds for upper triangular matrix

Determinant of triangular matrices forced by axioms

Determinant of any Matrix

- Given any matrix A

Transform to lower triangular by doing previous procedure "backwards"

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

Start in last column

Eliminate everything above main diagonal

$$\begin{vmatrix} a'_{11} & a'_{12} & a'_{13} & 0 \\ a'_{21} & a'_{22} & a'_{23} & 0 \\ a'_{31} & a'_{32} & a'_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

Determinant does not change

Many entries do change

Determinant of any Matrix

Repeat procedure until lower triangular

One last special case: if col has zero on diag

If rest of col above diag also zero, then okay

(because want lower triangular)

$$\begin{vmatrix} a'_{11} & a'_{12} & a'_{13} & 0 \\ a'_{21} & a'_{22} & a'_{23} & 0 \\ a'_{31} & a'_{32} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

If non-zero entry above diagonal, then swap rows

(which flips sign of determinant)

$$\begin{vmatrix} a'_{31} & a'_{32} & 0 & 0 \\ a'_{21} & a'_{22} & a'_{23} & 0 \\ a'_{11} & a'_{12} & a'_{13} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

Determinant of any Matrix

Have shown there is at most one function that satisfies all three axioms

- det for diagonal matrices forced by axioms
- det for triangular matrices forced by diag matrices and axioms
- det for arbitrary matrices forced by triangular matrices and axioms

Can also carefully check that det as defined actually satisfies all axioms

Have given procedure for computing determinant

Invertibility

- Claim: A singular if and only if $|A| = 0$

Pf: If A singular

rows of A are dependent

some row i of A lin. comb. of other rows

subtract those multiples of those rows from row i to get zero row

$|A| = 0$

Invertibility

- Claim: A singular if and only if $|A| = 0$

Pf: If $|A| = 0$

Do row operations to transform A into lower triangular L

since $|A| = 0$, L must have zero row

some row of A lin. comb. of other rows

rows of A are dependent

A singular

Product Rule

- Claim: $|AB| = |A||B|$

Pf: Easy case, if $|B| = 0$

Then B not invertible

So AB not invertible

$|AB| = 0$

Product Rule

- Claim: $|AB| = |A||B|$

Pf: interesting case, if $|B| \neq 0$

Define function $f(A) = \frac{|AB|}{|B|}$, holding B fixed

Want to show $f(A) = |A|$

Suffices to show f satisfies the three axioms

(determinant is unique)

Product Rule

- Define function $f(A) = \frac{|AB|}{|B|}$

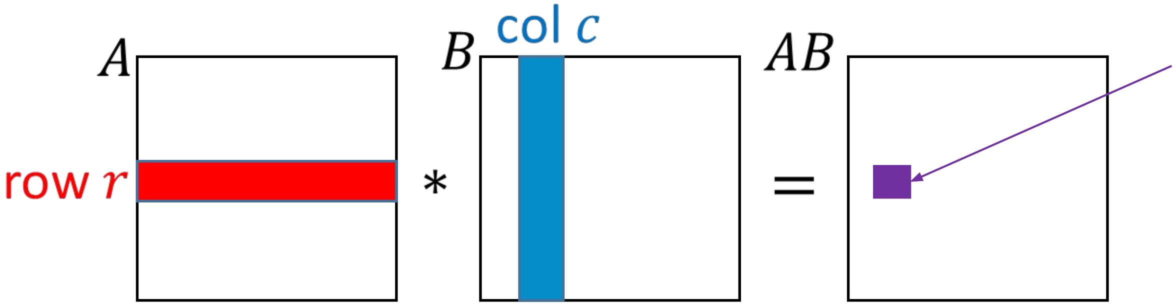
Property 1: $f(I) = \frac{|IB|}{|B|} = \frac{|B|}{|B|} = 1$

Product Rule

- Define function $f(A) = \frac{|AB|}{|B|}$

Property 2: Swap rows i and j of A to produce A'

Then $A'B$ obtained from AB by swapping rows i and j



Entry in row r and col c of AB is dot product of row r of A and col c of B

$$f(A') = \frac{|A'B|}{|B|} = -\frac{|AB|}{|B|} = -f(A)$$

Product Rule

- Define function $f(A) = \frac{|AB|}{|B|}$

Property 3:

Multiply row i of A by α to get A'

multiplies row i of AB by α

$$f(A') = \frac{|A'B|}{|B|} = \alpha \frac{|AB|}{|B|} = \alpha f(A)$$

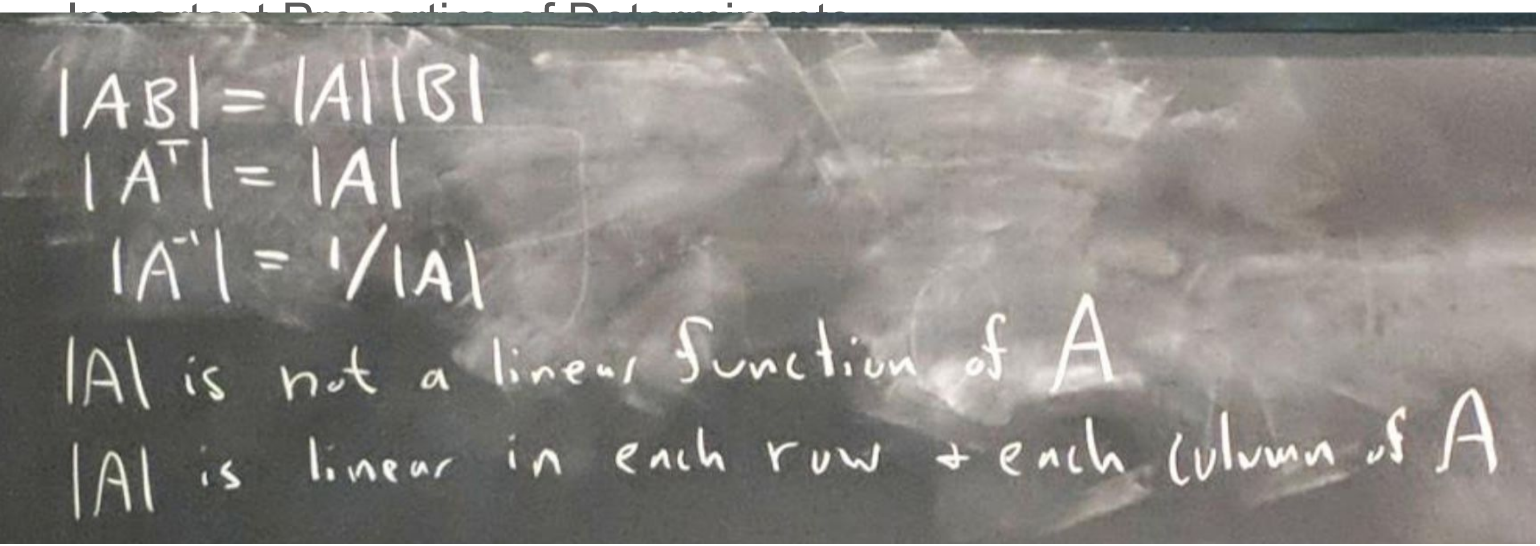
If have two matrices C, C' identical outside row i

$A = \mathrm{C} + $ row i of C'

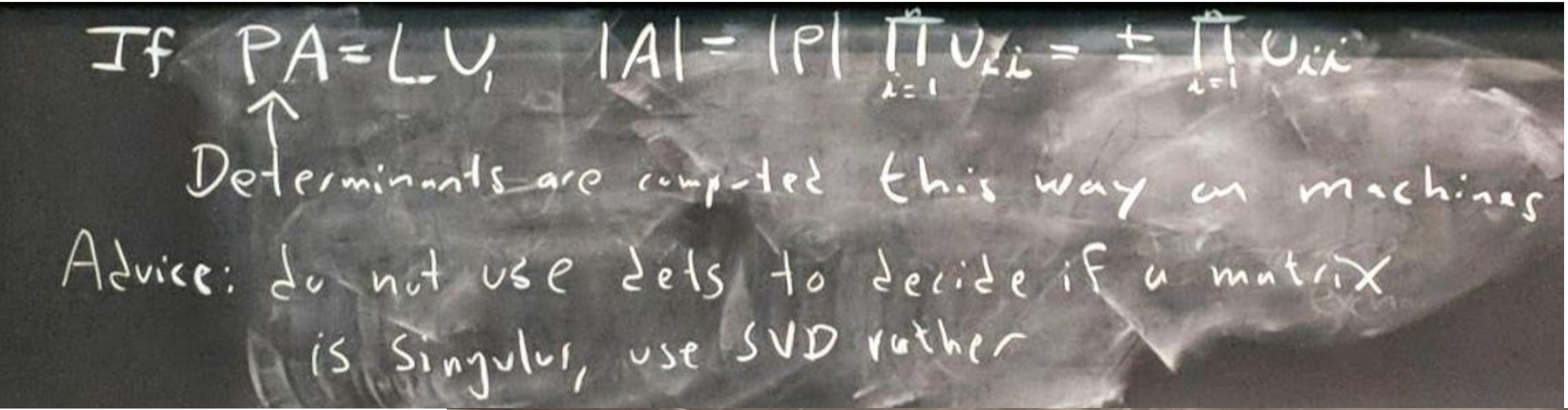
$$f(A) = f(C) + f(C')$$

21.Applications of Determinants

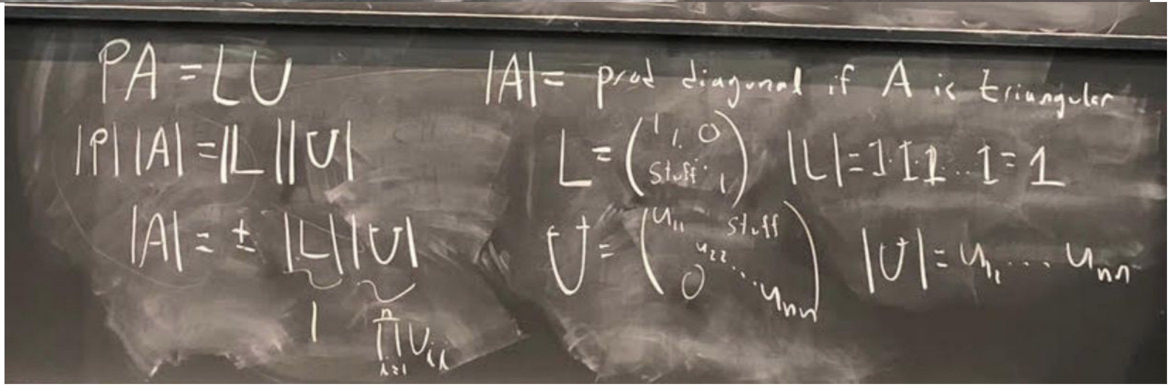
Important Properties of Determinants



How determinants are computed on machines



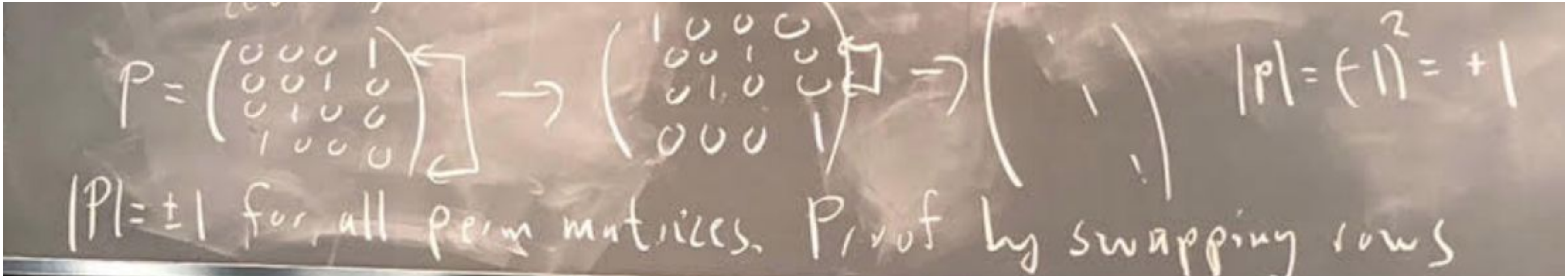
Detail:

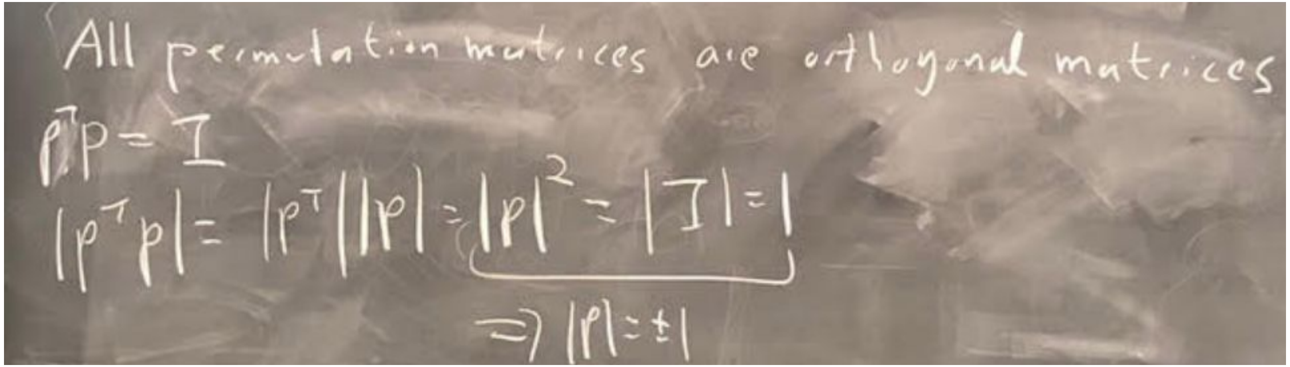


We don’t ask on a computer if a matrix is singular, we ask is it nearly singular enough to be treated as singular.

Note: We are using the capital Pi Notation

Permutation Matrices: Every row & col has a 1 and (n-1) 0's. The det is ±1. Proofs: swapping rows & orthogonality.





Application: the [15] or 16 puzzle

Encoding

Number the empty space with 16.

We encode the “state” of the puzzle with P_{ij}=1if tile i sits in slot j. The solved puzzle corresponds to I_{16}. Notice the bottom row of P indicates the location of the space.

A puzzle move is an exchange of tile 16 (the space) with a numbered tile. In the encoding, we exchange the bottom row of P with another row of P. This negates the determinant of P.

Any sequence that starts and ends with a solved puzzle has an even number of moves, because the #leftward moves = # rightward, #upward = # leftward. If somewhere in the middle, two physical tiles are exchanged, the determinant is out of parity and can not be restored.

Note: this argument does not apply if you exchange the space with a physical tile because the determinant would flip, but now an odd number of space moves would happen.

Putting it all together.

Suppose we make a sequence of moves numbered k=0,1,2,...,N starting with a solved puzzle at k=0 with the space in position (1,1), say. When k=N, the puzzle could be solved again, but at the very least we assume that the space is back in (1,1).

Let d(k) be the determinant of the permutation matrix that enclodes the state of the puzzle at step k. Let (i(k),j(k)) be the position of the space at step k.

Claim: $d(k) = (-1)^{i(k)+j(k)}$, d(0)=d(N)=1. Proof: When k=0, P=I so d(0)=1.
When we make a move from k to k+1, d(k) changes sign as we are switching another row with the bottom row of the matrix. Also $(-1)^{i(k)+j(k)}$ changes sign as well since we are either adding or subtracting one from either i or j with every move.

What if sneaky little sister or brother switches two physical pieces?

If at any point d(k) and $(-1)^{i(k)+j(k)}$ are unequal, but you say, I'm going to solve the puzzle anyway, you'll find you can't do it.

Why? If you have solved the puzzle at step N, then d(N) must be 1, as the permutation matrix is the identity. However with the switched piece, $(-1)^{i(n)+j(n)}$ is -1 as it is now always the opposite sign. … but then the space can't be in the (1,1) position because (i(n),j(n)) is exactly the location of the space.

What do we mean by the image of a shape under a transformation such as $x \rightarrow Ax$?

If one has a set S in R^n, one can talk about the image of S under a transformation. As a set this is {Ax: x in S}. It means apply A to every point in S.

If it feels too abstract, consider an example. Let S be the black nose of the corgi in the photo in the upper left of the lecture slide on transformations. That slide has four images of that "black nose" under A. Some are rotated, some are reflected, some are sheared, but they are all images of the black nose under A.

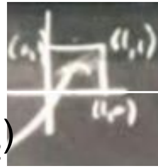
One often likes to consider the unit sphere or ball as a special set. This is the set of unit vectors. Then the image is {Ax: ||x||=1 } which is always some kind of ellipsoid like object.

Volumes

Note: 2d volume is called area.

If A is 2x2, a parallelogram may be described with linear algebra:
(Compare the definition of a subspace)
It says that a parallelogram is the image of the unit square.

$$\left\{ A\begin{pmatrix}c_1\\ c_2\end{pmatrix} : \begin{matrix}0\le c_1\le 1\\ 0\le c_2\le 1\end{matrix}\right\}$$



(c_1,c_2)

unit square = has area 1
$= \left\{ \begin{pmatrix}c_1\\ c_2\end{pmatrix} : \begin{matrix}0\le c_1\le 1\\ 0\le c_2\le 1\end{matrix}\right\}$

Suppose the columns of A are a1 and a2: $\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}$

The set of four vertices is $\{0,a_1,a_2,a_1+a_2\}$ which may be written:

Vertices: $\left\{ A\begin{pmatrix}c_1\\ c_2\end{pmatrix} : \begin{matrix}c_1\in\{0,1\}\\ c_2\in\{0,1\}\end{matrix}\right\}$

We emphasize that $\begin{matrix}0\le c_1\le 1\\ 0\le c_2\le 1\end{matrix}$ describes a filled in square while $c_1\in\{0,1\}$, $c_2\in\{0,1\}$ describes only the vertices.

The result is |A| is the signed area of the parallelogram. Note that $|cA|=c^2|A|$ confirms the scaling of area.

If A is diagonal we get length times width for the area of a rectangle lined up with the coordinate axes.

Parallelopiped In n dimensions

An n-dimensional parallelopiped may be described with a matrix A with columns

$a_1, a_2, \ldots, a_n \in R^n$. The filled region can be described by the following set of points:

$\{Ac : 0. \leq c. \leq 1\}$. (The notation means $c \in R^n$ and each component can go from 0 to 1.}

There are 2n vertices consisting of all possible sums of columns including the empty sum:

$\{Ac : c_i \in \{0,1\}$ for $i=1:n\}$

The signed volume is $|A|$. Note $|cA| = c^n|A|$ conforming to our intuition for scaling in 1,2,and 3 dimensions.

Triangles

Suppose the columns of A are a_1 and a_2: 

The triangle with vertices 0,a_1 and a_2 may also be described as a set with linear algebra notation:

$\{Ac : 0.\le c.\le 1,\ \Sigma c_i=1\ \}$

The area is ½|A|.

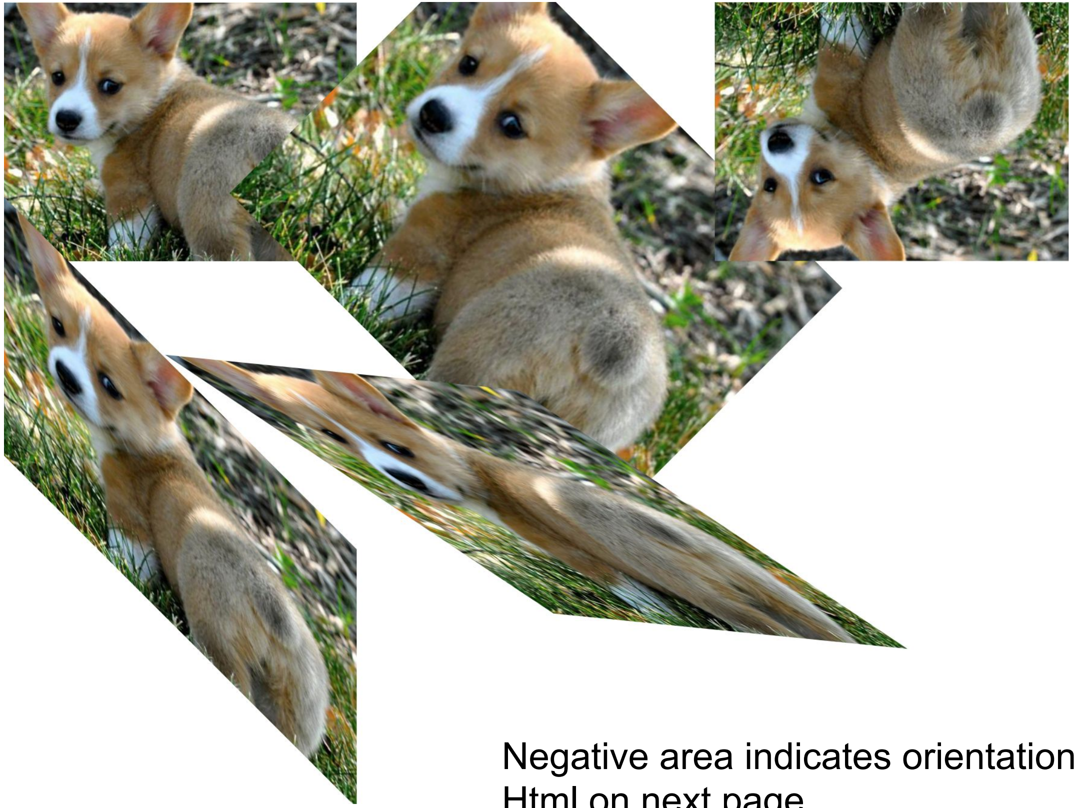
Simplices

In general you can take the n columns of A (nxn) and 0 as vertices and get the volume of a tetrahedron or simplex. One takes (1/n!)|A|.

Summary of volumes

1. Signed volume of a parallelopiped (parallelogram when n=2)
2. Signed volume of a simplex (triangle when n=2, tetrahedron when n=3)
3. det(A) = scale factor for the linear transformation $x \rightarrow Ax$
4. det(J) = local scale factor for a nonlinear transformation (J is the Jacobian)

Example of point 3 on web browser with fun html photo effects



If the area of the black corgi nose image is d in the first picture what is it in every other picture?
(answer: d * det(A))

If the area of the white region is d, what is it in every other picture?
(answer: d * det(A))

The matrices A in the five images are
[1 0;0 1] , [1 1;-1 1], [1 0; 0 -1],
[1 1;0 1], [1,1,.1,1.5]

Negative area indicates orientation (e.g. right and left) are reversed
Html on next page

You can paste the following html code in a file and open in your local browser.

Perhaps call it CorgiTransforms.html

And use File → Open File on your browser

```
<img src='https://i.barkpost.com/wp-content/uploads/2015/01/corgi2.jpg' >

<img src='https://i.barkpost.com/wp-content/uploads/2015/01/corgi2.jpg'
    style='transform: matrix(1,1,-1,1,  0, 200)' >

<img src='https://i.barkpost.com/wp-content/uploads/2015/01/corgi2.jpg'
    style='transform: matrix(1,0,0,-1,  0, 0)' > <br>

<img src='https://i.barkpost.com/wp-content/uploads/2015/01/corgi2.jpg'
    style='transform: matrix(1,1,0, 1,  0, 200)' >

<img src='https://i.barkpost.com/wp-content/uploads/2015/01/corgi2.jpg'
    style='transform: matrix(1,.1,1.5, 1,  0, 200)' > <br>
```

Transform(x) = Ax + b
Four Linear Transform parameters in A
Two Translation paramaters in b

HTML format:
matrix(a11,a12,a21,a22,b1,b2)

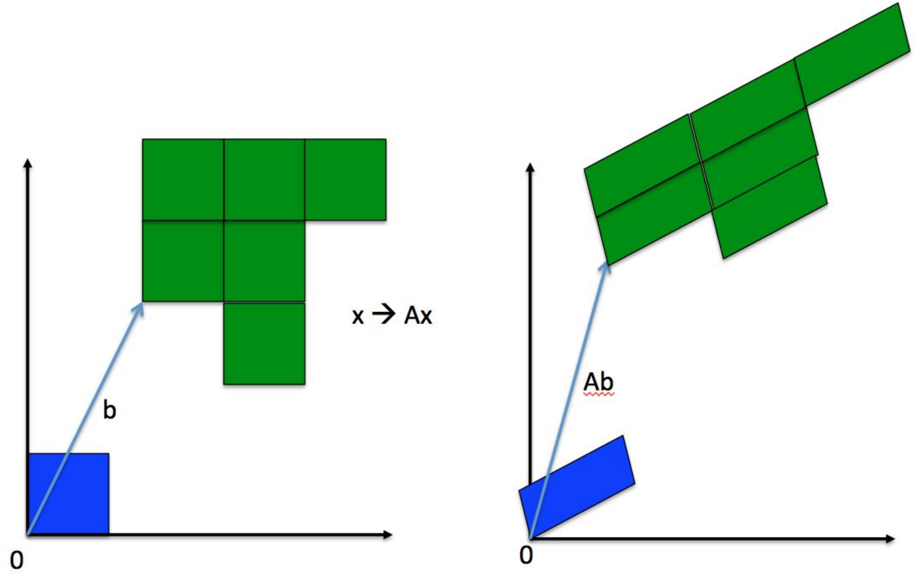
Documentation

Try your own matrices!

… but why is Det A the scaling factor for all volumes (areas?)

- You might say that you understand that the unit cube (with a vertex at the origin) becomes a parallelopiped (with a vertex at the origin) but what about a corgi nose becoming a distorted corgi nose. Why does that scale by Det A?
- Well the first fact is that any unit cube (does not have to be at the origin) has volume scaling by Det A, and then we do the “calculus thing” by pretending that every volume is the limit of lots of little cubes, all scaling by Det A.
- The fact is if you translate a cube by adding b to every point in the cube, and then apply A, linearity tells us, the set is the same as if you applied A to the unit cube with vertex at the origin, and then translating the resulting parallelopiped by Ab.

Diagram illustrating previous slide

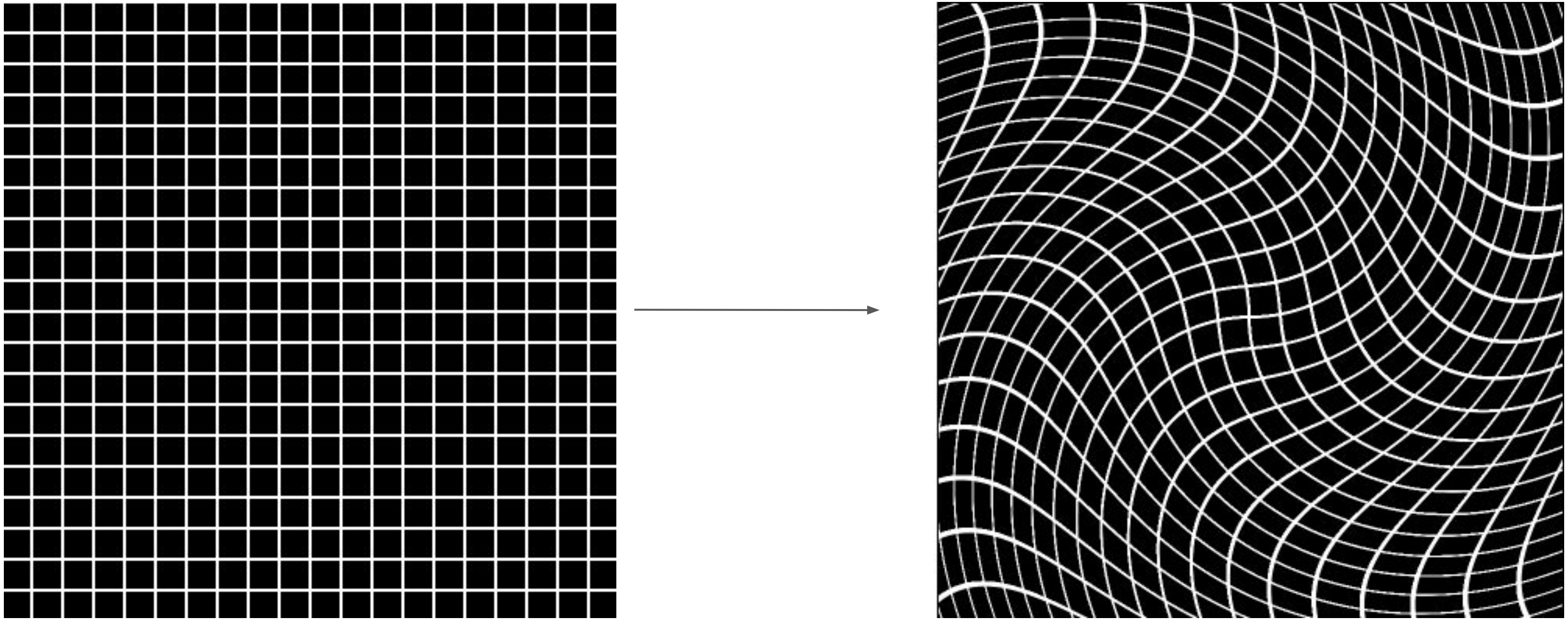


Congruent squares are mapped to congruent parallelograms
Scaled shapes map to scaled shapes

Nonlinear transformations

In a nonlinear transformation, little squares
Become little parallelograms in a variety
Of ways. The Jacobian describes the
Parallelogram at that point, and its determinant is
the local scaling of area.

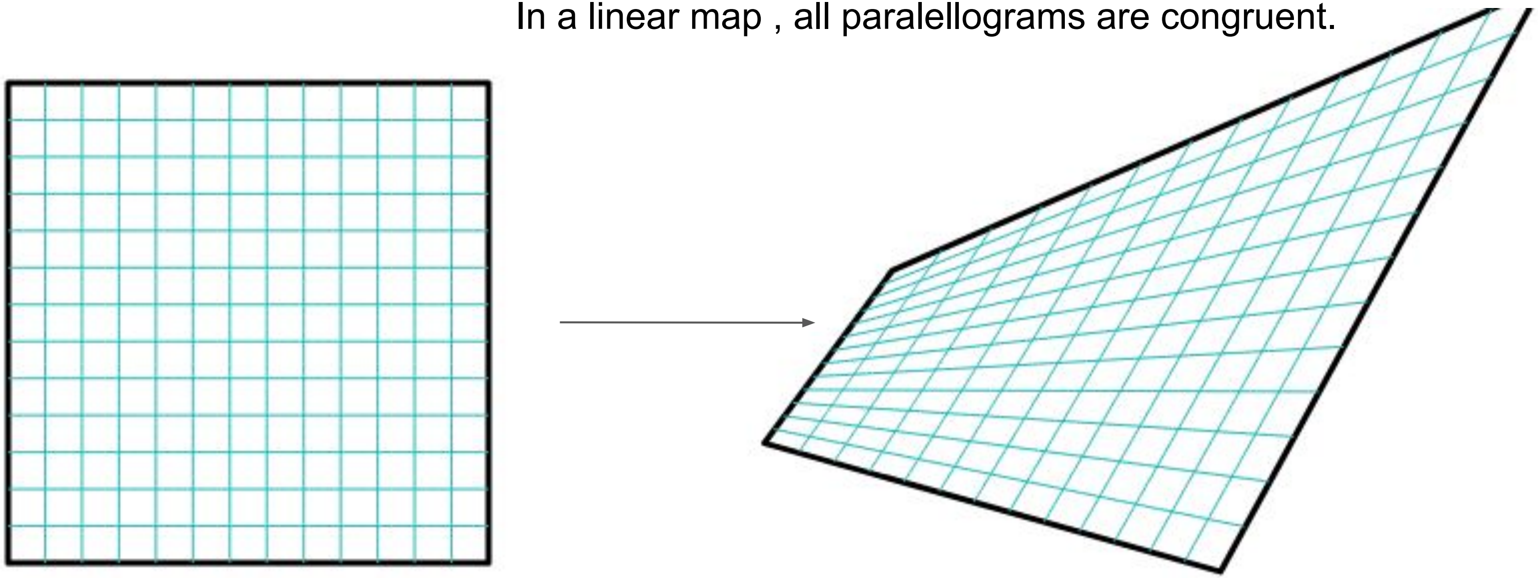
Nice slider you can play with



Another nonlinear map

Interactive Demo

If a square does not map into a parellogram, the map is Nonlinear. We see many different sized paralleograms. In a linear map , all paralellograms are congruent.

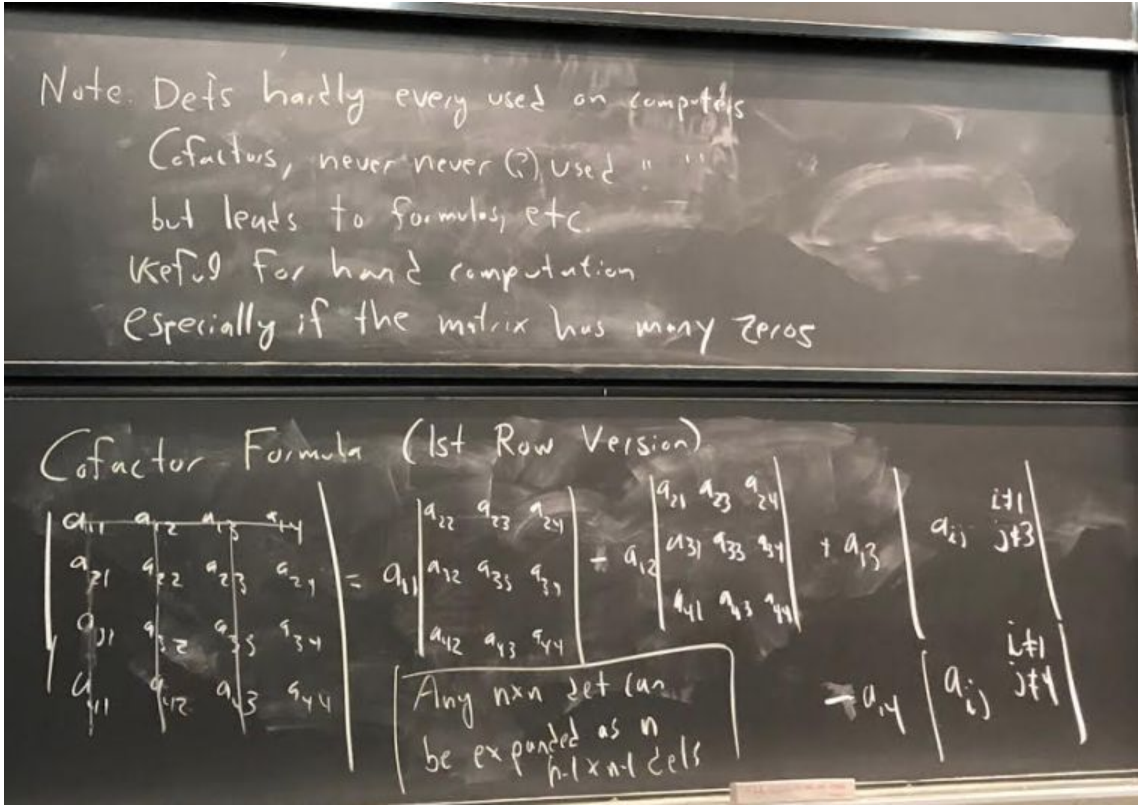


Nonlinear Transformations

If we nonlinearly transform the corgi, as when Phillip sits down and curls up, the area does not scale uniformly. Thus we need a local linear transformation to describe how boxes become parallelograms. This is what the jacobian does. It's determinant gives a local scaling of area.

(Need to draw a good nonlinear transformation of a corgi. Not sure html can do this? Anybody know?)

24. Cofactors



Cofactors

Given nxn A, let $M_{ij} \in R$ be the (n-1) by (n-1) determinant of A with row i and column j deleted. Let $C_{ij} = M_{ij}(-1)^{i+j}$

We can compute determinants and inverses using cofactors (but not advised for large matrices on a computer). A formula to be proved soon is $A^{-1} = C^T/\det(A)$

```
# need the following packages
using InvertedIndices, LinearAlgebra
A = rand(3,3)
M = [ det( A[Not(i),Not(j)] ) for i=1:3,j=1:3 ]
C = [ (-1)^(i+j) * det( A[Not(i),Not(j)] ) for i=1:3,j=1:3 ]
```

```
3×3 Array{Float64,2}:
  0.330627   -0.376808    0.177844
 -0.0945505   0.703736   -0.343431
 -0.0113083  -0.0324163   0.216193
```

```
inv(A) ≈ C'/det(A)
```

```
true
```

In Julia, the InvertedIndices package Lets us write A[Not(i),Not(j)] for the Matrix A with row i and col j deleted

Let’s see some cofactor matrices symbolically

```
using SymPy,LinearAlgebra
n=2
A = [ symbols("a_$i$j",real=true) for i=1:n,j=1:n]
display(A)
C = [ (-1)^(i+j) * det( A[Not(i),Not(j)] ) for i=1:n,j=1:n ]
```

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

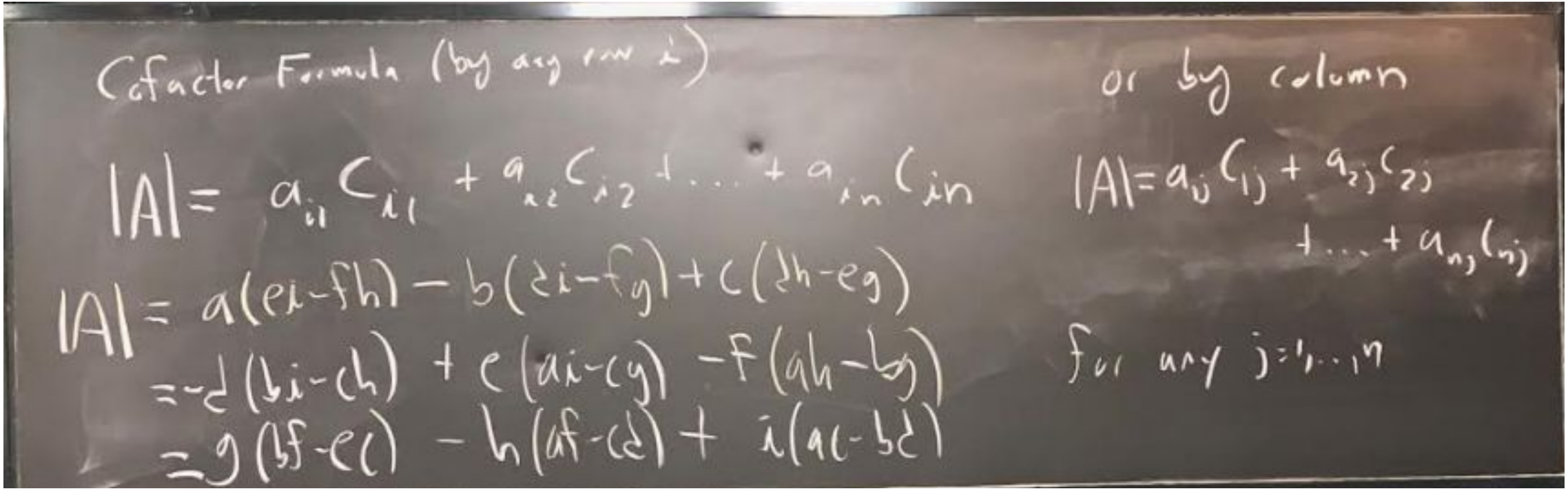
$$\begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix}$$

```
n=3
A = [ symbols("a_$i$j",real=true) for i=1:n,j=1:n]
display(A)
C = [ (-1)^(i+j) * det( A[Not(i),Not(j)] ) for i=1:n,j=1:n ]
```

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\begin{bmatrix} a_{22}a_{33} - a_{23}a_{32} & -a_{21}a_{33} + a_{23}a_{31} & a_{21}a_{32} - a_{22}a_{31} \\ -a_{12}a_{33} + a_{13}a_{32} & a_{11}a_{33} - a_{13}a_{31} & -a_{11}a_{32} + a_{12}a_{31} \\ a_{12}a_{23} - a_{13}a_{22} & -a_{11}a_{23} + a_{13}a_{21} & a_{11}a_{22} - a_{12}a_{21} \end{bmatrix}$$

Cofactor Formula (for any row or column)



Let's check the six cofactor det formulas for n=3

These six symbolic formulas are the same

```
A[1,:]·C[1,:]
```

$a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(-a_{21}a_{33} + a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$

```
A[2,:]·C[2,:]
```

$a_{21}(-a_{12}a_{33} + a_{13}a_{32}) + a_{22}(a_{11}a_{33} - a_{13}a_{31}) + a_{23}(-a_{11}a_{32} + a_{12}a_{31})$

```
A[3,:]·C[3,:]
```

$a_{31}(a_{12}a_{23} - a_{13}a_{22}) + a_{32}(-a_{11}a_{23} + a_{13}a_{21}) + a_{33}(a_{11}a_{22} - a_{12}a_{21})$

```
A[:,1]·C[:,1]
```

$a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{21}(-a_{12}a_{33} + a_{13}a_{32}) + a_{31}(a_{12}a_{23} - a_{13}a_{22})$

```
A[:,2]·C[:,2]
```

$a_{12}(-a_{21}a_{33} + a_{23}a_{31}) + a_{22}(a_{11}a_{33} - a_{13}a_{31}) + a_{32}(-a_{11}a_{23} + a_{13}a_{21})$

```
A[:,3]·C[:,3]
```

$a_{13}(a_{21}a_{32} - a_{22}a_{31}) + a_{23}(-a_{11}a_{32} + a_{12}a_{31}) + a_{33}(a_{11}a_{22} - a_{12}a_{21})$

A numerical example

```
A = rand(3,3)
C = [ (-1)^(i+j) * det( A[Not(i),Not(j)] ) for i=1:n,j=1:n ]
```

```
3×3 Array{Float64,2}:
  0.28262    -0.458966   0.33968
 -0.0999143   0.440471  -0.353387
 -0.30642     0.442968  -0.153385
```

```
A[1,:]·C[1,:]
```

```
0.11529967670281088
```

```
A[2,:]·C[2,:]
```

```
0.115299676702810
```

```
A[3,:]·C[3,:]
```

```
0.11529967670281084
```

```
A[:,1]·C[:,1]
```

```
0.115299676702810
```

```
A[:,2]·C[:,2]
```

```
0.11529967670281083
```

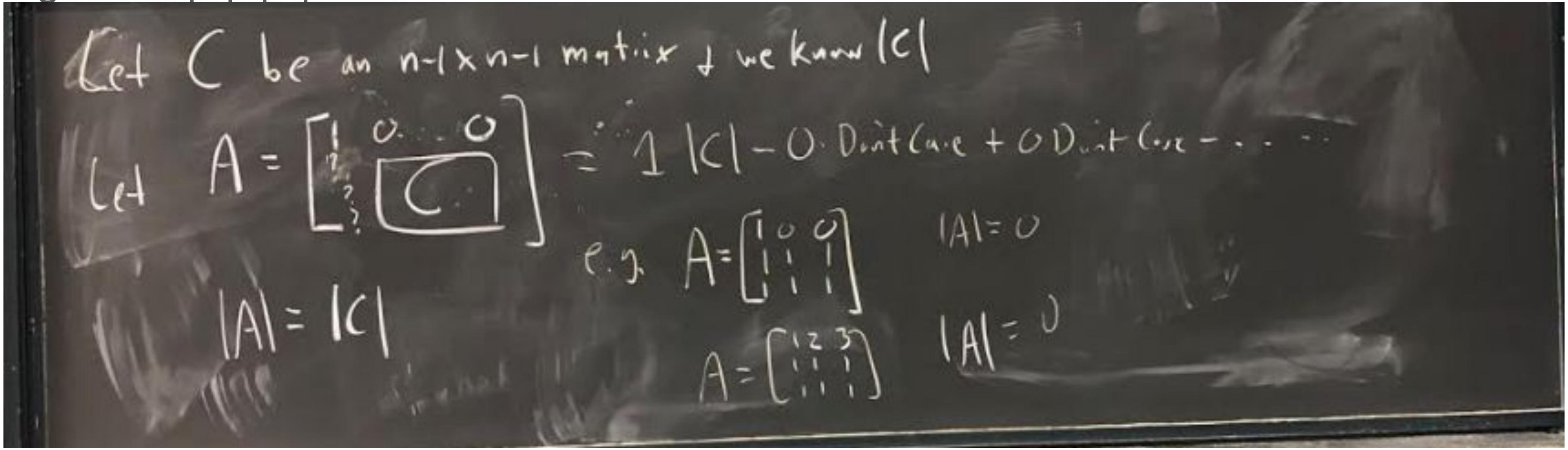
```
A[:,3]·C[:,3]
```

```
0.115299676702810
```

Example Use

Suppose C is an n-1 x n-1 matrix and we know |C|

We can augment to an n x n A with A_{11}=1 and Don't Cares below the 1. We will get that |A|=|C|



Same idea numerically

```
C = rand(1:9,3,3)
```

```
3×3 Array{Int64,2}:
 1  8  2
 5  5  4
 9  2  8
```

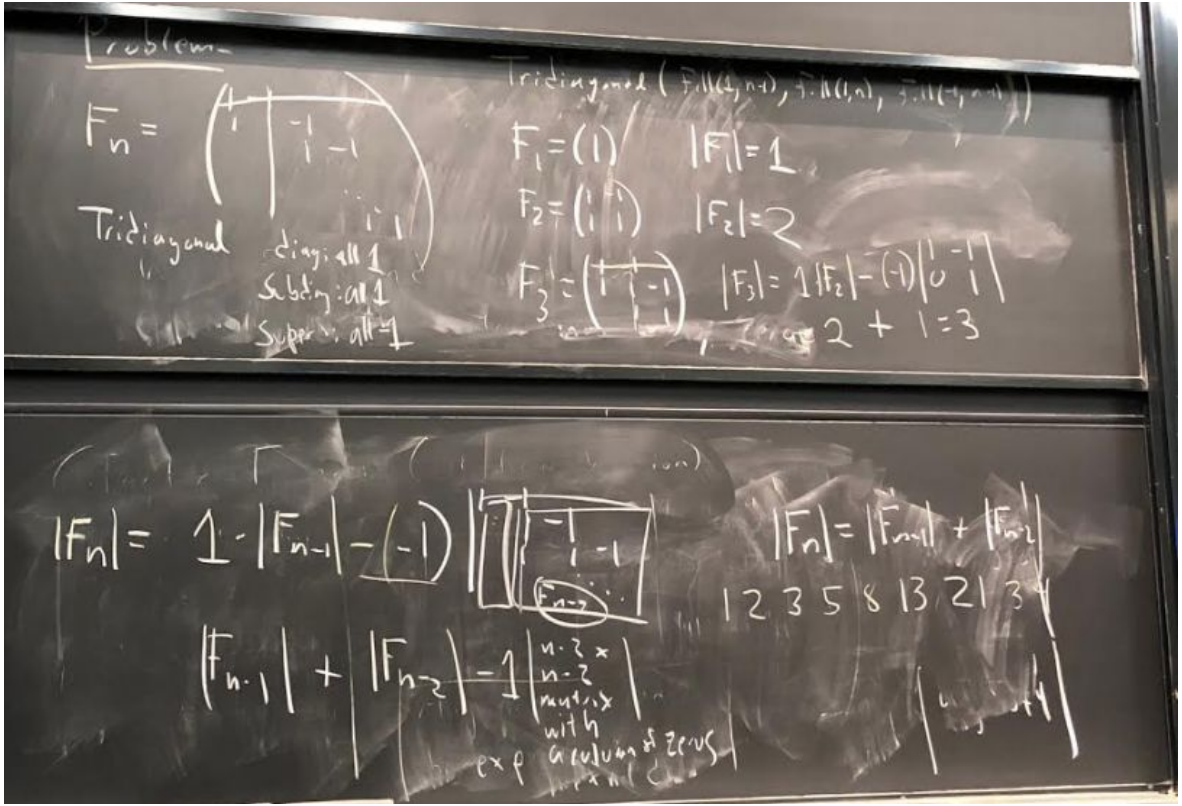
```
A = [ 1 0 0 0;rand(1:9,3) C]
```

```
4×4 Array{Int64,2}:
 1  0  0  0
 4  1  8  2
 2  5  5  4
 3  9  2  8
```

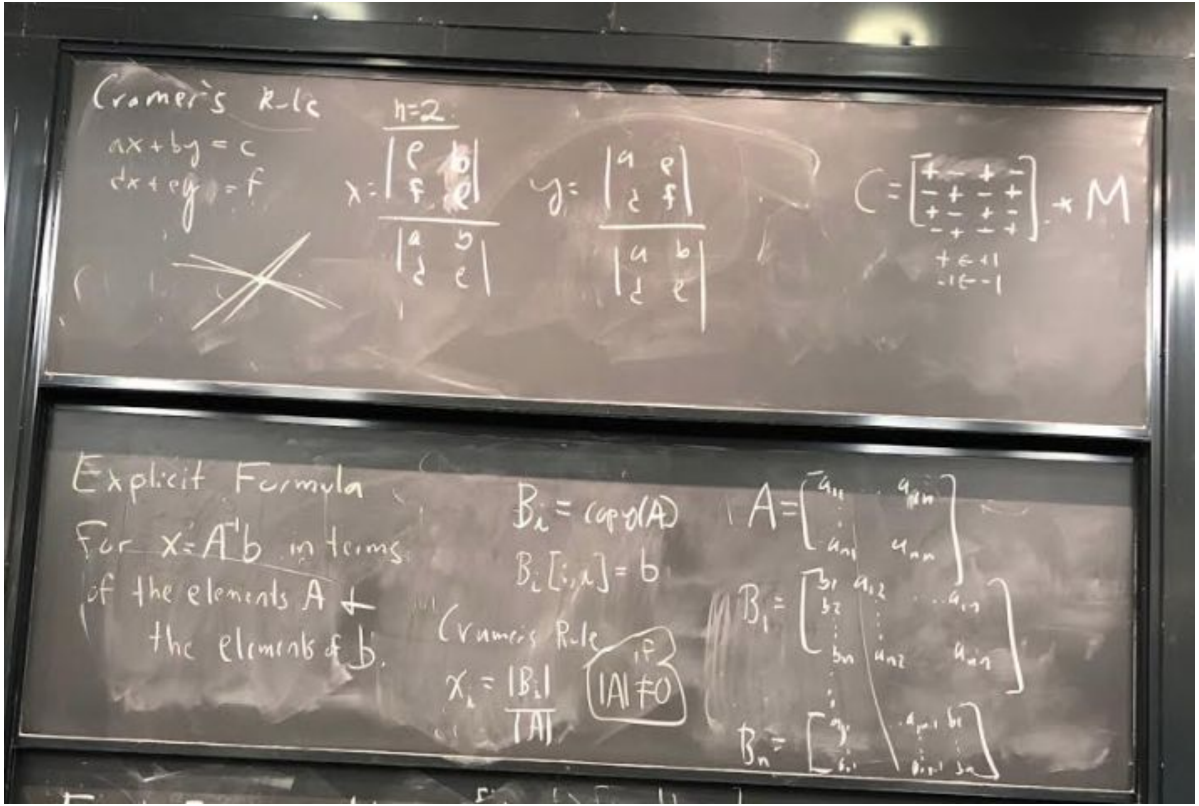
```
det(C), det(A)
```

```
(-70.0, -70.0)
```

Example Application: Fibonnaci



Cramer’s Rule for Ax=b



A few of you may have seen [Cramer’s rule for small Systems](#) in high school but It seems it is not that common.

The rule is that you create B_i by taking A and replacing column i with b.

Then $x_i = |B_i|/|A|$

Cramer's Rule numerically

```
A = rand(1:9,3,3)
```

```
3×3 Array{Int64,2}:
 8  7  1
 6  1  9
 2  8  3
```

```
b = rand(1:9,3)
```

```
3-element Array{Int64,1}:
 3
 5
 9
```

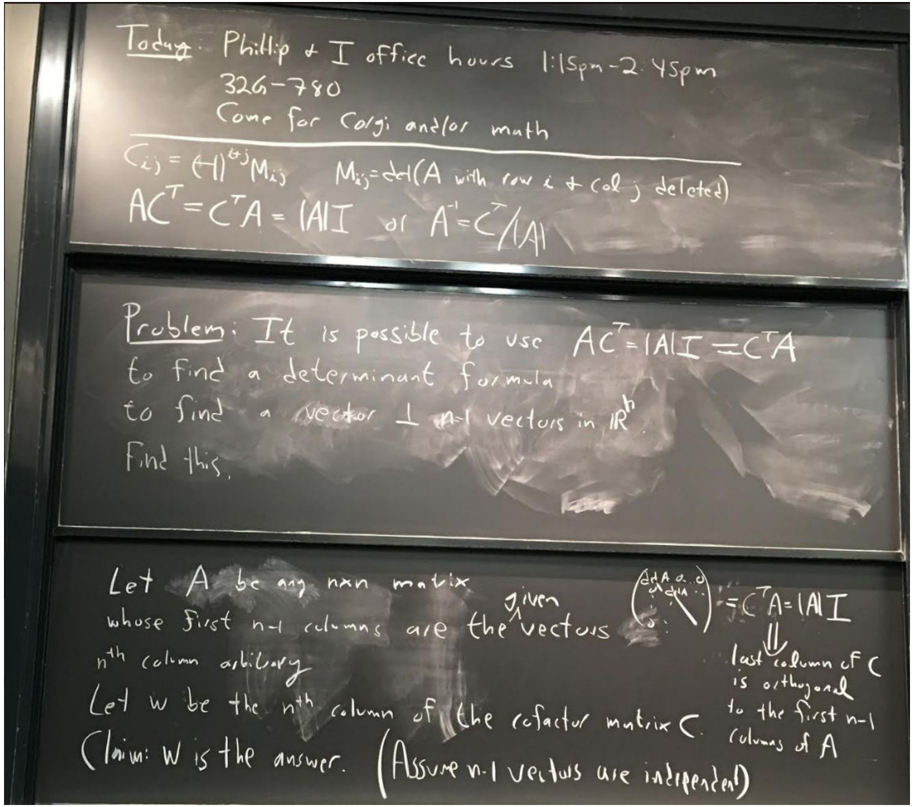
```
x = A\b # solution using built in function
```

```
3-element Array{Float64,1}:
 -0.5652173913043478
  0.9565217391304347
  0.8260869565217391
```

```
for i=1:3
  B = copy(A)
  B[:,i] = b
  println( det(B)/det(A))
end
```

```
-0.5652173913043479
0.9565217391304349
0.8260869565217391
```

25. More applications and some proofs



What is W explicitly?

$$W_1 = M_{1,n} = \begin{vmatrix} a_{21} & \cdots & a_{2,n-1} \\ \vdots & & \vdots \\ a_{n1} & & a_{n,n-1} \end{vmatrix}$$

$$W_2 = -M_{2,3} = -\begin{vmatrix} a_{11} & a_{1,n-1} \\ a_{31} & a_{3,n-1} \\ a_{41} & a_{4,n-1} \\ \vdots & \\ a_{n1} & a_{n,n-1} \end{vmatrix}$$

$$W_3 = M_{3,3}$$

$\vdots$

$(-1)^{i+n}$ A with i^{th} deleted & n^{th} column deleted

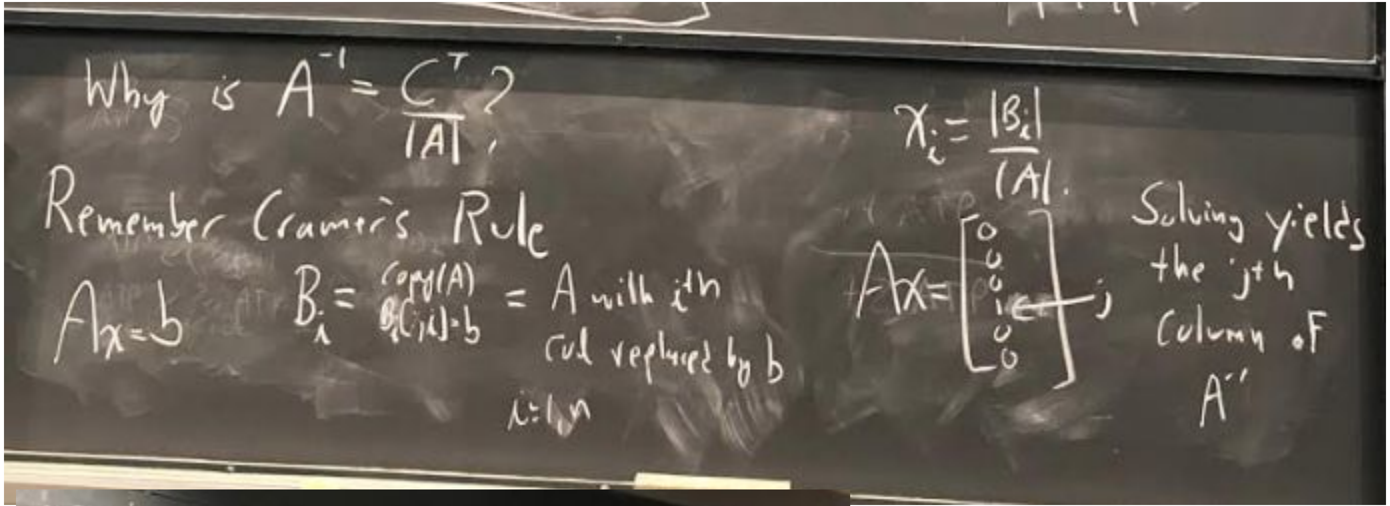
$$\begin{vmatrix} \cancel{a_1} & \cancel{b_1} & ? \\ a_2 & b_2 & ? \\ a_3 & b_3 & ? \end{vmatrix}$$

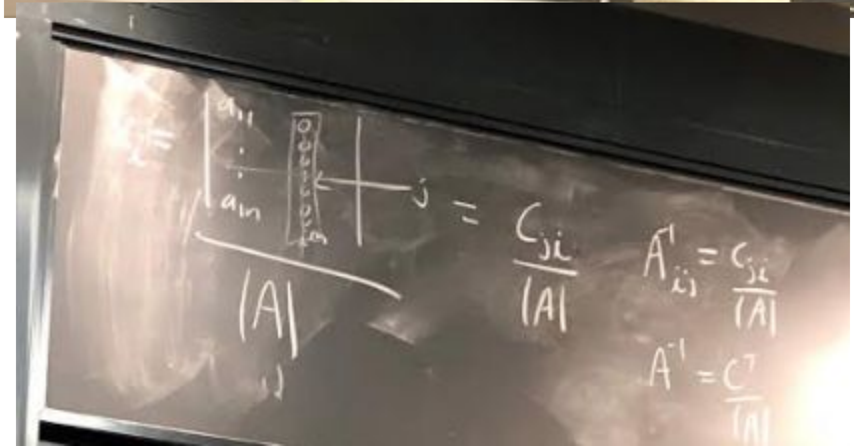
$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\begin{vmatrix} \cancel{a_1} & \cancel{b_1} & \cancel{x} \\ \cancel{a_2} & \cancel{b_2} & \cancel{x} \\ \cancel{a_3} & \cancel{b_3} & \cancel{x} \end{vmatrix}$$

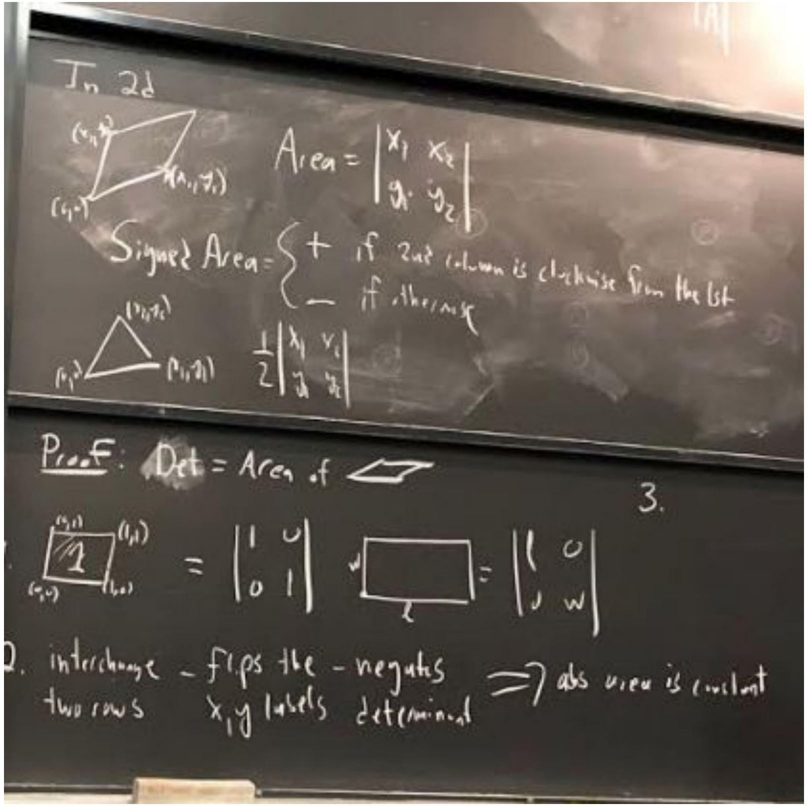
$$W = \begin{pmatrix} \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix} \\ -\begin{vmatrix} a_1 & b_1 \\ a_3 & b_3 \end{vmatrix} \\ \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \end{pmatrix}$$

Derivation of cofactor formula for A inverse

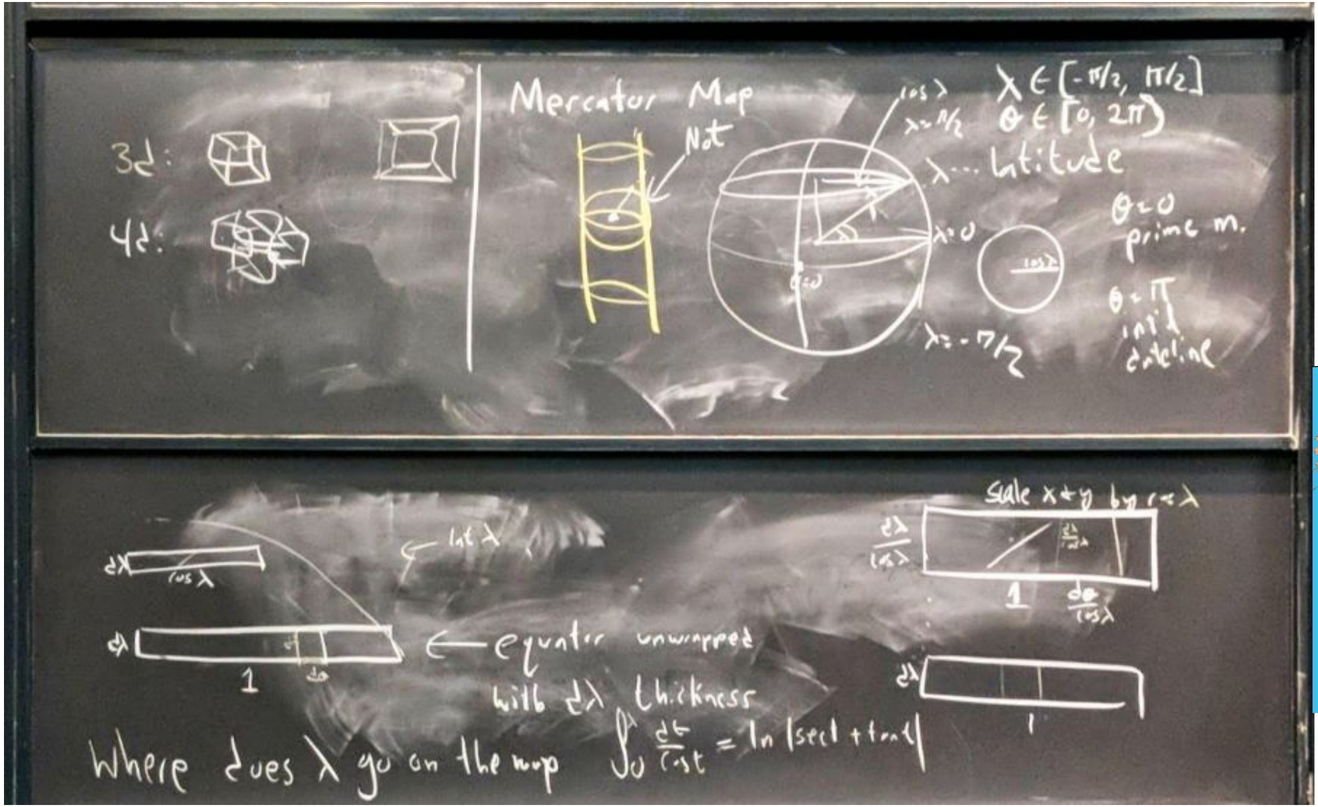




Proof of Parallelogram (Parallelopiped) Formulas



26. Applications to Maps and Jacobians



The mercator map is
In every school in the
World. Here is the
mathematics.

Jacobians

If we have a map from R^n to R^n

The jacobian matrix is the “derivative”.

Sometimes people talk about the derivative of a vector with respect to a vector.

Jacobians: the 3d example generalizes to any number of dimensions. It is the parallelopiped formula locally.

Slide source

3D Jacobian

$$x = x(u,v,w), y = y(u,v,w), z = z(u,v,w)$$

- maps volumes (consisting of small cubes of volume $\mathrm{d}x\mathrm{d}y\mathrm{d}z$
- ....to small cubes of volume $\mathrm{d}u\mathrm{d}v\mathrm{d}w$

$$\int\int\int_V f(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z = \int\int\int_{V'} F(u,v,w)\left|\frac{\partial(x,y,z)}{\partial(u,v,w)}\right|\mathrm{d}u\mathrm{d}v\mathrm{d}w$$

- Where

$$\frac{\partial(x,y,z)}{\partial(u,v,w)} = \begin{vmatrix} x_u & x_v & x_w \\ y_u & y_v & y_w \\ z_u & z_v & z_w \end{vmatrix}$$

f maps R^n to R^n

$y = f(x)$

One can think of an input box consisting of

$x + dx_1\ e_1$, $x + dx_2\ e_2$, …, $x + dx_n\ e_n$

Where the dx_i are “little” scalar perturbations and the e_i are the coord directions

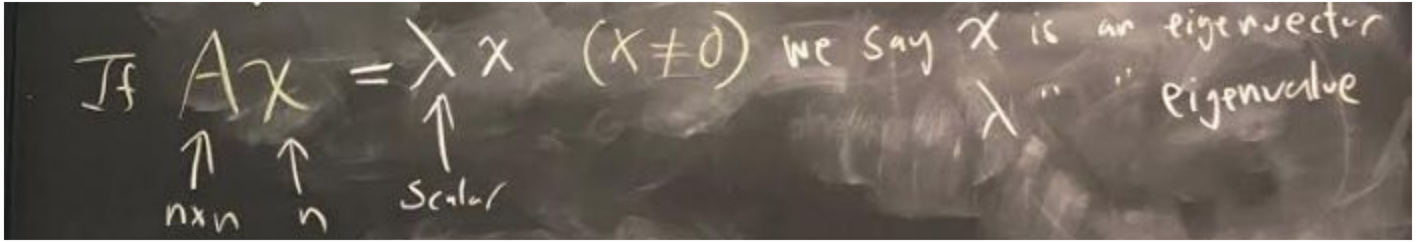
The output box is

$y + dx_1(\partial y_1/\partial x_1\ \ \partial y_2/\partial x_{1,}\ ,,,\ \partial y_n/\partial x_1)^T + ... + dx_n(\partial y_1/\partial x_n\ \ \partial y_2/\partial x_{n,}\ ,,,\ \partial y_n/\partial x_n)^T$

Thus the volume of the input box is $dx_1 dx_{2...} dx_n$

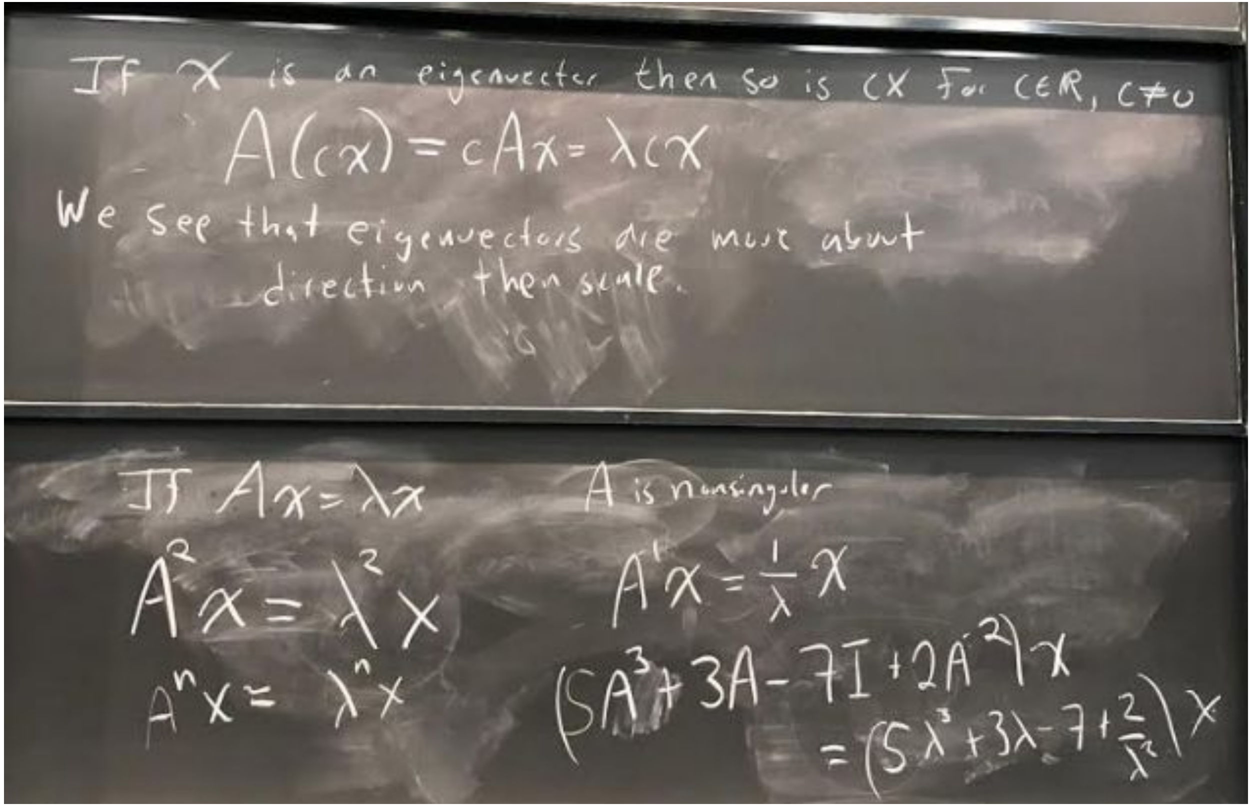
And the volume of the output parallelopiped is $\det(\ \partial y_i/\partial x_j)\ \ dx_1 dx_{2...} dx_n$

27. Eigenvalues & Eigenvectors

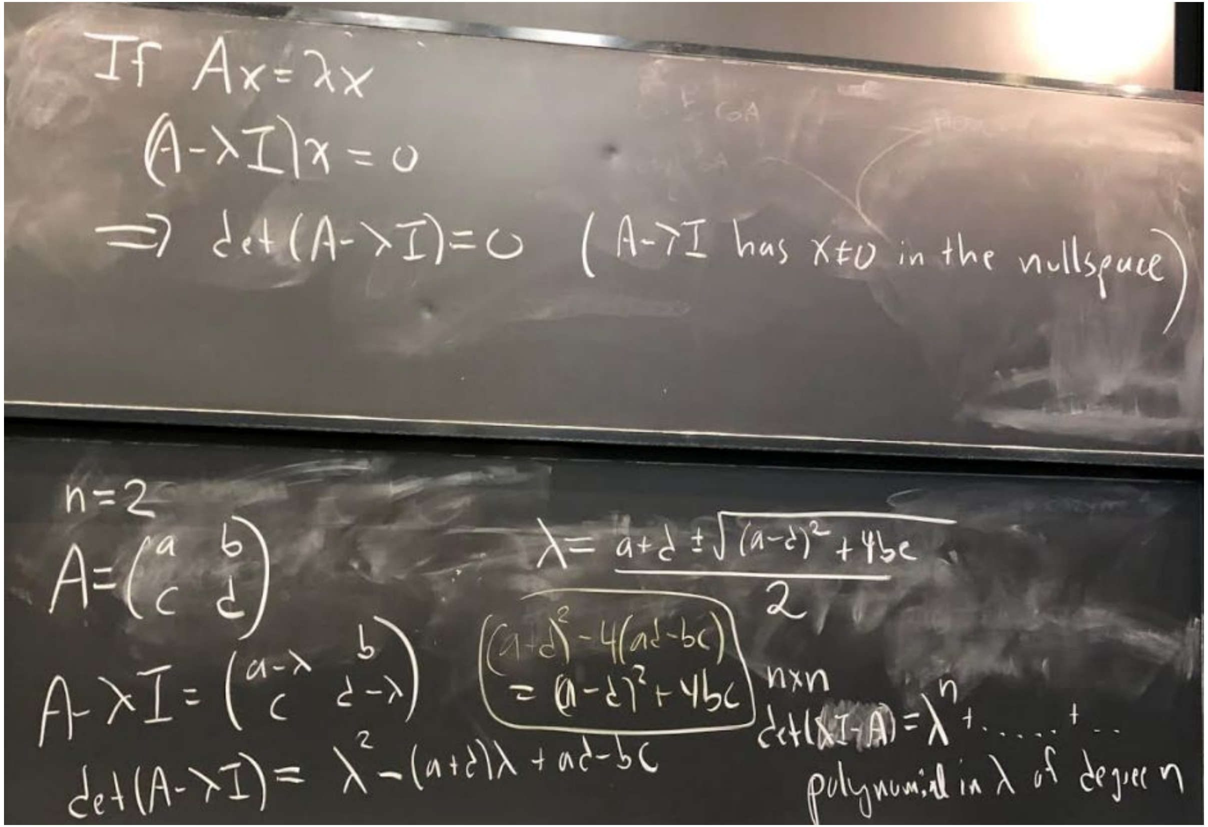


Nice intro video online: https://www.youtube.com/watch?v=PFDu9oVAE-g

Simple eigen properties - Eigenvectors are more about direction than the scaling



Eigenvalues are solutions to det(A-λI)=0.



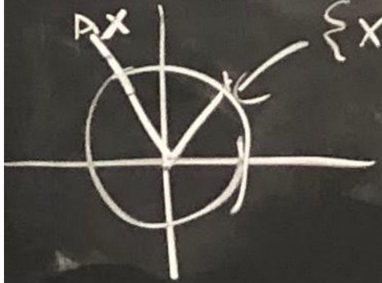
The irony of computation

Eigenvalues are always taught as solutions to a polynomial equation, so many students and even mathematicians believe that this is how eigenvalues are computed on machines.

In fact, it is often the other way around. Modern polynomial solvers are often built on top of eigensolvers !

The Basics

2×2 picture



$\{x : \|x\| = 1,\ x \in \mathbb{R}^2\}$

Usually x & Ax do not line up

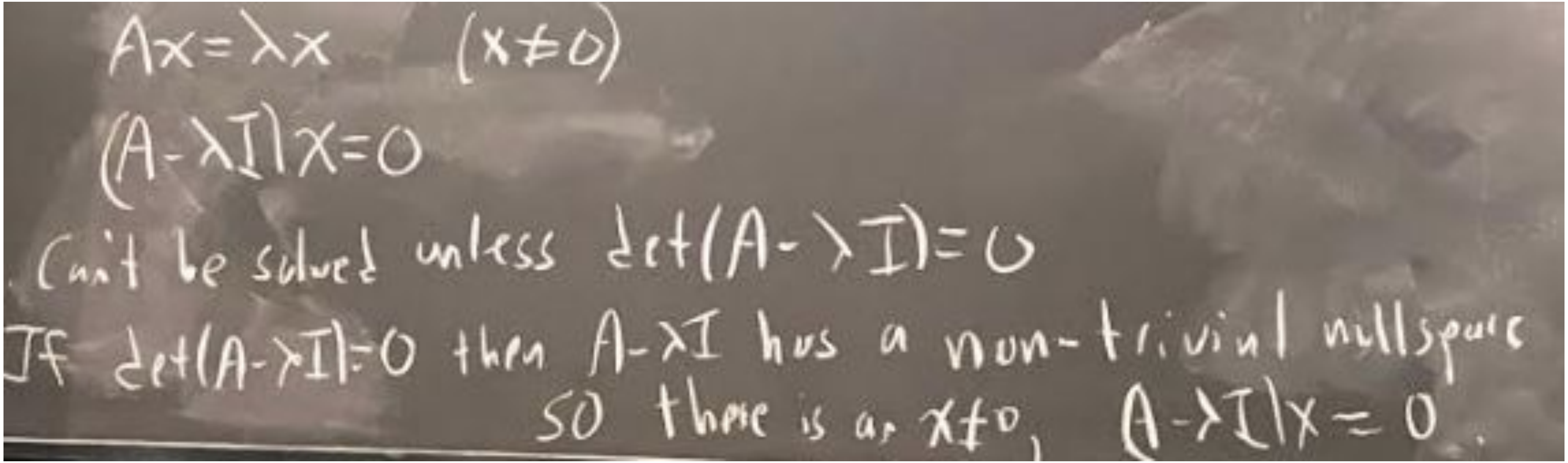
but if $Ax = \lambda x$,

then x is an eigenvector

and λ is an eigenvalue

28. Hand computation of 2x2 eigenproblems

Basic facts



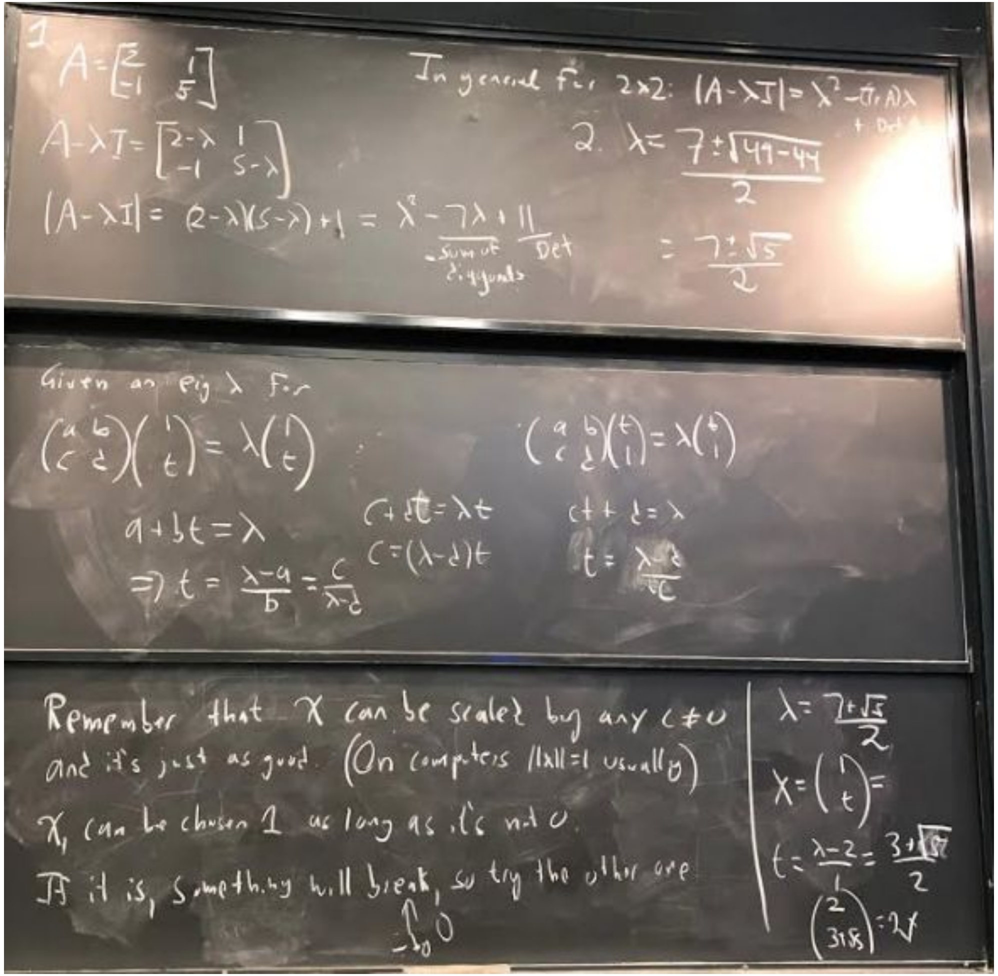
This x is an eigenvector.

When might you be able to do by hand?

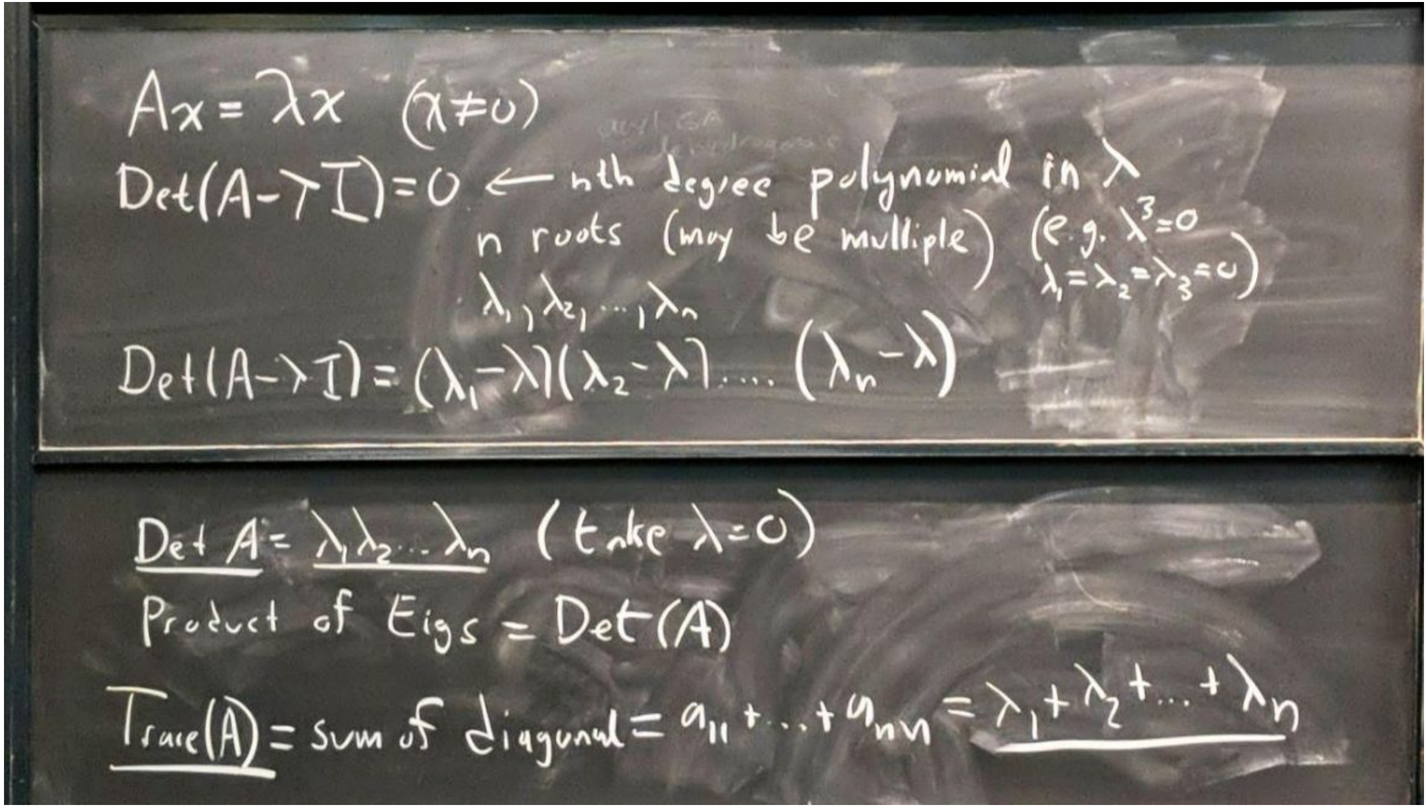
1. 2x2
2. Triangular
3. Some very special cases

Otherwise must be done on a computer

Example


1. A = [2 1; -1 5]
In general for 2x2: |A-λI| = λ² - (Tr A)λ + Det A
A - λI = [2-λ 1; -1 5-λ]
2. λ = (7 ± √(49-44))/2
|A-λI| = (2-λ)(5-λ)+1 = λ² - 7λ + 11
Sum of diagonals
Det
= (7 ± √5)/2
Given an eig λ for
(a b; c d)(1; t) = λ(1; t)
(a b; c d)(t; 1) = λ(t; 1)
a + bt = λ
c + dt = λt
c = (λ-d)t
=> t = (λ-a)/b = c/(λ-d)
ct + d = λ
t = (λ-d)/c
Remember that x can be scaled by any c ≠ 0
and it's just as good (On computers ||x|| = 1 usually)
x₁ can be chosen 1 as long as it's not 0.
If it is, something will break, so try the other one
λ = (7+√5)/2
x = (1; t) =
t = (λ-2)/1 = (3+√5)/2

Eigenvalue Facts



Direct check when n=2 of the trace and det = sum and product , respectively

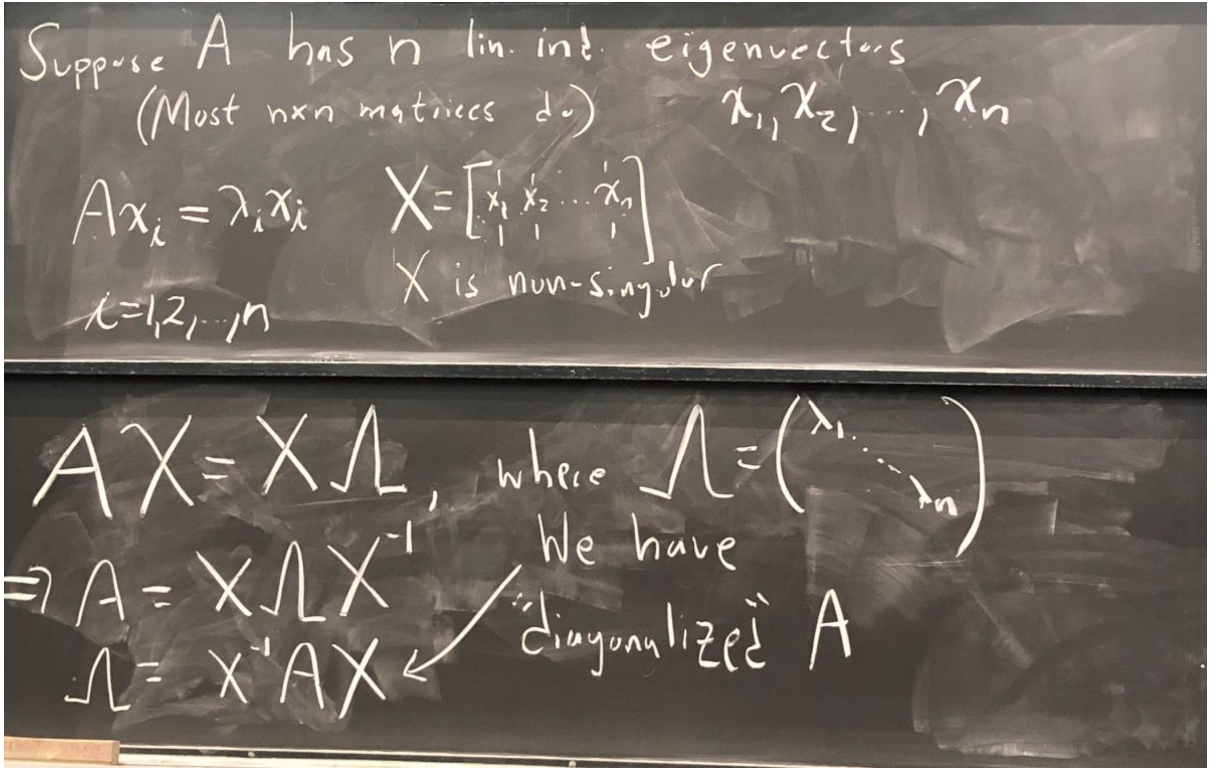
$n=2$

$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$$\det(A-\lambda I) = \lambda^2 - (a+d)\lambda + ad-bc$$
$$= (\lambda-\lambda_1)(\lambda-\lambda_2)$$
$$= \lambda^2 - (\lambda_1+\lambda_2)\lambda + \lambda_1\lambda_2$$

$$\Longrightarrow \lambda_1+\lambda_2 = a+d = \text{Trace}$$
$$\lambda_1\lambda_2 = ad-bc = \text{Det}$$

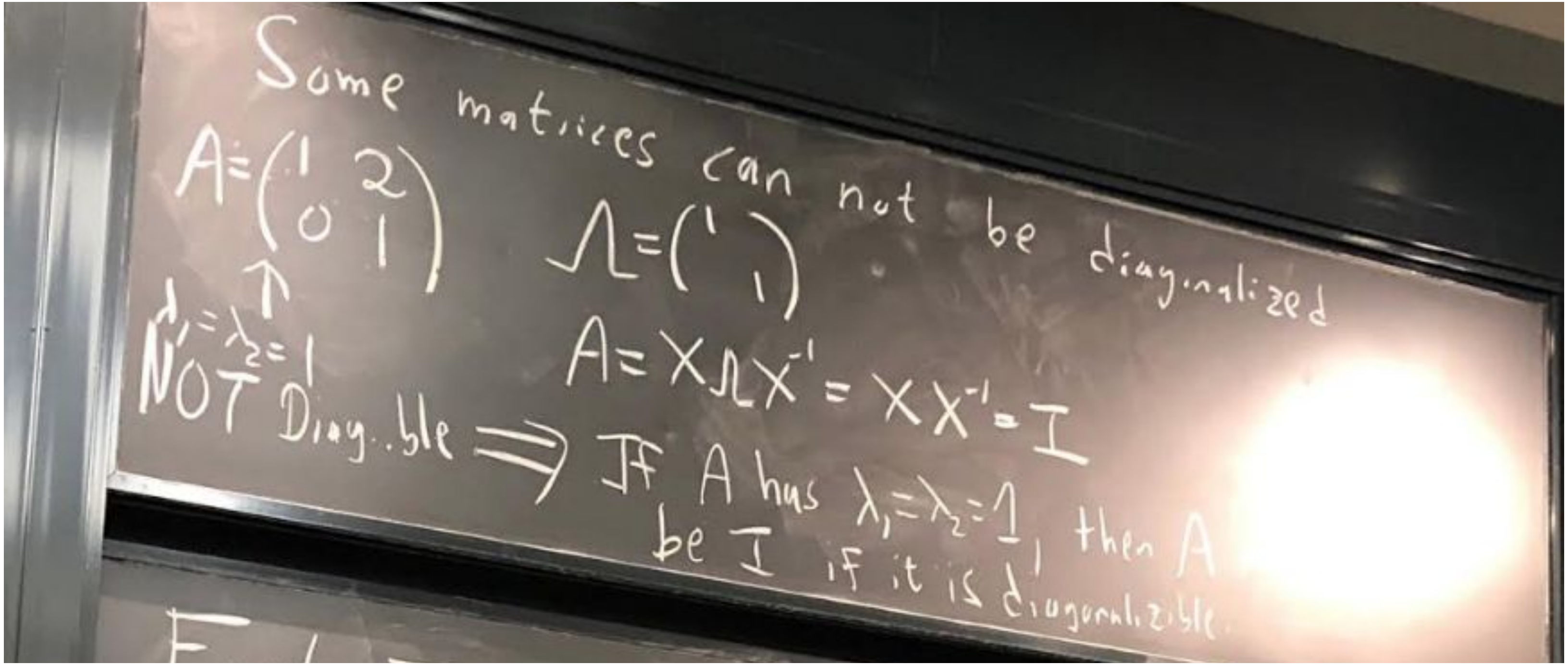
29. Diagonalizing a matrix (when possible)



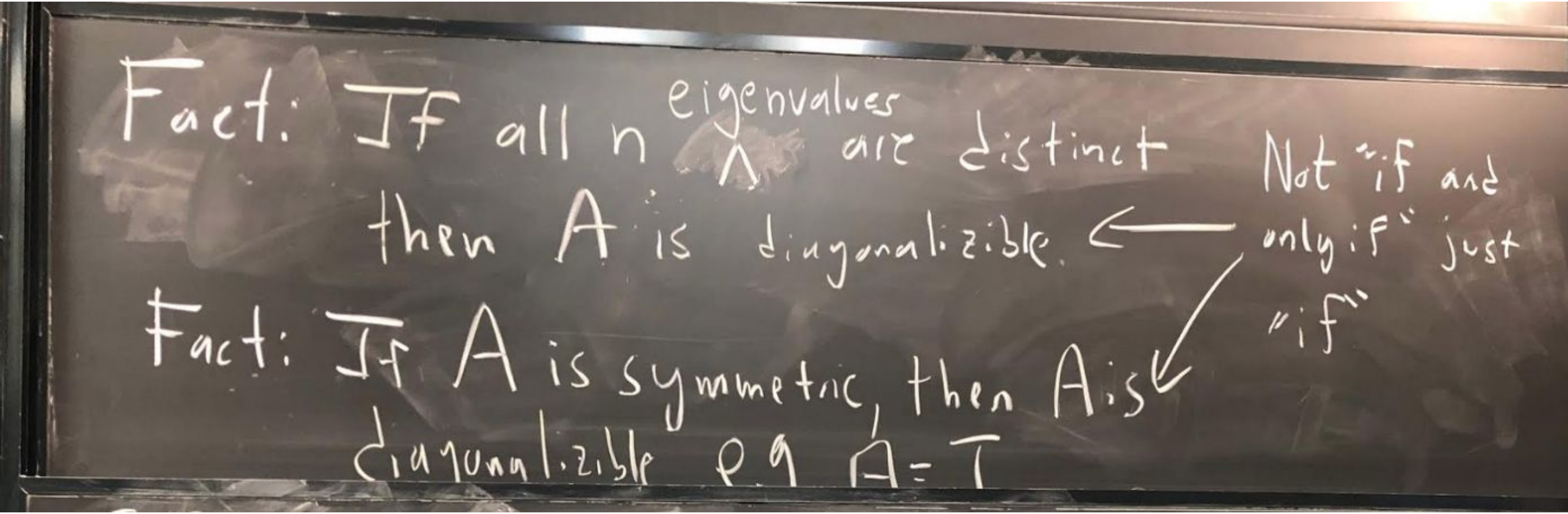
Note: there is no standard
Order to the eigenvalues
Unlike singular values

The greek letter Λ
Is pronounced Lambda.

Some matrices can not be diagonalized



Matrices that are definitely diagonalizable



E.g. A=I

Similarity

Let C be any n xn matrix.

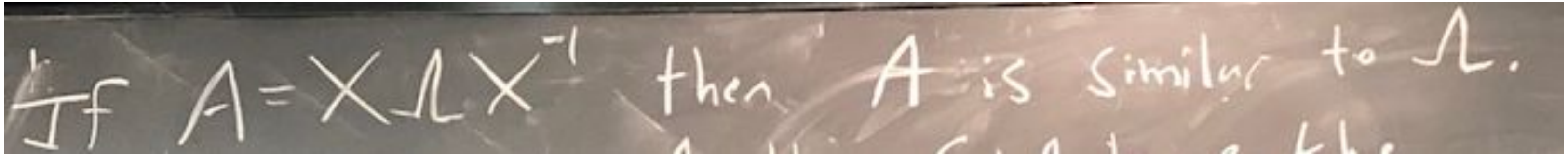
We say that C is similar to A if there is an invertible B such that

$A=BCB^{-1}$.

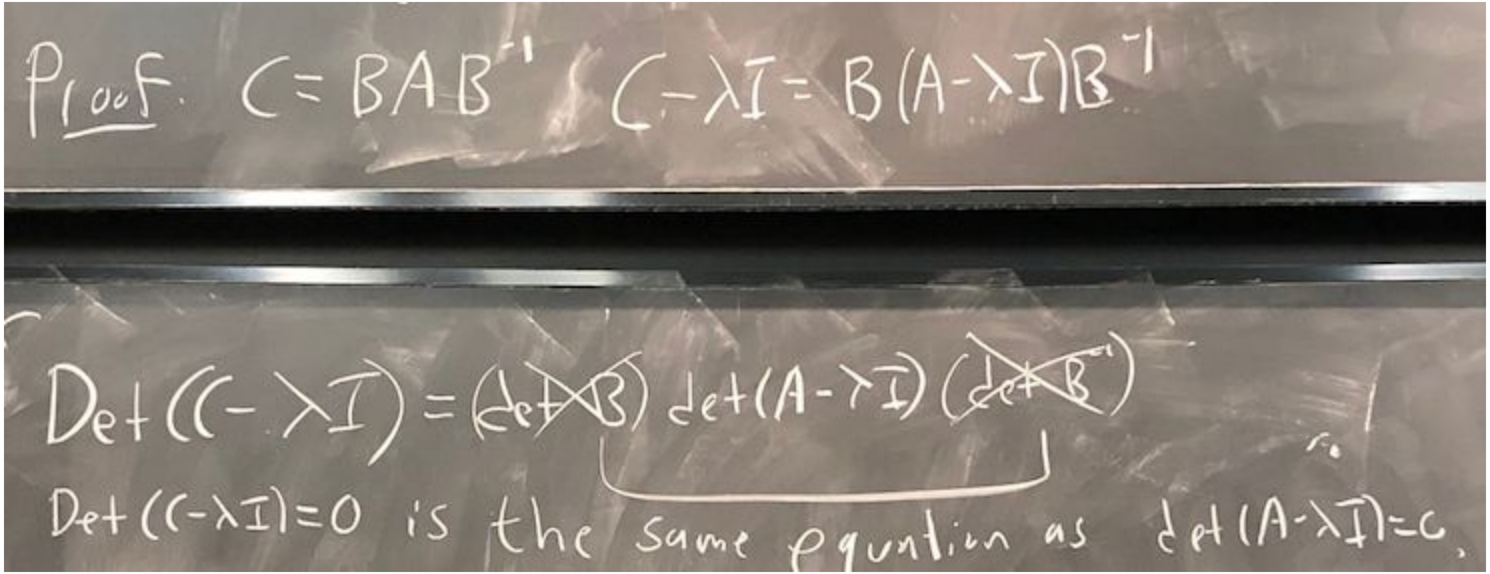
The set of matrices similar to C may be written

$\{BCB^{-1} : B \text{ is non-singular}\}$

Diagonalizable matrices are similar to their diagonal form



If C is similar to A then C and A have the same eigenvalues (even if not diagonalizable)



Corollary: det A = det C , trace A = trace C

Suppose you know A and C, how do you find B?

If A is 2x2 and diagonalizable, by hand you can go back and forth through the eigenvectors of A and C.

More generally, if you know the eigenvectors of A and C (if diagonalizable) then again you can go back and forth.

Otherwise, except for very special cases, it's so nuisancy to do by hand, that it's almost never been asked in traditional linear algebra classes.

Example of how to find B on a computer

```julia
using LinearAlgebra

# setup
X₁,X₂ = randn(4,4),randn(4,4)
Λ  = Diagonal(1:4)
A,C = X₁\Λ*X₁, X₂\Λ*X₂;
```

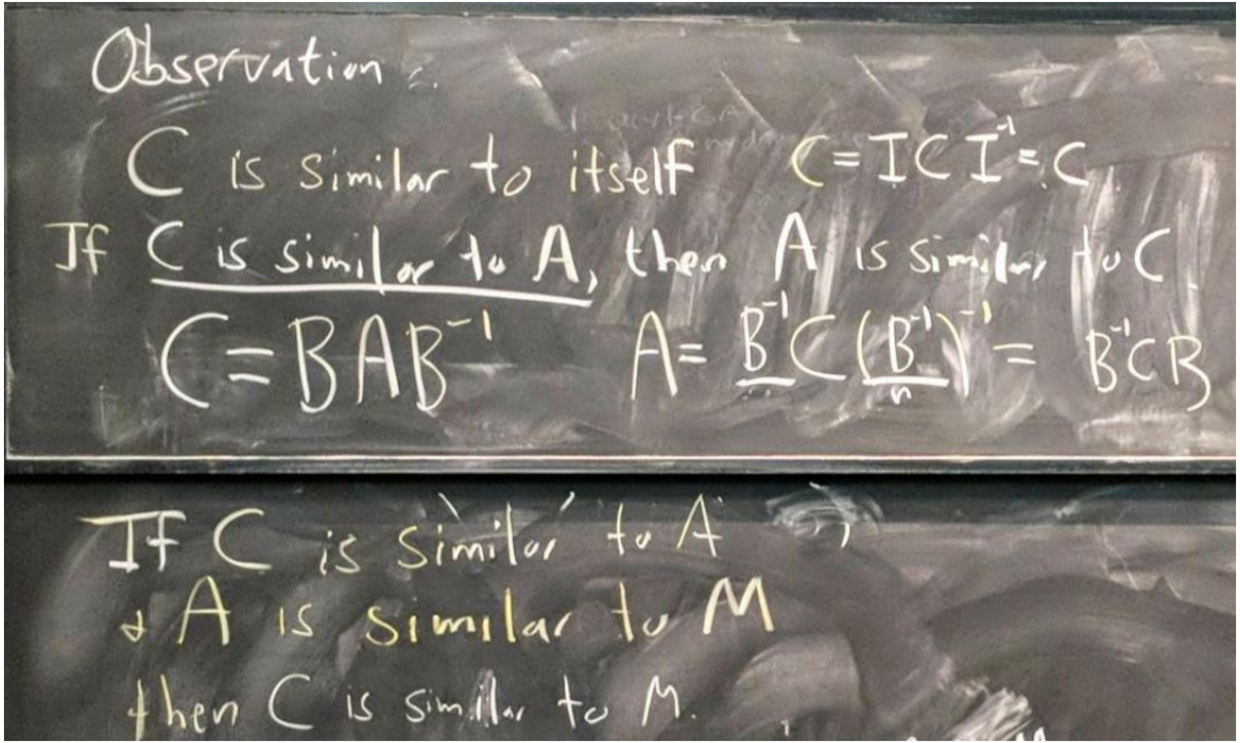
Given A and C find B such that $C = BAB^{-1}$

```julia
Λ₁,M₁ = eigen(A)
Λ₂,M₂ = eigen(C)
# let's sort the eigenvectors
M₁ = M₁[:,sortperm(Λ₁)]
M₂ = M₂[:,sortperm(Λ₂)]
B = M₂/M₁

C ≈ B*A*inv(B)
```

```julia
#copyable text
using LinearAlgebra

# setup
X₁,X₂ = randn(4,4),randn(4,4)
Λ  = Diagonal(1:4)
A,C = X₁\Λ*X₁, X₂\Λ*X₂;

# Given A and C find B such that C = BAB^-1
Λ₁,M₁ = eigen(A)
Λ₂,M₂ = eigen(C)
# let's sort the eigenvectors
M₁ = M₁[:,sortperm(Λ₁)]
M₂ = M₂[:,sortperm(Λ₂)]
B = M₂/M₁

C ≈ B*A*inv(B)
```

Similarity Properties

This means similarity is an "equivalence relationship"



← reflexive property of similarity
← symmetric property

← Transitive property

Exercise: prove the transitive property

Connections to the SVD

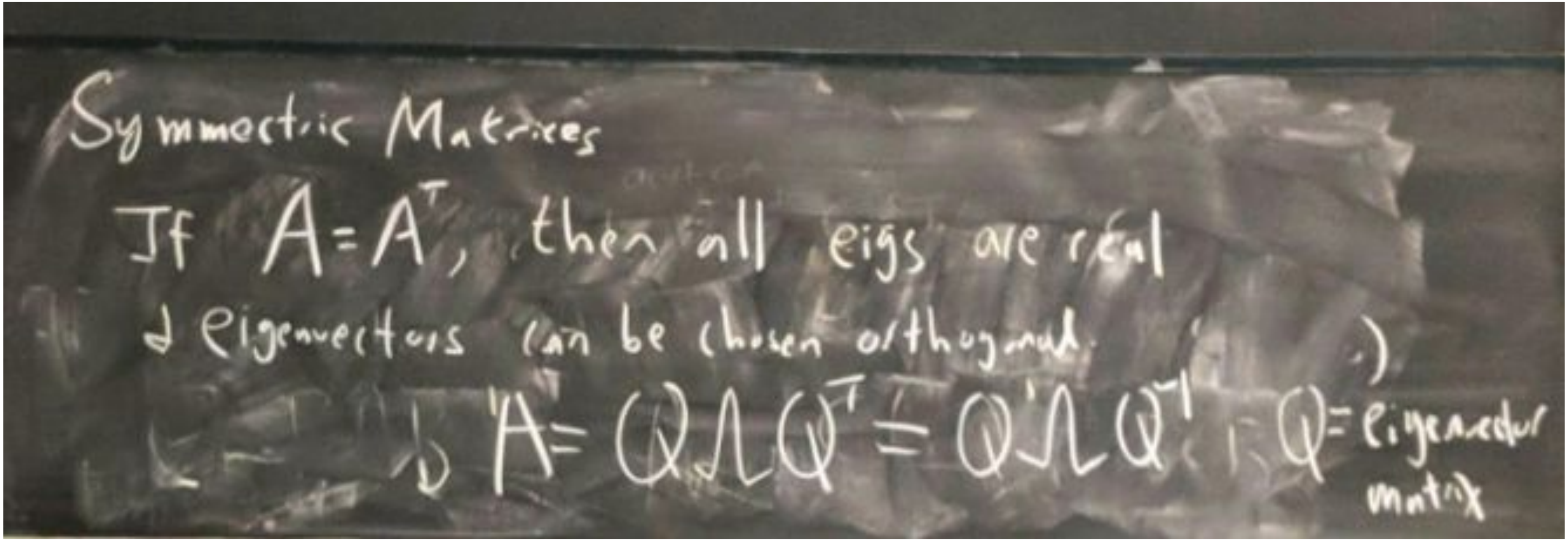
If $A = U\Sigma U^T$ is a full svd, then we have an eigendecomposition of A. (U=V)

This can only happen if A is symmetric with positive eigenvalues. We can allow 0 on the diagonal of Σ, as A could have 0 eigenvalues if it is singular.

If A is symmetric it can always be decomposed as $Q\Lambda Q^T$, but this would only be an svd if the diagonal of Λ is positive. One can convert to an svd by negating some columns of Q corresponding to negative eigenvalues, and recognizing the zeros.

In general A^TA and AA^T are symmetric with positive eigenvalues. Their eigendecompositions can give the svd, but this is not how the svd is computed on computers. Still somewhat useful to know.

Repeat about symmetric matrix eigendecomposition



Summary of relationship of svd to eigenvalues

- If A is symmetric with positive eigenvalues: $Q\Lambda Q^T$ is an SVD
- If A is symmetric and has positive and 0 eigenvalues: $Q\Lambda Q^T$ is still a full svd, the compact SVD would hide the nullspace but it could be recovered
- If A is symmetric, $Q(\Lambda D)(QD)^T$ is an svd where D is diagonal with +1 and -1 rigged to make ΛD have only non-negative entries on the diagonal
- If A is singular it has at least one zero eigenvalue and at least one singular value gets wiped out.
- It is always the case that $A^TA=V(\Sigma^T\Sigma)V^T$ is an eigendecomposition
- It is always the case that $AA^T=U(\Sigma\Sigma^T)U^T$ is an eigendecomposition
- The above two formulas say that the singular values are sqrts of the eigenvalues of A^TA and AA^T with some chatter about the zeros. This is how eigenvalue-first classes sometimes define singular values.
-

30. Matrix Differential Equations and the Matrix Exp

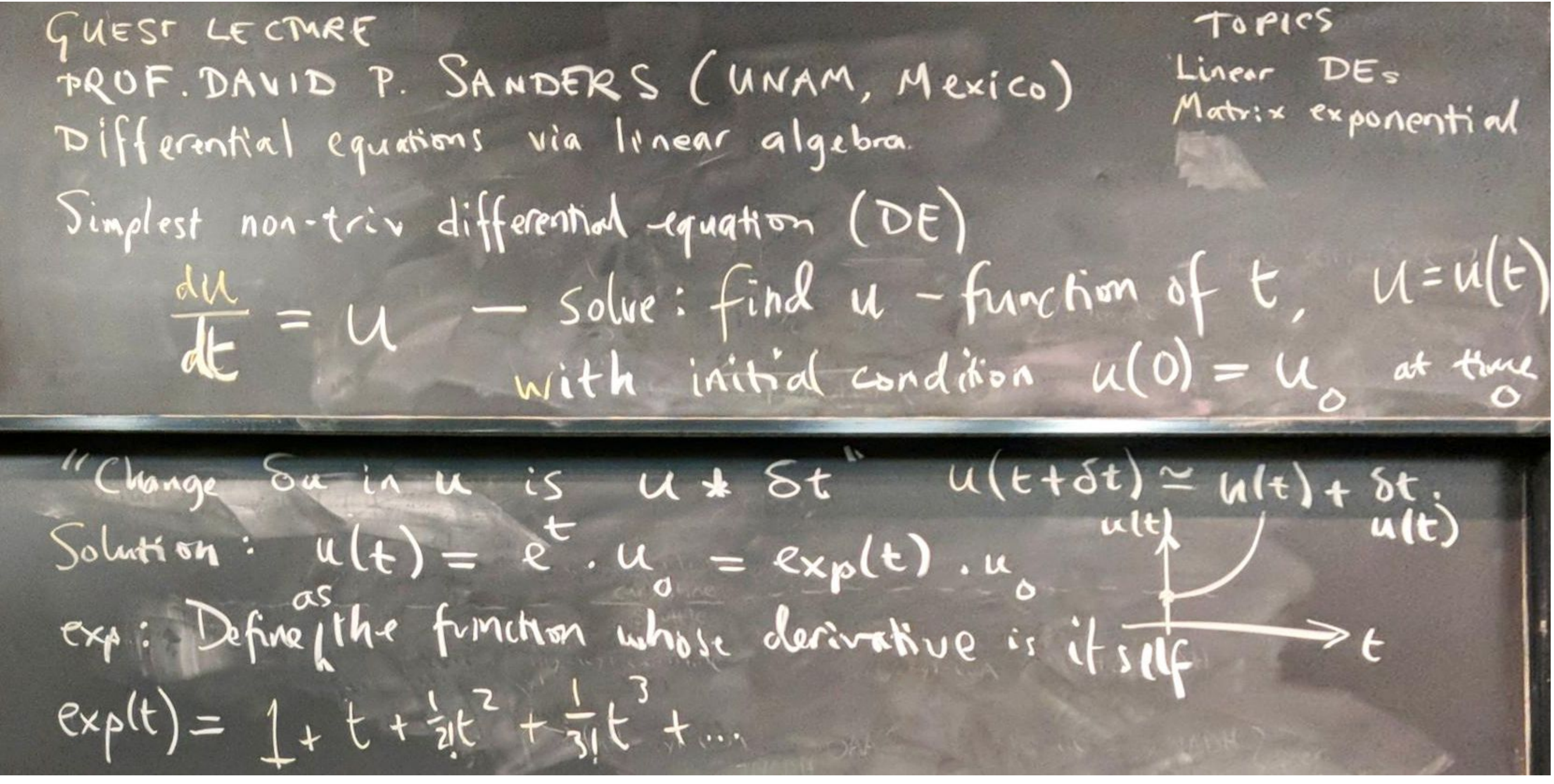
Linear algebra and differential equations (DEs) may seem to be unrelated, but in fact they are strongly interconnected: linear algebra gives us a direct way to solve systems of linear differential equations with constant coefficients.

The simplest non-trivial differential equation is $$\frac{du}{dt} = u$$

Let's recall the meaning of this: we are thinking of u as a function of t, sometimes written u = u(t). We want to find the function u(t) such that its derivative (with respect to t) is the same function.

We remember from calculus that the exponential function satisfies this (in fact, except for multiplying by a constant factor, it is the only such non-trivial function).

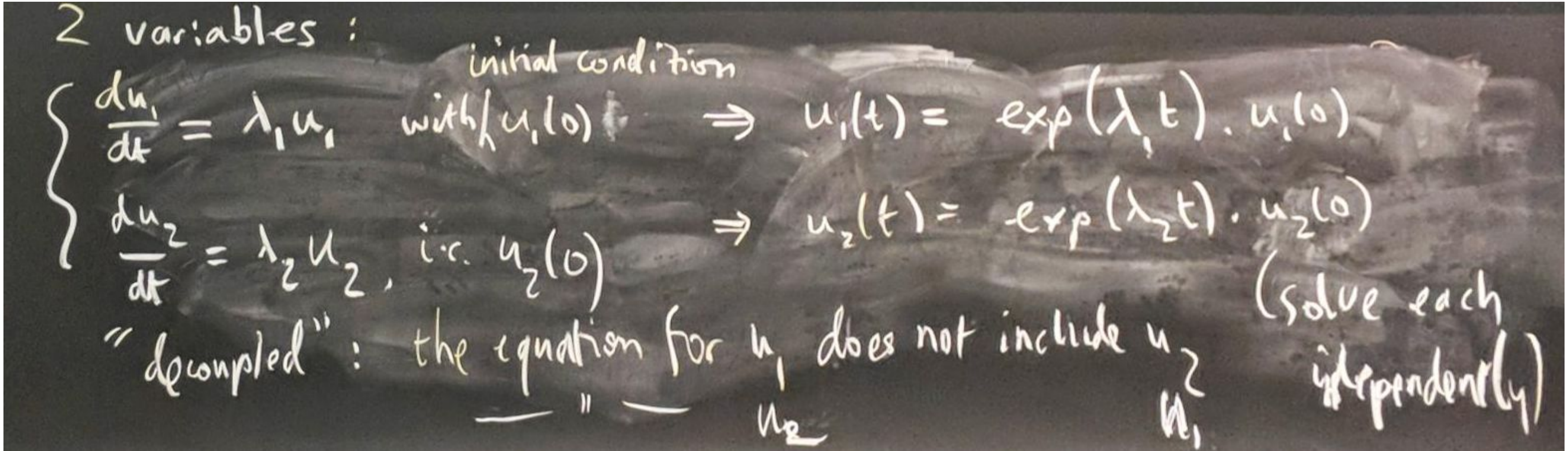
Simplest differential equation; exponential function



On the previous slide, we recall the definition of the exponential function exp(t), or e^t, as a **power series** in t, i.e. a sum of powers multiplied by coefficients, of the form a_n t^n, for n from 0 to infinity.

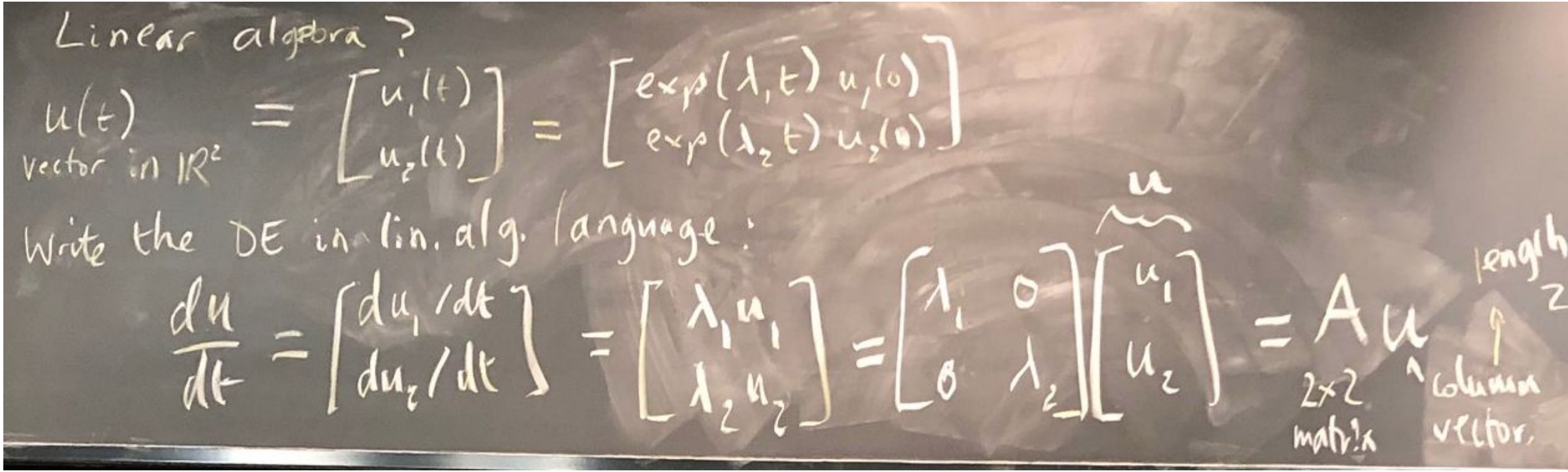
If we take exp(λt) instead, that satisfies the differential equation du/dt = λu.

What if we have two variables but equations that are uncoupled (the two variables "don't mix"? Then we can immediately find the solution:

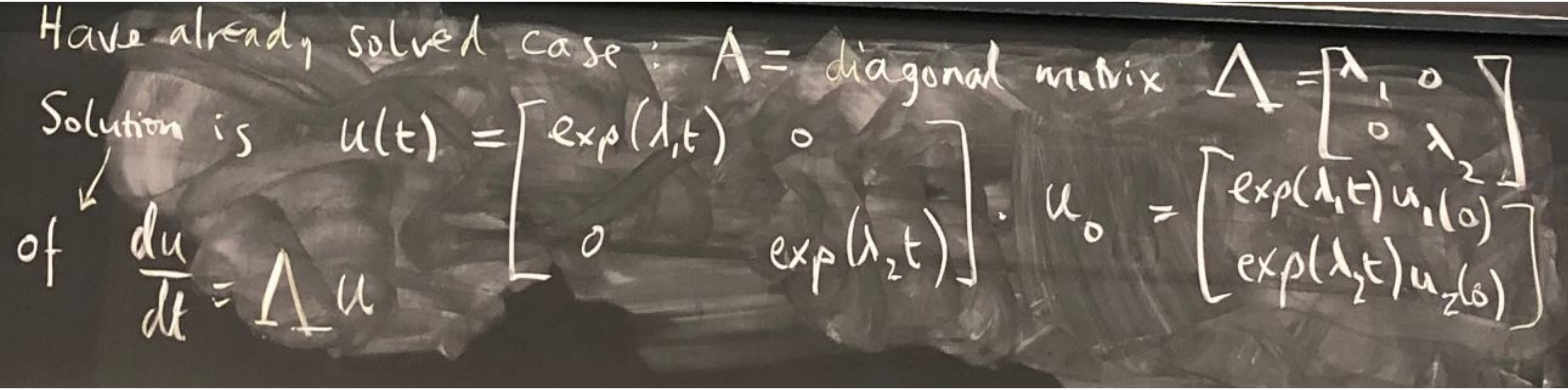


Uncoupled DEs: linear algebra language

Let's use some linear algebra language for the same result, by introducing a vector u = (u1, u2). We see that uncoupled equations corresponds to a **diagonal matrix.** The previous slide thus gives the solution for the linear differential equation du/dt = Au when u is a vector and A is a diagonal matrix!

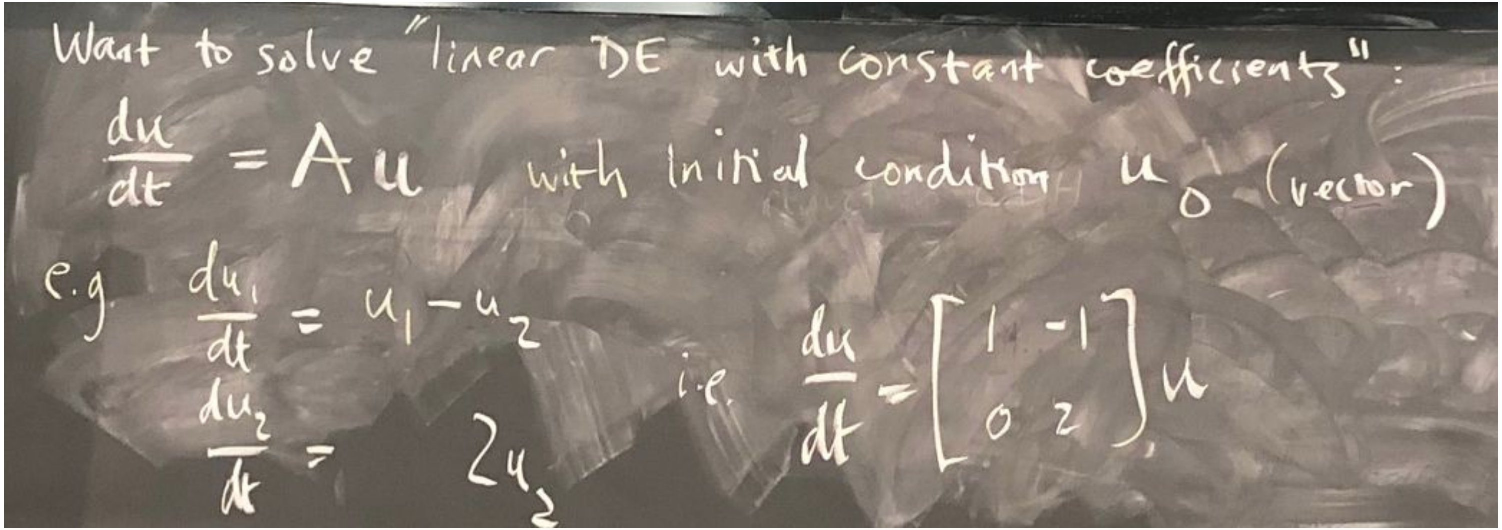


Solution of du/dt = Au for diagonal matrix A



General (coupled) linear DEs

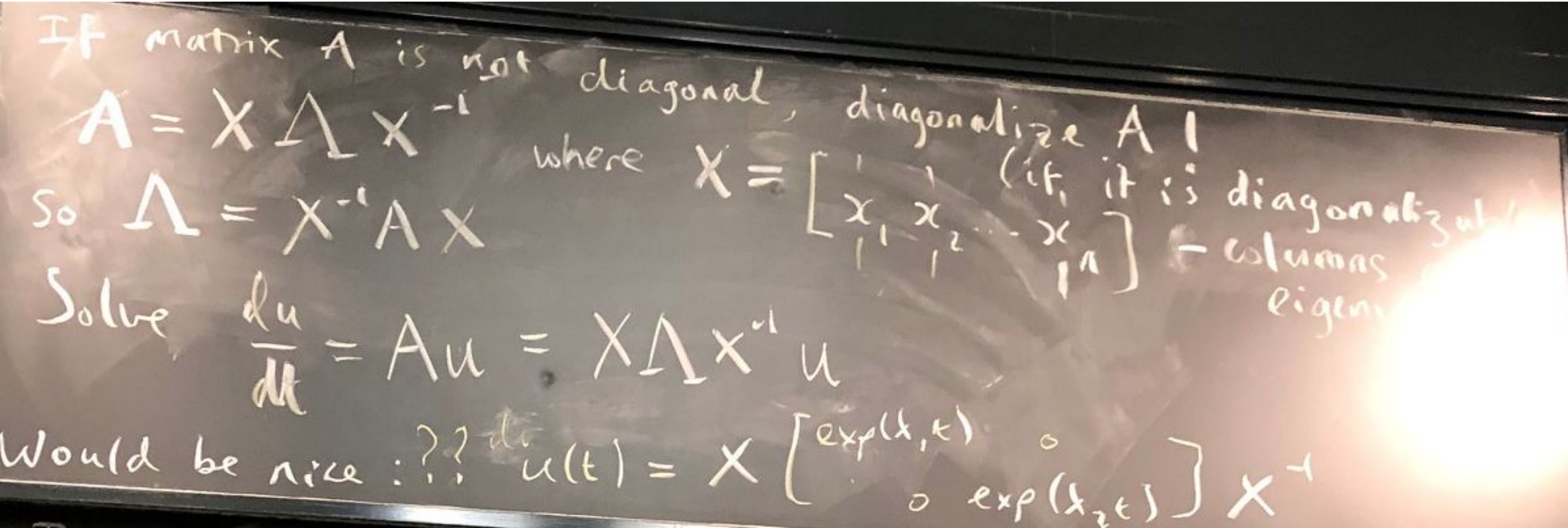
What if the equations are coupled? E.g.



We would like to use the solution that we just found, so we try to **diagonalize** the matrix (if possible)!

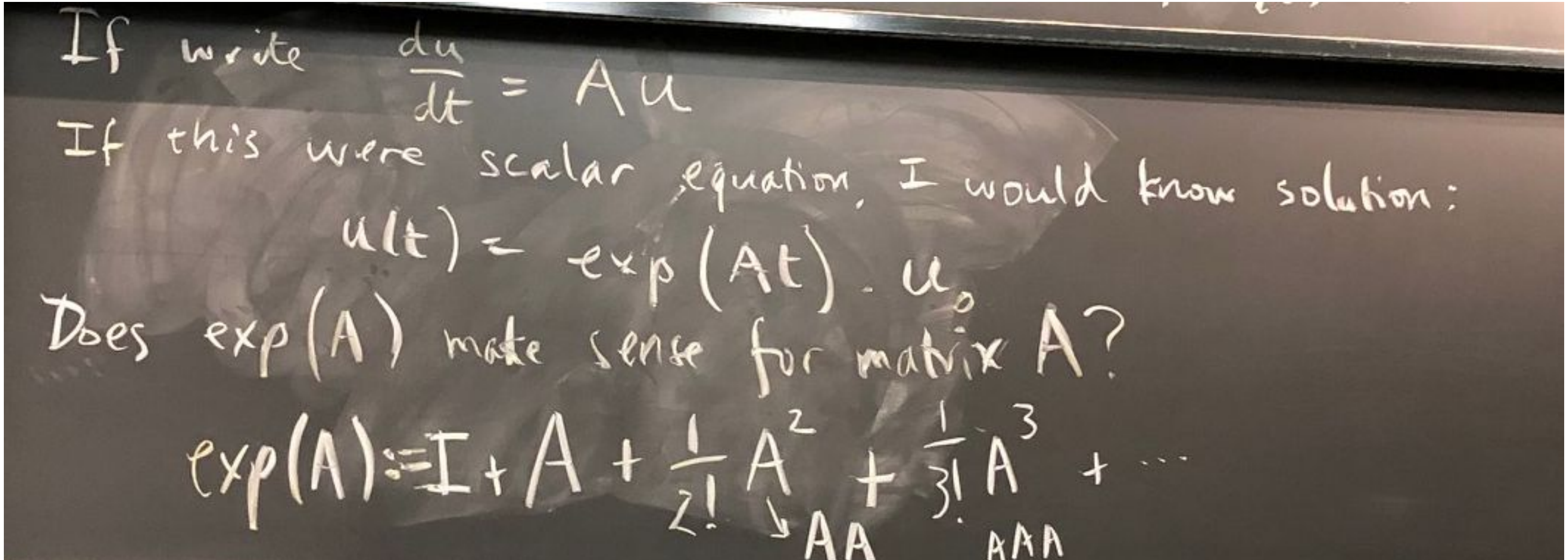
Diagonalize A to solve du/dt = Au

We diagonalize A using the eigenvalues and eigenvectors. If we are lucky then the solution will be a simple transformation of the diagonal solution. We are just guessing at this point, but it turns out that this is correct as we will see below!



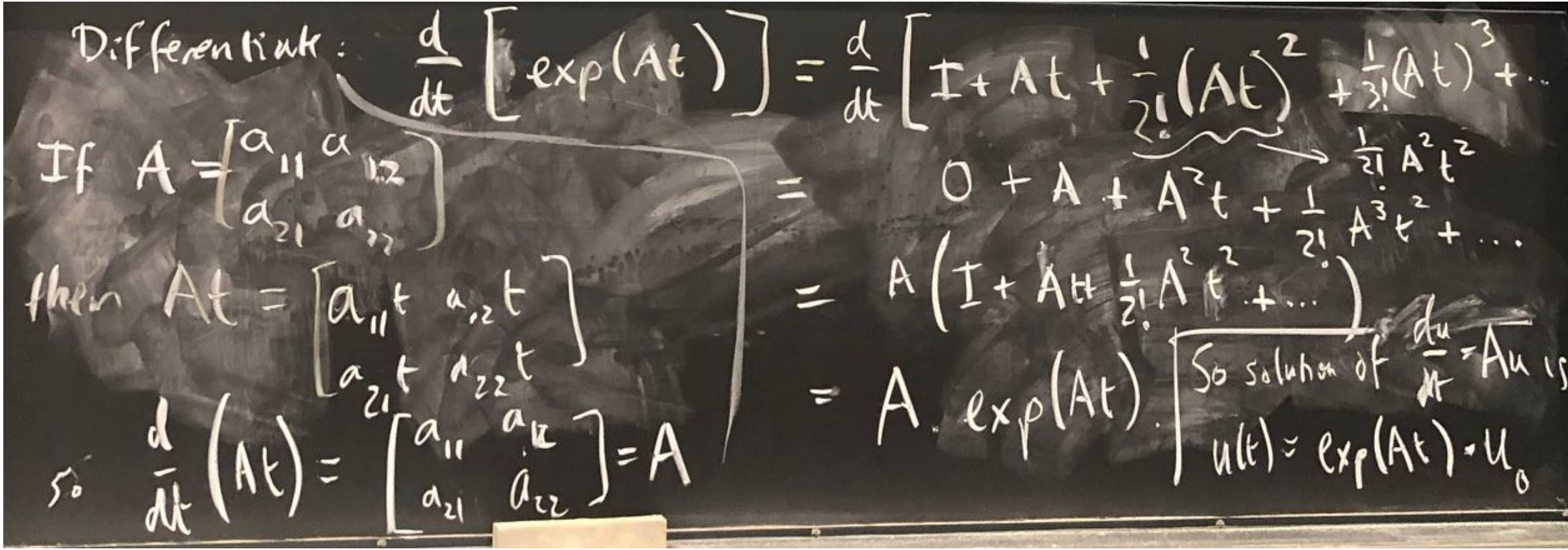
Solution of du/dt = Au: matrix exponential

Let's try to solve the *vector* equation du/dt = Au, where A is a **matrix**, by analogy with the scalar case. If u and A were both scalars, we know the solution is exp(At) * u0 (where u0 is the initial condition). Maybe we can define exp(At) when A is a matrix in the same way, i.e. via a series expansion, but now a series of powers of the matrix A!:



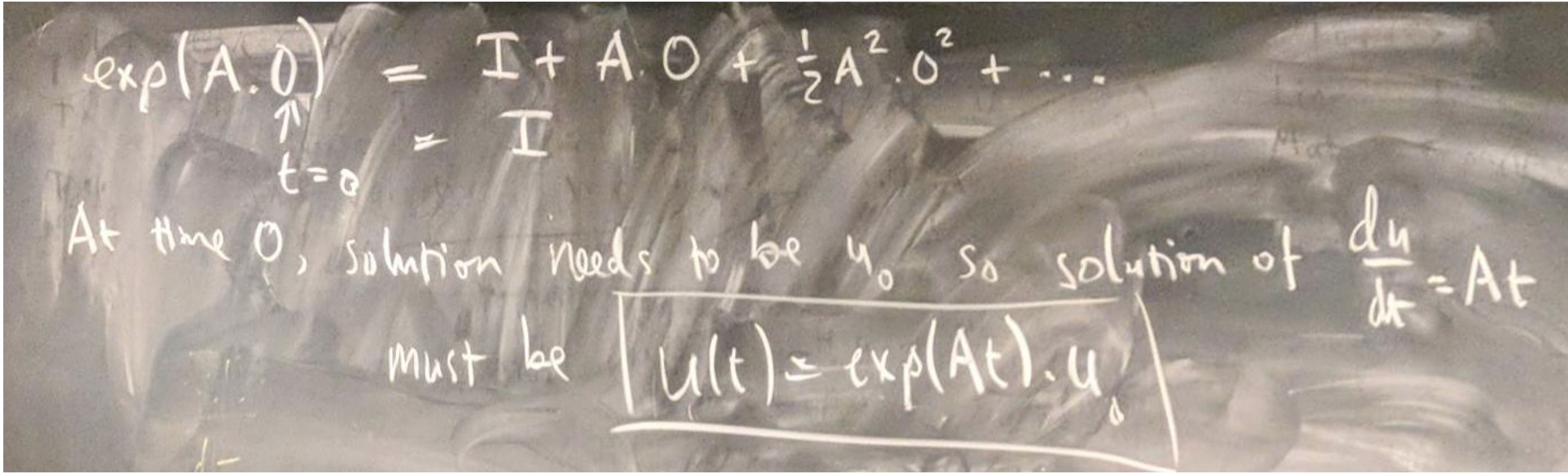
Differentiating the matrix exponential

Let's check by differentiating that exp(At) really is a solution:



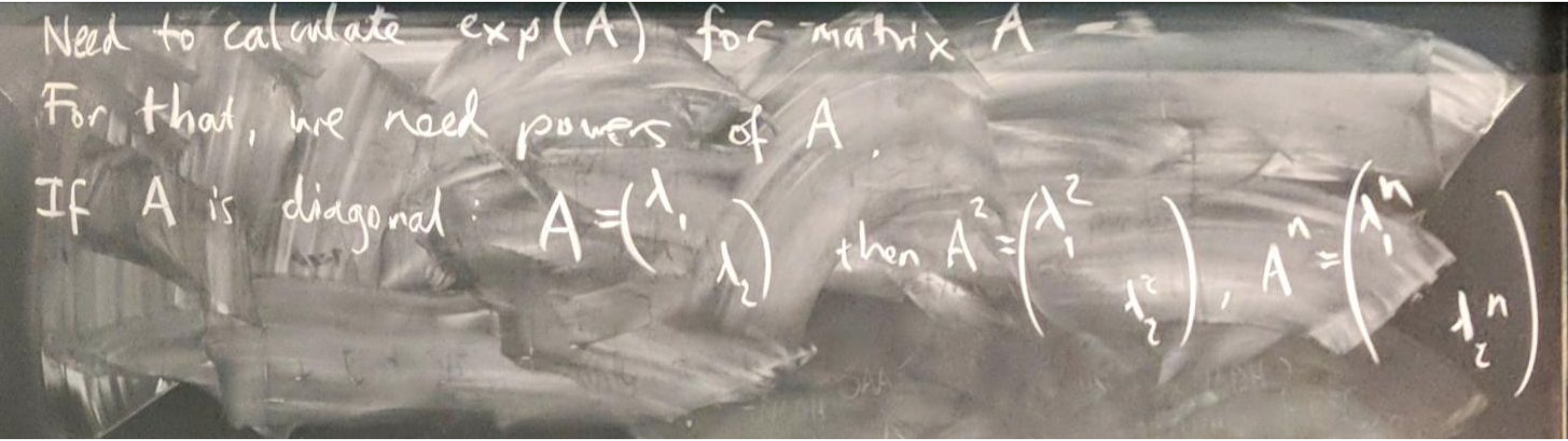
Initial condition

When t=0, we want to recover the initial condition u0 (this is a vector). Since at t=0 we have exp(At) = exp(0) = I, we need to multiply the matrix exp(At) by the initial vector u0 to get the full solution:



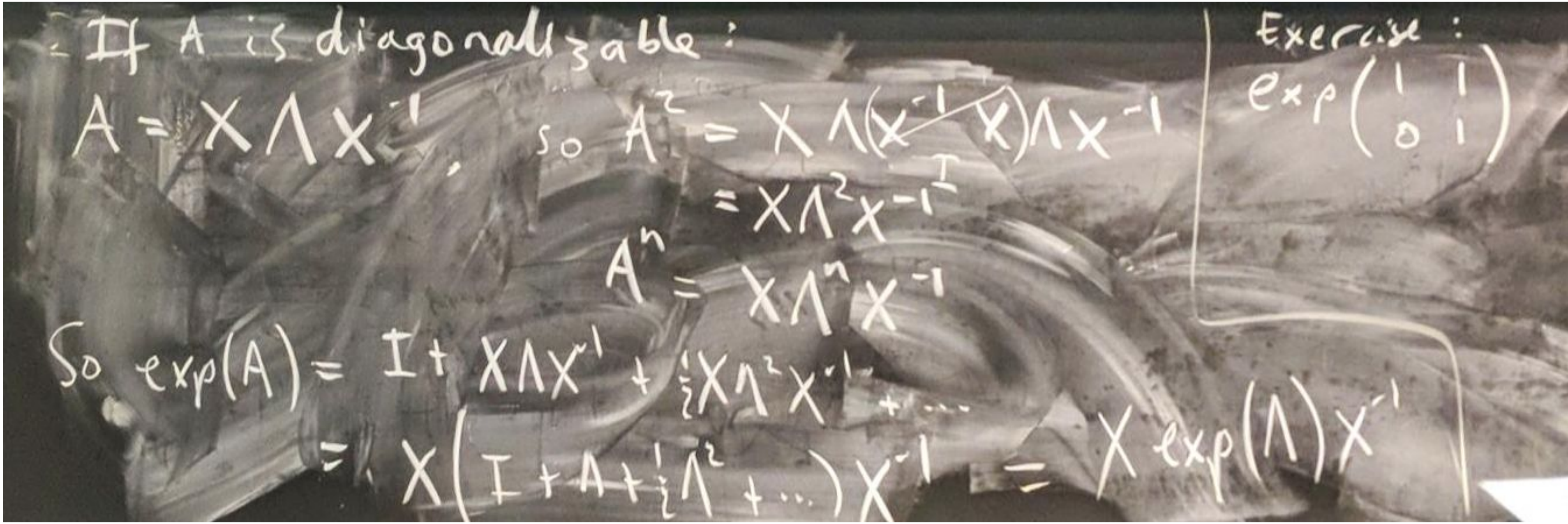
Calculation of exp(A)

To calculate exp(A), we have the series definition, for which we need powers of a matrix. This is easy if A is diagonal:



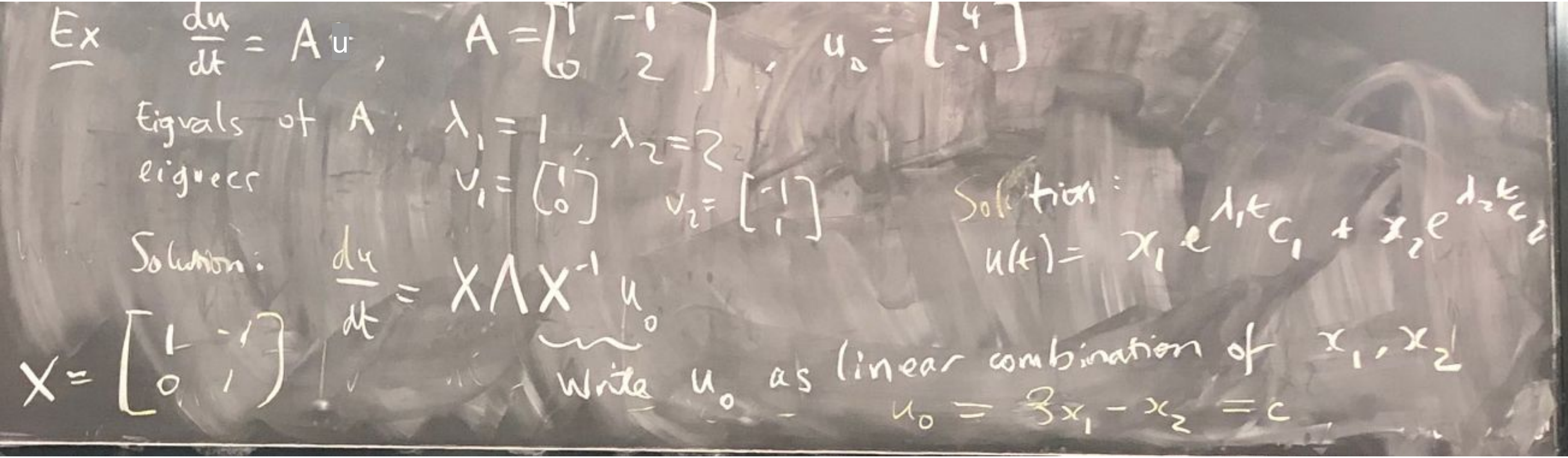
A^n and exp(A) for diagonalizable matrix

If A is not diagonal, we try to diagonalize it (if possible). This gives a nice way to calculate powers of A, and hence the exponential:



Complete solution of du/dt = Au

We can put this all together to give a recipe to solve any linear differential equation with constant coefficients, e.g.



31. Eigenvalue Proofs

- Distinct Eigenvalues ⇒ linearly independent eigenvectors
- Symmetric Matrices have real eigenvalues
- Symmetric Matrices -- can choose eigenvector matrix to be orthogonal

Distinct Eigs, Linearly independent eigenvectors n=2

n=2: A:2x2 $Ax_i=\lambda_i x_i$ for i=1,2

Want: If $c_1 x_1 + c_2 x_2 =0$ then $c_1 = c_2 =0$. (This is the very def of linear independence)

Warmup:

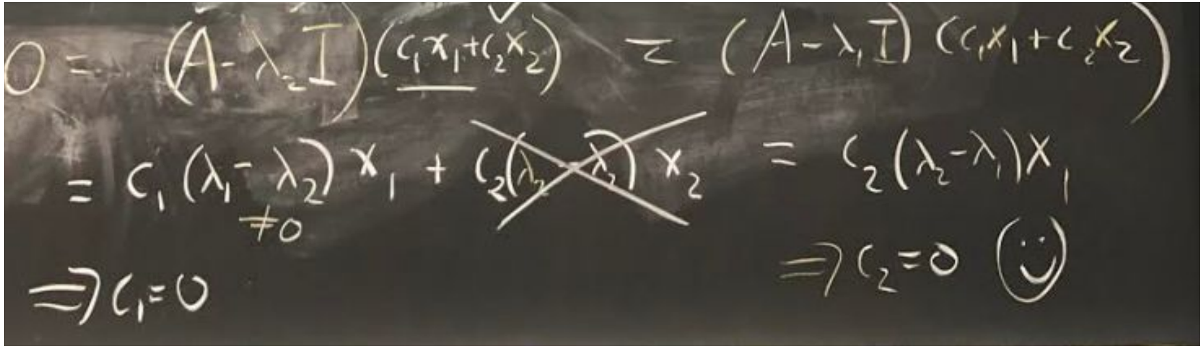
$(A-3I)x_1=(\lambda_1-3)x_1$

$(A-3I)x_2=(\lambda_2-3)x_2$

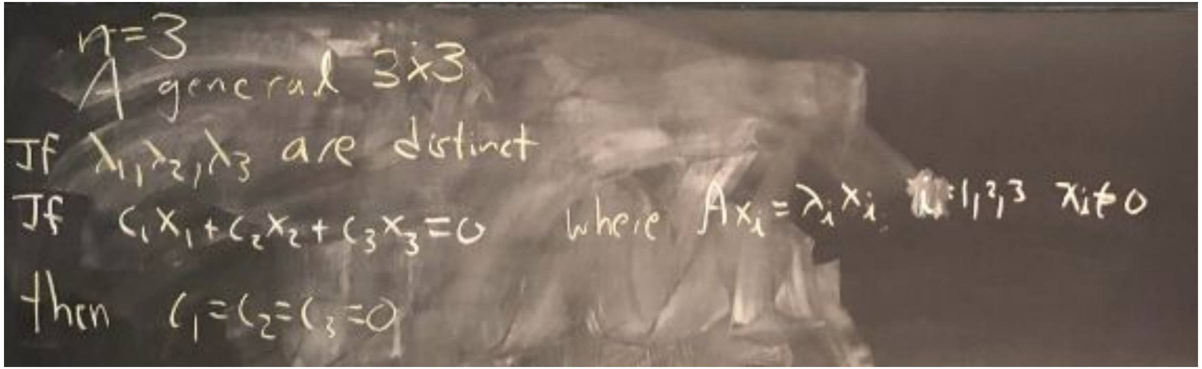
$(A-3I)(A-2I)x_1=(\lambda_1-3)(\lambda_1-2)x_1$

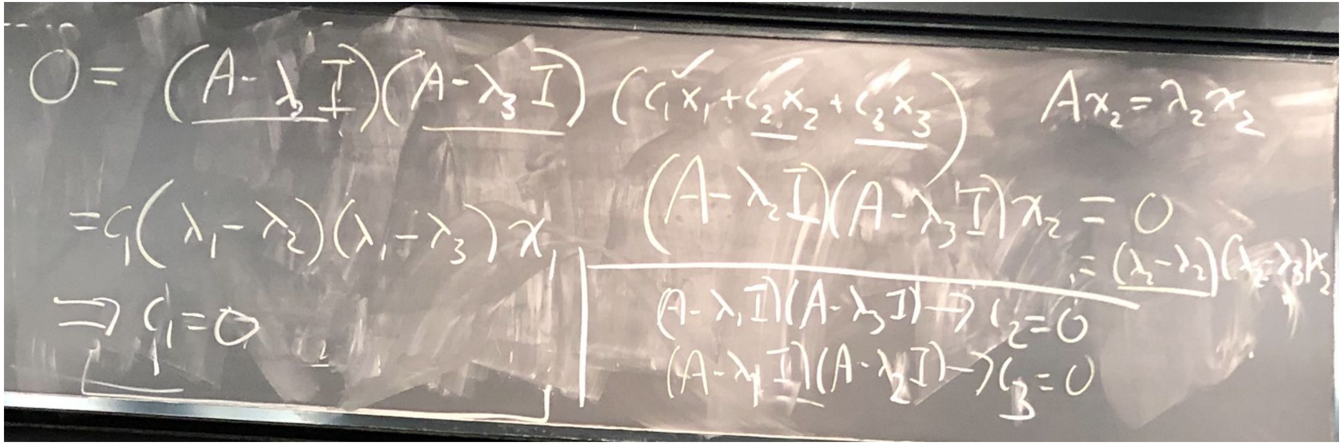
$=(A-2I)(A-3I)x_1=(\lambda_1-3)(\lambda_1-2)x_1$

Proof: We first show that $c_1=0$ and then point out the c_2 proof is similar



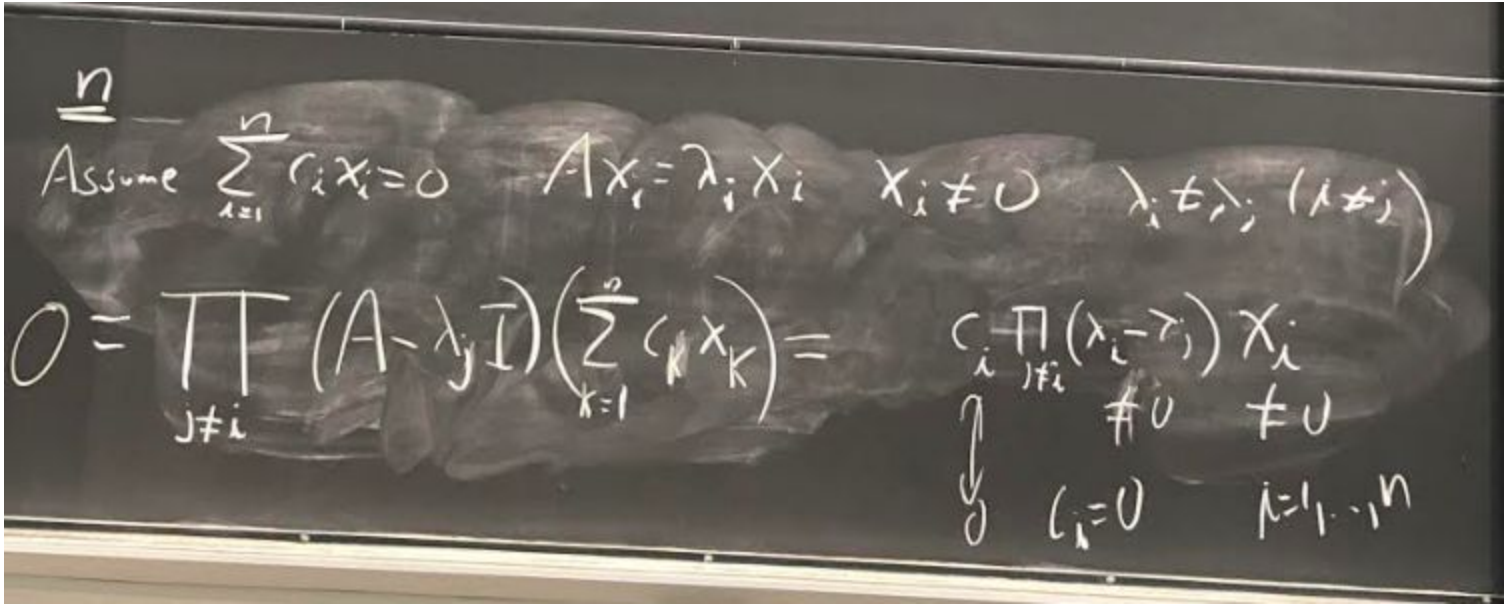
Distinct Eigs = Lin ind Eigenvectors n=3





Distinct eigenvalues = lin independent eigenvectors

General n



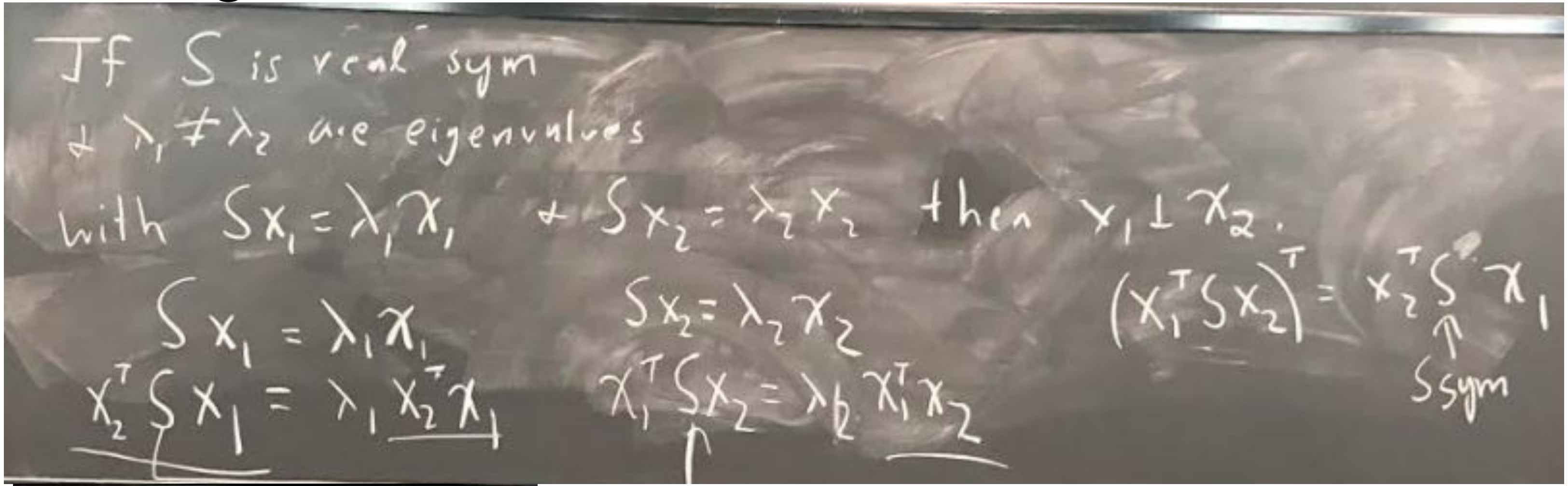
Sym matrices have real eigenvalues

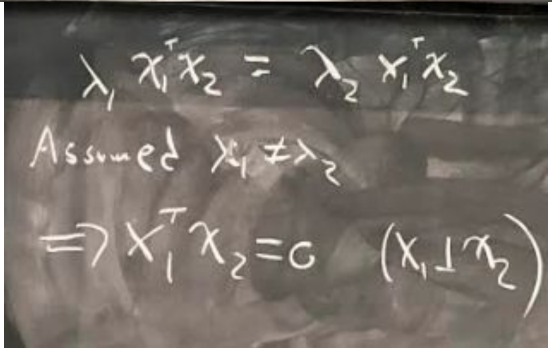
$Sx = \lambda x \Rightarrow S\bar{x} = \bar{\lambda}\,\bar{x} \Rightarrow \quad x^T S\bar{x} = \bar{\lambda} x^T \bar{x}$

But $x^T S = \lambda\, x^T$ so $x^T S\bar{x} = \lambda\, x^T\,\bar{x}$

Hence $\lambda = \bar{\lambda}$

Symmetric Matrices - can choose eigenvectors orthogonal

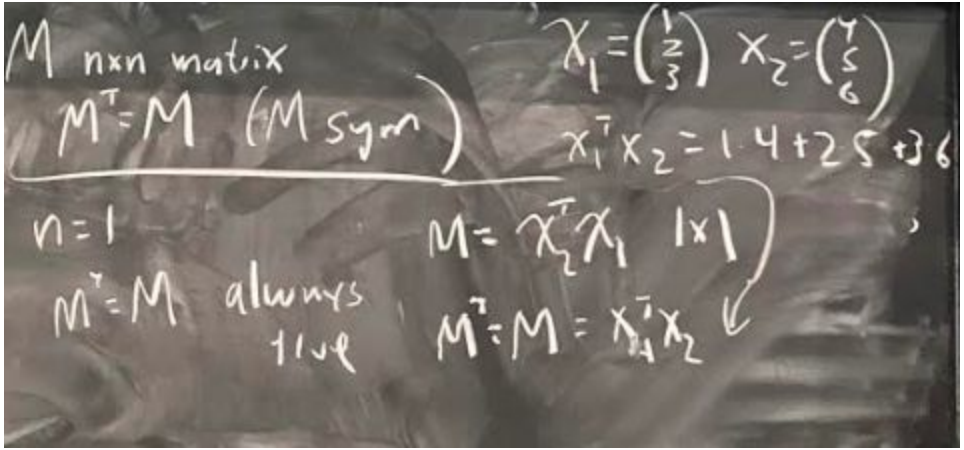




This can be extended to all n eigenvectors and the case of eigenvalues not being distinct. Thus $S=Q\Lambda Q^T$
Is an eigenfactorization of any symmetric matrix.

The extension to the non distinct eigenvalue case requires continuity arguments and will not be discussed in this class.

Reminders on symmetric matrices



Often the easiest way to see a matrix
Is symmetric is to check M^T=M.

Also recognizing that a 1x1 matrix is always
symmetric is handy.

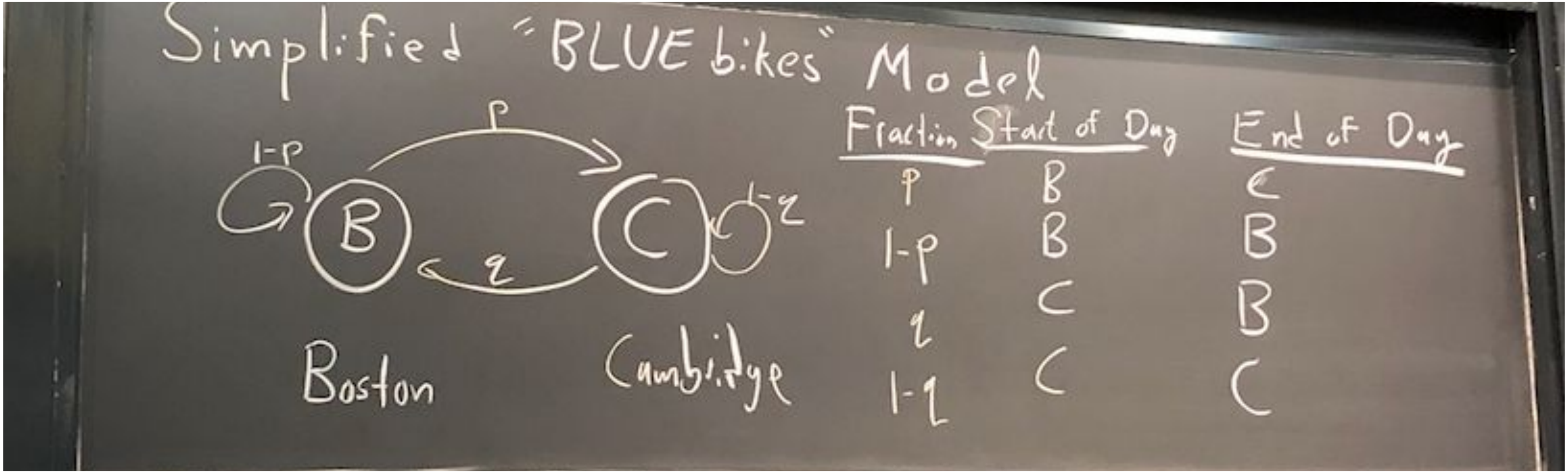
32. Markov Matrices

Note: Strang mentions positive Markov which guarantees one eigenvalue 1 and rest of smaller magnitude. More commonly Markov refers to non-negative and thus comes with no such guarantee. We will call the first a positive Markov matrix. A general Markov matrix can have multiple eigenvalues with absolute value 1.

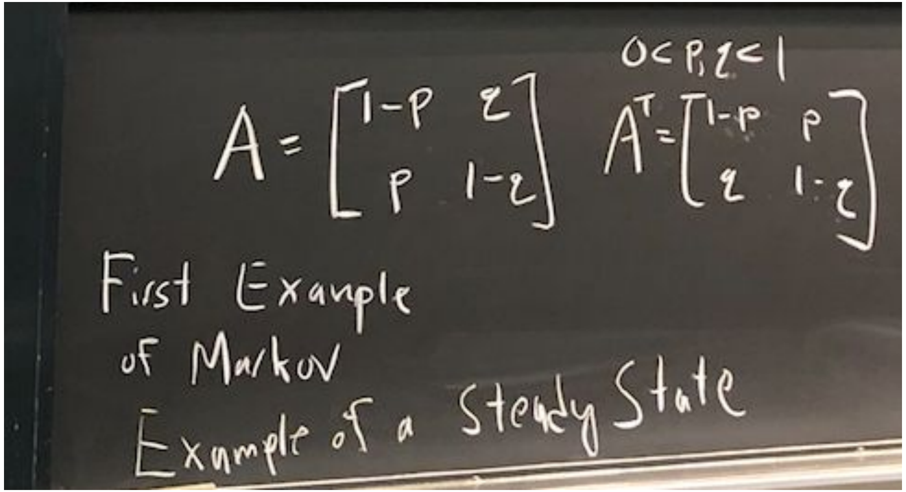
Cosider the identity or [0 1;1 0]

Simplified “BLUEbikes” model

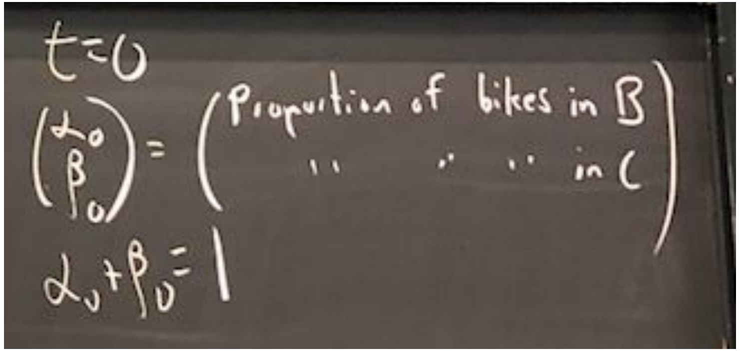
Imagine there are two bike stations, One in Boston, one in Cambridge. and a fraction, p, of the bikes every day go from Boston to Cambridge,While q go the other way.
Assume no trucks move the bikes back overnight, what is the steady State?
This kind of transition situation shows up in many, many applications.



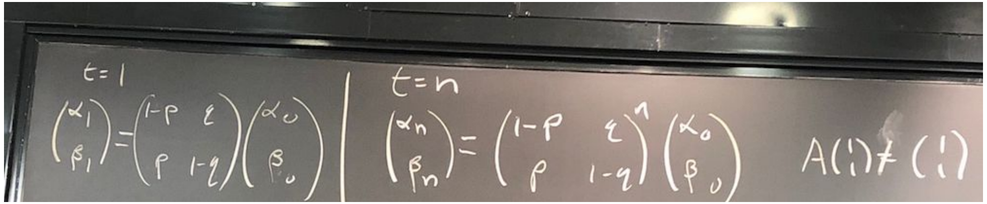
Use matrices



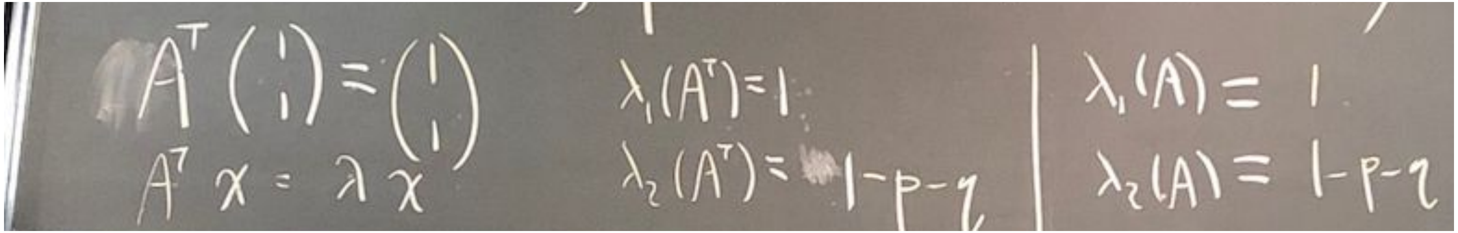
Initial Conditions:



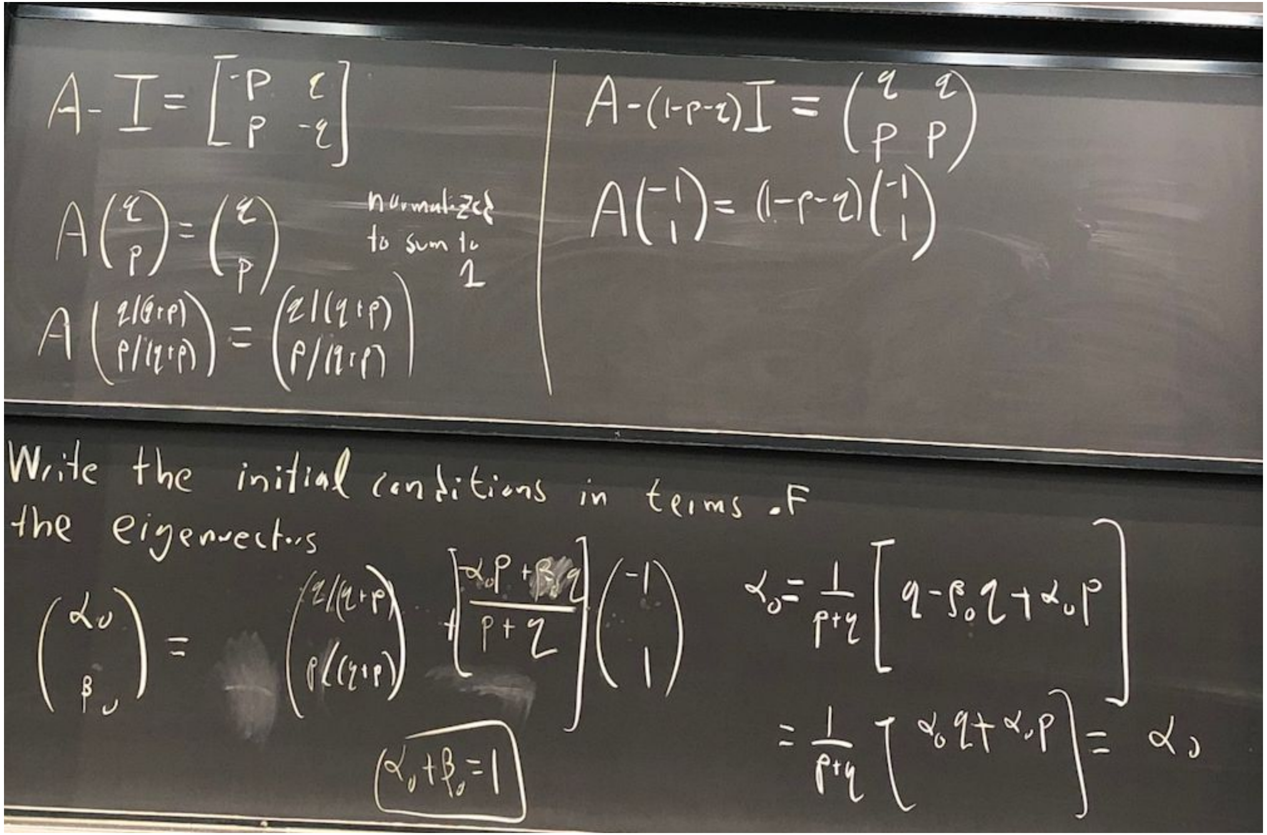
Bikes on day n



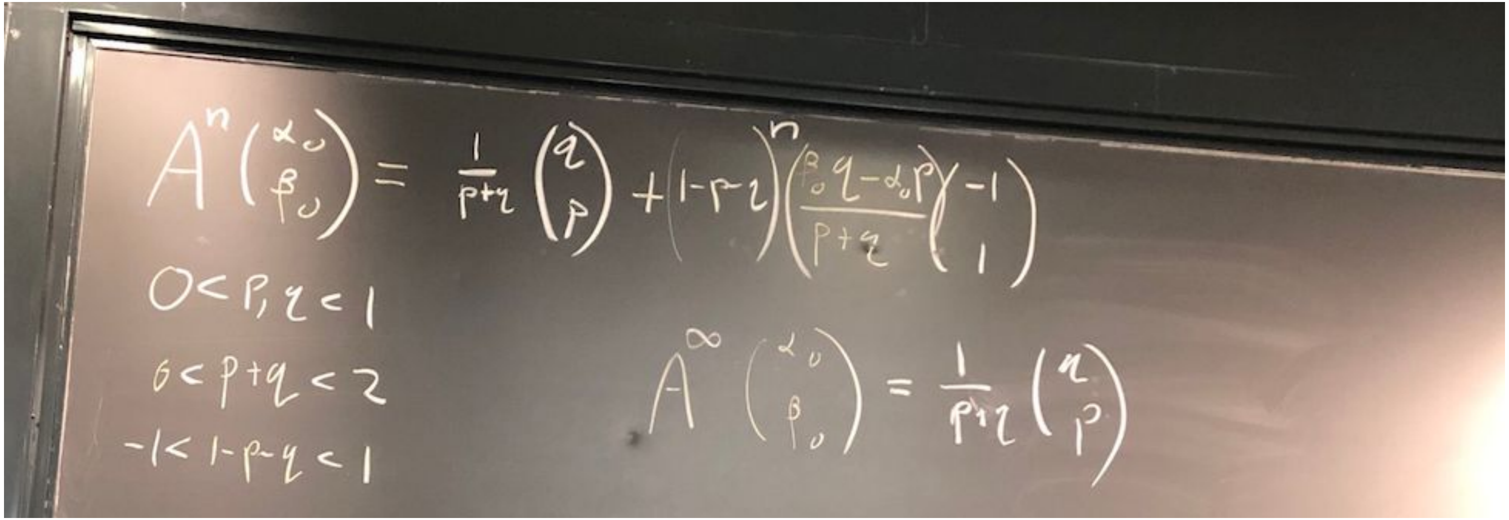
Eigenvalues



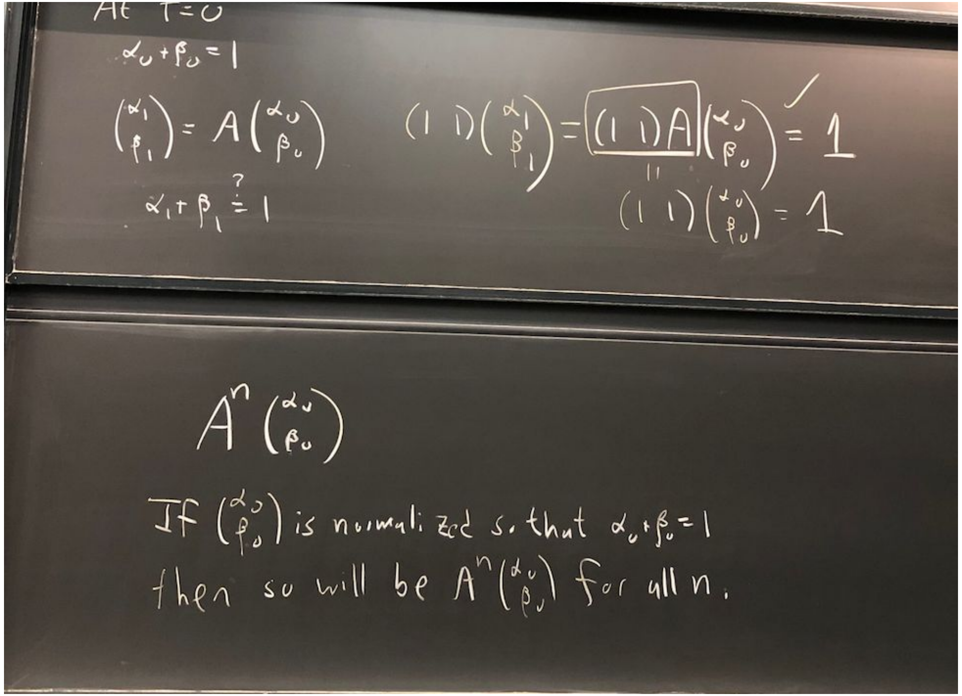
Eigenvectors



Explicit formula for nth state and steady state



Word about normalization



With markov, it’s often preferred to have vectors sum to 1 then have sum of squares to 1. It is an arbitrary normalization but it is convenient.

Markov Analysis of Chutes and Ladders

Prof Steven Johnson has put together a great notebook analyzing the kids board game of chutes and ladders with Markov matrices

A Julia notebook for Markov Matrices by Prof Johnson

35. Positive Definite Matrices

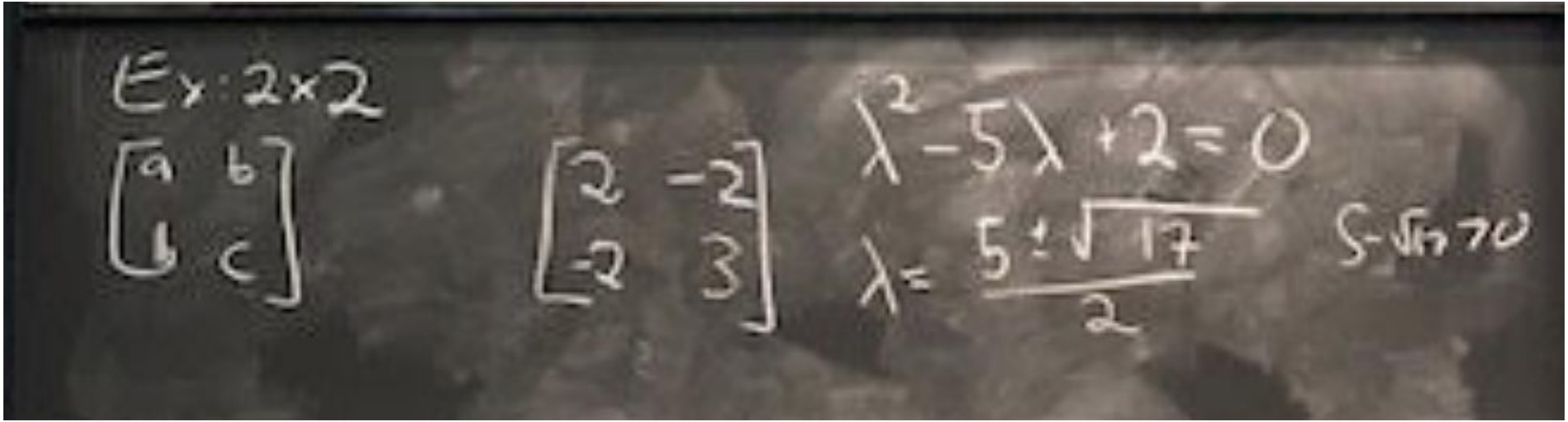
A symmetric matrix is said to be positive definite if all of its eigenvalues are positive.

Warning: There is an extended definition that says a matrix A is positive definite if the so-called symmetric part: $A+A^T$ is positive definite. We will stick to the first definition and if we ever use the general definition we will be clear about it.

Wikipedia starts with the same definition as ours, but also discusses the extended definition. Wolfram mathworld does it the other way around.

More notes coming...

Example



5 - sqrt(17) > 0 so the two eigenvalues are positive

2x2 Pos Def Matrix Test

If $a>0$ and $ac-b^2>0$ then $S=\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is pos def

If S is pos def, then $a>0$ + $ac-b^2>0$

Proof: Assume $\underline{a>0}$ + $ac>b^2$, prove that λ_1, λ_2 are positive

$|S| = ac-b^2 > 0$

$\quad \| $

$\lambda_1 \lambda_2$

$c > \frac{b^2}{a} \geq 0 \quad \Rightarrow \lambda_1 > 0$

$\qquad\qquad\qquad\quad \lambda_2 > 0$

$\Rightarrow c > 0$

$\Rightarrow a + c = \lambda_1 + \lambda_2 > 0$

Assume $\lambda_1 > 0$, $\lambda_2 > 0$, prove that $ac-b^2>0$ + $a>0$

$ac-b^2 = \lambda_1 \lambda_2 > 0$

For sure $a+c>0$ since $a+c = \lambda_1 + \lambda_2$

We can rule out $a<0$, $c<0$. We have $ac > b^2 \geq 0$

so we can't have $a>0$ + $c \leq 0$ or $a \leq 0$ + $c > 0$

$\Rightarrow a>0$, $c>0$

$n=2$ $LU + LDL^T$ pos def

$$S = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ b/a & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} a & b \\ 0 & \frac{ac-b^2}{a} \end{pmatrix}}_{U}$$

$\det S = (\det L)(\det U)$

$= 1 \cdot a \times U_{22}$

$$= \begin{pmatrix} 1 & 0 \\ b/a & 1 \end{pmatrix} \begin{pmatrix} a & \\ & \frac{ac-b^2}{a} \end{pmatrix} \begin{pmatrix} 1 & b/a \\ 0 & 1 \end{pmatrix} = LDL^T$$

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ b/a & 1 \end{pmatrix} \begin{pmatrix} \sqrt{a} & \\ & \sqrt{\frac{ac-b^2}{a}} \end{pmatrix}$$

$S = \tilde{L} \tilde{L}^T$ Cholesky Decomposition

lower but not unit

For $n=2$

$$\boxed{D = \begin{bmatrix} a & \\ & \frac{ac-b^2}{a} \end{bmatrix}}$$

S is SPD

$\Downarrow$

In general D is diagonal with pos diag for all n

Equivalent Conditions for Positive Definite Matrices

1. All eigenvalues > 0
2. S = LU diagonal of U positive (L unit lower triangular)
3. $S = LDL^T$ diagonal D all positive (L unit lower triangular)
4. All n upper left determinants positive
5. Energy : $x^TSx > 0$ unless $x=0$
6. $S = A^TA$ for A with independent columns
7. The compact SVD = the full SVD = an eigenfactorization

The ultimate “if and only if”. You can find out eigenvalues are positive without even knowing their values!

Proof of the equivalence

(we did not prove the equivalence in this class, but it is good to mention a few)

Proof that (1) $\Rightarrow$ (5)

If a symmetric matrix S has positive eigenvalues it can be factored $S=Q\Lambda Q^T$, with the diagonal of Λ positive. Then $x^TSx = x^TQ\Lambda Q^Tx = \Sigma\, \lambda_i(Q^Tx)_i^2$. If x is not 0, (Q^Tx) is not 0, and the sum must be positive.

Proof that (5) $\Rightarrow$ (1)

If $Sx = \lambda x$ then $\lambda = x^TSx \,/\, x^Tx > 0$

That (7) and (1) equivalent is straight from the definitions.

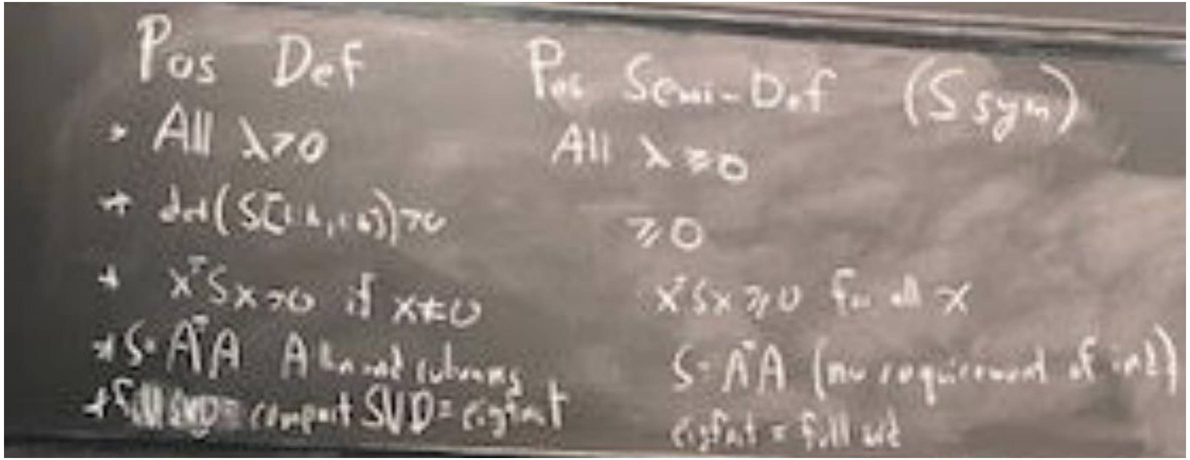
More Proofs of equivalence

(6) ⇒ (5): $x^T S x = ||Ax||^2 > 0$ unless $x = 0$.

Other arguments a bit trickier and will not be covered.

36. Positive Semidefinite and the mean removing projection

Semidefinite relaxes positve (>) to non-negative (≥), etc:

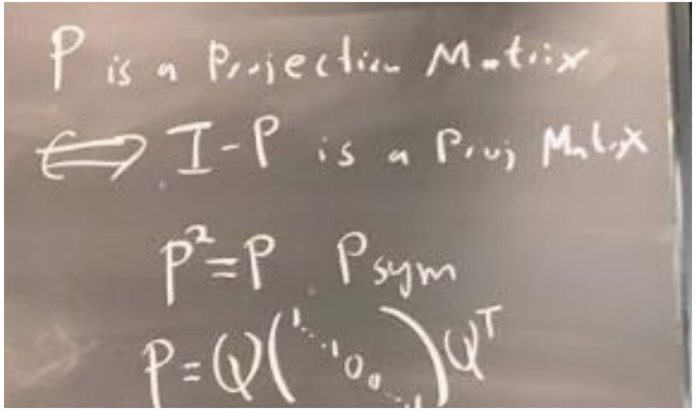


For semi definite $S=A^TA$ does not require A be independent, not does it require that the full SVD be the compact SVD

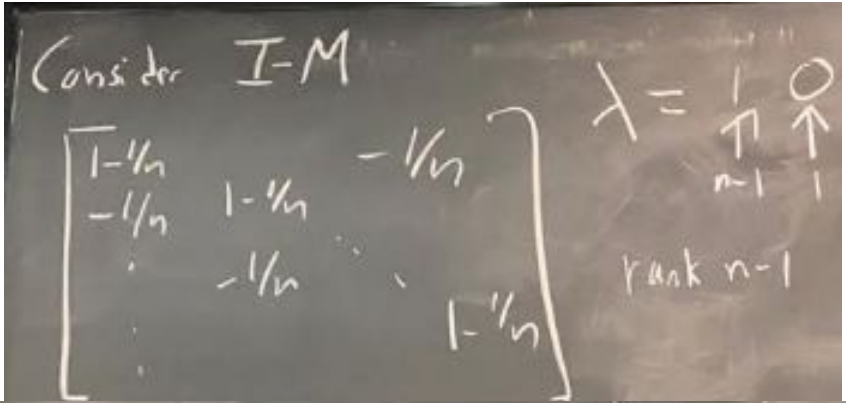
The nxn matrix M with all entries 1/n

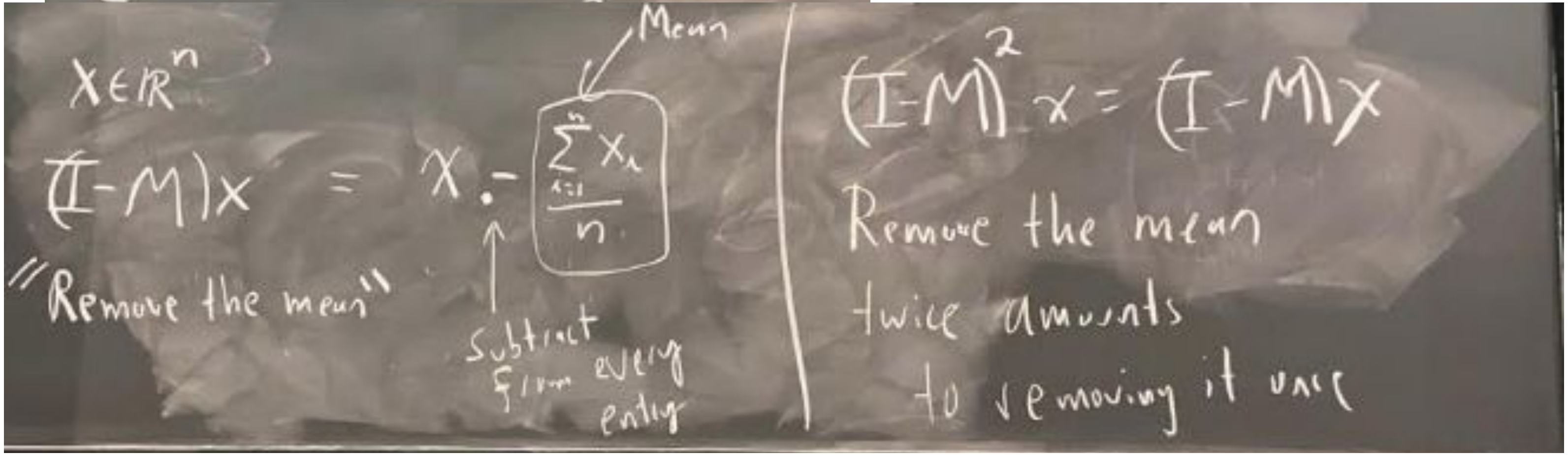
- Symmetric
- Semi-Definite
- Markov
- Projection Matrix
- Rank 1
- Eigenvalues: one 1, rest 0

Reminder about projection matrices in general

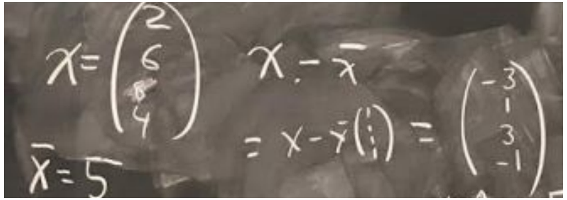


The mean subtracting matrix I-M

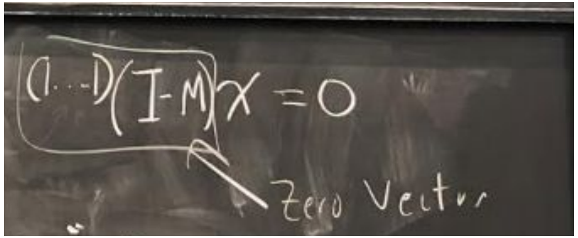




Example of Mean subtracting



← This vector is “mean free” (it sums to 0)



← Why is (I-M)x mean free? Multiplication on the left by the ones vector proves it.

37. Linear Algebra for Statistics

The normal distribution -- an idealized random number

The “standard” normal is generated in julia with randn()

0. The normal distribution

```
randn()
```

0.440810677059617

```
randn()
```

-0.9012911518994019

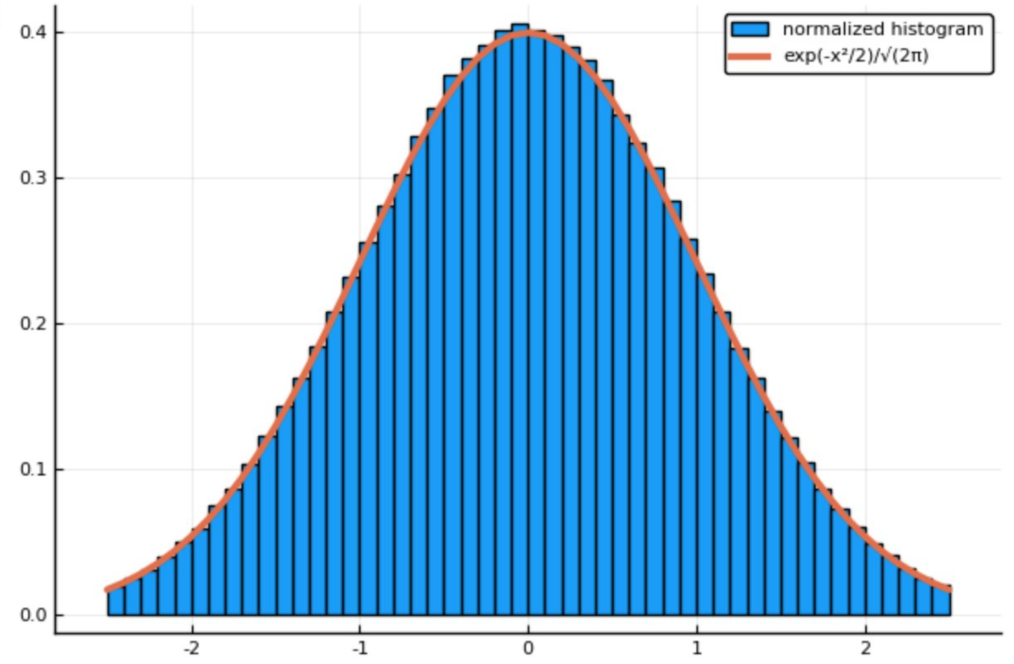
```
randn()
```

1.2505464003852211

Every time you type randn() you get a new scalar random number out. This number may be positive or negative.

Histogramming the standard normal

```
x = -2.5:.1:2.5; N = 1_000_000
histogram(randn(N), bins=x, normalized=true, label="normalized histogram")
plot!(x,exp.(-x.^2/2)./√(2π), linewidth=3, label="exp(-x²/2)/√(2π)")
```



A normalized histogram counts the number of “hits” in each interval and rescales to make the total area of the bars 1. Julia does this automatically for you with the normalized=true option to histogram in the plots package.

If you did this yourself, you would divide the count by (N*h) so the total sum of the verticals would be 1/h, which multiplied by the width would give 1.

On the left, N = total samples = 1,000,000.
And h = bin width = .1 .

The standard normal has mean 0 and variance 1.

Here are some experiments, taking the mean of 100,000,000 normals and the mean square of 100,000,000 normals.

```
N = 100_000_000
x = randn(N)
sum(x)/N, sum(x.^2)/N
```

```
(1.3429369941805908e-5, 0.9999799972744179)
```

```
x = randn(N)
sum(x)/N, sum(x.^2)/N
```

```
(1.2607419023385073e-5, 1.000060923953871)
```

```
x = randn(N)
sum(x)/N, sum(x.^2)/N
```

```
(5.105633150334836e-5, 0.9999638965369215)
```

```
x = randn(N)
sum(x)/N, sum(x.^2)/N
```

```
(-0.00021861216321156037, 0.9999782960223332)
```

```
x = randn(N)
sum(x)/N, sum(x.^2)/N
```

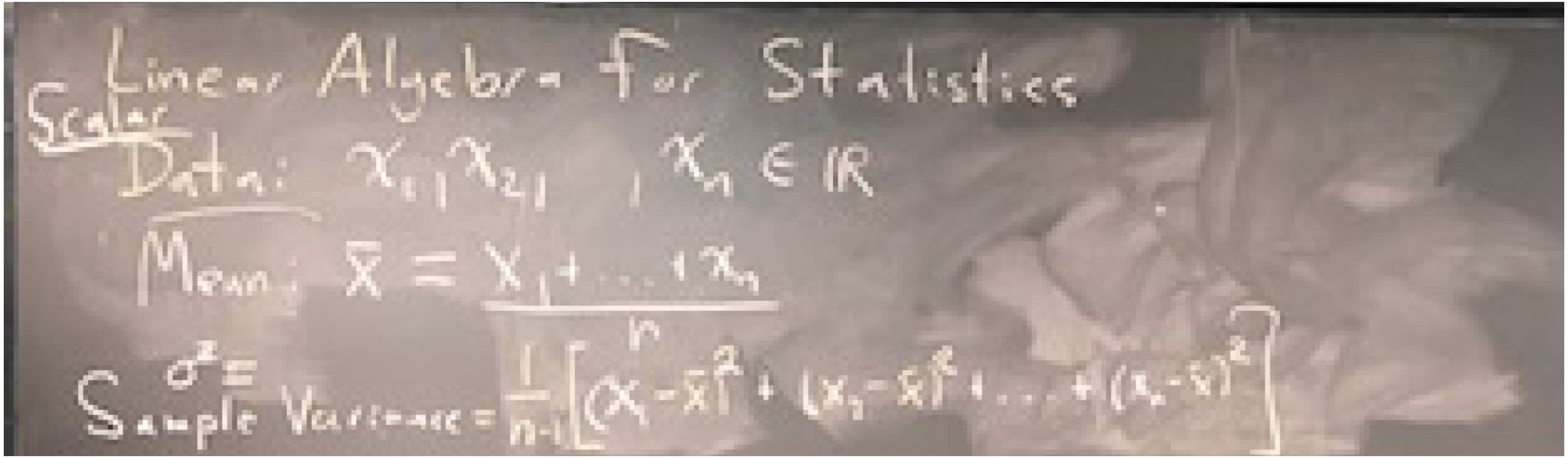
```
(-3.2452654789061673e-6, 1.0000202384290602)
```

We notice through the experiment that on average a standard normal is 0 and the square is 1 at least to 5ish decimal places.

The normal with mean μ and variance σ^2

The random variable σ * randn() + μ is a normal with mean μ and if you subtract the mean the expected square is σ^2 .

Scalar data, mean and sample variance

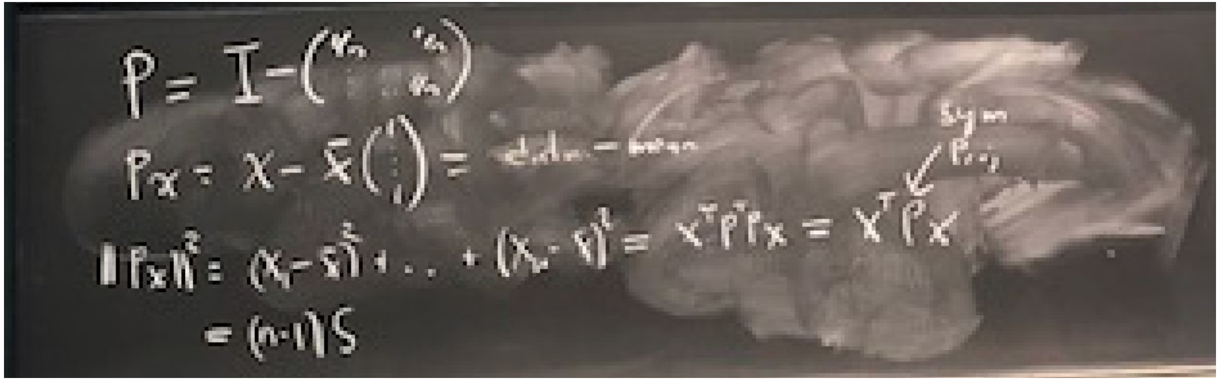


The mean subtracting matrix

P = the mean subtracting matrix (symmetric and a projection)

Px = data - mean

$x^T Px = ||Px||^2$ = is the numerator in sample variance



randn(n) vector of n ind standard normals

Any orthogonal Q (non-random)

Q * randn(n) has the same distribution as a random object

$$P = Q \Lambda Q^T \quad \text{eigen}$$

$$\Lambda = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} \text{orth to } 1\text{s} & 1/\sqrt{n} \end{pmatrix}$$

$$x^T P x = (x^T Q) \Lambda (Q^T x)$$

$$= \underbrace{(Q^T x)_1^2}_{1} + \underbrace{(Q^T x)_2^2}_{1} + \dots + (Q^T x)_{n-1}^2$$

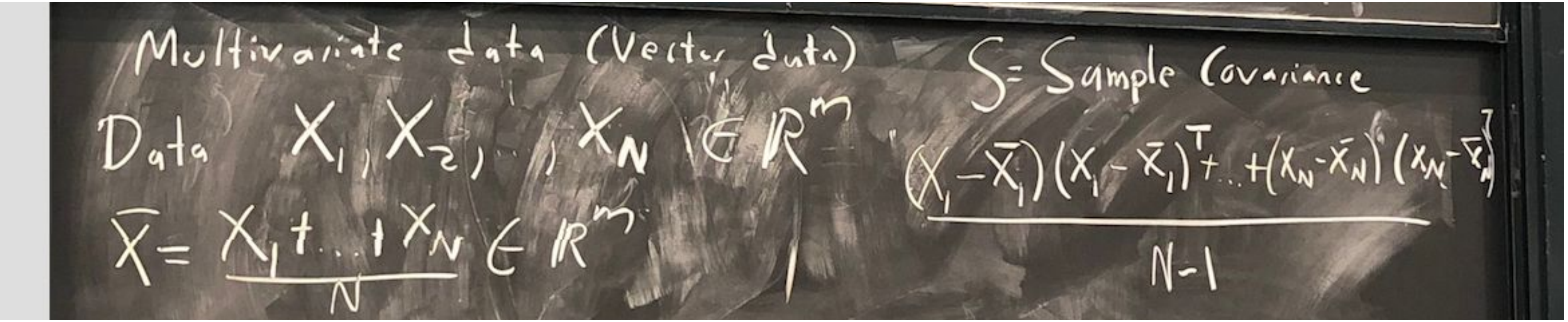
1st component of an orth times randn(n)

$x^T P x$ on average is $\underline{n-1}$

$\frac{1}{n-1} x^T P x$ = sample variance

$= 1$ correct for normals

Multivariate (vector data)

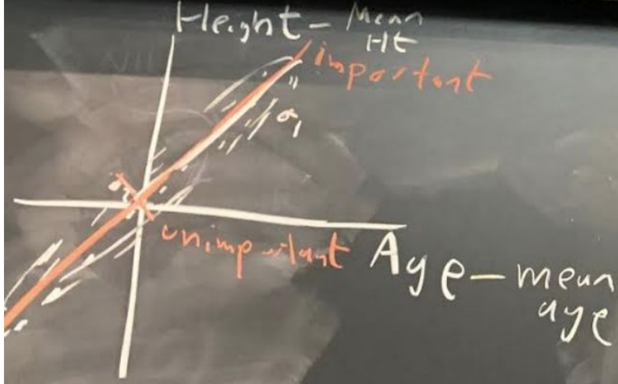


Idealized:

Mult Normal $\quad \underset{\substack{\uparrow \\ m\times m}}{A} * \text{randn}(m) + \underset{\substack{\uparrow \\ m}}{\mu}$

Mean $\mu \in \mathbb{R}^m$

Covariance $C = A^T A$



Ideal:

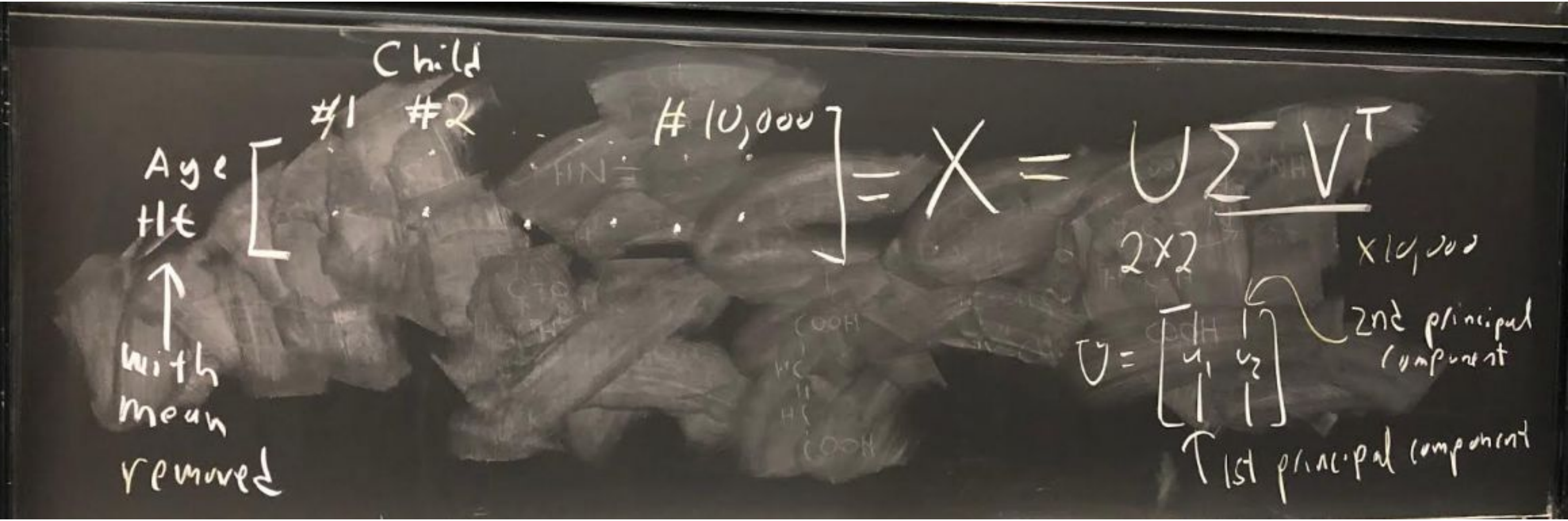
$A = U \Sigma V^T$

$$\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}$$

$\uparrow$ imp $\quad \uparrow$ unimp

$m \times m$

You decide which σ_i to keep as important



If you take the svd of the data matrix, the U are the principal directions. The singular values squared are the variance due to those directions.

The high variance directions are hoped to represent the important information.