Linear Algebra & Vector Calculus

- Linear Algebra Review
- Vector Calculus Review

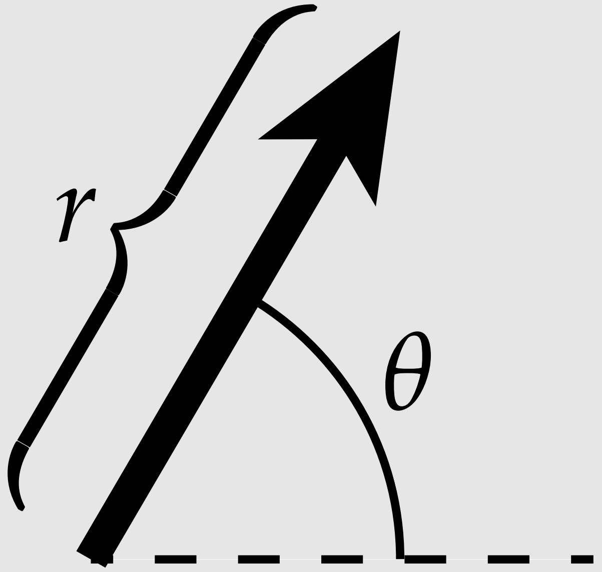
What Is A Vector?

- Intuitively, a vector is a little arrow
  - Encoded as direction + magnitude

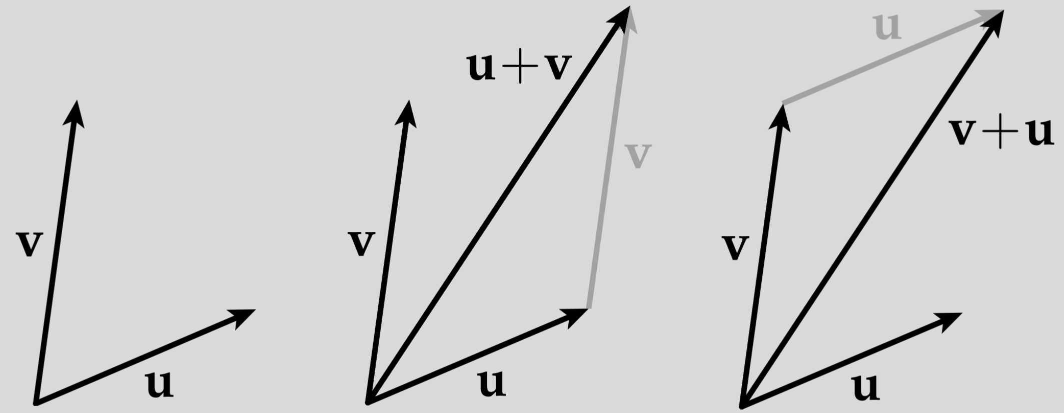
- Many types of data can be represented as vectors
    - Polynomials
    - Images
    - Radiance

- Vectors are functions of their coordinate system
  - Can't directly compare coordinates in different systems!
    - **Example:** polar and cartesian

- Why start with a vector when talking about Linear Algebra?
  - Most of linear algebra can be explained with vectors



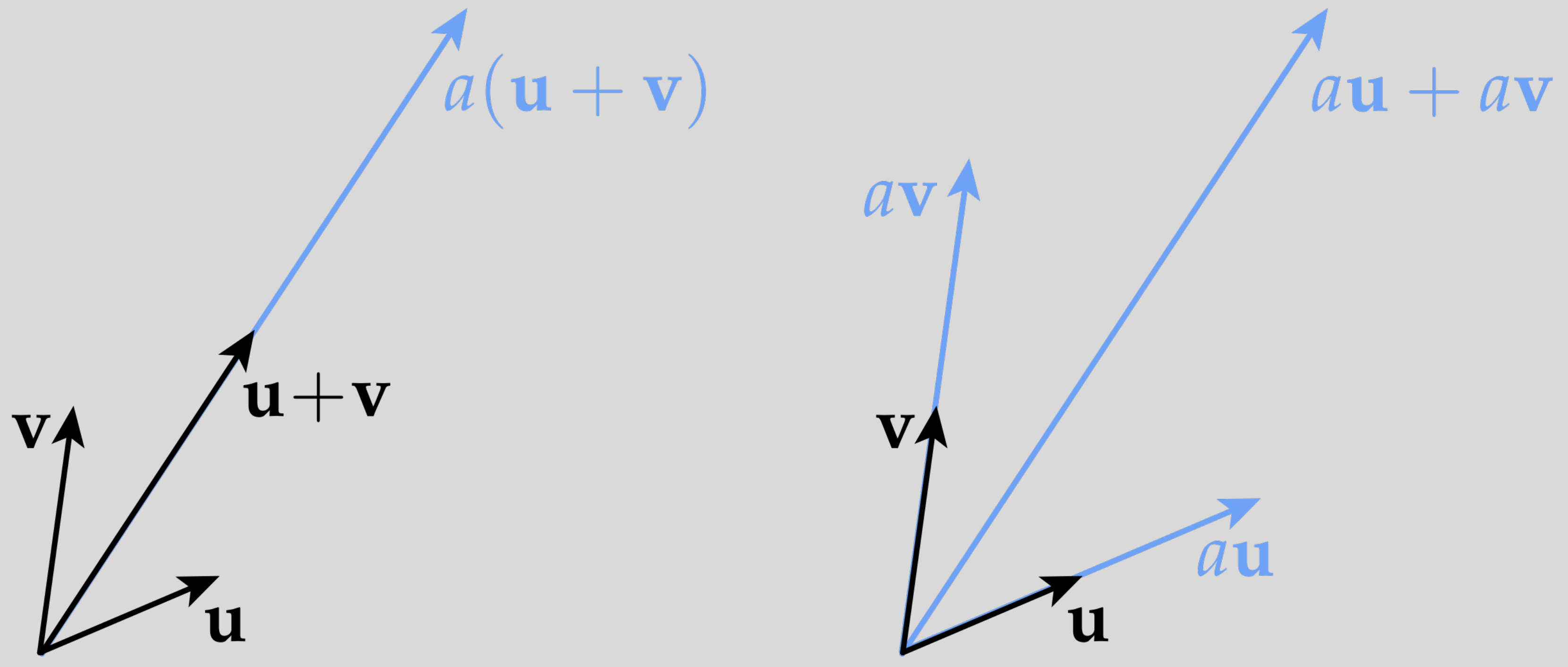
Basic Vector Operations



Vector addition: ***u + v = v + u***"
"commutative" or "abelian"

Vector multiplication: *a(b**u**) = (ab)**u***

Basic Vector Operations



Order of operations for adding and scaling do not matter

$a(\mathbf{u}+\mathbf{v}) = a\mathbf{u} + a\mathbf{v}$

Formal Vector Space Definition

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b:

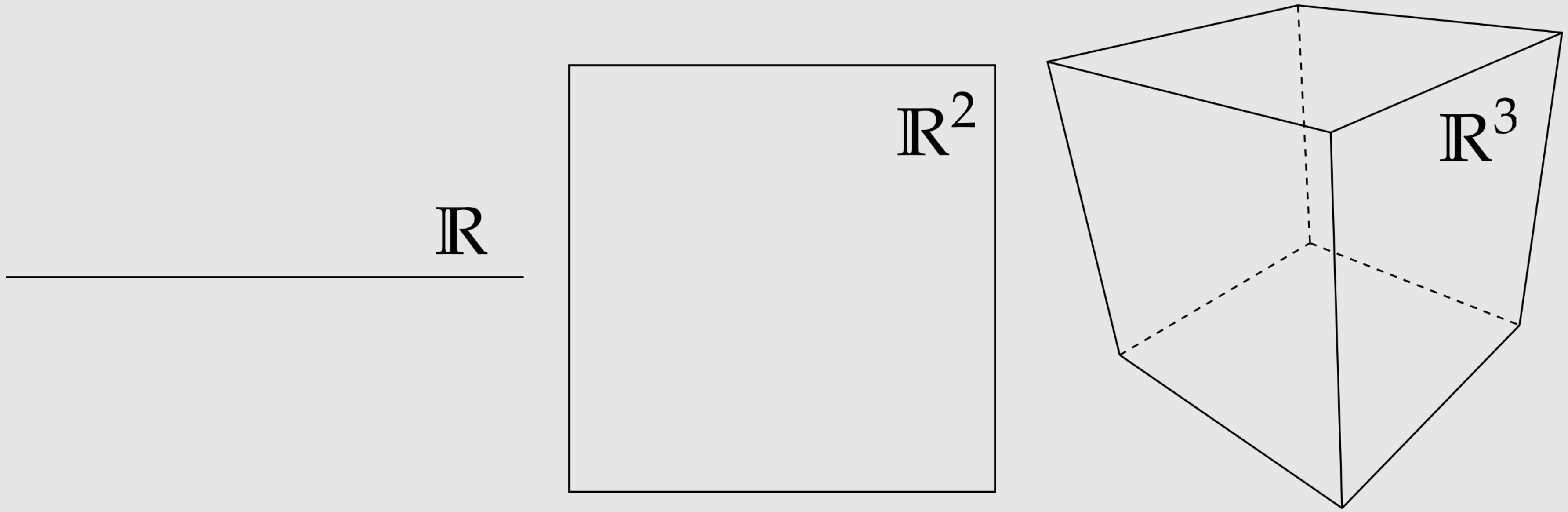
- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a *zero vector* "$\mathbf{0}$" such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every $\mathbf{v}$ there is a vector "$-\mathbf{v}$" such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

These rules did not "fall out of the sky!" Each one comes from the geometric behavior of "little arrows." (Can you draw a picture for each one?)

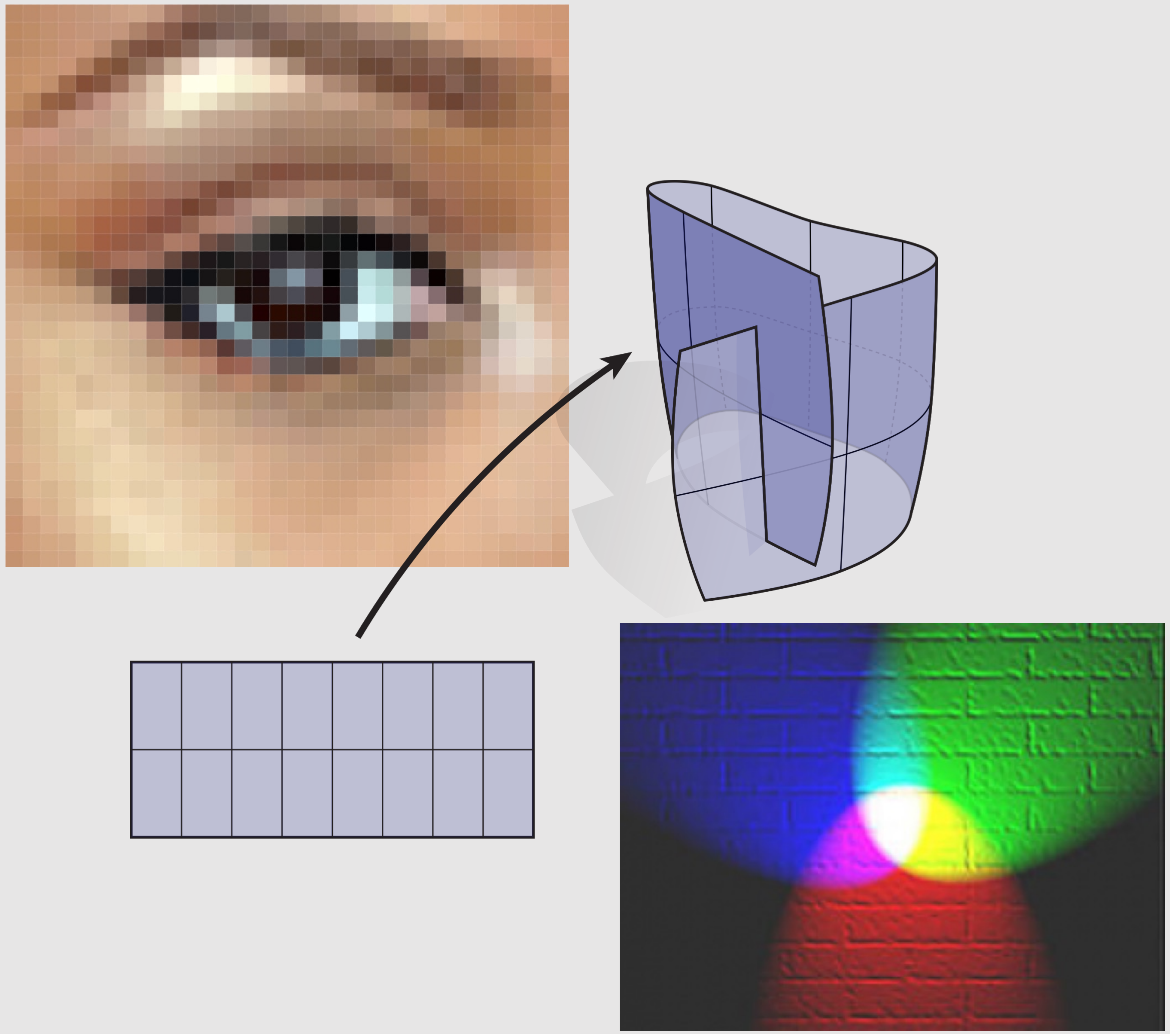
Any collection of objects satisfying all of these properties is a vector space.

Euclidean Vector Space

- Typically denoted by $\mathbb{R}^n$, meaning "n real numbers"
  - **Example:** (1.23, 4.56, π/2) is a point in $\mathbb{R}^3$



Functions as Vectors

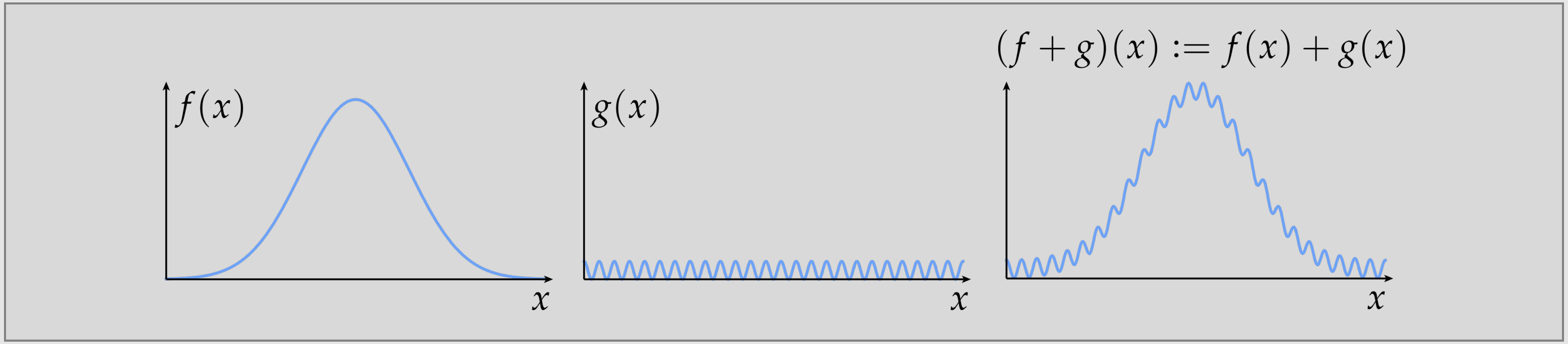


- Functions also behave like vectors
- Functions are all over graphics!
  - **Example:** images
  - $I(x, y)$ takes in coordinates and returns the pixel color in the image
- Representing functions as vectors allow us to apply vector operations

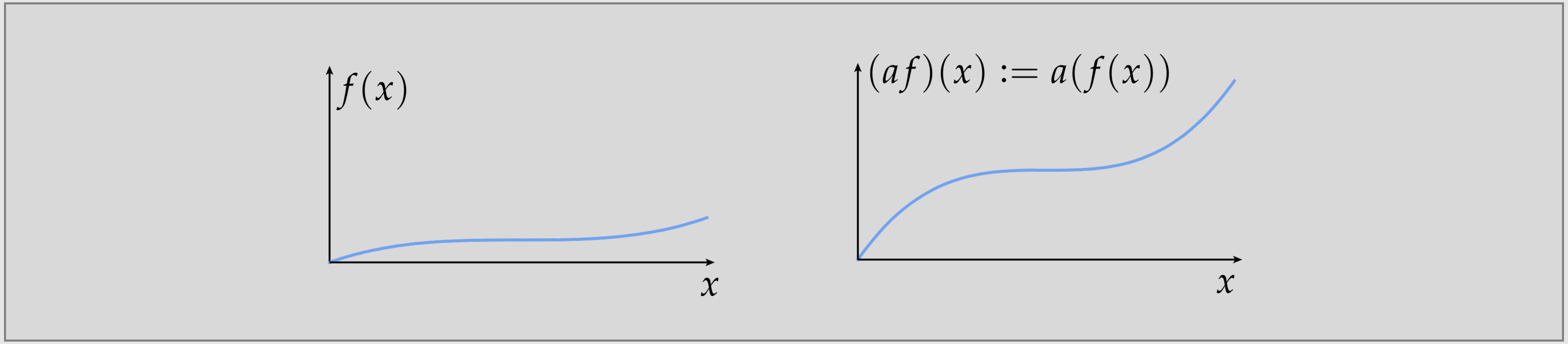
Functions as Vectors

Do functions exhibit the same behavior as "little arrows?"

Well, we can certainly add two functions:



We can also scale a function:



Functions as Vectors

What about the rest of these rules?

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b:

- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a *zero vector* "$\mathbf{0}$" such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every $\mathbf{v}$ there is a vector "$-\mathbf{v}$" such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

Try it out at home! (E.g., the "zero vector" is the function equal to zero for all x)

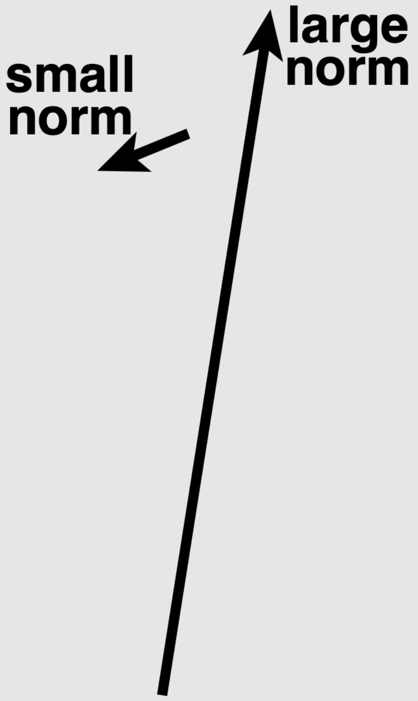
Short answer: yes, functions are vectors! (Even if they don't look like "little arrows")

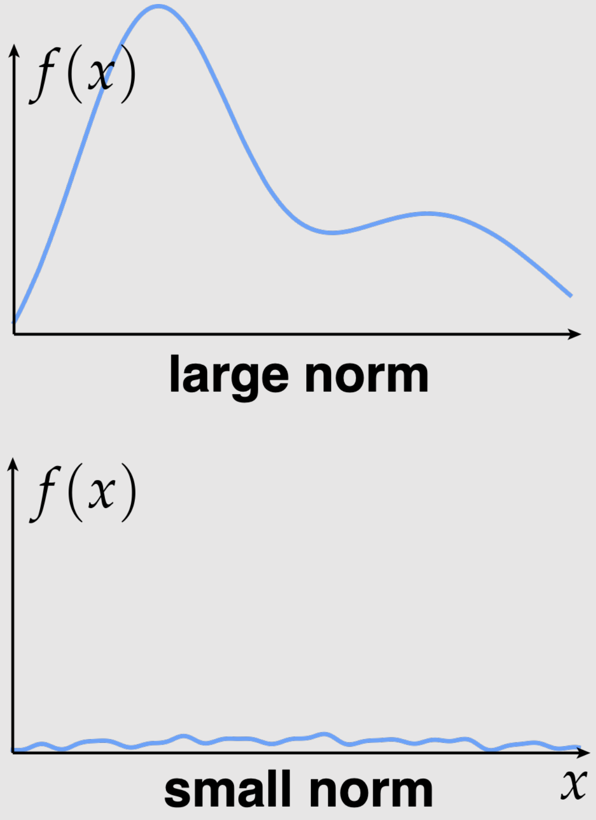
Never blindly accept a rule given by authority.

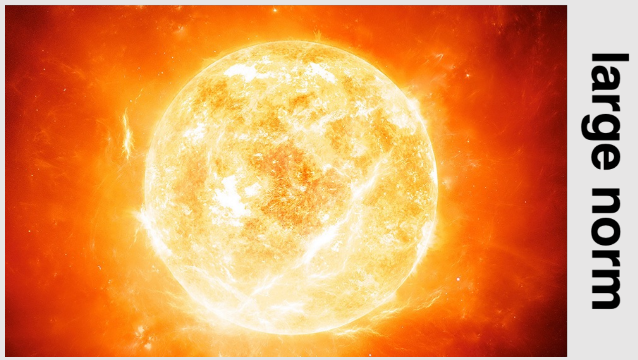
Always ask: where does this rule come from?
What does it mean geometrically? (Can you draw a picture?)

Norm of a Vector

For a given vector v, $|v|$ is its **length** / **magnitude** / **norm**.
Intuitively, this captures how "big" the vector is





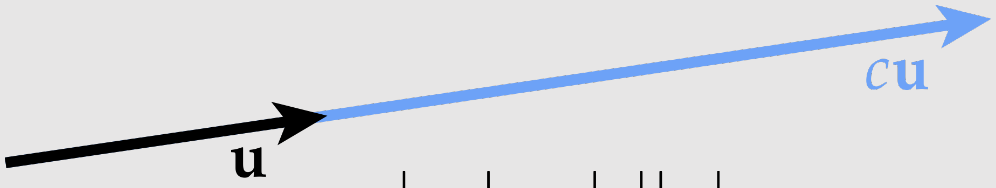


Norm Properties

For one thing, it shouldn't be negative!

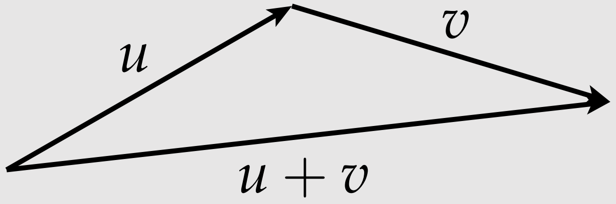
$$|\mathbf{u}| \geq 0 \qquad\qquad |\mathbf{u}| = 0 \iff \mathbf{u} = \mathbf{0}$$

Also, if we scale a vector by a scalar c, its norm should scale by the same amount.



$$|c\mathbf{u}| = |c||\mathbf{u}|$$

Finally, we know that the shortest path between two points is always along a straight line.**



$$|\mathbf{u}| + |\mathbf{v}| \geq |\mathbf{u} + \mathbf{v}|$$

**sometimes called the "triangle inequality" since the diagram looks like a triangle

Norm Definition

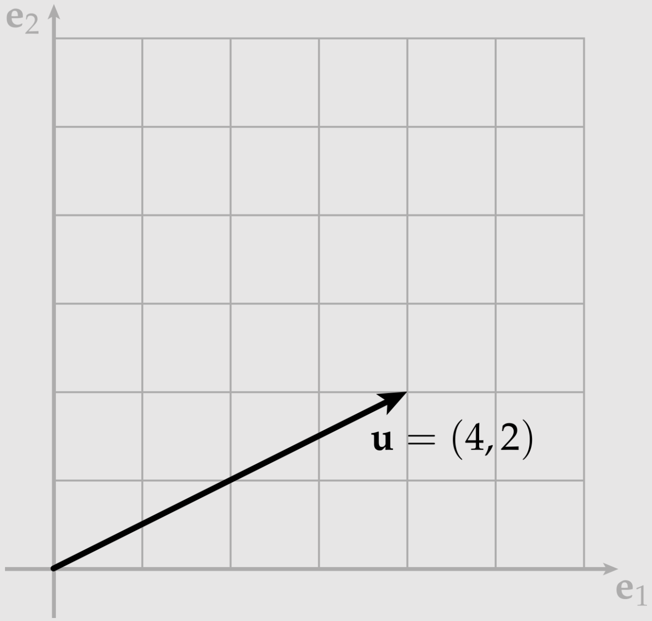
A **norm** is any function that assigns a number to each vector and satisfies the following properties for all vectors ***u, v,*** and all scalars *a*

- $|\mathbf{v}| \geq 0$
- $|\mathbf{v}| = 0 \iff \mathbf{v} = \mathbf{0}$
- $|a\mathbf{v}| = |a||\mathbf{v}|$
- $|\mathbf{u}| + |\mathbf{v}| \geq |\mathbf{u} + \mathbf{v}|$

Euclidean Norm in Cartesian Coordinates

A standard norm is the so-called **Euclidean norm** of n-vectors

$$|\mathbf{u}| = |(u_1, \ldots, u_n)| := \sqrt{\sum_{i=1}^{n} u_i^2}$$



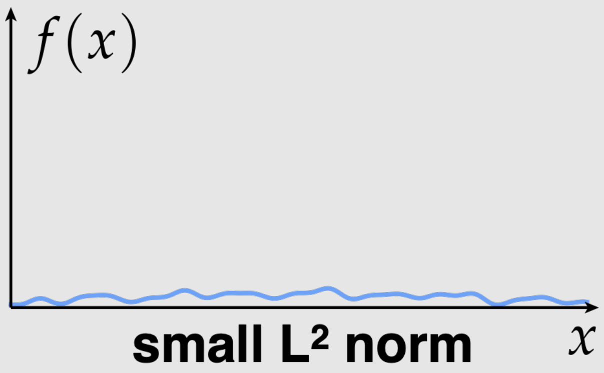
$$|\mathbf{u}| = \sqrt{4^2 + 2^2} = 2\sqrt{5}$$

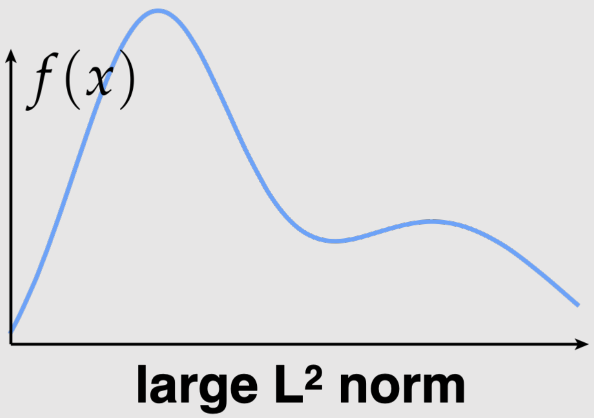
L^2 Norm Of Functions

- L2 norm measures the total magnitude of a function
- Consider real-valued functions on the unit interval [0,1] whose square has a well-defined integral. The L2 norm is defined as:

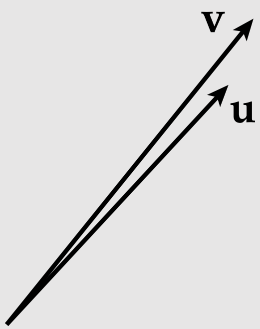
$$||f|| := \sqrt{\int_0^1 f(x)^2 \, dx}$$

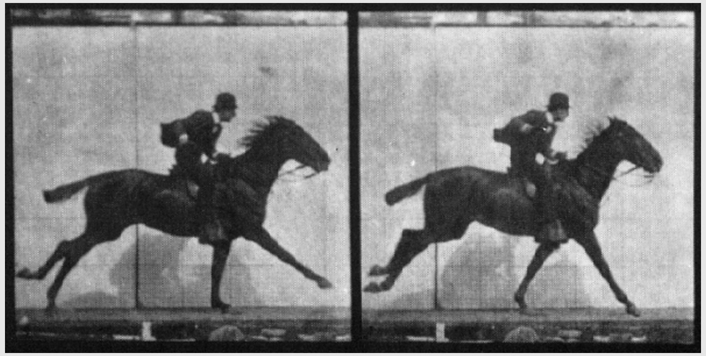
- Not too different from the Euclidean norm
  - We just replaced a sum with an integral
- **Careful!** does the formula above exactly satisfy all our desired properties for a norm?



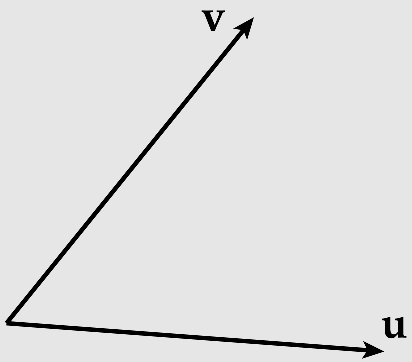


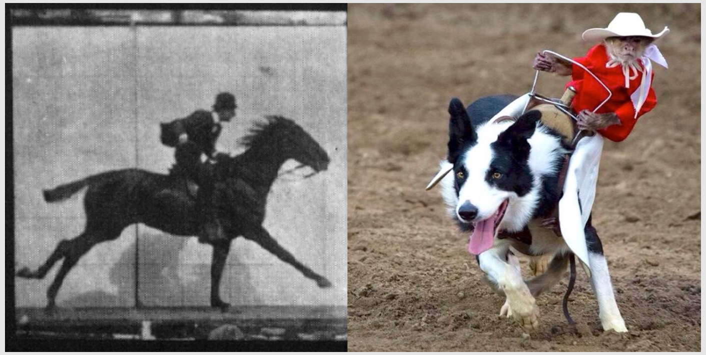
Inner Product





[similar]



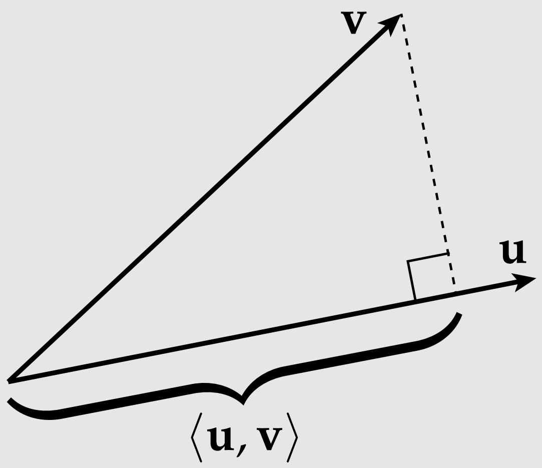


[different]

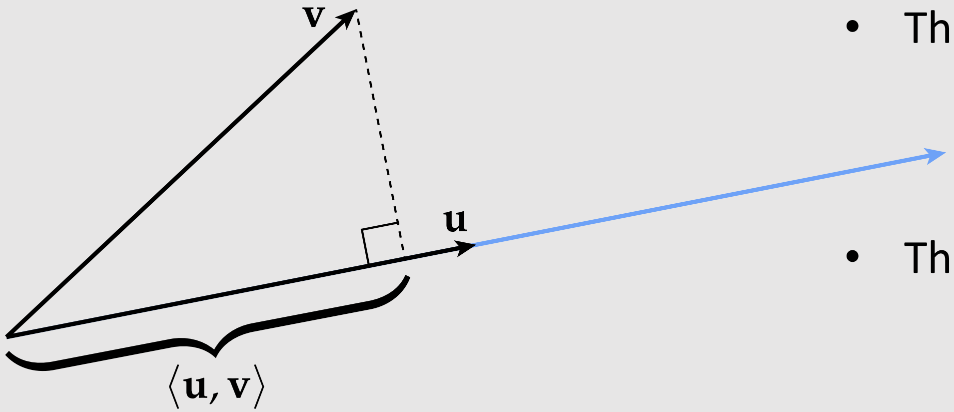
- **Inner product** measures the *"similarity"* of vectors, or how well vectors *"line up"*
- The dot product of two vectors is commutative:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$$

Inner Product



[no scale]



[scaling u or v]

- For unit vectors $|u|=|v|=1$, an inner product measures the extent, or percent, of one vector along the direction of the other. If we scale either vector, the inner product also scales:

$$\langle 2\mathbf{u}, \mathbf{v} \rangle = 2\langle \mathbf{u}, \mathbf{v} \rangle$$

  - Vectors need to be normalized when computing similarity!

- Any vector will always be aligned with itself:

$$\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$$

- The dot product of any unit vector with itself is:

$$\langle \mathbf{u}, \mathbf{u} \rangle = 1$$

- Thus for a unit vector $\hat{\mathbf{u}} := \mathbf{u}/|\mathbf{u}|$

$$\langle \mathbf{u}, \mathbf{u} \rangle = \langle |\mathbf{u}|\hat{\mathbf{u}}, |\mathbf{u}|\hat{\mathbf{u}} \rangle = |\mathbf{u}|^2 \langle \hat{\mathbf{u}}, \hat{\mathbf{u}} \rangle = |\mathbf{u}|^2 \cdot 1 = |\mathbf{u}|^2$$

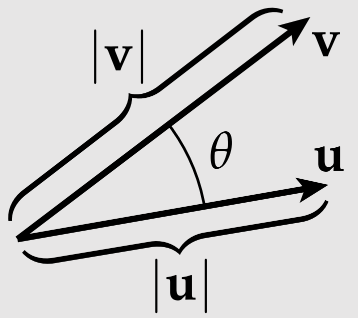
Inner Product Formal Definition

An inner product is any function that assigns to any two vectors u,v a number <u,v> satisfying the following properties:

- $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
- $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$
- $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \iff \mathbf{u} = \mathbf{0}$
- $\langle a\mathbf{u}, \mathbf{v} \rangle = a\langle \mathbf{u}, \mathbf{v} \rangle$
- $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$

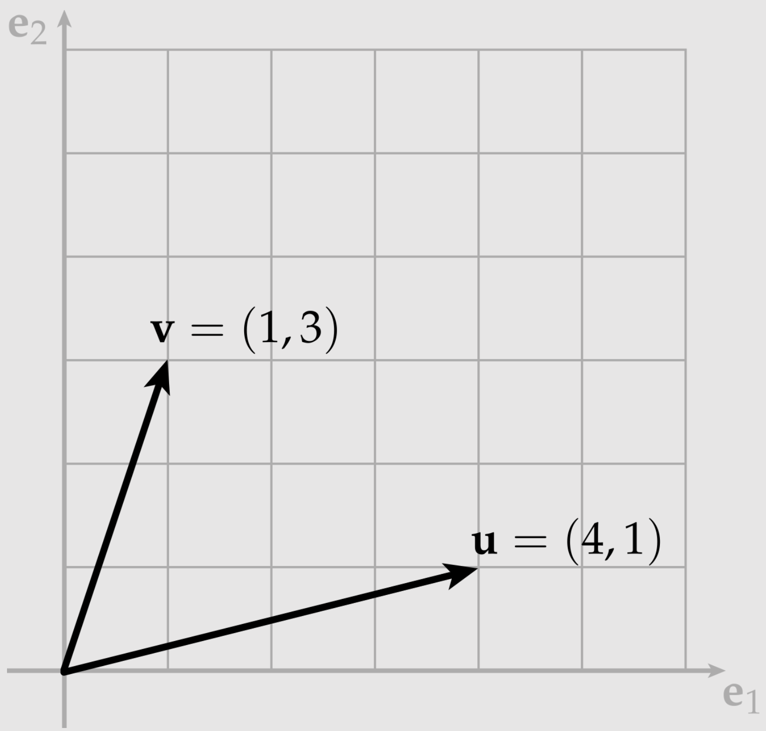
[Euclidean inner product] $\langle \mathbf{u}, \mathbf{v} \rangle := |\mathbf{u}||\mathbf{v}|\cos(\theta)$

[Cartesian inner product] $\mathbf{u} \cdot \mathbf{v} := u_1 v_1 + \cdots + u_n v_n$



Inner Product In Cartesian Coordinates

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle (u_1, \ldots, u_n), (v_1, \ldots, v_n) \rangle := \sum_{i=1}^{n} u_i v_i$$



$$\langle \mathbf{u}, \mathbf{v} \rangle = 4 \cdot 1 + 1 \cdot 3 = 7$$

L^2 Inner Product Of Functions

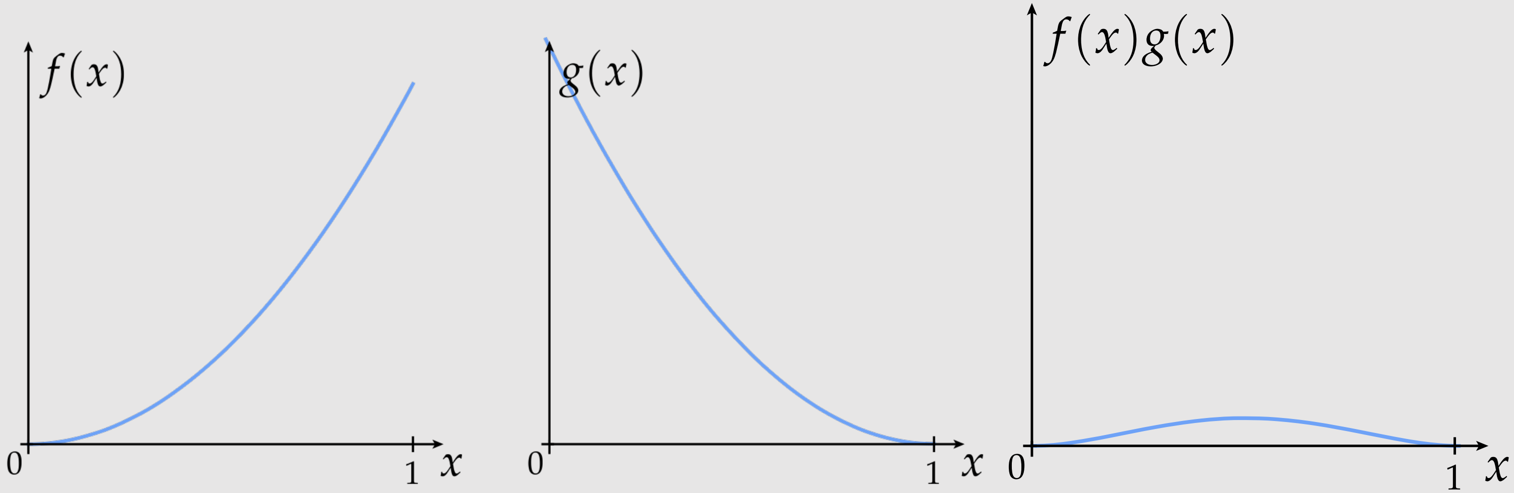
$$\langle\langle f,g \rangle\rangle := \int_0^1 f(x)g(x)\,dx$$

Example:

$$f(x) := x^2, \quad g(x) := (1-x)^2$$

$$\langle\langle f,g \rangle\rangle = \int_0^1 x^2(1-x)^2\,dx = \cdots = \frac{1}{30}$$

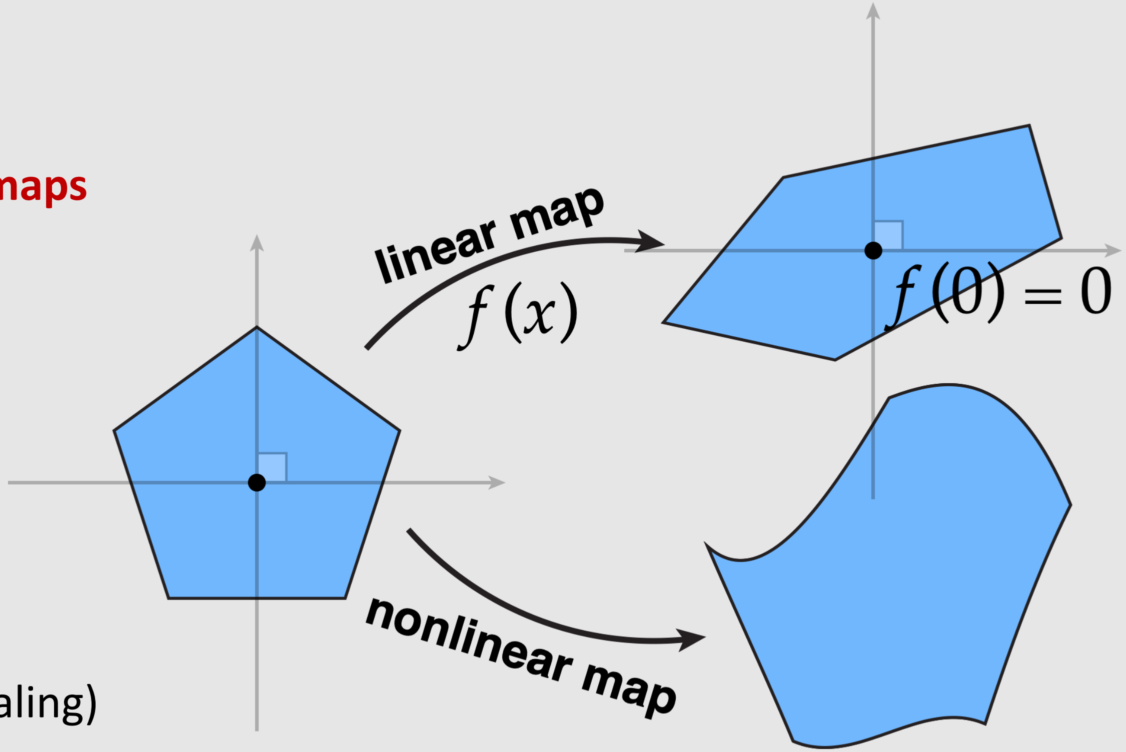
small number



functions don't line up much

Linear Maps

- Linear algebra is study of **vector spaces** and **linear maps** between them
- Linear maps have 2 characteristics:
  - Converts lines to lines
  - Keeps the origin fixed
- Linear map benefits:
  - Easy to solve systems of linear equations.
  - Basic transformations (rotation, translation, scaling) can be expressed as linear maps
  - All maps can be approximated as linear maps over a short distance/short time. (Taylor's theorem)
    - This approximation is used all over geometry, animation, rendering, image processing

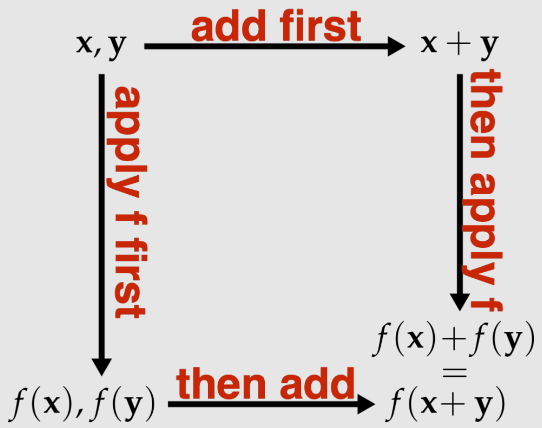


Linear Maps

A map ***f*** is **linear** if it maps vectors to vectors, and if for all vectors ***u,v*** and scalars a we have:

$$f(\mathbf{u}+\mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$
$$f(a\mathbf{u}) = af(\mathbf{u})$$

It doesn't matter whether we add the vectors and then apply the map, or apply the map and then add the vectors (and likewise for scaling):

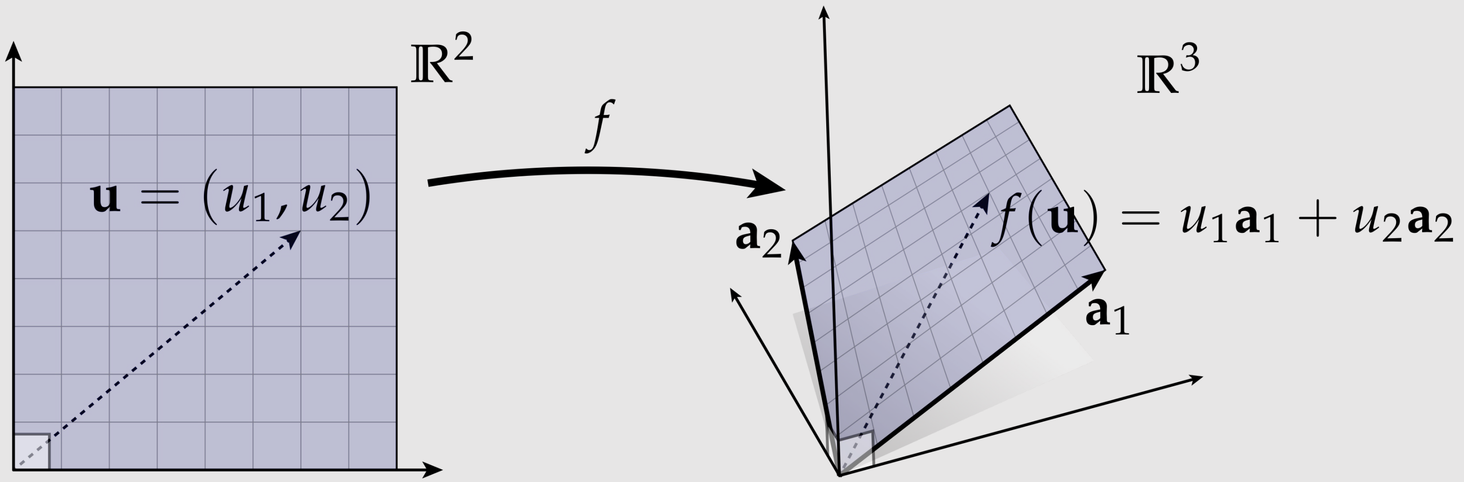


Linear Maps

For maps between $\mathbb{R}^n$ and $\mathbb{R}^m$ (e.g., a map from 2D to 3D), a map is linear if it can be expressed as

$$f(u_1, \ldots, u_m) = \sum_{i=1}^{m} u_i \mathbf{a}_i$$

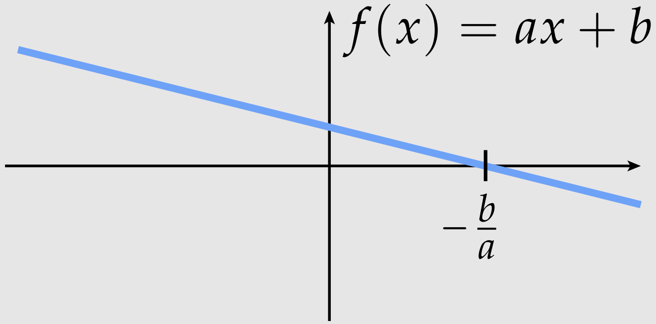
In other words, if it is a linear combination of a fixed set of vectors a_i:



Is $f(x) = ax + b$ a linear map?

Linear vs. Affine Maps

No! but it is easy to be fooled since it looks like a line.
However, it does not keep the origin fixed ($f(x) \neq 0$)



Another way to see it's not linear? It doesn't preserve sums:

$$f(x_1 + x_2) = a(x_1 + x_2) + b = ax_1 + ax_2 + b$$
$$f(x_1) + f(x_2) = (ax_1 + b) + (ax_2 + b) = ax_1 + ax_2 + 2b$$

This is called an affine map.

We will see a trick on how to turn affine maps into linear maps using homogeneous coordinates in a future lecture.

Is $\boldsymbol{f(u) = \int_0^1 u(x)dx}$ a linear map?

This will be on your homework?**

** hint: consider $u(x) = x$

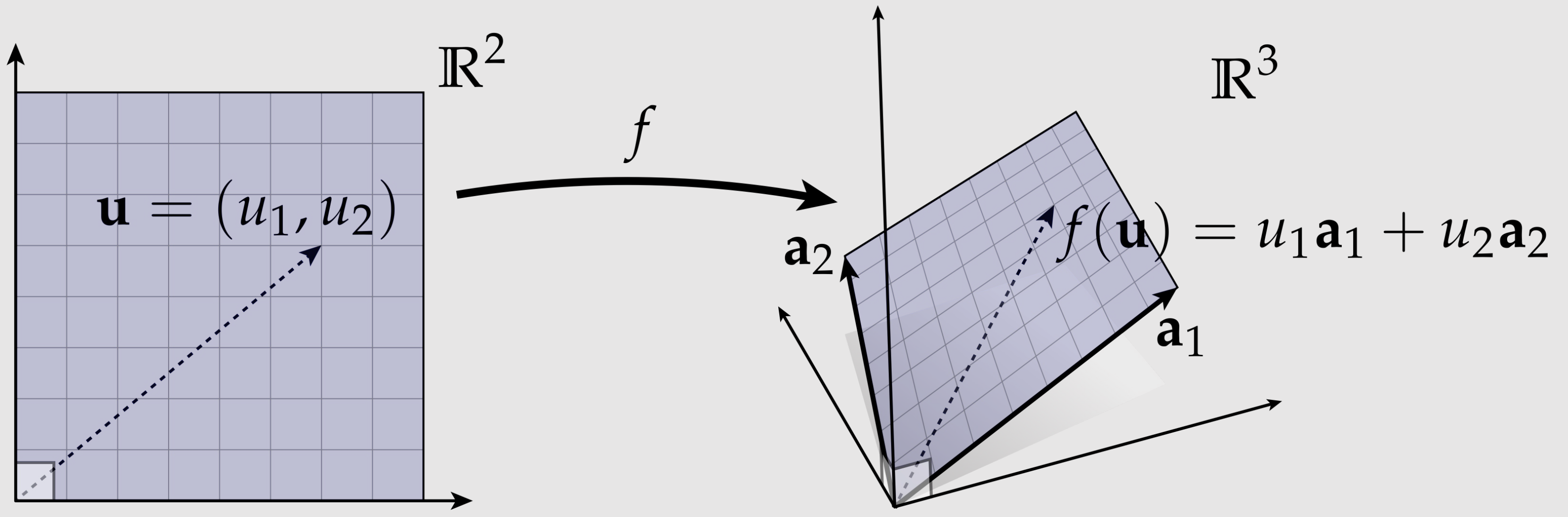
Span

The **span** of a set of vectors S_1 is the set of all vectors S_2 that can be written as a linear combination of the vectors in S_1

$$\text{span}(\mathbf{u}_1, \ldots, \mathbf{u}_k) = \left\{ \mathbf{x} \in V \,\middle|\, \mathbf{x} = \sum_{i=1}^{k} a_i \mathbf{u}_i,\ a_1, \ldots, a_k \in \mathbb{R} \right\}$$

Span & Linear Maps

The **image** of any **linear map** is the **span** of the **vectors** from applying the linear map.



The **image** of any **function** is the **codomain** of the **inputs** from applying the function.

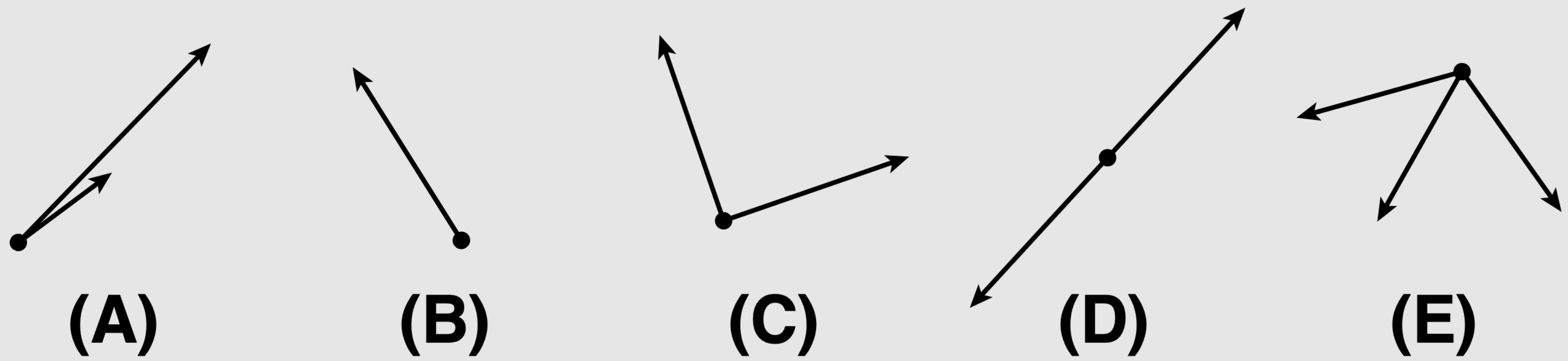
Orthonormal Basis

If we have exactly n vectors $e_1, \ldots, e_n$ such that:

$$\text{span}(\mathbf{e}_1, \ldots, \mathbf{e}_n) = \mathbb{R}^n$$

Then we say that these vectors are a basis for $\mathbb{R}^n$.

Note that there are many different choices of bases for $\mathbb{R}^n$!



Which of the following are bases for $\mathbb{R}^2$?

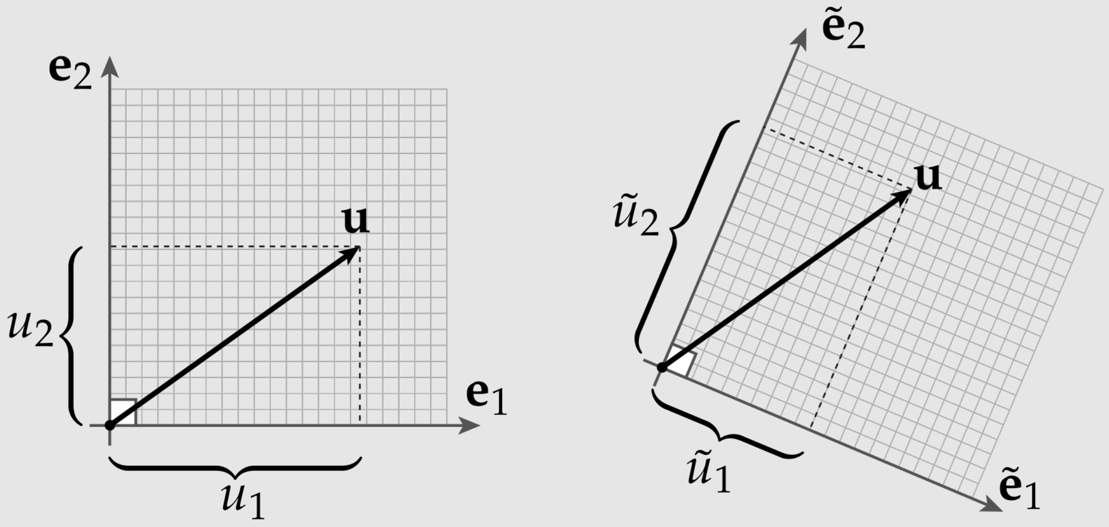
Orthonormal Basis

Most often, it is convenient to have to basis vectors that are:

- (i) unit length
- (ii) mutually orthogonal

In other words, if $e_1, \ldots, e_n$ are our basis vectors, then:

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases}$$



*Common bug: projecting a vector onto a basis that is NOT orthonormal while continuing to use standard norm / inner product.

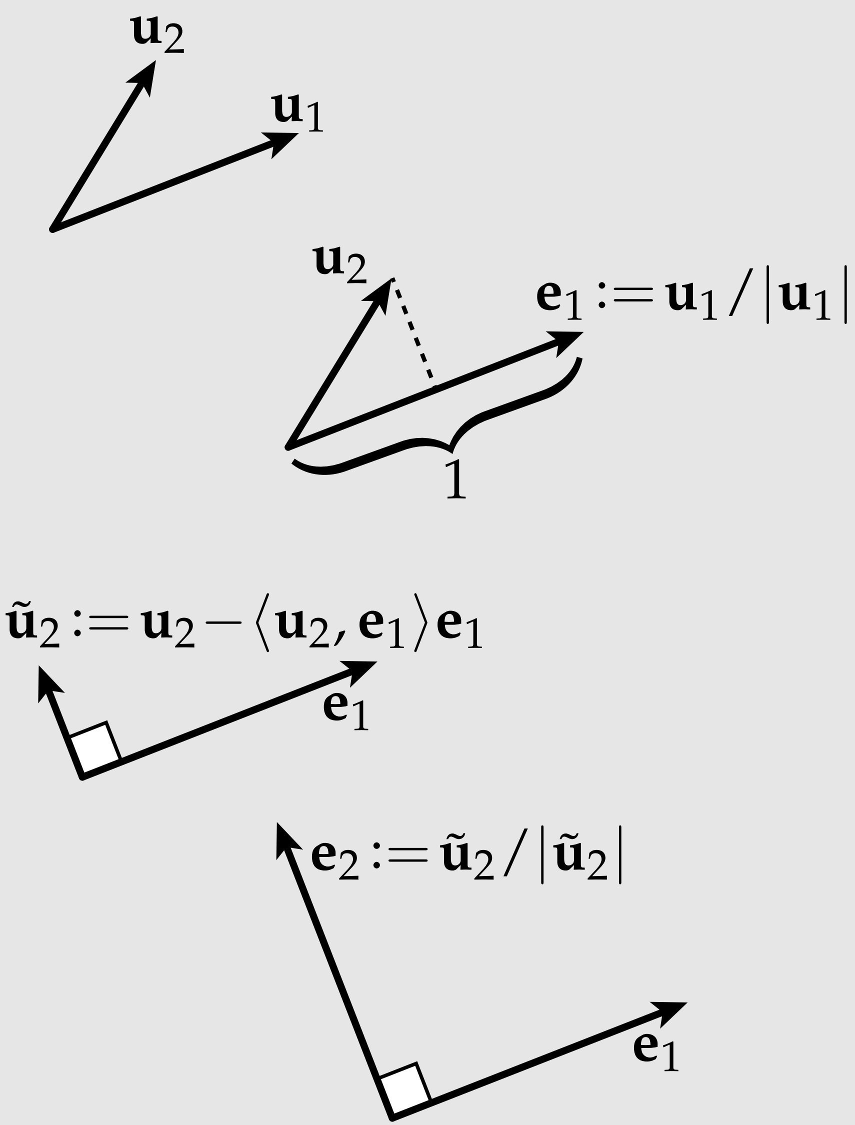
Gram-Schmidt

Given a collection of basis vectors $a_1, \dots, a_n$, we can find an orthonormal basis $e_1, \dots, e_n$ using the **Gram-Schmidt** method

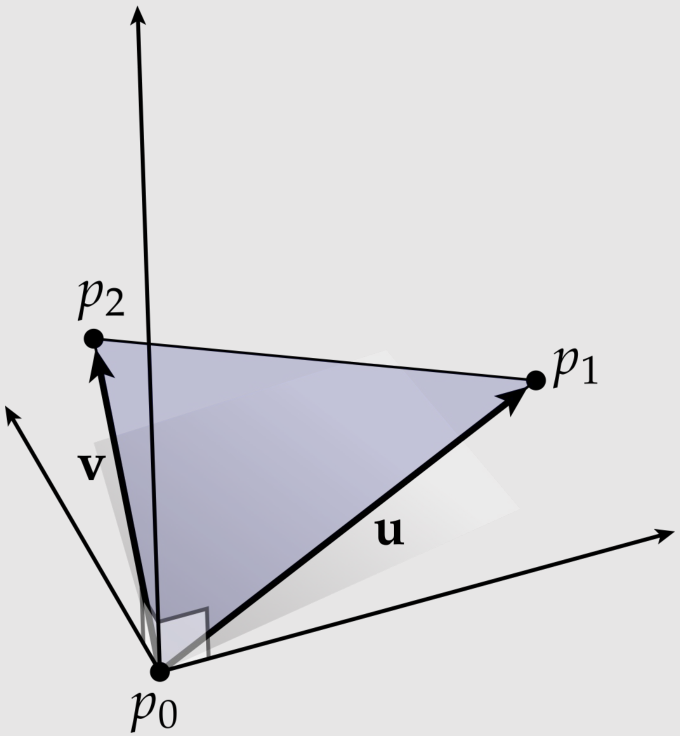
Gram-Schmidt algorithm:

- Normalize the 1st vector
- Subtract any component of the 1st vector from the 2nd one
- Normalize the 2nd one
- Repeat, removing components of first k vectors from vector k+1

- **Caution!** Does not work well for large sets of vectors or nearly parallel vectors
  - Modified Gram-Schmidt algorithms exist



Gram-Schmidt Example



Common task: have a triangle in 3D, need orthonormal basis for the plane containing the triangle

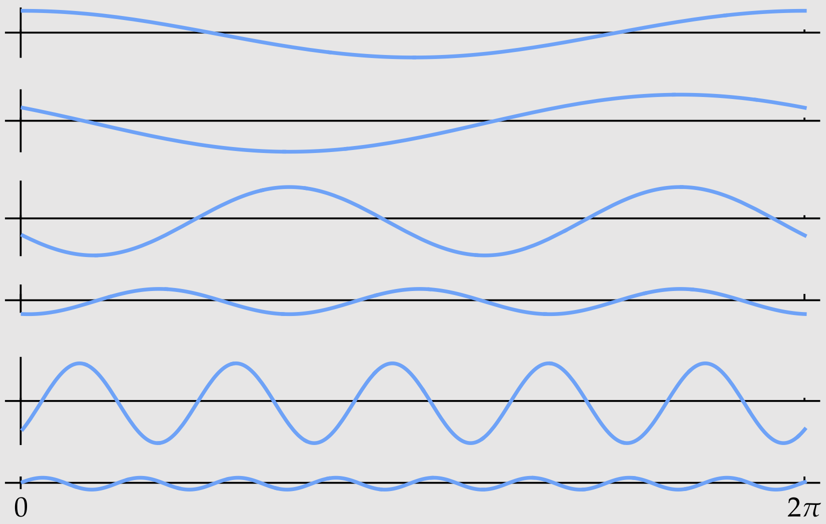
Strategy: apply Gram-Schmidt to (any) pair of edge vectors

$$\mathbf{u} := p_1 - p_0$$
$$\mathbf{v} := p_2 - p_0$$
$$\mathbf{e}_1 := \mathbf{u}/|\mathbf{u}|$$
$$\tilde{\mathbf{v}} := \mathbf{v} - \langle \mathbf{v}, \mathbf{e}_1 \rangle \mathbf{e}_1$$
$$\mathbf{e}_2 := \tilde{\mathbf{v}}/|\tilde{\mathbf{v}}|$$

Does the order matter? *(Ex: if we swapped u and v in the above equation, what happens?)*

Fourier Transform

[lower frequency]



[higher frequency]

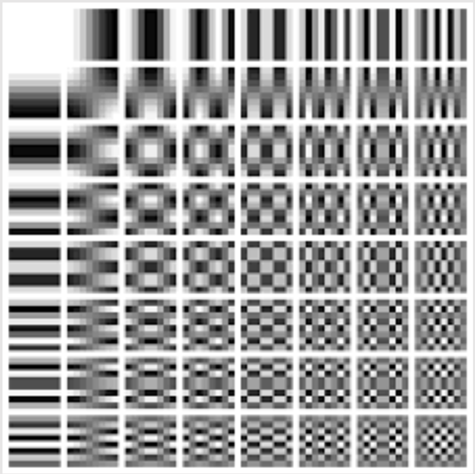
- Functions are also vectors, meaning they have an orthonormal basis known as a **Fourier transform**
  - Example: functions that repeat at intervals of 2π

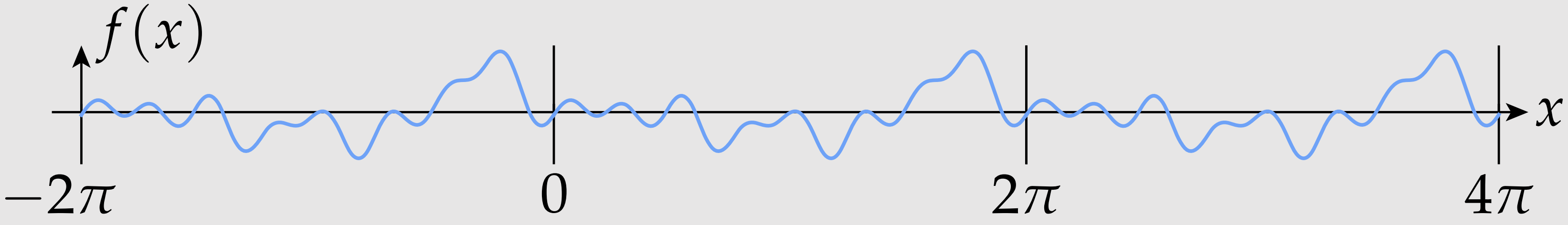
- Can project onto basis of sinusoids:

$$\cos(nx), \sin(mx),\ m, n \in \mathbb{N}$$

- Fundamental building block for many graphics algorithms:
  - Example: JPEG Compression

- More generally, this idea of projecting a signal onto different “frequencies” is known as **Fourier decomposition**





System Of Linear Equations

- A system of linear equations is a bunch of equations where left-hand side is a linear function, right hand side is constant.
  - Unknown values are called **degrees of freedom (DOFs)**
  - Equations are called **constraints**

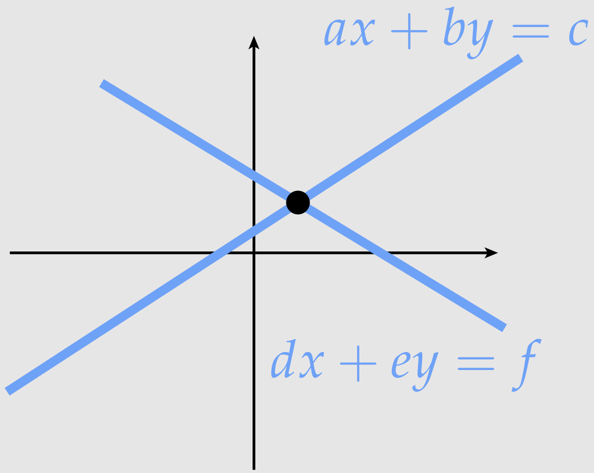
- We can use linear systems to solve for:
  - The point where two lines meet
  - Given a point b, find the point x that maps to it

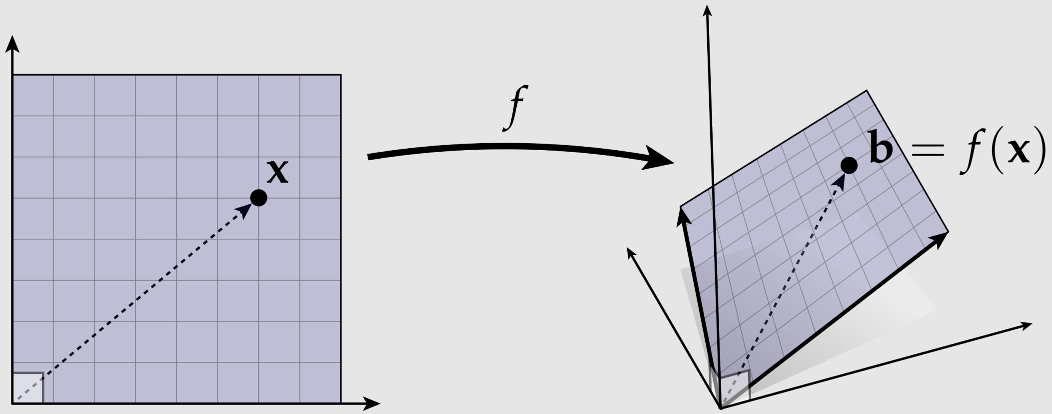
$$\begin{aligned} x + 2y &= 3 \\ 4x + 5y &= 6 \end{aligned}$$

$$x = 3 - 2y$$

$$4(3 - 2y) + 5y = 6$$

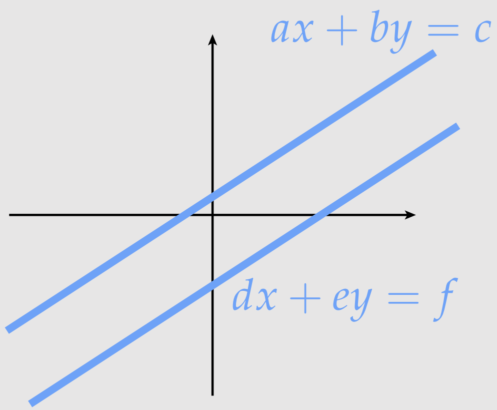
$$\boxed{\begin{aligned} y &= 2 \\ x &= -1 \end{aligned}}$$



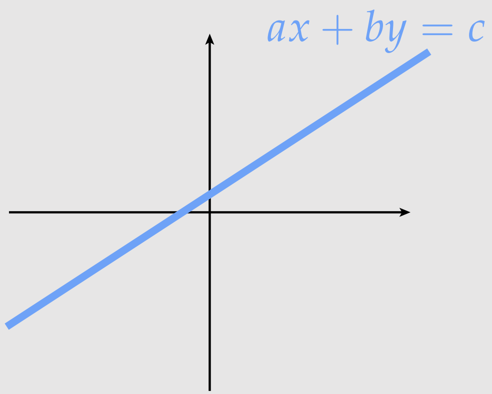


Existence of Solutions

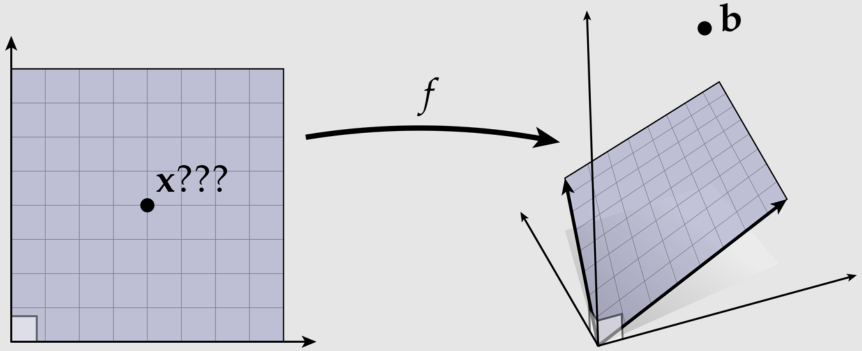
Of course, not all linear systems can be solved!
(And even those that can be solved may not have a unique solution.)



[no solution]



[many solution]



[no solution]

Matrices

- We've gone this far without talking about a matrix
  - But linear algebra is not fundamentally about matrices.
  - We can understand almost all the basic concepts without ever touching a matrix!

- Still, VERY useful!
  - Symbolic manipulation
  - Easy to store
  - Fast to compute
    - (Sometimes) hardware support for matrix ops

- Some of the (many) uses for matrices:
  - Transformations
  - Coordinate System Conversions
  - Compression
  - Gram-Schmidt

$$\begin{bmatrix} 1 & 7 & 3 \\ 4 & 9 & 2 \\ 0 & 1 & 1 \end{bmatrix}$$

What does this little block of funny numbers do?

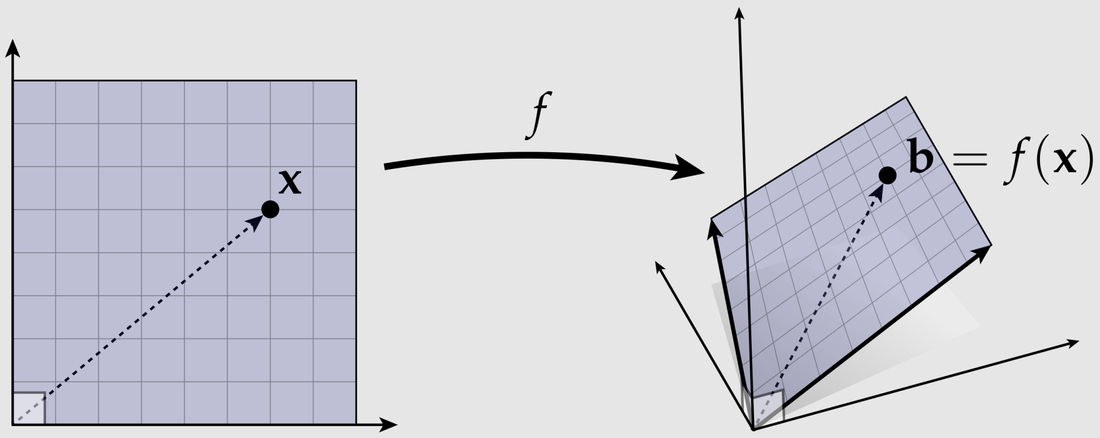
Linear Maps As Matrices

Example: consider the linear map:

$$f(\mathbf{u}) = u_1\mathbf{a}_1 + u_2\mathbf{a}_2$$

$\boldsymbol{a}$ vectors become columns in the matrix:

$$A := \begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix}$$



Multiplying the original vector $\boldsymbol{u}$ maps it to $\boldsymbol{f}(\boldsymbol{u})$:

$$\begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_{1,x}u_1 + a_{2,x}u_2 \\ a_{1,y}u_1 + a_{2,y}u_2 \\ a_{1,z}u_1 + a_{2,x}u_2 \end{bmatrix} = u_1\mathbf{a}_1 + u_2\mathbf{a}_2$$

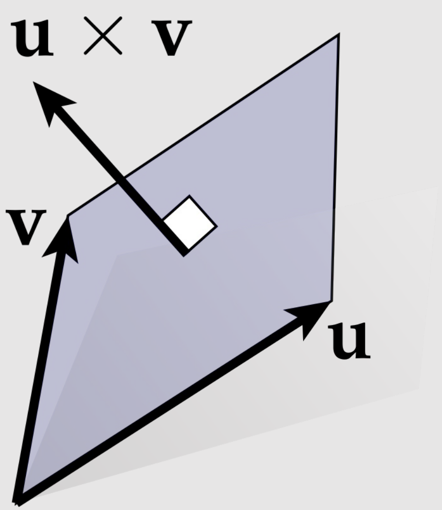
How to map $\boldsymbol{f}(\boldsymbol{u})$ back to $\boldsymbol{u}$? Take the inverse of the matrix!

- ~~Linear Algebra Review~~

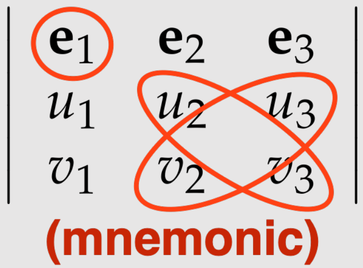
- Vector Calculus Review

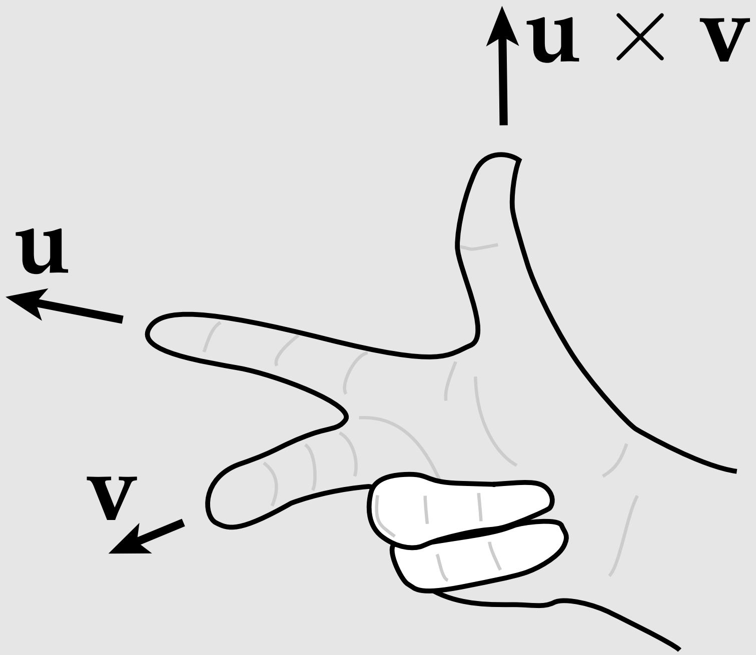
Cross Product

- Inner product takes two vectors and produces a scalar
  - Cross product takes two vectors and produces a vector
- Geometrically:
  - Magnitude equal to parallelogram area
  - Direction orthogonal to both vectors
  - …but which way?
    - Use "right hand rule"
    - Only works in 3D



$$\sqrt{\det(\mathbf{u}, \mathbf{v}, \mathbf{u} \times \mathbf{v})} = |\mathbf{u}||\mathbf{v}|\sin(\theta)$$





- θ is angle between u and v
- "det" is determinant of three column vectors

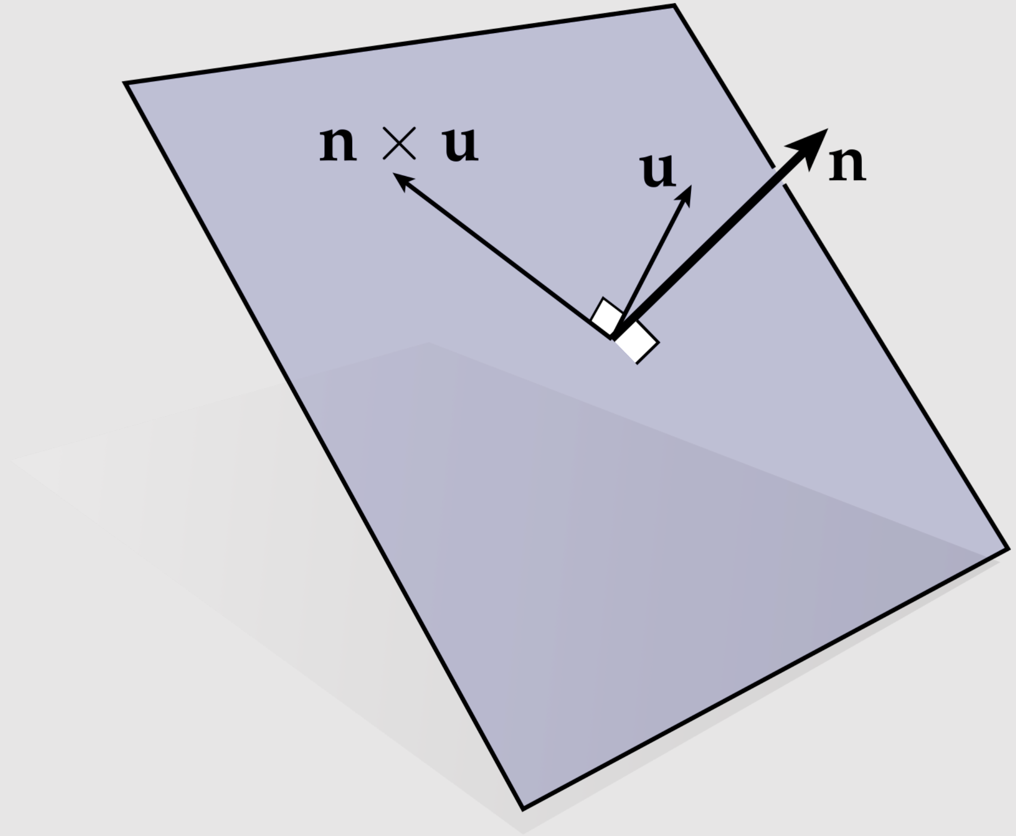
Cross Product In 2D

$$\mathbf{u} \times \mathbf{v} := \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}$$

We can abuse notation in 2D and write it as:

$$\mathbf{u} \times \mathbf{v} := u_1v_2 - u_2v_1$$

Cross Product As A Quarter Rotation



- In 3D, cross product with a unit vector N is equivalent to a quarter-rotation in the plane with normal N.
  - Use the right hand rule :)

  - What is $n \times (n \times u)$?

Dot And Cross Products

Dot product as a matrix multiplication:

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^{\mathsf{T}}\mathbf{v} = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \sum_{i=1}^{n} u_i v_i$$

Cross product as a matrix multiplication:

$$\mathbf{u} := (u_1, u_2, u_3) \Rightarrow \widehat{\mathbf{u}} := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

$$\mathbf{u} \times \mathbf{v} = \widehat{\mathbf{u}}\mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Dot And Cross Products

Useful to notice $\boldsymbol{u} \times \boldsymbol{v} = -\boldsymbol{v} \times \boldsymbol{u}$

This means:

$$\mathbf{v} \times \mathbf{u} = -\widehat{\mathbf{u}}\mathbf{v} = \widehat{\mathbf{u}}^{\mathsf{T}}\mathbf{v}$$

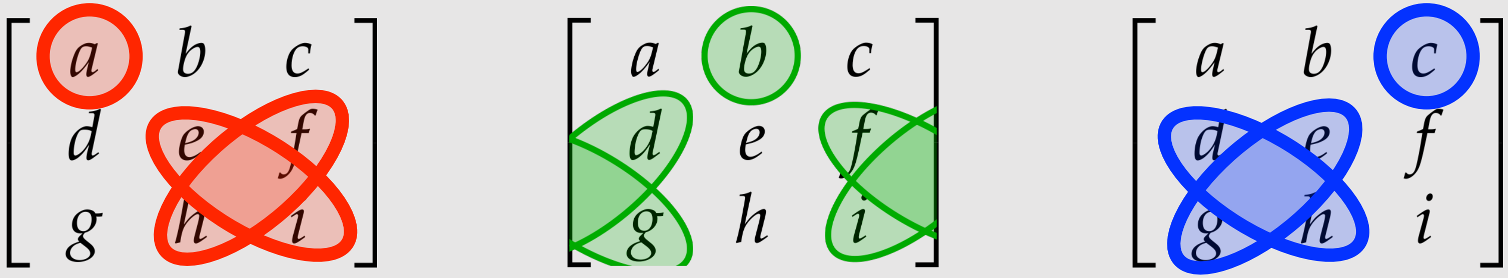
$$\mathbf{u} := (u_1, u_2, u_3) \Rightarrow \widehat{\mathbf{u}} := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

$$\mathbf{u} \times \mathbf{v} = \widehat{\mathbf{u}}\mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Determinant

$$\mathbf{A} := \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

The determinant of A is:

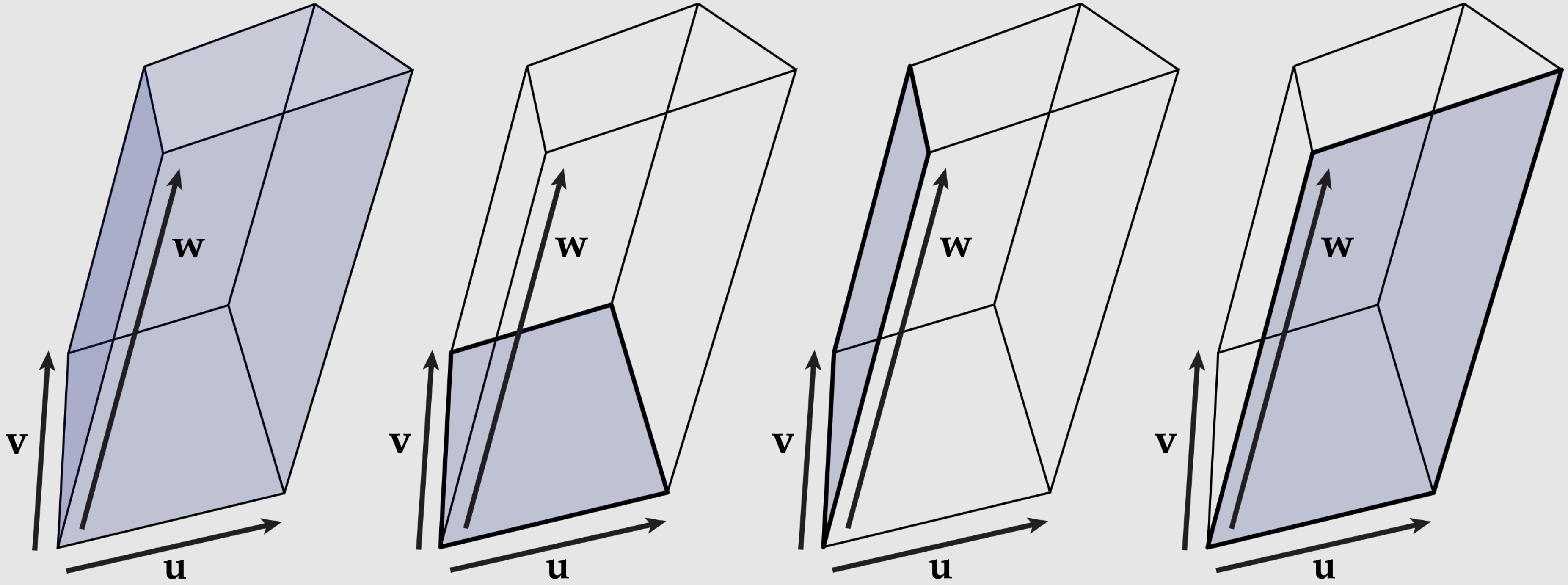


$$\det(\mathbf{A}) = a(ei - fh) + b(fg - di) + c(dh - eg)$$

Great, but what does that mean?

Determinant

det(u,v,w) encodes **signed volume** of parallelepiped with edge vectors u, v, w.



$$\det(\mathbf{u},\mathbf{v},\mathbf{w}) = (\mathbf{u}\times\mathbf{v})\cdot\mathbf{w} = (\mathbf{v}\times\mathbf{w})\cdot\mathbf{u} = (\mathbf{w}\times\mathbf{u})\cdot\mathbf{v}$$

What happens if we reverse the order of the vectors in the cross product?

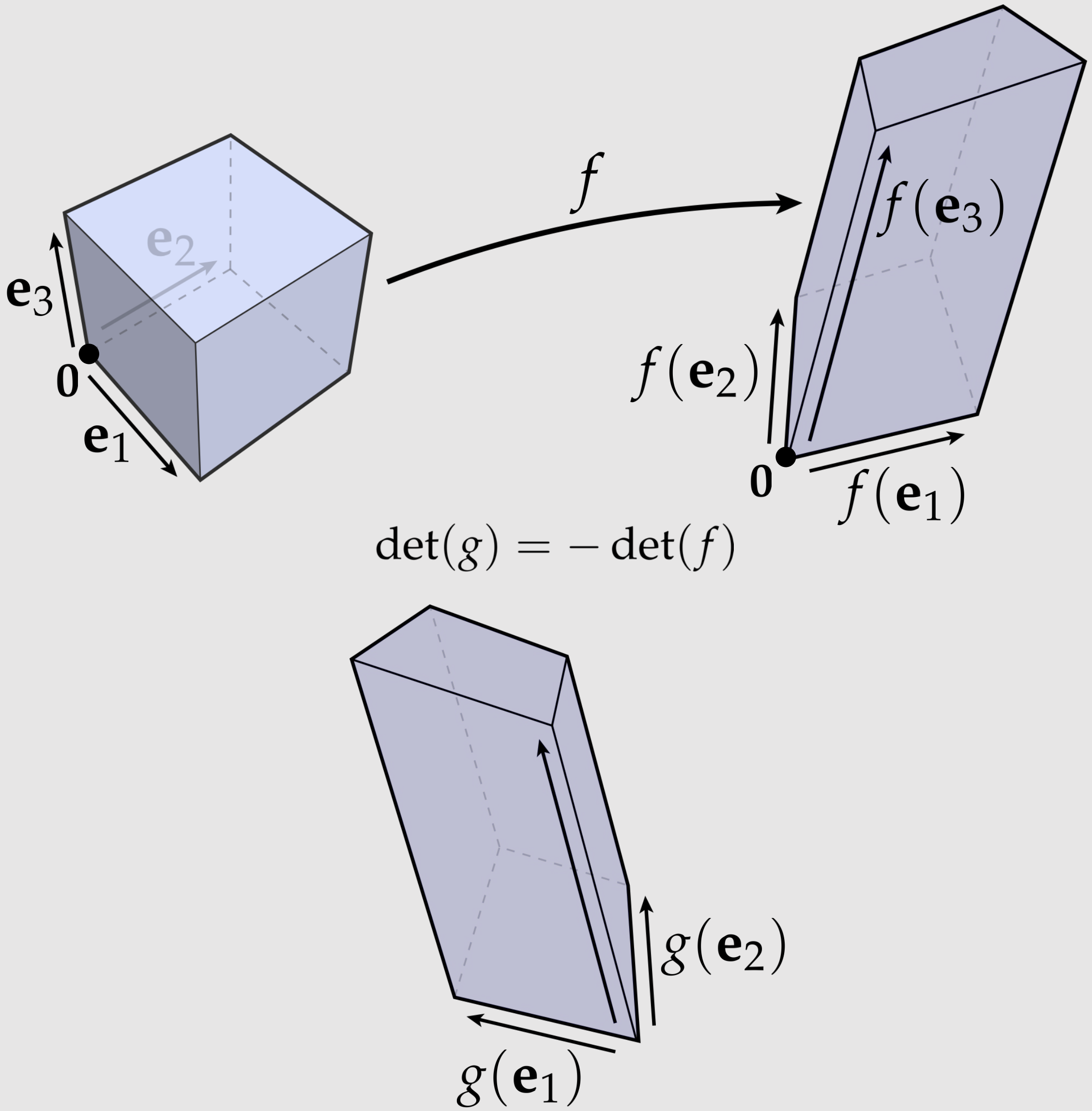
Determinant of a Linear Map

- Recall that a linear map is a transformation from one coordinate space to another and is defined by a set of vectors $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$ ...

$$f(\mathbf{u}) = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + u_3\mathbf{a}_3$$

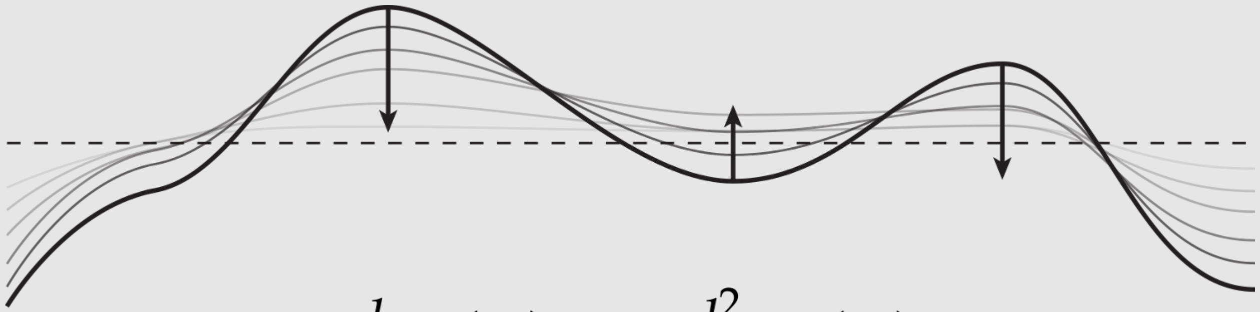
$$A := \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} a_{1,x} & a_{2,x} & a_{3,x} \\ a_{1,y} & a_{2,y} & a_{3,y} \\ a_{1,z} & a_{2,z} & a_{3,z} \end{bmatrix}$$

- The $\boldsymbol{det(A)}$ here measures the change in volume between spaces.
    - The sign tells us whether the orientation was reversed.

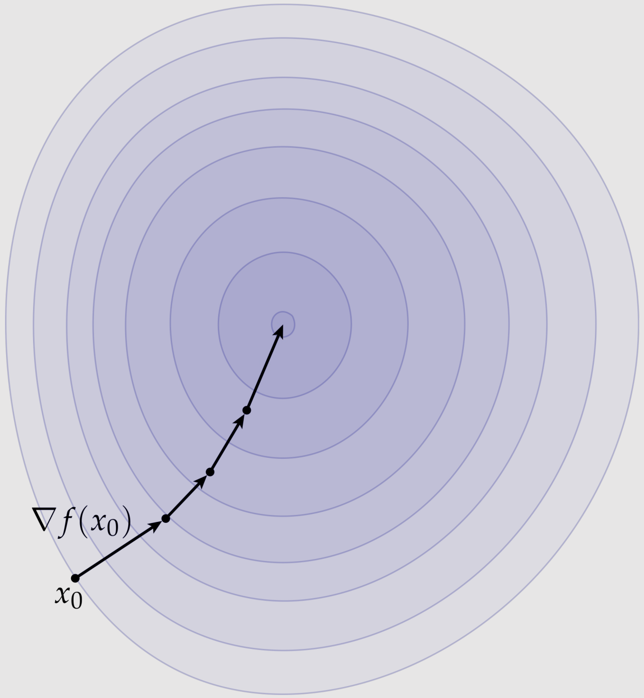


Differential Operators

- Many uses for computer graphics:
  - Expressing physical/geometric problems in terms of related rates of change (ODEs, PDEs)
  - Numerical optimization – minimizing the cost relative to some objective



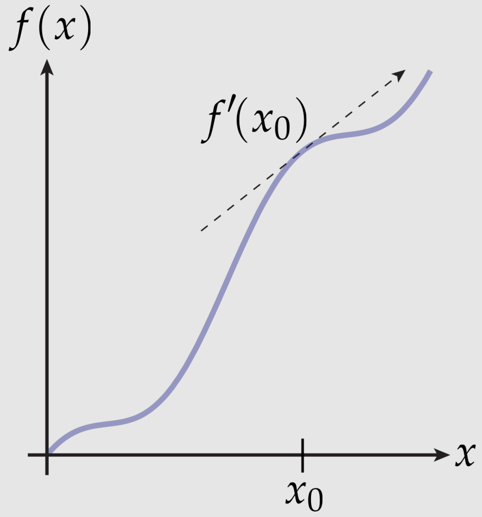
$$\frac{d}{dt}\phi(x) = \frac{d^2}{dx^2}\phi(x)$$



Derivative of a Slope

Measures the amount of change for an infinitesimal step:

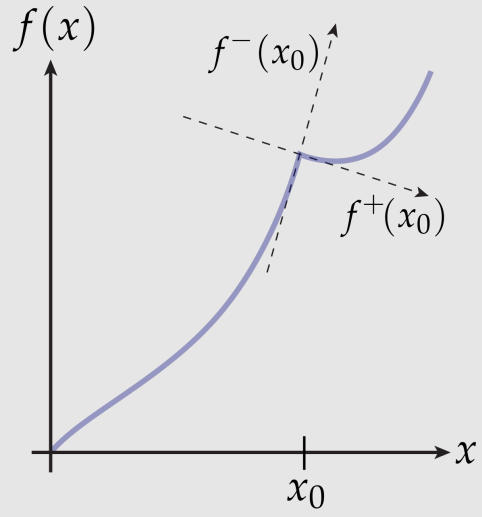
$$f'(x_0) := \lim_{\varepsilon \to 0} \frac{f(x_0 + \varepsilon) - f(x_0)}{\varepsilon}$$



What if the slopes do not match if we change directions?

$$f^+(x_0) := \lim_{\varepsilon \to 0} \frac{f(x_0 + \varepsilon) - f(x_0)}{\varepsilon}$$

$$f^-(x_0) := \lim_{\varepsilon \to 0} \frac{f(x_0) - f(x_0 - \varepsilon)}{\varepsilon}$$



Differentiable** only if $f^+ = -f^-$

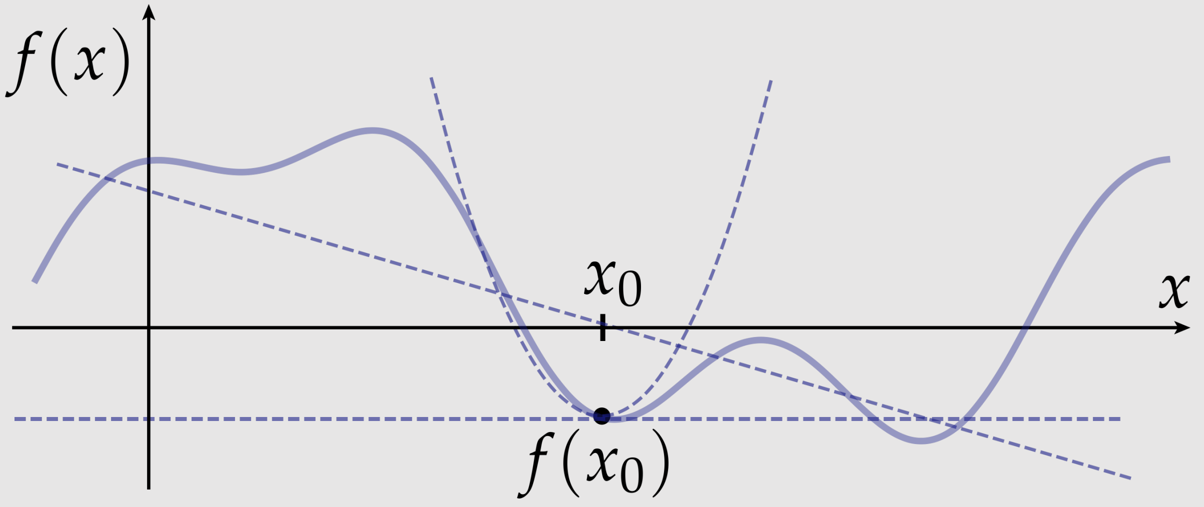
**Many functions in graphics are not differentiable!

Derivative as Best Linear Approximation

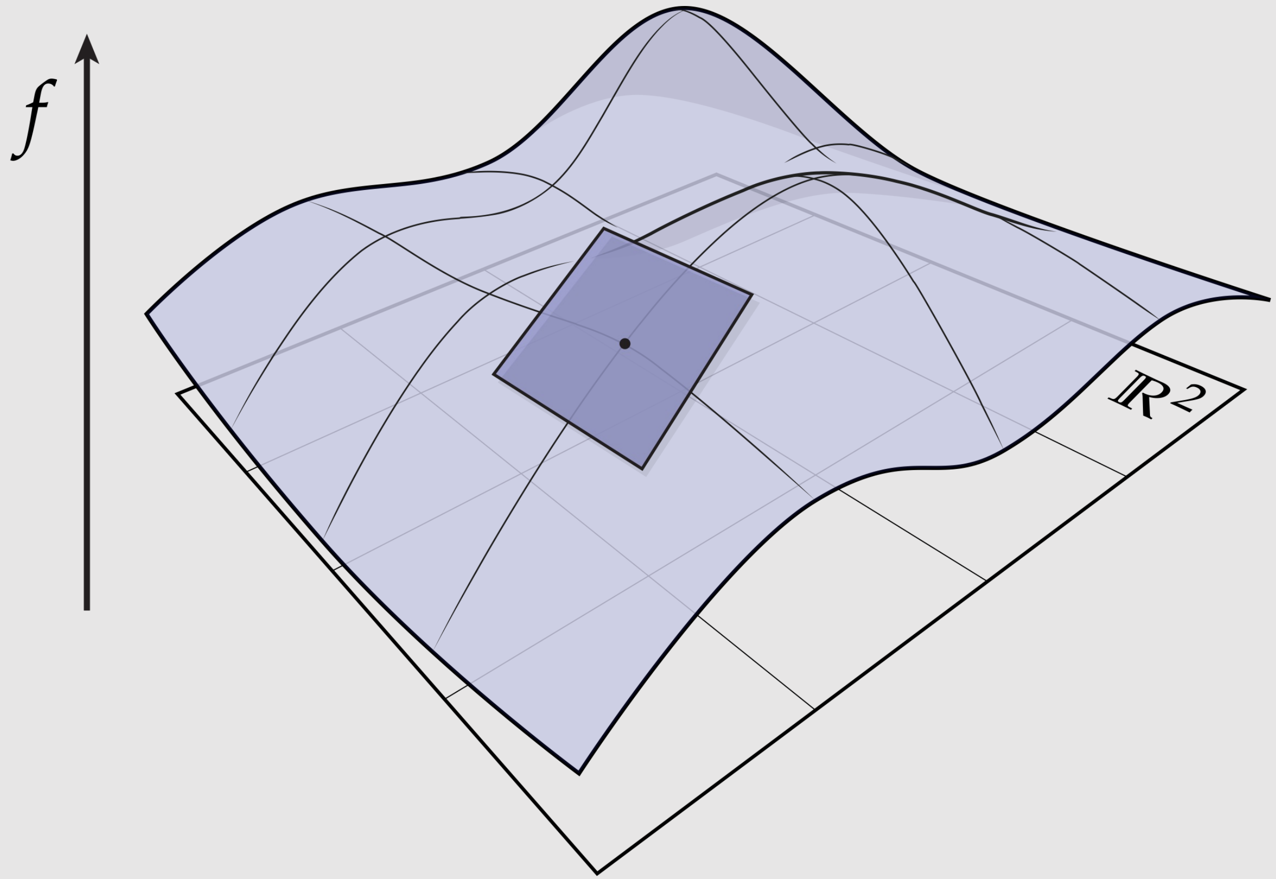
Any smooth function can be expressed as a **Taylor series:**

[constant] [linear] [quadratic]

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \cdots$$

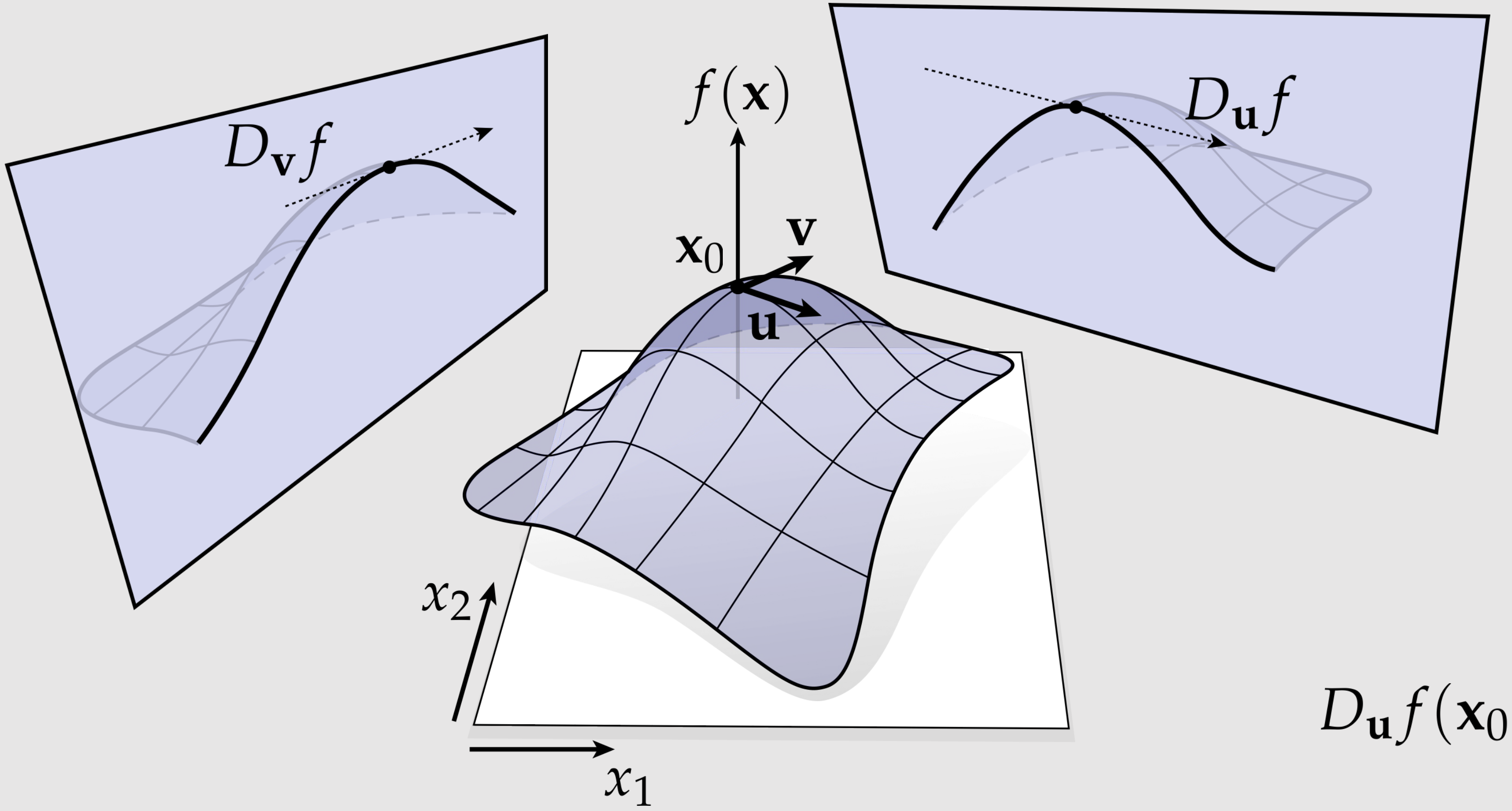


Derivative as Best Linear Approximation



Can be applied for multi-variable functions too.

Directional Derivative

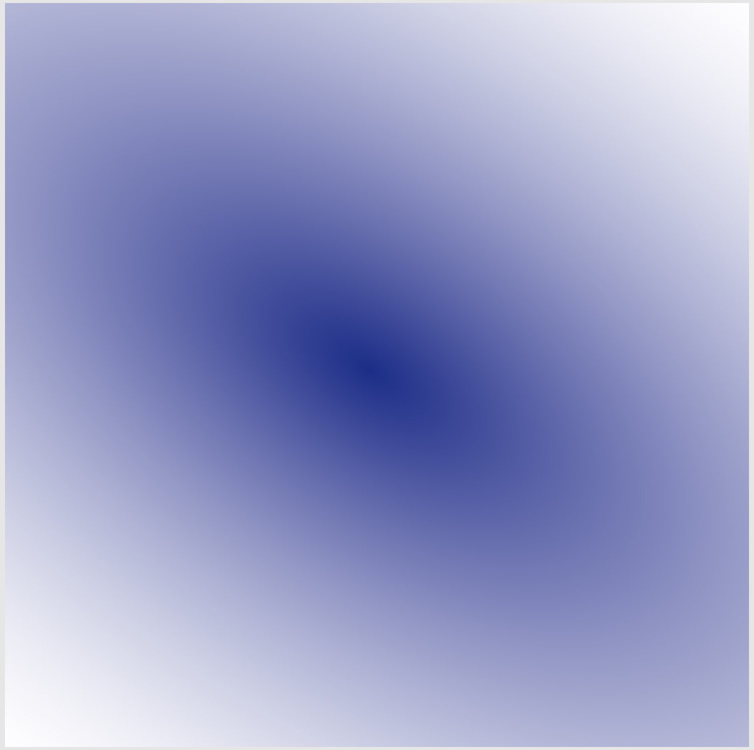


For multi-variable functions, we can take a slice of the function in the direction of vector $\boldsymbol{u}$ and compute the derivative from the resulting 2D function.

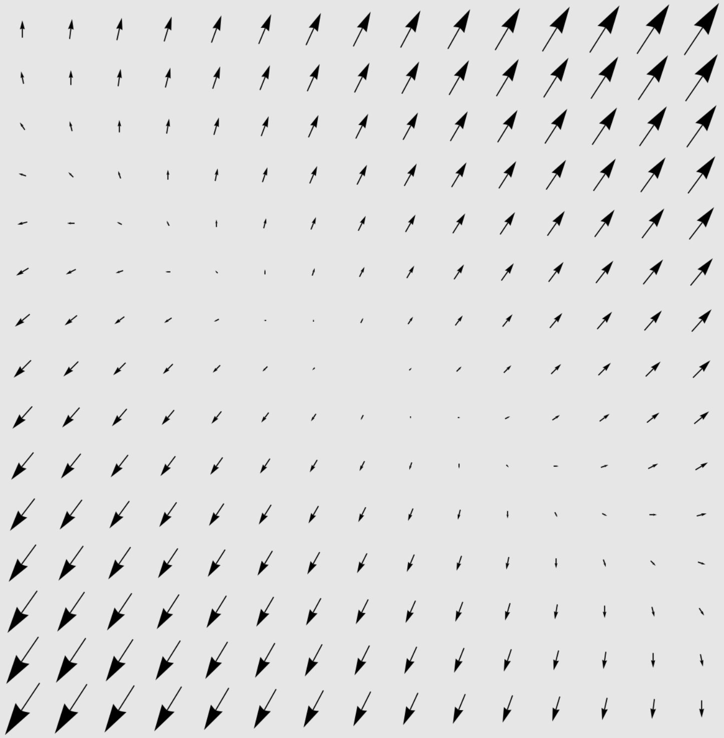
$$D_{\mathbf{u}}f(\mathbf{x}_0) := \lim_{\varepsilon \to 0} \frac{f(\mathbf{x}_0 + \varepsilon\mathbf{u}) - f(\mathbf{x}_0)}{\varepsilon}$$

Gradient

Given a multivariable function, we compute a vector at each location.



$$f(\mathbf{x})$$

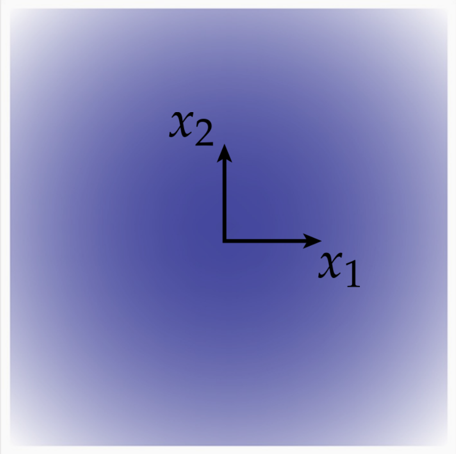


$$\nabla f(\mathbf{x})$$

[nabla]

Gradient in Coordinates

$$\nabla f = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix}$$



$f(\mathbf{x})$

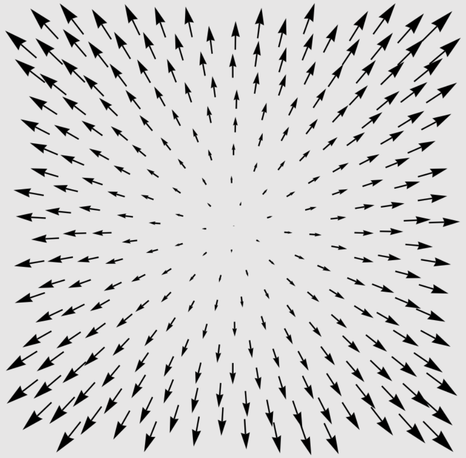
Example:

$$f(\mathbf{x}) := x_1^2 + x_2^2$$

$$\frac{\partial f}{\partial x_1} = \frac{\partial}{\partial x_1} x_1^2 + \frac{\partial}{\partial x_1} x_2^2 = 2x_1 + 0$$

$$\frac{\partial f}{\partial x_2} = \frac{\partial}{\partial x_2} x_1^2 + \frac{\partial}{\partial x_2} x_2^2 = 0 + 2x_2$$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = 2\mathbf{x}$$

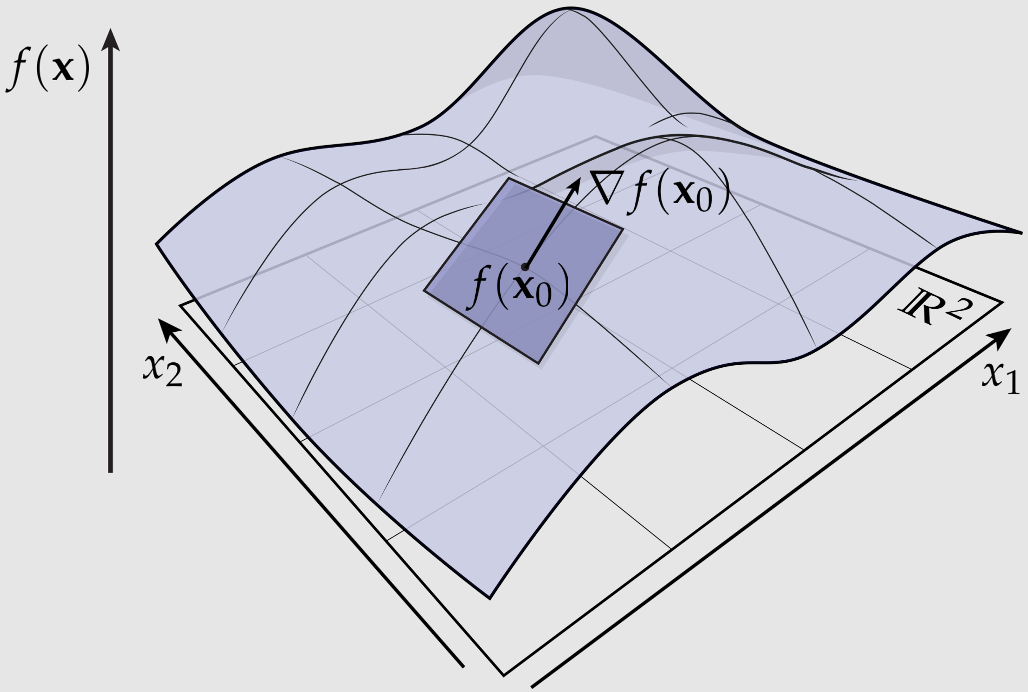


$\nabla f(\mathbf{x})$

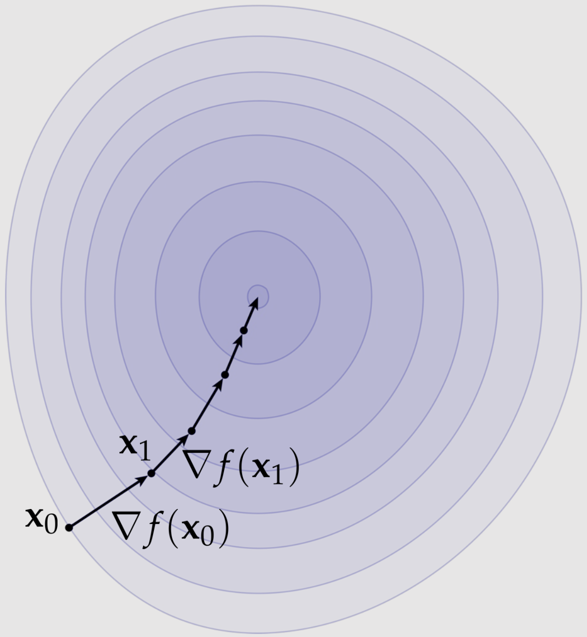
Gradient as Best Linear Approximation

- Gradient tells us the direction of steepest ascent.
  - Steepest descent if negative direction
  - No change if orthogonal direction

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle$$



- We can take multiple small steps to arrive at the maximum
  - How we make that step is its own field of research known as ‘optimization’



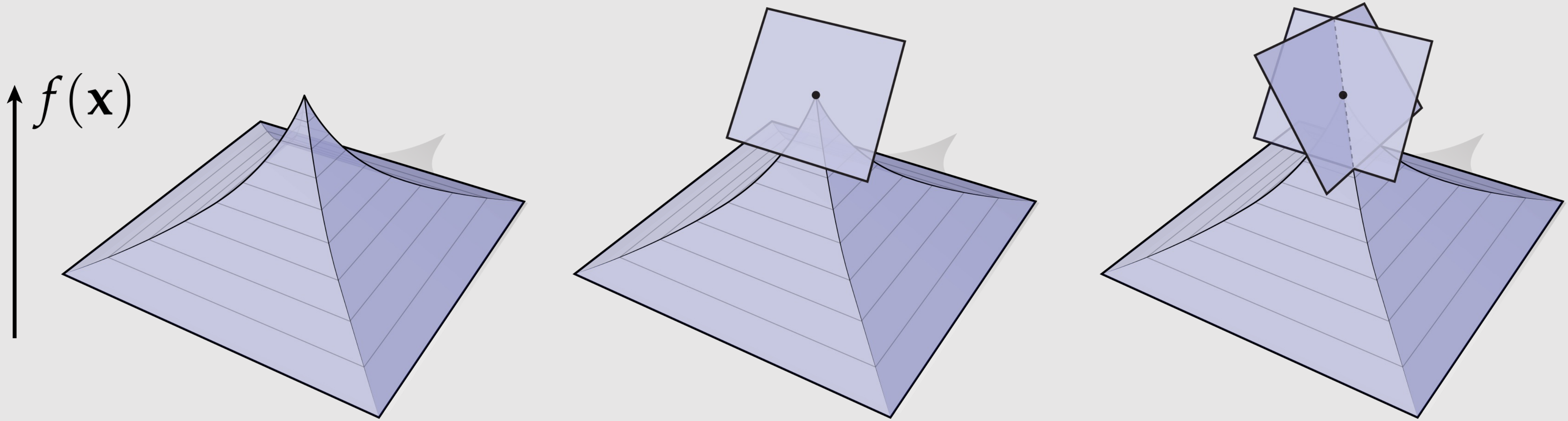
Gradient & Directional Derivative

The gradient $\nabla f(\mathbf{x})$ is a unique vector

$$\langle \nabla f(x), \mathbf{u} \rangle = D_{\mathbf{u}} f(\mathbf{x})$$

such that taking the inner product of the gradient along any direction gives the directional derivative.

Only works if function is differentiable!



Gradient of Dot Product

$$f := \mathbf{u}^\mathsf{T}\mathbf{v} = \sum_{i=1}^{n} u_i v_i$$

$$\frac{\partial}{\partial u_k}\sum_{i=1}^{n} u_i v_i = \sum_{i=1}^{n} \frac{\partial}{\partial u_k}(u_i v_i) = v_k$$

(equals zero unless i = k)

$$\Rightarrow \nabla_{\mathbf{u}} f = \begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix}$$

Gradient:

$$\boxed{\nabla_{\mathbf{u}}(\mathbf{u}^\mathsf{T}\mathbf{v}) = \mathbf{v}}$$

Not so different from $\frac{d}{dx}(xy) = y$

Gradients of Matrix-Valued Expressions**

For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and **symmetric** matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

Matrix Derivative	Looks Like
$\nabla_{\mathbf{x}}(\mathbf{x}^T \mathbf{y}) = \mathbf{y}$	$\frac{d}{dx} xy = y$
$\nabla_{\mathbf{x}}(\mathbf{x}^T \mathbf{x}) = 2\mathbf{x}$	$\frac{d}{dx} x^2 = 2x$
$\nabla_{\mathbf{x}}(\mathbf{x}^T A \mathbf{y}) = A\mathbf{y}$	$\frac{d}{dx} axy = ay$
$\nabla_{\mathbf{x}}(\mathbf{x}^T A \mathbf{x}) = 2A\mathbf{x}$	$\frac{d}{dx} ax^2 = 2ax$
$\ldots$	$\ldots$

**Excellent resource: Petersen & Pedersen, "The Matrix Cookbook"

L^2 Gradient

- Consider a function $F(f)$ that has an input function f
  - **Same idea:** the gradient of F with respect to f measures how changing the function f best increases F
    - Example:

$$F(f) := \langle\langle f, g \rangle\rangle$$

    - I claim the gradient is:

$$\nabla F = g$$

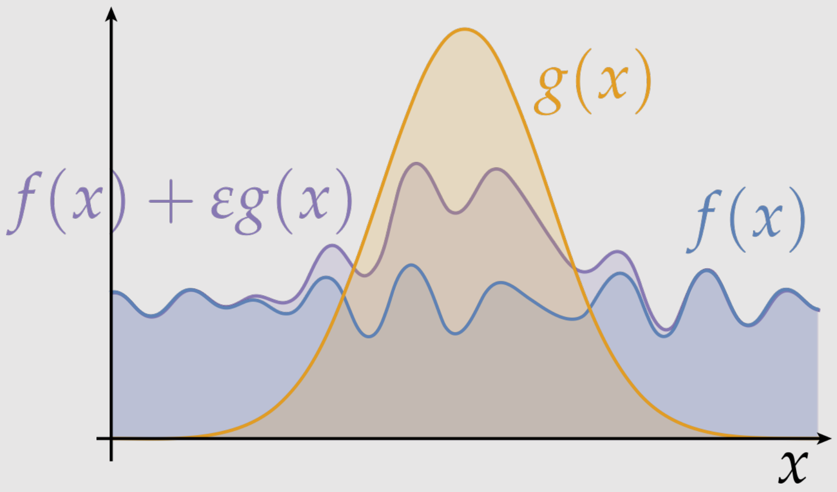
    - This means adding more of g to f increases ∇F
      - This is true for inner products!

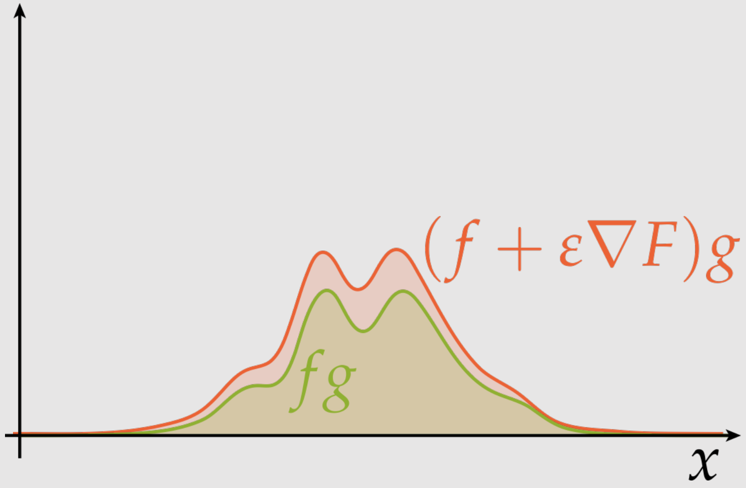
- How do we compute the gradient in general?
  - Look for a function ∇F such that:

$$\langle\langle \nabla F, u \rangle\rangle = D_u F$$

  - Where the directional derivative is:

$$D_u F(f) = \lim_{\varepsilon \to 0} \frac{F(f + \varepsilon u) - F(f)}{\varepsilon}$$





L^2 Gradient Example

Consider:

$$F(f) := ||f||^2$$

Apply the directional derivative formula for a given direction u:

$$\langle\langle \nabla F(f_0), u \rangle\rangle = \lim_{\varepsilon \to 0} \frac{F(f_0 + \varepsilon u) - F(f_0)}{\varepsilon}$$

Substitute F and expand the numerator $F(f_0 + \varepsilon u)$:

$$||f_0 + \varepsilon u||^2 = ||f_0||^2 + \varepsilon^2 ||u||^2 + 2\varepsilon \langle\langle f_0, u \rangle\rangle$$

Subtract the remaining $F(f_0)$ and divide by ε:

$$\lim_{\varepsilon \to 0} (\varepsilon ||u||^2 + 2\langle\langle f_0, u \rangle\rangle) = 2\langle\langle f_0, u \rangle\rangle$$

Set equal to the gradient term:

$$\langle\langle \nabla F(f_0), u \rangle\rangle = 2\langle\langle f_0, u \rangle\rangle$$

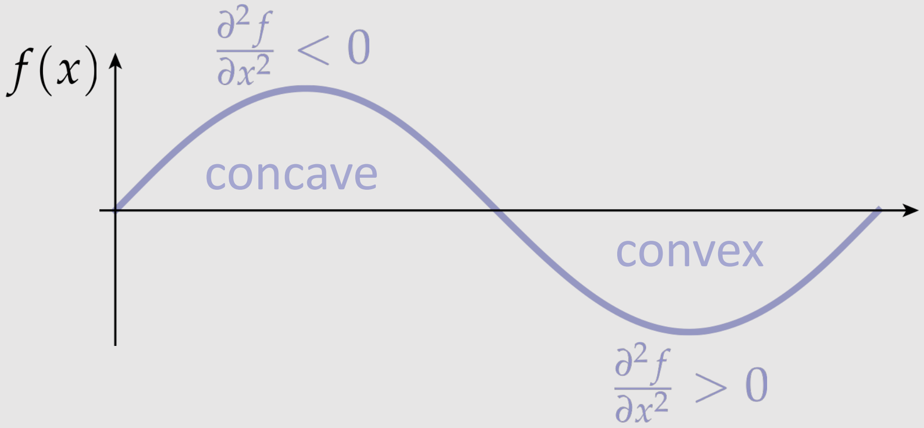
Solution:

$$\boxed{\nabla F(f_0) = 2f_0}$$

kinda looks like $\frac{d}{dx} x^2 = 2x$

Laplacian

- Measures the **curvature** of a function



- Several ways to calculate:
  - Divergence of gradient *(outside course scope)*:

$$\Delta f := \nabla \cdot \nabla f = \mathrm{div}(\mathrm{grad} f)$$

  - Sum of 2nd partial derivative:

$$\Delta f := \textstyle\sum_{i=1}^{n} \partial^2 f / \partial x_i^2$$

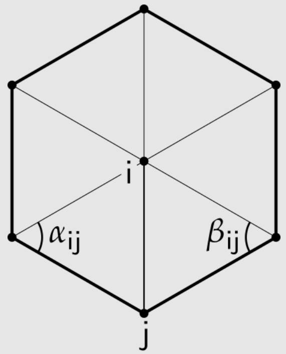
  - Gradient of Dirichlet energy *(outside course scope)*:

$$\Delta f := -\nabla_f \left(\tfrac{1}{2} ||\nabla f||^2\right)$$

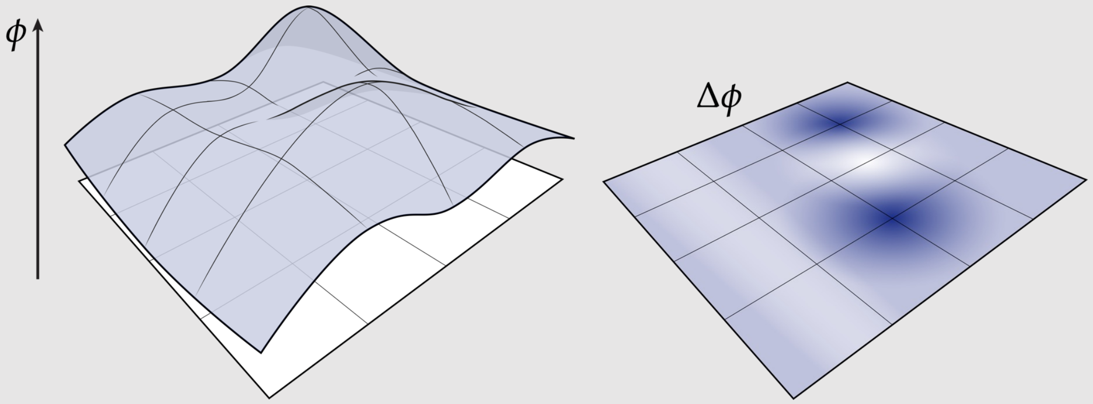
  - Variation of Surface Area:

	1	
1	-4	1
	1	

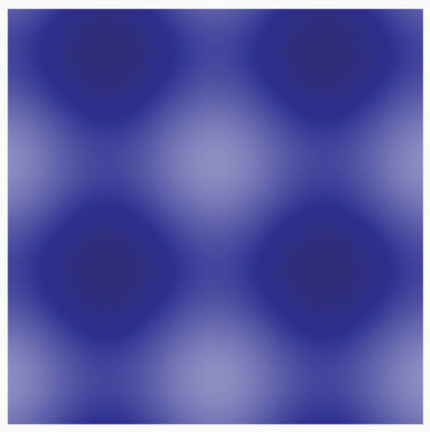
$$\frac{4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}}{h^2}$$



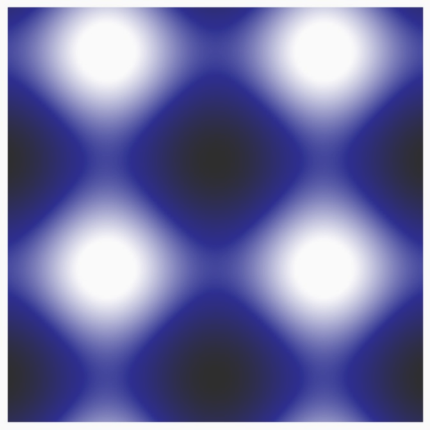
$$\frac{1}{2}\sum_j (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - u_i)$$



Laplacian Example



f



Δf

Consider:

$$f(x_1, x_2) := \cos(3x_1) + \sin(3x_2)$$

Using the following equation:

$$\Delta f := \sum_i \partial^2 f / \partial x_i^2$$

Compute the first partial:

$$\frac{\partial^2}{\partial x_1^2} f = \frac{\partial^2}{\partial x_1^2} \cos(3x_1) + \cancelto{0}{\frac{\partial^2}{\partial x_1^2} \sin(3x_2)} =$$

$$-3 \frac{\partial}{\partial x_1} \sin(3x_1) = -9 \cos(3x_1).$$

And the second:

$$\frac{\partial^2}{\partial x_2^2} f = -9 \sin(3x_2).$$

Add together:

$$\Delta f = -9(\cos(3x_1) + \sin(3x_2)) = -9f$$

When does this happen?

Hessian

- A matrix representing a gradient to the gradient
  - Matrix is always **symmetric**
    - Order of partial derivatives does not matter given f is continuous
- A gradient was a vector that gives us partial derivatives of the function
  - A hessian is an operator that gives us partial derivatives of the gradient:

$$(\nabla^2 f)\mathbf{u} := D_{\mathbf{u}}(\nabla f)$$

$$\nabla^2 f := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Taylor Series For Multivariate Functions

Using the **Hessian**, we can now write 2nd-order approximation of any smooth, multivariable function $f(x)$ around some point x_0:

[constant] [linear] [quadratic]

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \cdots$$

$$f(\mathbf{x}) \approx \underbrace{f(\mathbf{x}_0)}_{c \in \mathbb{R}} + \langle \underbrace{\nabla f(\mathbf{x}_0)}_{\mathbf{b} \in \mathbb{R}^n}, \mathbf{x} - \mathbf{x}_0 \rangle + \langle \underbrace{\nabla^2 f(x_0)}_{\mathbf{A} \in \mathbb{R}^{n \times n}}(\mathbf{x} - \mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle / 2$$

In matrix form:

$$f(\mathbf{u}) \approx \tfrac{1}{2}\mathbf{u}^\mathsf{T}\mathbf{A}\mathbf{u} + \mathbf{b}^\mathsf{T}\mathbf{u} + c, \quad \mathbf{u} := \mathbf{x} - \mathbf{x}_0$$

Recap

Charlie Brown (1984) Charles Schulz

- That was a lot of math
  - But now you should have the proper mathematical background to complete this course
- We will use **Linear Algebra**…
  - As an effective bridge between geometry, physics, computation, etc.
  - As a way to formulate a problem. Write the problem as Ax=b and ask the computer to solve
- We will use **Vector Calculus**…
  - As a basic language for talking about spatial relationships, transformations, etc.
  - For much of modern graphics (physically-based animation, geometry processing, etc.) formulated in terms of partial differential equations (PDEs) that use div, curl, Laplacian, and so on
- A0.0 will reinforce the content taught in this lecture
  - Be sure to refer back to the slides for help