# Introduction to Matrix Calculus

Zayn Patel

# Brief reminders about linear transformations as matrices

- We can define *any* input and output basis we want as long as there is a corresponding change of basis matrix (and it's invertible) to transform between bases.

# Brief reminders about linear transformations as matrices

- We can define *any* input and output basis we want as long as there is a corresponding change of basis matrix (and it's invertible) to transform between bases.
- We can define a linear map f: $R^m$ -> $R^n$ if m > n, m < n, m = n as long as the transformation matrix between these spaces m and n shares the same rank.

# Brief reminders about linear transformations as matrices

- We can define *any* input and output basis we want as long as there is a corresponding change of basis matrix (and it's invertible) to transform between bases.
- We can define a linear map f: $R^m$ -> $R^n$ if m > n, m < n, m = n as long as the transformation matrix between these spaces m and n shares the same rank.
- When we define a matrix A, the entries of this matrix transform the standard basis[1] vectors, this is how we get the transformation.

# Brief reminders about linear transformations as matrices

- We can define *any* input and output basis we want as long as there is a corresponding change of basis matrix (and it's invertible) to transform between bases.
- We can define a linear map f: $R^m$ -> $R^n$ if m > n, m < n, m = n as long as the transformation matrix between these spaces m and n shares the same rank.
- When we define a matrix A, the entries of this matrix transform the standard basis vectors, this is how we get the transformation.
- Linear means the transformation satisfies the following two conditions:

$$L[v_1 + v_2] = L[v_1] + L[v_2] \textbf{ and } c[Lv] = L[cv]$$

# Brief reminders about linear transformations as matrices

- On an earlier slide we talked about *why* we can think of linear transformations as matrices. Recall:
  - Start with basis vectors
  - Some transformation (rotation by 90 degrees clockwise) has column vectors which define how these change the basis vectors
  - We put these column vectors into a matrix, A
  - We conclude that this linear transformation (rotation) is represented by the column vectors in A
    - Therefore we can represent this linear transformation as a matrix.

# Brief reminders about linear transformations as matrices

- On an earlier slide we talked about *why* we can think of linear transformations as matrices. Recall:
  - Start with basis vectors
  - Some transformation (rotation by 90 degrees clockwise) has column vectors which define how these change the basis vectors
  - We put these column vectors into a matrix, A
  - We conclude that this linear transformation (rotation) is represented by the column vectors in A
    - Therefore we can represent this linear transformation as a matrix.

**Key idea:** We can generalize this idea to any linear transformation and represent it as a matrix. Rotations, shears, projections are easy transformations to visualize but there are many more.

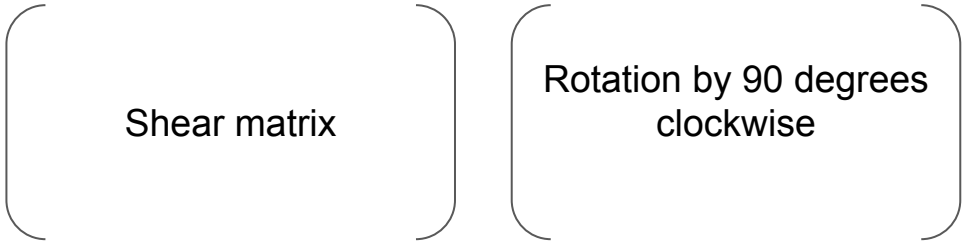
# Brief reminders about linear transformations as matrices

- On an earlier slide we talked about *why* we can think of linear transformations as matrices. Recall:
  - Start with basis vectors
  - Some transformation (rotation by 90 degrees clockwise) has column vectors which define how these change the basis vectors
  - We put these column vectors into a matrix, A
  - We conclude that this linear transformation (rotation) is represented by the column vectors in A
    - Therefore we can represent this linear transformation as a matrix.

**Key idea 2:** Every time you write see a matrix or write one down you should think of it as a linear transformation instead of some m x n set of values.

# Brief reminders about linear transformations as matrices

To make this more concrete let’s think of a 2x2 rotation matrix and a 2x2 shear matrix:



# Brief reminders about linear transformations as matrices

To make this more concrete let's think of a 2x2 rotation matrix and a 2x2 shear matrix:

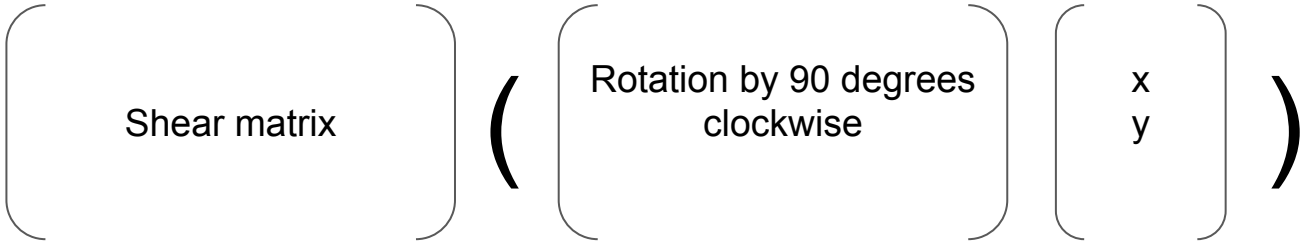
$$\begin{pmatrix} \text{Shear matrix} \end{pmatrix} \quad \begin{pmatrix} \text{Rotation by 90 degrees clockwise} \end{pmatrix}$$

Now if we apply the rotation to a vector in $R^2$ we get a new matrix, R:

$$\begin{pmatrix} \text{Rotation by 90 degrees clockwise} \end{pmatrix} \quad \begin{pmatrix} x \\ y \end{pmatrix} \quad = \quad \begin{pmatrix} \mathbf{R} \end{pmatrix}$$

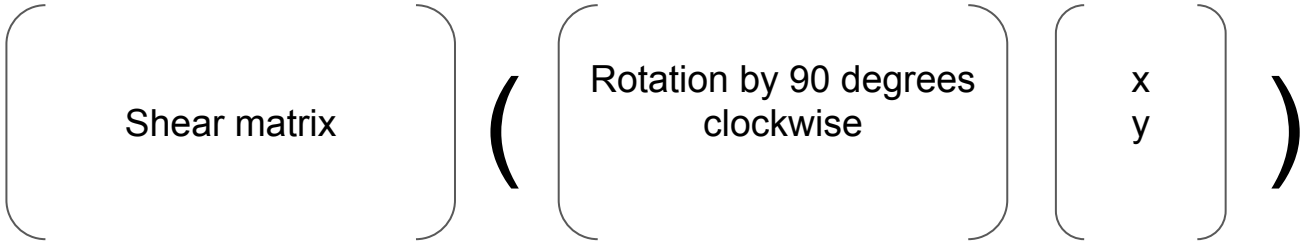
# Brief reminders about linear transformations as matrices

If we apply the shear to this matrix R we can look at this as:

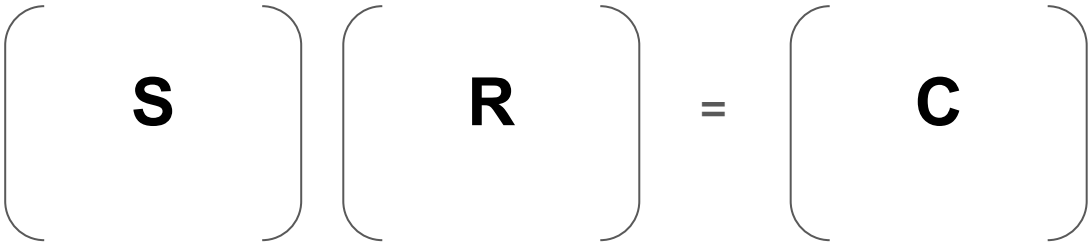


# Brief reminders about linear transformations as matrices

If we apply the shear to this matrix R we can look at this as:

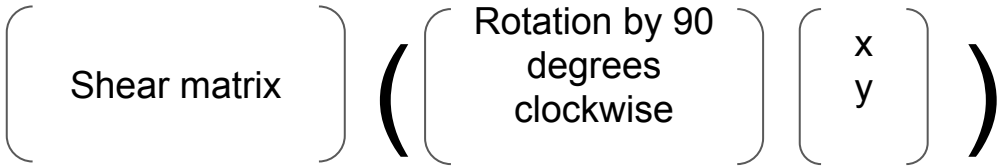



And we can look at this as a matrix product of shear := S and R which gives us C:



# Brief reminders about linear transformations as matrices

If we apply the shear to this matrix R we can look at this as:




And we can look at that as a matrix product of shear := S and R which gives us C:


**Key idea:** When we think of linear transformations as matrices we get **matrix multiplication as matrix composition.** We are applying one transformation to another.

# Brief reminders about linear transformations as matrices

**Key idea cont.:** If we think about the transformation on the previous slide: rotation then shear unless we had a *super special case (?) and it might still be unlikely to work* this transformation is different from shear then rotation.

So:

R @ S ≠ S @ R

# Brief reminders about linear transformations as matrices

**Key idea cont.:** If we think about the transformation on the previous slide: rotation then shear unless we had a *super special case (?) and it might still be unlikely to work* this transformation is different from shear then rotation.

So:

R @ S ≠ S @ R

**Matrix multiplication is not communitive *because* we are applying transformations and the order of these transformations matters.**

*Note: @ is matmul notation in Python which I am using above.*

# Relation to matrix calculus

In this presentation we will be thinking about how to apply rules from differential calculus to matrices. There are two reasons to bring up linear transformations.

# Relation to matrix calculus

In this presentation we will be thinking about how to apply rules from differential calculus to matrices. There are two reasons to bring up linear transformations.

1. We will be thinking of f'(x) as a linear operator. f'(x) will be some object[1], and it will "operate" on an object representing infinitesimal change[2] to produce an output, df. The previous slides motivate transformations from one vector space to another.

[1](scalar, vector, matrix)
[2] [dx] or [dA]

# Relation to matrix calculus

In this presentation we will be thinking about how to apply rules from differential calculus to matrices. There are two reasons to bring up linear transformations.

1. We will be thinking of f'(x) as a linear operator. f'(x) will be some object[1], and it will "operate" on an object representing infinitesimal change[2] to produce an output, df. The previous slides motivate transformations from one vector space to another.
2. While we will derive rules like the power rule and chain rule for vectors and matrices the fact that we are not working with scalars, only, for calculus means **communitivity matters**. The theory behind linear transformations built up to a nice result of why matmul does not commute.

[1](scalar, vector, matrix)
[2] [dx] or [dA]

# Derivatives as linear operators: scalar valued functions

- The definition of the derivative we will use in this presentation is:
  $df = f(x + dx) - f(x) = f'(x)\,[dx]$

  Note: **Every time you see f'(x) is a linear operator I mean f'(x) maps [dx] to df.**

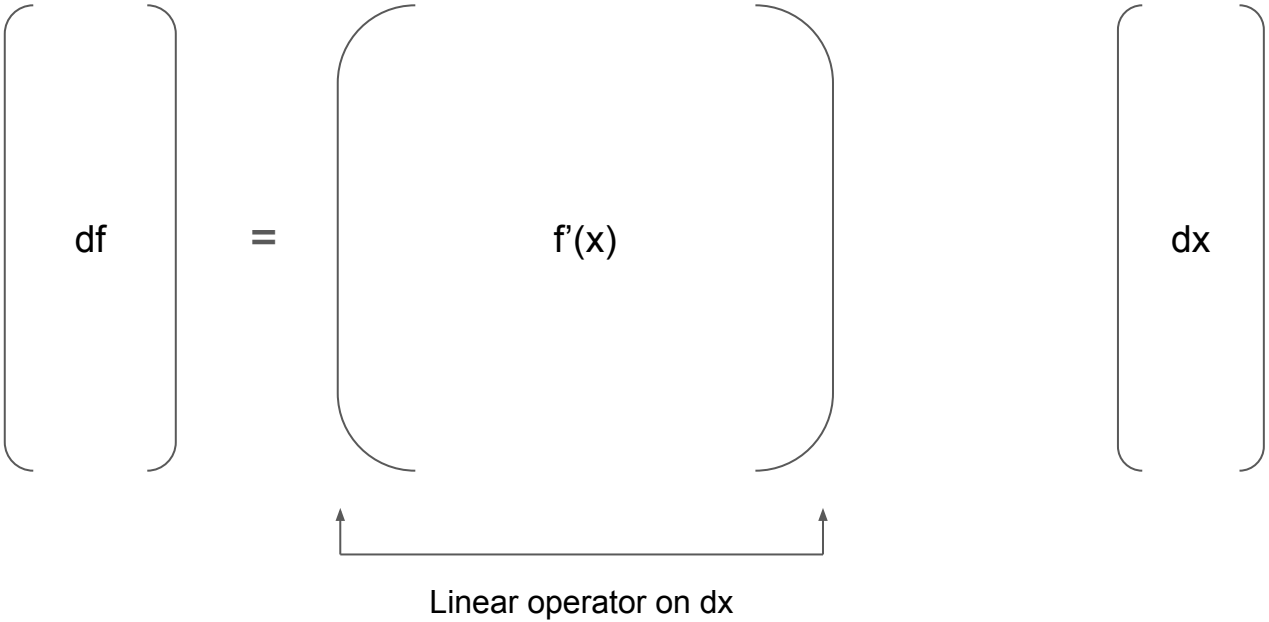
# Derivatives as linear operators: scalar valued functions

- The definition of the derivative we will use in this presentation is:
  $df = f(x + dx) - f(x) = f'(x)\,[dx]$

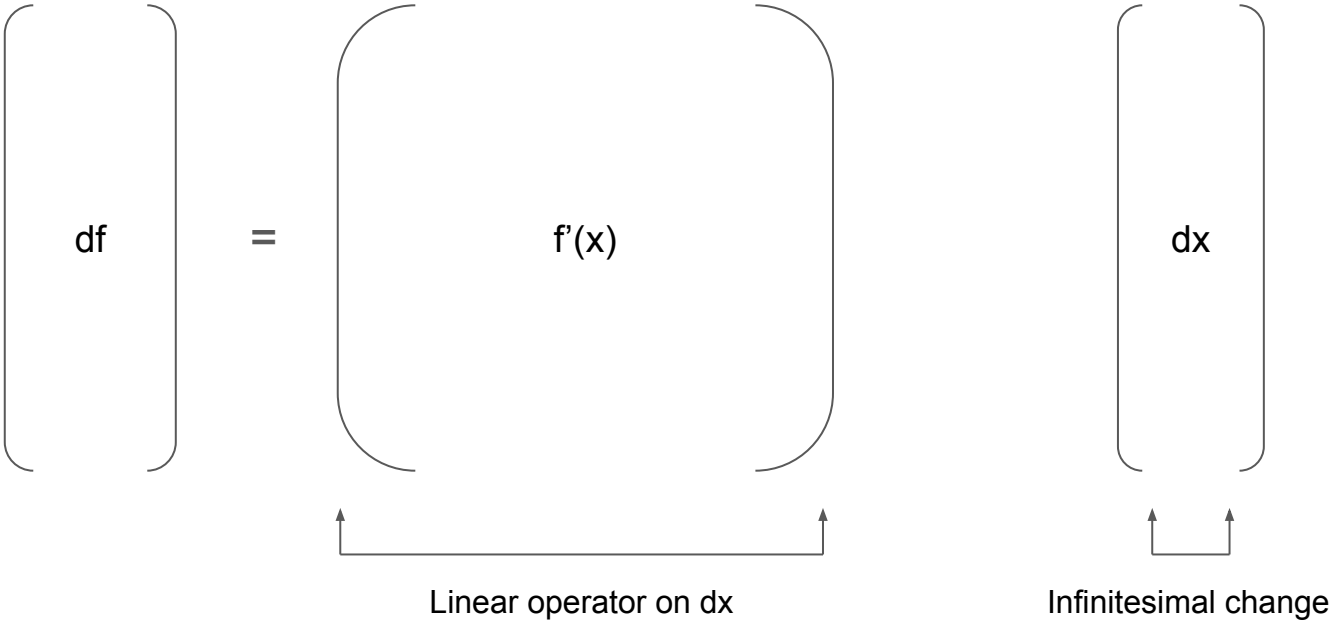
  Note: **Every time you see f'(x) is a linear operator I mean f'(x) maps [dx] to df.**

- In the definition of the derivative we defined [dx] to be a column vector. (Recall that since [dx] is a vector it represents some infinitesimal change in a direction). If we want a scalar output f'(x) needs to be in the form of a row vector. This row vector will be called $(\text{grad } f)^T$.
  - Why? Because x (input to the f'(x)) is a column vector itself. When we apply the derivative to the x we are taking *all partial derivatives of the components*. So this *is the gradient* as learned in calculus. Now because we are using vectors we transpose it so the dimensions match to get a scalar.

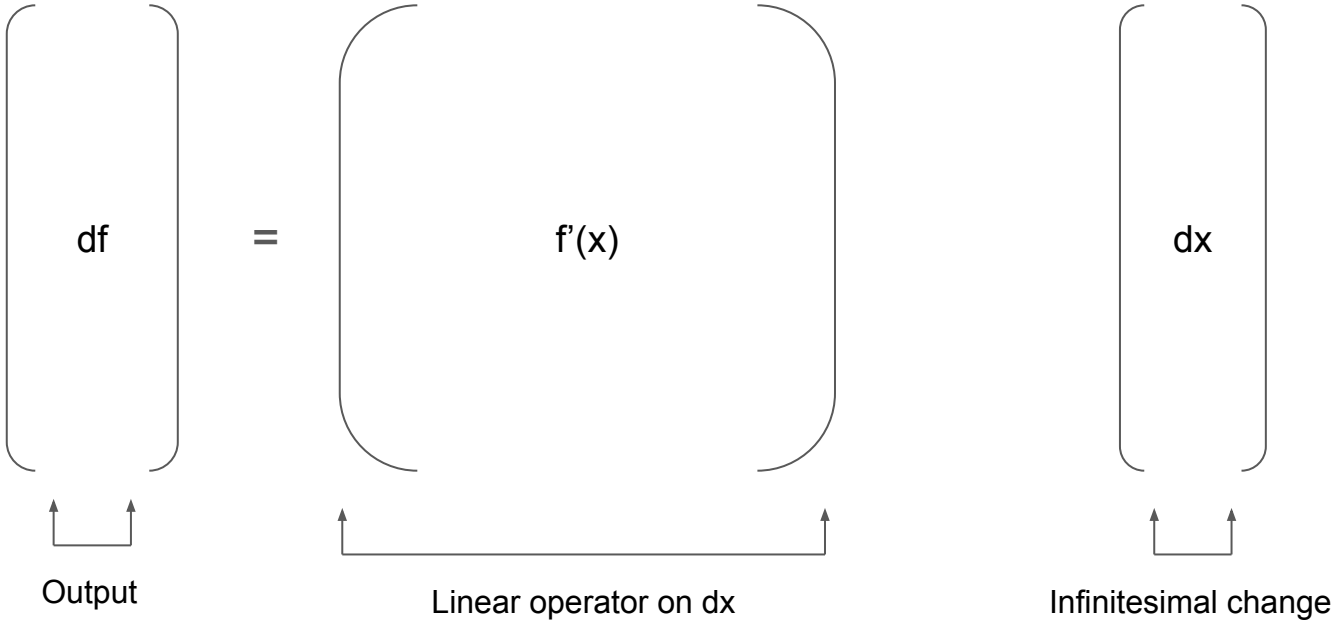
# Derivatives as linear operators: vector valued functions



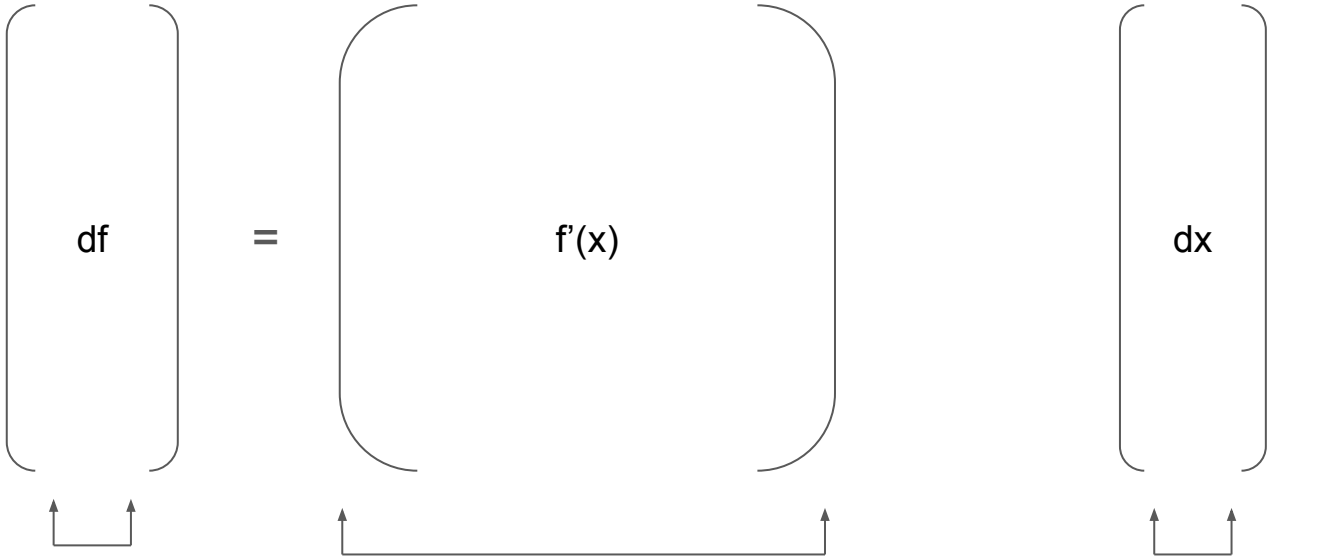
# Derivatives as linear operators: vector valued functions



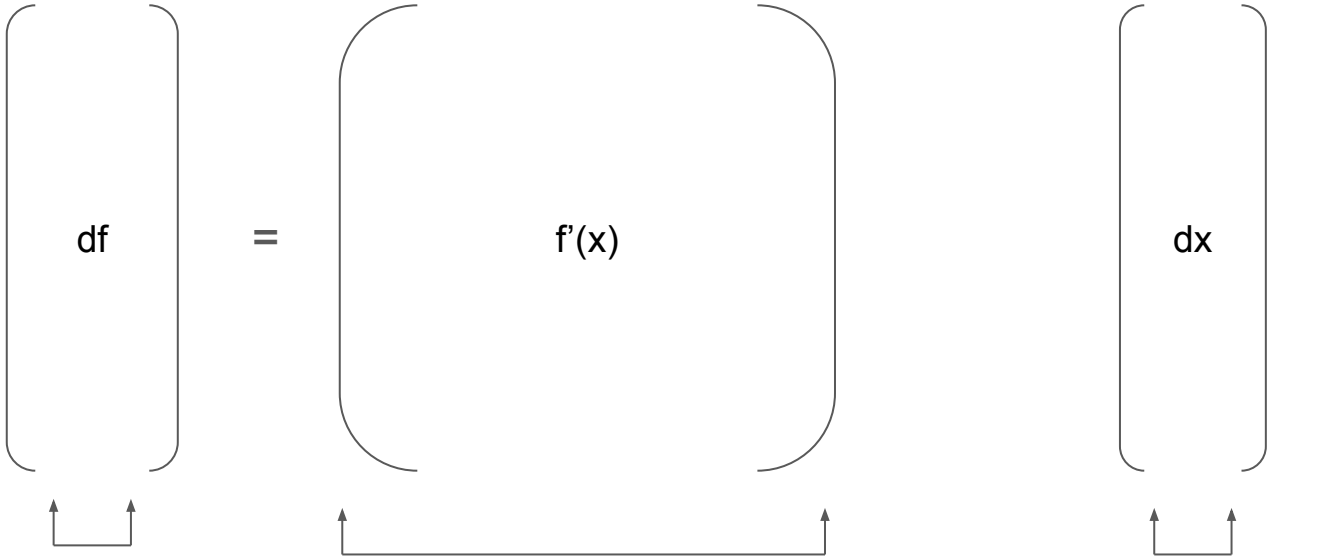
# Derivatives as linear operators: vector valued functions



# Derivatives as linear operators: vector valued functions



# Derivatives as linear operators: vector valued functions



# Why the Jacobian is the linear operator for vec-valued fcn

- One perspective is dimension. If we have [dx] as a column vector with n components and [df] as a column vector with m components *then* we need a matrix (m x n) to *operate* on [dx] to produce [df]. This is **defined** to be the Jacobian.
    - Recall from Calc III that the Jacobian is a matrix with the partial derivatives w.r.t each component in each direction.

**Key idea:** First, our linear operator thinking is important because we don't care what the grad or Jacobian components are. *Just that these objects take [dx] and map it to a different vector space with [df].*

# Why the Jacobian is the linear operator for vec-valued fcn

- One perspective is dimension. If we have [dx] as a column vector with n components and [df] as a column vector with m components *then* we need a matrix (m x n) to *operate* on [dx] to produce [df]. This is **defined** to be the Jacobian.
    - Recall from Calc III that the Jacobian is a matrix with the partial derivatives w.r.t each component in each direction.

**Key idea:** First, our linear operator thinking is important because we don't care what the grad or Jacobian components are. *Just that these objects take [dx] and map it to a different vector space with [df].* Second, we can use the Jacobian to think about the chain rule for matrices (or vectors).

# *Very quick reminder*: Jacobian matrix in a picture

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Source: Wikipedia (Jacobian matrix and determinant)

# Using the Jacobian in the chain rule for matrices

Let's first derive the chain rule for functions on arbitrary vector spaces. If f(x) = g(h(x)) and g,h are matrices, x is a vector:

*df* = f(x + dx) - f(x)

= g'(h(x)) [h(x + dx) - h(x)] ←

We rewrite this using vector space notation with *ideas about chain rule for scalar functions.*

# Using the Jacobian in the chain rule for matrices

Let's first derive the chain rule for functions on arbitrary vector spaces. If f(x) = g(h(x)) and g,h are matrices, x is a vector:

*df* = f(x + dx) - f(x)

= g'(h(x)) [h(x + dx) - h(x)] ← We rewrite this using vector space notation with *ideas about chain rule for scalar functions.*

= g'(h(x)) [h(x + dx) - h(x)] ← Apply the definition of the derivative for h(x). Recall: *dh* = h(x + dx) - h(x) = h'(x)[dx].

# Using the Jacobian in the chain rule for matrices

Let's first derive the chain rule for functions on arbitrary vector spaces. If f(x) = g(h(x)) and g,h are matrices, x is a vector:

*df* = f(x + dx) - f(x)

= g'(h(x)) [h(x + dx) - h(x)] ← We rewrite this using vector space notation with *ideas about chain rule for scalar functions.*

= g'(h(x)) [h'(x)[dx]] ← Apply the definition of the derivative for h(x). Recall: *dh* = h(x + dx) - h(x) = h'(x)[dx].

# Using the Jacobian in the chain rule for matrices

Let's first derive the chain rule for functions on arbitrary vector spaces. If f(x) = g(h(x)) and g,h are matrices, x is a vector:

*df* = f(x + dx) - f(x)

= g'(h(x)) [h(x + dx) - h(x)] ← We rewrite this using vector space notation with *ideas about chain rule for scalar functions.*

= g'(h(x)) [h'(x)[dx]] ← Apply the definition of the derivative for h(x). Recall: *dh* = h(x + dx) - h(x) = h'(x)[dx].

= g'(h(x)) h'(x)[dx] ← Rewrite this so we clearly see a **composition of g' and h' matrices.**

# Using the Jacobian in the chain rule for matrices

Let's first derive the chain rule for functions on arbitrary vector spaces. If f(x) = g(h(x)) and g,h are matrices, x is a vector:

*df* = f(x + dx) - f(x)

= g'(h(x)) [h(x + dx) - h(x)] ← We rewrite this using vector space notation with *ideas about chain rule for scalar functions.*

= g'(h(x)) [h'(x)[dx]] ← Apply the definition of the derivative for h(x). Recall: *dh* = h(x + dx) - h(x) = h'(x)[dx].

= g'(h(x)) h'(x)[dx] ← Rewrite this so we clearly see a **composition of g' and h' matrices.**

**Key idea:** Our output, *df* (a.k.a f'(x) dx) is an m x n Jacobian matrix. So, g'(h(x)) is an m x p Jacobian and h'(x) is a p x n Jacobian. See notebook for a numerical calculation of this product.

# Calculating the derivative of $A^2$ and $A^3$

Let’s transition to show our first result of calculus on matrices by differentiating $A^2$.

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our first result of calculus on matrices by differentiating $A^2$.

Single variable calculus version

$f(x) = x^2$ ← Define function

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our first result of calculus on matrices by differentiating $A^2$.

Single variable calculus version

$f(x) = x^2$ ← Define function.

$f'(x) = dx\ x + x\ dx$ ← Recall product rule is u' * v + u * v'

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our first result of calculus on matrices by differentiating $A^2$.

Single variable calculus version

$f(x) = x^2$ ← Define function.

$f'(x) = dx\ x + x\ dx$ ← Recall product rule is u' * v + u * v'

Matrix calculus version

$f(x) = A^2 = A * A$ ← Define function.

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our first result of calculus on matrices by differentiating $A^2$.

Single variable calculus version

$f(x) = x^2$ ← Define function.

$f(x) = dx * x + x * dx$ ← Recall product rule is u' * v + u * v'

Matrix calculus version

$f(x) = A^2 = A * A$ ← Define function.

$f'(x) = dA\,A + A\,dA$ ← Same product rule as before but in single variable calculus the dx * x can have different orders because these commute. dA * A can't change order.

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

Single variable calculus version

$f(x) = x^3 = x*(x*x)$ ← Define function.

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

<u>Single variable calculus version</u>

$f(x) = x^3 = x*(x*x)$ ← Define function.

$f'(x) = (x*x)' * x + (x*x) * dx$ ← Recall product rule is u' * v + u * v'

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

<u>Single variable calculus version</u>

$f(x) = x^3 = x*(x*x)$ ← Define function.

$f'(x) = (x*x)' * x + (x*x) * dx$ ← Recall product rule is u' * v + u * v'

$f'(x) = (x*dx + dx*x) * x + (x*x) * dx$ ← Apply the product rule again to (x*x)'

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

<u>Single variable calculus version</u>

$f(x) = x^3 = x*(x*x)$ ← Define function.

$f'(x) = (x*x)' * x + (x*x) * dx$ ← Recall product rule is u' * v + u * v'

$f'(x) = (x*dx + dx*x) * x + (x*x) * dx$ ← Apply the product rule again to (x*x)'

$f'(x) = x^2\ dx + dx\ x^2 + x^2\ dx$ ← Simplify (assume * between terms still)

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

<u>Single variable calculus version</u>

$f(x) = x^3 = x*(x*x)$ ← Define function.

$f'(x) = (x*x)' * x + (x*x) * dx$ ← Recall product rule is u' * v + u * v'

$f'(x) = (x*dx + dx*x) * x + (x*x) * dx$ ← Apply the product rule again to (x*x)'

$f'(x) = x^2\ dx + dx\ x^2 + x^2\ dx$ ← Simplify (assume * between terms still)

$f'(x) = x^2\ dx + x^2\ dx + x^2\ dx = 3x^2\ dx$ ← Final version which aligns with our "power rule"

# Calculating the derivative of $A^2$ and $A^3$

Let's transition to show our second result of calculus on matrices by differentiating $A^3$.

Matrix calculus version

$f(x) = A^3 = A*(A*A)$

$f'(x) = (A*A)' * A + dA * (A*A)$

$f'(x) = (A*dA + dA*A) * A + dA * (A*A)$

$f'(x) = A^2\, dA + A\, dA\, A + dA\, A^2$

**Key idea:** Communitivity, so far, is the key difference between single variable calculus and matrix calculus. It's why we have A dA A instead of another $A^2$ dA term.

# Jacobians and Kronecker Products

We calculated the derivative of $A^2$ and $A^3$ because the rest of our examples in this section rely on the derivative of $A^2$, specifically.

# Jacobians and Kronecker Products

We calculated the derivative of $A^2$ and $A^3$ because the rest of our examples in this section rely on the derivative of $A^2$, specifically. In this notebook, which I recommend you reference for an introduction to this section, I do an example that confirms the derivative of $A^2$ (in vectorized form) is equal to the Jacobian multiplied by dA (in vectorized form).

# Jacobians and Kronecker Products

We calculated the derivative of $A^2$ and $A^3$ because the rest of our examples in this section rely on the derivative of $A^2$, specifically. In this notebook which I recommend you reference for an introduction to this section I do an example that confirms the derivative of $A^2$ (in vectorized form) is equal to the Jacobian multiplied by dA (in vectorized form).

This is because $f(A) = A^2$. So $f'(A) = d(A^2)$.

# Jacobians and Kronecker Products

We calculated the derivative of $A^2$ and $A^3$ because the rest of our examples in this section rely on the derivative of $A^2$, specifically. In this notebook which I recommend you reference for an introduction to this section I do an example that confirms the derivative of $A^2$ (in vectorized form) is equal to the Jacobian multiplied by dA (in vectorized form).

This is because $f(A) = A^2$. So $f'(A) = d(A^2)$.

Thinking of the Jacobian as a linear operator we can write this as: $f'(A)[dA]$.

# Jacobians and Kronecker Products

We calculated the derivative of $A^2$ and $A^3$ because the rest of our examples in this section rely on the derivative of $A^2$, specifically. In this notebook which I recommend you reference for an introduction to this section I do an example that confirms the derivative of $A^2$ (in vectorized form) is equal to the Jacobian multiplied by dA (in vectorized form).

This is because $f(A) = A^2$. So $f'(A) = d(A^2)$.

Thinking of the Jacobian as a linear operator we can write this as: $f'(A)[dA]$.

And we have previously shown that $f'(A)$ is $dA\,A + A\,dA$.

# Jacobians and Kronecker Products

Now that we have an introductory understanding of the Jacobian for $A^2$ and the Kronecker product let's state a key identity and derive it. We will use this identity to show we can write the Jacobian in Kronecker Product notation.

# Jacobians and Kronecker Products

Now that we have an introductory understanding of the Jacobian for $A^2$ and the Kronecker product let's state a key identity and derive it. We will use this identity to show we can write the Jacobian in Kronecker Product notation.

**Key identity:**

$$(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$$

*Assumption: This identity **requires** compatible sized matrices. We cannot do $BCA^T$ if the dimensions do not match.*

# Jacobians and Kronecker Products

Let’s begin with the RHS and suppose $A = I$.

# Jacobians and Kronecker Products

Let’s begin with the RHS and suppose A = I.

This turns $BCA^T$ into BC. So we have vec(BC).

# Jacobians and Kronecker Products

Let’s begin with the RHS and suppose A = I.

This turns $BCA^T$ into BC. So we have vec(BC).

We want to figure out what BC is so we know what vec(BC) is and we can relate this to the LHS. So it helps to have an understanding of the terms and multiplication of this matrix product. Using some knowledge from slides 12 and 13 we have matrix multiplication/composition. We can think of the multiplication as:

# Jacobians and Kronecker Products

Let's begin with the RHS and suppose A = I.

This turns $BCA^T$ into BC. So we have vec(BC).

We want to figure out what BC is so we know what vec(BC) is and we can relate this to the LHS. So it helps to have an understanding of the terms and multiplication of this matrix product. Using some knowledge from slides 12 and 13 we have matrix multiplication/composition. We can think of the multiplication as:

$B\,(c_1, c_2, c_3, \ldots) = (Bc_1, Bc_2, Bc_3) \Rightarrow \text{vec}(BC) =$

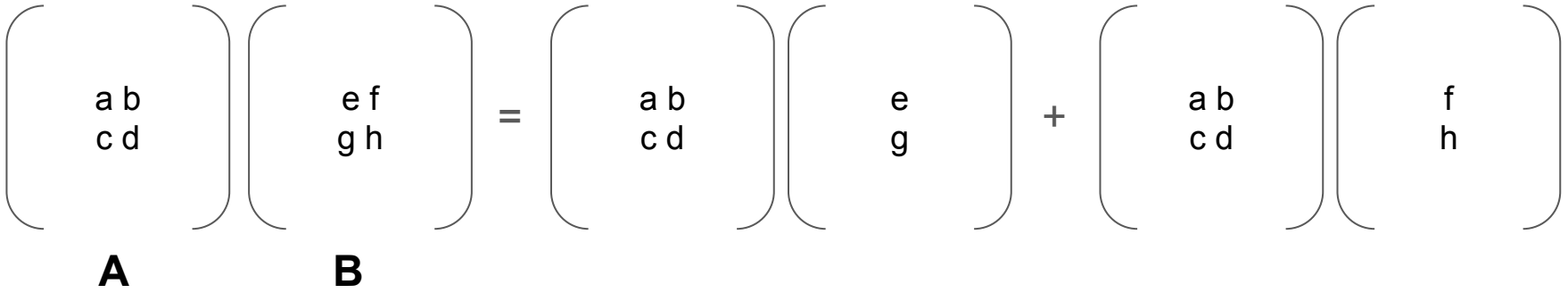
$$\begin{pmatrix} Bc_1 \\ Bc_2 \\ \vdots \end{pmatrix}$$

# Jacobians and Kronecker Products

Just in case this isn’t clear the following is a 2x2 * 2x2 example that uses the same concept as the BC case.

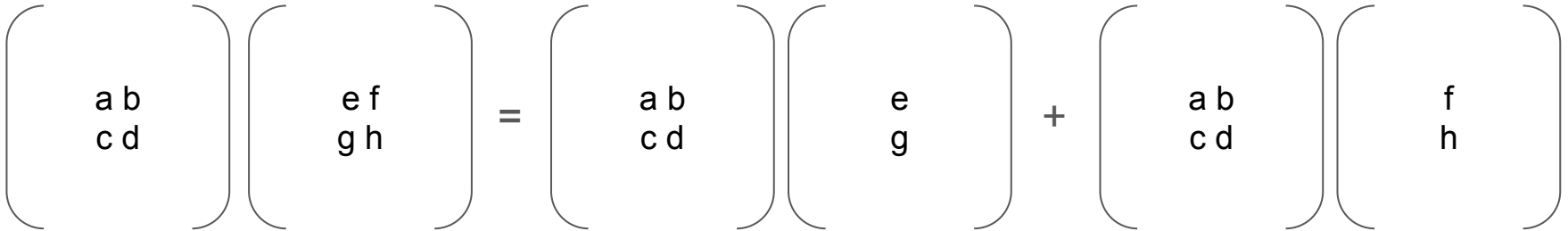
# Jacobians and Kronecker Products

Just in case this isn't clear the following is a 2x2 * 2x2 example that uses the same concept as the BC case.



# Jacobians and Kronecker Products

Just in case this isn't clear the following is a 2x2 * 2x2 example that uses the same concept as the BC case.

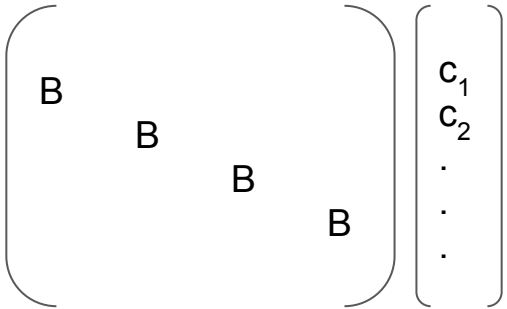



**A** **B**

**Key idea:** We take each column of B and multiply it by the matrix A. This is exactly what we are doing to compute the matrix product BC on the previous slides.

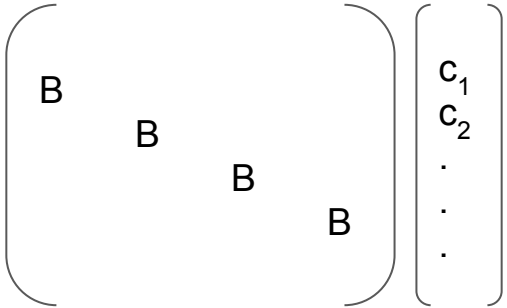
# Jacobians and Kronecker Products

Returning to the vec(BC) example we can express vec(BC) as (I $\otimes$ B) * vec(C).



# Jacobians and Kronecker Products

Returning to the vec(BC) example we can express vec(BC) as $(I \otimes B) * vec(C)$.



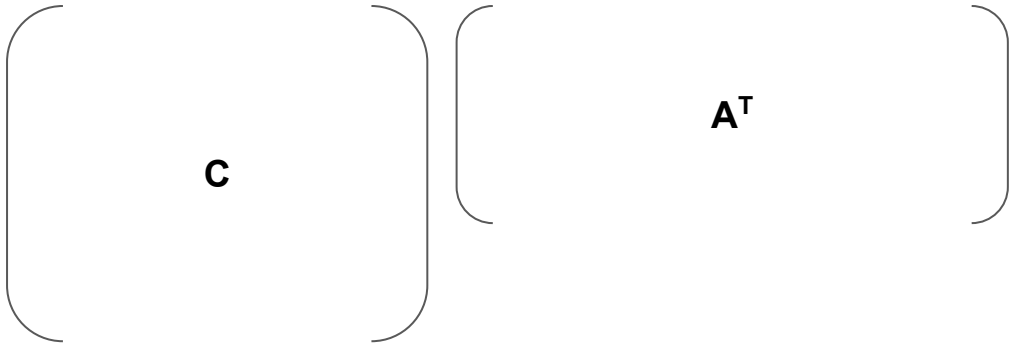

**Key idea:** Beginning with A = I and using facts about matrix product BC gives us: $(I \otimes B)vec(C) = vec(BC)$.

# Jacobians and Kronecker Products

Now let's assume B = I to understand where the $A^T$ term comes from. We will again be interested in understanding the RHS term $CA^T$, without vectorization first, then using $\text{vec}(CA^T)$ to relate it to the RHS.

# Jacobians and Kronecker Products

Now let's assume B = I to understand where the $A^T$ term comes from. We will again be interested in understanding the RHS term $CA^T$, without vectorization first, then using $vec(CA^T)$ to relate it to the RHS.



# Jacobians and Kronecker Products

Let's build our own 2x2 $CA^T$ to understand what this looks like symbolically. Then we can generalize this to arbitrarily sized C and $A^T$.

$$\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \begin{pmatrix} a_1 & a_3 \\ a_2 & a_4 \end{pmatrix} = \begin{pmatrix} (c_1a_1 + c_2a_2) & (c_1a_3 + c_2a_4) \\ (c_3a_1 + c_4a_2) & (c_3a_3 + c_4a_4) \end{pmatrix}$$

# Jacobians and Kronecker Products

Let's build our own 2x2 $CA^T$ to understand what this looks like symbolically. Then we can generalize this to arbitrarily sized C and $A^T$.

$$\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \begin{pmatrix} a_1 & a_3 \\ a_2 & a_4 \end{pmatrix} = a_1 \begin{pmatrix} c_1 \\ c_3 \end{pmatrix} + a_2 \begin{pmatrix} c_2 \\ c_4 \end{pmatrix} = \begin{pmatrix} a_1c_1 + a_2c_2 \\ a_1c_3 + a_2c_4 \end{pmatrix}$$

# Jacobians and Kronecker Products

Let's build our own 2x2 $CA^T$ to understand what this looks like symbolically. Then we can generalize this to arbitrarily sized C and $A^T$.

$$\begin{pmatrix} a_1c_1 + a_2c_2 \\ a_1c_3 + a_2c_4 \end{pmatrix} \qquad \begin{pmatrix} (c_1a_1 + c_2a_2) & (c_1a_3 + c_2a_4) \\ (c_3a_1 + c_4a_2) & (c_3a_3 + c_4a_4) \end{pmatrix}$$

**Key idea:** The first column of $CA^T$ is a linear combination of the columns of $A^T$ and the columns of C. The *coefficients* are given by the columns of $A^T$.

# Jacobians and Kronecker Products

What's true in our 2x2 case is true for compatibly sized matrices C and $A^T$. So, in general, we end up with our *typical* matrix multiplication summation, **a** in this summation represents the first row of A..

$$\operatorname{vec}(CA^T) = \begin{pmatrix} \sum_j a_{1j}\vec{c}_j \\ \sum_j a_{2j}\vec{c}_j \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11}\,\mathrm{I} & a_{12}\,\mathrm{I} & \cdots \\ a_{21}\,\mathrm{I} & a_{22}\,\mathrm{I} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}}_{A \otimes \mathrm{I}} \underbrace{\begin{pmatrix} \vec{c}_1 \\ \vec{c}_2 \\ \vdots \end{pmatrix}}_{\operatorname{vec} C},$$

Source: Matrix Calc Notes

# Jacobians and Kronecker Products

**Key idea:** For each row in $A^T$ we take linear combinations of its columns with c's columns. This forms vec($CA^T$). In Kronecker product notation this is the same thing as ($A \otimes I$) * vec(C). We see those linear combinations are the same but vec() stacks.

$$\mathrm{vec}(CA^T) = \begin{pmatrix} \sum_j a_{1j}\vec{c}_j \\ \sum_j a_{2j}\vec{c}_j \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11}\,\mathrm{I} & a_{12}\,\mathrm{I} & \cdots \\ a_{21}\,\mathrm{I} & a_{22}\,\mathrm{I} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}}_{A \otimes \mathrm{I}} \underbrace{\begin{pmatrix} \vec{c}_1 \\ \vec{c}_2 \\ \vdots \end{pmatrix}}_{\mathrm{vec}\, C},$$

# Jacobians and Kronecker Products

Now that we understand how to derive the key Kronecker identity let's discuss *how* we can write the Jacobian of $A^2$ in Kronecker product notation.

# Jacobians and Kronecker Products

Now that we understand how to derive the key Kronecker identity let's discuss *how* we can write the Jacobian of $A^2$ in Kronecker product notation.

Recall the key identity: $(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$

# Jacobians and Kronecker Products

Now that we understand how to derive the key Kronecker identity let's discuss *how* we can write the Jacobian of $A^2$ in Kronecker product notation.

Recall the key identity: $(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$

Let's write $d(A^2) = A\,dA + dA\,A$ as:

$d(A^2) = A\,dA\,I + I\,dA\,A$

# Jacobians and Kronecker Products

Now that we understand how to derive the key Kronecker identity let's discuss *how* we can write the Jacobian of $A^2$ in Kronecker product notation.

Recall the key identity: $(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$

Let's write $d(A^2) = A\,dA + dA\,A$ as:

$d(A^2) = A\,dA\,I + I\,dA\,A$

We want to do this because our Kronecker identity requires three matrices: B, C, and A. The identity is our method of adding another matrix without changing the algebraic structure of $d(A^2)$.

# Jacobians and Kronecker Products

Recall the key identity: $(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$

Let's write $d(A^2) = A\,dA + dA\,A$ as:

$d(A^2) = A\,dA\,I + I\,dA\,A$

Recall that in our derivation we were thinking of how the RHS $\text{vec}(BCA^T)$ related to the Kronecker notation on the LHS. At risk of sounding obvious I think we do this because $BCA^T$ is simpler to conceptualize. Most of intro linear algebra has matrix products; vec() stacks the columns of this product on each other. Kronecker notation is newer *but* since we have an equality we rearrange the RHS pieces to look like the left.

**Now, let's let dA be C in our key Kronecker identity and take $\text{vec}(dA^2)$.**

# Jacobians and Kronecker Products

Recall the key identity with dA in place of C: $(A \otimes B)\mathrm{vec}(\mathbf{dA}) = \mathrm{vec}(B\mathbf{dA}A^T)$

Now let's look at the A dA I first and ask ourselves which "Kronecker product" we should use:

$\mathrm{vec}(dA^2) = A\,dA\,I + I\,dA\,A$

# Jacobians and Kronecker Products

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

Now let's look at the A dA I first and ask ourselves which "Kronecker product" we should use:

$\text{vec}(dA^2) = A\,dA\,I + I\,dA\,A$

If we look at $\text{vec}(BdAA^T)$ and set $B = A$ and $A^T = I$ then we have:

$(I \otimes A)\text{vec}(dA) = \text{vec}(AdAI^T)$

Note: We know $I^T = I$ so we can avoid writing $(I^T \otimes A)\text{vec}(dA)$ on the LHS.

# Jacobians and Kronecker Products

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

Now let's look at the A dA I first and ask ourselves which "Kronecker product" we should use:

$\text{vec}(dA^2) = A\,dA\,I + I\,dA\,A$

If we look at $\text{vec}(BdAA^T)$ and set $B = I$ and $A^T = A^T$ then we have:

$(A^T \otimes I)\text{vec}(dA) = \text{vec}(IdAA^T)$

# Jacobians and Kronecker Products

The full Kronecker product is:

$((I \otimes A) + (A^T \otimes I)) * \text{vec}(dA)$ (vec version)

$((I \otimes A) + (A^T \otimes I)) * [dA]$ (linear operator version)

**Key idea:** The Kronecker product identity abstracts away the need to compute element-wise partial derivatives using the Jacobian – giving us the *same* answer but with simpler operations.

# Jacobians and Kronecker Products in Julia

```
X^2
```

$$\begin{bmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{bmatrix}$$

```
# Take the Jacobian of Y, vector valued f

jac(Y, X) = Symbolics.jacobian(vec(Y), ve
```

```
jac (generic function with 1 method)
```

```
# I think we would get the same answer if
# ... of each term in X^2 are in the firs
# ... and 2,1 entry of X^2 is the second

J = jac(X^2, X)
```

$$\begin{bmatrix} 2a & b & c & 0 \\ c & a+d & 0 & c \\ b & 0 & a+d & b \\ 0 & b & c & 2d \end{bmatrix}$$

```
begin
    I2 = [1 0; 0 1]
    kron(I2,X) + kron(X', I2)
end
```

$$\begin{bmatrix} 2a & b & c & 0 \\ c & a+d & 0 & c \\ b & 0 & a+d & b \\ 0 & b & c & 2d \end{bmatrix}$$

Source: Deliverable 3

# Jacobians and Kronecker Products – Extra Fun $d(A^3)$

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

On slide 43 we derived the $d(A^3)$ so I plug that result in below.

$\text{vec}(dA^3) = A^2\, dA + A\, dA\, A + dA\, A^2$

# Jacobians and Kronecker Products – Extra Fun $d(A^3)$

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

On slide 43 we derived the $d(A^3)$ so I plug that result in below.

$\text{vec}(dA^3) = A^2\, dA + A\, dA\, A + dA\, A^2$

Just as we did before we can rewrite this with the identity matrix:

$\text{vec}(dA^3) = A^2\, dA\, I + (A\, dA\, A)\, I + dA\, A^2\, I$

# Jacobians and Kronecker Products – Extra Fun d($A^3$)

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

$\text{vec}(dA^3) = A^2\, dA\, I + (A\, dA\, A)\, I + dA\, A^2\, I$

If we look at $\text{vec}(BdAA^T)$ and set $B = \mathbf{A^2}$ and $A^T = \mathbf{I}$ then we have:

$(I \otimes A^2)\text{vec}(dA) = \text{vec}(A^2 dA I^T)$

# Jacobians and Kronecker Products – Extra Fun d($A^3$)

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

$\text{vec}(dA^3) = A^2\, dA\, I + (A\, dA\, A)\, I + dA\, A^2\, I$

If we look at $\text{vec}(BdAA^T)$ and set $B = \mathbf{A^2}$ and $A^T = \mathbf{I}$ then we have:

$(I \otimes A^2)\text{vec}(dA) = \text{vec}(A^2 dA I^T)$

# Jacobians and Kronecker Products – Extra Fun $d(A^3)$

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

$\text{vec}(dA^3) = A^2\, dA\, I + (A\, dA\, A)\, I + dA\, A^2\, I$

If we look at $\text{vec}(BdAA^T)$ and set $B = \mathbf{A}$ and $A^T = \mathbf{A}^T$ then we have:

$(A^T \otimes A)\text{vec}(dA) = \text{vec}(AdAA^T)$

# Jacobians and Kronecker Products – Extra Fun $d(A^3)$

Recall the key identity with dA in place of C: $(A \otimes B)\text{vec}(\mathbf{dA}) = \text{vec}(B\mathbf{dA}A^T)$

$\text{vec}(dA^3) = A^2\, dA\, I + (A\, dA\, A)\, I + I\, dA\, A^2$

If we look at $\text{vec}(BdAA^T)$ and set $B = \mathbf{I}$ and $A^T = \mathbf{(A^2)^T}$ then we have:

$((A^2)^T \otimes I)\text{vec}(dA) = \text{vec}((A^2)^T dAI)$

# Jacobians and Kronecker Products – Extra Fun $d(A^3)$

The full Kronecker product is:

$((I \otimes A^2) + (A^T \otimes A) + ((A^2)^T \otimes I)) * \text{vec}(dA)$ (vec version)

$((I \otimes A^2) + (A^T \otimes A) + ((A^2)^T \otimes I)) * [dA]$ (linear operator version)

**Key idea:** The $d(A^3)$ example should feel more "plug in values" than $d(A^2)$ since we now *really understand* how to use the Kronecker identity.

# Jacobians and Kronecker Products in Julia

```
J_2 = jac(X^3, X)
```

$$\begin{bmatrix} 3a^2 + 2bc & ab + (a+d)\,b & 2ac + cd & bc \\ 2ac + cd & a^2 + (a+d)\,d + 2bc & c^2 & ac + 2cd \\ 2ab + bd & b^2 & a\,(a+d) + 2bc + d^2 & ab + 2bd \\ bc & ab + 2bd & (a+d)\,c + cd & 2bc + 3d^2 \end{bmatrix}$$

```
begin
    kron(I2, X2) + kron(X', X) + kron(X2', I2)
end
```

$$\begin{bmatrix} 3a^2 + 2bc & 2ab + bd & 2ac + cd & bc \\ 2ac + cd & a^2 + ad + 2bc + d^2 & c^2 & ac + 2cd \\ 2ab + bd & b^2 & a^2 + ad + 2bc + d^2 & ab + 2bd \\ bc & ab + 2bd & ac + 2cd & 2bc + 3d^2 \end{bmatrix}$$

**Key idea:** With some distribution in the Kronecker product solution – these are the same. We can do d($A^n$) with the same process computing the Jacobian and Kronecker products to see they are the same.

Source: Deliverable 3

# Final notes

- We can use our chain rule derivation from slide 32 to consider a function: a(b(c(x))) and take its derivative so it looks like: a’(b(c(x))b’(c(x))c’(x).
- If we cared about getting the gradient of a function (1 x n row vector):

  a’ = 1 x n

  b’ = n x n

  c’ = n x n

  We would also care about *how* we do this matrix multiplication. This is where we exploit associativity.

# Final notes

- If we cared about getting the gradient of a function (1 x n row vector):

  a’ = 1 x n; b’ = n x n; c’ = n x n

  We would also care about *how* we do this matrix multiplication. This is where we exploit associativity.

  (a’ b’) * c → (row vector * matrix) ⇒ (row vector * matrix)

  a’ (b’ c’) → (matrix * matrix) ⇒ (row vector * matrix)

  Without getting too detailed there are two observations I want to share.

  1. Associativity *matters* in ML (and all matmul) because it has different comp costs.
  2. Left to right multiplication is called *reverse mode auto differentiation (a.k.a backpropagation)*. Right to left multiplication is called *forward mode auto differentiation.*