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: $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
“commutative” or “abelian”

Vector multiplication: $a(b\mathbf{u}) = (ab)\mathbf{u}$
Basic Vector Operations

Order of operations for adding and scaling do not matter

\[ a(u + v) = au + av \]
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 “0” such that \( \mathbf{v} + 0 = 0 + \mathbf{v} = \mathbf{v} \)
- For every \( \mathbf{v} \) there is a vector “\(-\mathbf{v}\)” such that \( \mathbf{v} + (-\mathbf{v}) = 0 \)
- \( 1\mathbf{v} = \mathbf{v} \)
- \( a(b\mathbf{v}) = (ab)\mathbf{v} \)
- \( a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v} \)
- \( (a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v} \)

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:** $(1.23, 4.56, \pi/2)$ 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:

\[(f + g)(x) := f(x) + g(x)\]

We can also scale a function:

\[(af)(x) := a(f(x))\]
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 “0” such that $\mathbf{v} + 0 = 0 + \mathbf{v} = \mathbf{v}$
- For every $\mathbf{v}$ there is a vector “$-\mathbf{v}$” such that $\mathbf{v} + (-\mathbf{v}) = 0$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

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 $\mathbf{v}$, $|\mathbf{v}|$ is its **length** / **magnitude** / **norm**. Intuitively, this captures how “big” the vector is.
Norm Properties

For one thing, it shouldn’t be negative!

\[ |\mathbf{u}| \geq 0 \quad |\mathbf{u}| = 0 \iff \mathbf{u} = \mathbf{0} \]

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.**

**sometimes called the “triangle inequality” since the diagram looks like a triangle

\[ |\mathbf{u}| + |\mathbf{v}| \geq |\mathbf{u} + \mathbf{v}| \]
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 \( a \):

- \( |v| \geq 0 \)
- \( |v| = 0 \iff v = 0 \)
- \( |av| = |a||v| \)
- \( |u| + |v| \geq |u + v| \)
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

- **Inner product** measures the “similarity” of vectors, or how well vectors “line up.”

- The dot product of two vectors is commutative:

\[ \langle u, v \rangle = \langle v, u \rangle \]
Inner Product

- 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:
  $$\langle 2u, v \rangle = 2\langle u, v \rangle$$
- Vectors need to be normalized when computing similarity!

- Any vector will always be aligned with itself:
  $$\langle u, u \rangle \geq 0$$
- The dot product of any unit vector with itself is:
  $$\langle u, u \rangle = 1$$
- Thus for a unit vector $\hat{u} := u / |u|$
  $$\langle u, u \rangle = \langle |u|\hat{u}, |u|\hat{u} \rangle = |u|^2\langle \hat{u}, \hat{u} \rangle = |u|^2 \cdot 1 = |u|^2$$
Inner Product Formal Definition

An inner product is any function that assigns to any two vectors $u, v$ a number $\langle u, v \rangle$ satisfying the following properties:

- $\langle u, v \rangle = \langle v, u \rangle$
- $\langle u, u \rangle \geq 0$
- $\langle u, u \rangle = 0 \iff u = 0$
- $\langle au, v \rangle = a \langle u, v \rangle$
- $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$

- **Euclidean inner product**
  \[ \langle u, v \rangle := |u||v| \cos(\theta) \]

- **Cartesian inner product**
  \[ u \cdot v := u_1v_1 + \cdots + u_nv_n \]
Inner Product In Cartesian Coordinates

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

\[ \langle \mathbf{u}, \mathbf{v} \rangle = 4 \cdot 1 + 1 \cdot 3 = 7 \]
L² Inner Product Of Functions

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

Example:

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

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

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:

\[
\begin{align*}
f(u + v) &= f(u) + f(v) \\
f(au) &= af(u)
\end{align*}
\]

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

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

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 $f(u) = \int_0^1 u(x)dx$ 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 $S_1$

$$\text{span}(u_1, \ldots, u_k) = \left\{ x \in V \mid x = \sum_{i=1}^{k} a_i u_i, \ a_1, \ldots, a_k \in \mathbb{R} \right\}$$
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:

\[
\text{span}(e_1, \ldots, e_n) = \mathbb{R}^n
\]

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 \( \mathbb{R}^2 \)?

(A)  
(B)  
(C)  
(D)  
(E)  

Which of the following are bases for \( \mathbb{R}^2 \)?
Orthonormal Basis

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

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

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

$$\langle e_i, e_j \rangle = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases}$$

*Common bug: projecting a vector onto a basis that is NOT orthonormal while continuing to use standard norm / inner product.*
Given a collection of basis vectors \( a_1, \ldots, a_n \), we can find an orthonormal basis \( e_1, \ldots, e_n \) 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

\[
\begin{align*}
\mathbf{u} & := p_1 - p_0 \\
\mathbf{v} & := p_2 - p_0 \\
\mathbf{e}_1 & := \mathbf{u} / |\mathbf{u}| \\
\mathbf{\tilde{v}} & := \mathbf{v} - \langle \mathbf{v}, \mathbf{e}_1 \rangle \mathbf{e}_1 \\
\mathbf{e}_2 & := \mathbf{\tilde{v}} / |\mathbf{\tilde{v}}|
\end{align*}
\]

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

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

- 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**
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

\[
\begin{align*}
x + 2y &= 3 \\
4x + 5y &= 6 \\
x &= 3 - 2y \\
4(3 - 2y) + 5y &= 6 \\
y &= 2 \\
x &= -1
\end{align*}
\]
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.)
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

What does this little block of funny numbers do?
**Linear Maps As Matrices**

Example: consider the linear map:

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

\( \mathbf{a} \) 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 \( \mathbf{u} \) maps it to \( f(\mathbf{u}) \):

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

How to map \( f(\mathbf{u}) \) back to \( \mathbf{u} \)? 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

\[ \sqrt{\text{det}(u, v, u \times v)} = |u||v| \sin(\theta) \]

- \( \theta \) is angle between \( u \) and \( v \)
- “det” is determinant of three column vectors

\[ \begin{vmatrix} e_1 & e_2 & e_3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} \]

(mnemonic)
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:

\[ \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} = \sum_{i=1}^{n} u_i \mathbf{v}_i \]

Cross product as a matrix multiplication:

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

\[ \mathbf{u} \times \mathbf{v} = \mathbf{\hat{u}} \mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} \]
Dot And Cross Products

Useful to notice \( \mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u} \)

This means:

\[
\mathbf{v} \times \mathbf{u} = -\hat{\mathbf{u}} \mathbf{v} = \hat{\mathbf{u}}^T \mathbf{v}
\]

\[
\mathbf{u} := (u_1, u_2, u_3) \Rightarrow \hat{\mathbf{u}} := \begin{bmatrix}
0 & -u_3 & u_2 \\
-3 & 0 & -u_1 \\
u_2 & u_1 & 0
\end{bmatrix}
\]

\[
\mathbf{u} \times \mathbf{v} = \hat{\mathbf{u}} \mathbf{v} = \begin{bmatrix}
0 & -u_3 & u_2 \\
u_2 & 0 & -u_1 \\
-u_2 & u_1 & 0
\end{bmatrix} \begin{bmatrix}
v_1 \\
v_2 \\
v_3
\end{bmatrix}
\]
Determinant

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

The determinant of \( \mathbf{A} \) is:

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

Great, but what does that mean?
Determinant

\[ \text{det}(u,v,w) \text{ encodes signed volume of parallelepiped with edge vectors } u, v, w. \]

\[
\text{det}(u,v,w) = (u \times v) \cdot w = (v \times w) \cdot u = (w \times u) \cdot v
\]

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 \( \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \ldots \)

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

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

- The \( \text{det}(A) \) 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**:

\[
f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x-x_0)^2}{2!}f''(x_0) + \cdots
\]
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 $\mathbf{u}$ and compute the derivative from the resulting 2D function.

$$D_{\mathbf{u}}f(x_0) := \lim_{\varepsilon \to 0} \frac{f(x_0 + \varepsilon \mathbf{u}) - f(x_0)}{\varepsilon}$$
Gradient

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

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \]

Example:

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

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

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

\[ \nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = 2\mathbf{x} \]
Gradient as Best Linear Approximation

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

\[ f(x) \approx f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle \]

- 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 $\nabla f(x)$ is a unique vector

$$\left< \nabla f(x), u \right> = D_uf(x)$$

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

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

$$(\text{equals zero unless } i = k)$$

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

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

Gradient:

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

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 $\mathbf{A} \in \mathbb{R}^{n \times n}$:

<table>
<thead>
<tr>
<th>MATRIX DERIVATIVE</th>
<th>LOOKS LIKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nabla \mathbf{x} (\mathbf{x}^T \mathbf{y}) = \mathbf{y}$</td>
<td>$\frac{d}{dx} \mathbf{x} \mathbf{y} = \mathbf{y}$</td>
</tr>
<tr>
<td>$\nabla \mathbf{x} (\mathbf{x}^T \mathbf{x}) = 2\mathbf{x}$</td>
<td>$\frac{d}{dx} \mathbf{x}^2 = 2\mathbf{x}$</td>
</tr>
<tr>
<td>$\nabla \mathbf{x} (\mathbf{x}^T \mathbf{A} \mathbf{y}) = \mathbf{A} \mathbf{y}$</td>
<td>$\frac{d}{dx} \mathbf{a} \mathbf{x} \mathbf{y} = \mathbf{a} \mathbf{y}$</td>
</tr>
<tr>
<td>$\nabla \mathbf{x} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = 2\mathbf{A} \mathbf{x}$</td>
<td>$\frac{d}{dx} \mathbf{a} \mathbf{x}^2 = 2\mathbf{a} \mathbf{x}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Excellent resource: Petersen & Pedersen, “The Matrix Cookbook”**
L² 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:
      $$F(f) := \langle f, g \rangle$$
      • I claim the gradient is:
        $$\nabla F = g$$
      • This means adding more of $g$ to $f$ increases $\nabla F$
        • This is true for inner products!

• How do we compute the gradient in general?
  • Look for a function $\nabla F$ such that:
    $$\langle \nabla F, u \rangle = D_u F$$
  • Where the directional derivative is:
    $$D_u F(f) = \lim_{\varepsilon \to 0} \frac{F(f + \varepsilon u) - F(f)}{\varepsilon}$$
L² 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 \( \varepsilon \):

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

Set equal to the gradient term:

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

Solution:

\[
\nabla F(f_0) = 2f_0
\]

kinda looks like \( \frac{d}{dx} x^2 = 2x \)
Laplacian

- Measures the \textit{curvature} of a function

- Several ways to calculate:
  - Divergence of gradient (\textit{outside course scope}):
    \[
    \Delta f := \nabla \cdot \nabla f = \text{div}(\text{grad} f)
    \]
  - Sum of 2\textsuperscript{nd} partial derivative:
    \[
    \Delta f := \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}
    \]
  - Gradient of Dirichlet energy (\textit{outside course scope}):
    \[
    \Delta f := - \nabla_f \left( \frac{1}{2} \| \nabla f \|^2 \right)
    \]
  - Variation of Surface Area:
    \[
    \phi
    \]

\[
4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j+1} - u_{i,j-1} \quad \frac{1}{h^2} \sum_j \left( \cot \alpha_{ij} + \cot \beta_{ij} \right) (u_j - u_i)
\]
Laplacian Example

Consider:

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

Using the following equation:

\[ \Delta f := \sum_i \frac{\partial^2 f}{\partial x_i^2} \]

Compute the first partial:

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

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

And the second:

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

Add together:

\[ \Delta f = -9(\cos(3x_1) + \sin(3x_2)) = -9f \]
Hessian

- A matrix representing a gradient to the gradient
  - Matrix is always \textit{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:

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

\[
(\nabla^2 f) u := D_u(\nabla f)
\]
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 \( x_0 \):

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

In matrix form:

\[
f(u) \approx \frac{1}{2} u^T A u + b^T u + c, \quad u := x - x_0
\]
Recap

• 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