A Bigger Mathematical Picture for Computer Graphics

Eric Lengyel, PhD
Terathon Software

Common math in computer graphics

- Dot / cross products, scalar triple product
- Planes as 4D vectors
- Homogeneous coordinates
- Plücker coordinates for 3D lines
- Transforming normal vectors and planes with the inverse transpose of a matrix

Common math in computer graphics

- These concepts often used without a complete understanding of the big picture
- Can be used in a way that is not natural
- Different pieces used separately without knowledge of the connection among them

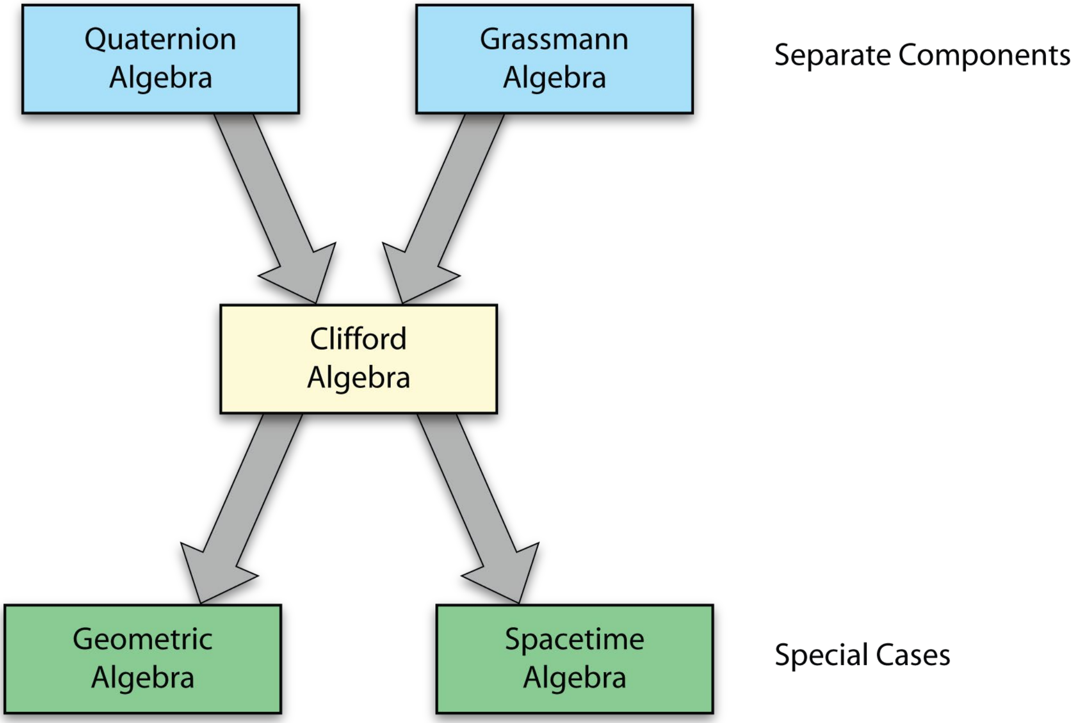
There is a bigger picture

- All of these arise as part of a single mathematical system discovered by Hermann Grassmann.
- Understanding the big picture provides deep insights into seemingly unusual properties
- Knowledge of the relationships among these concepts makes better 3D programmers

History

Hamilton 1843	Grassmann 1844	Clifford 1878
		
Quaternions	Exterior algebra	Clifford algebra

History


Quaternion Algebra
Grassmann Algebra
Separate Components
Clifford Algebra
Geometric Algebra
Spacetime Algebra
Special Cases

Outline

- Grassmann algebra in 3-4 dimensions
  - Wedge product, bivectors, trivectors...
  - Transformations
  - Homogeneous model
  - Geometric computation
  - Programming considerations

The wedge product

- Also known as:
  - The progressive product
  - The exterior product

- Gets name from symbol: $\mathbf{a} \wedge \mathbf{b}$

- Read “**a** *wedge* **b**”

The wedge product

- Operates on scalars, vectors, and more
  - Ordinary multiplication for scalars *s* and *t*:

$$s \wedge t = t \wedge s = st$$

$$s \wedge \mathbf{v} = \mathbf{v} \wedge s = s\mathbf{v}$$

  - The square of a vector **v** is always zero:

$$\mathbf{v} \wedge \mathbf{v} = 0$$

Wedge product anticommutativity

- Zero square implies vectors anticommute

$$(\mathbf{a}+\mathbf{b})\wedge(\mathbf{a}+\mathbf{b})=0$$

$$\mathbf{a}\wedge\mathbf{a}+\mathbf{a}\wedge\mathbf{b}+\mathbf{b}\wedge\mathbf{a}+\mathbf{b}\wedge\mathbf{b}=0$$

$$\mathbf{a}\wedge\mathbf{b}+\mathbf{b}\wedge\mathbf{a}=0$$

$$\mathbf{a}\wedge\mathbf{b}=-\mathbf{b}\wedge\mathbf{a}$$

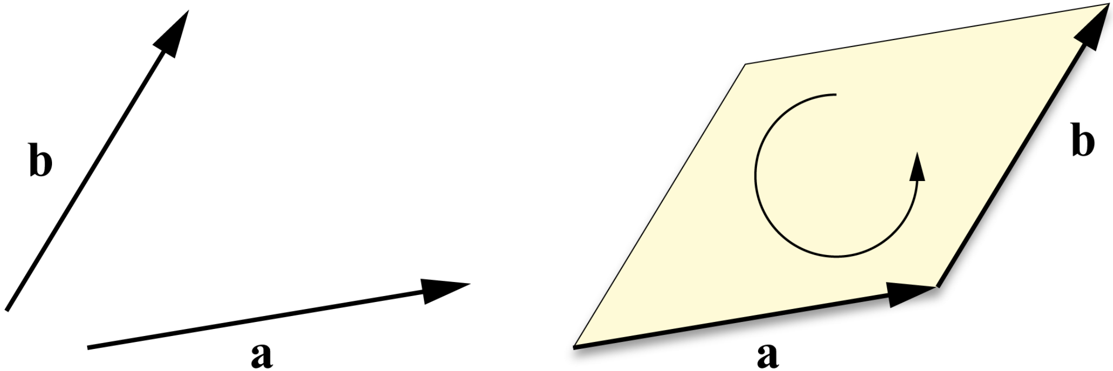
Bivectors

- Wedge product between two vectors produces a “bivector”
  - A new mathematical entity
  - Distinct from a scalar or vector
  - Represents an oriented 2D area

  - A vector represents an oriented 1D direction
  - Scalars are zero-dimensional values

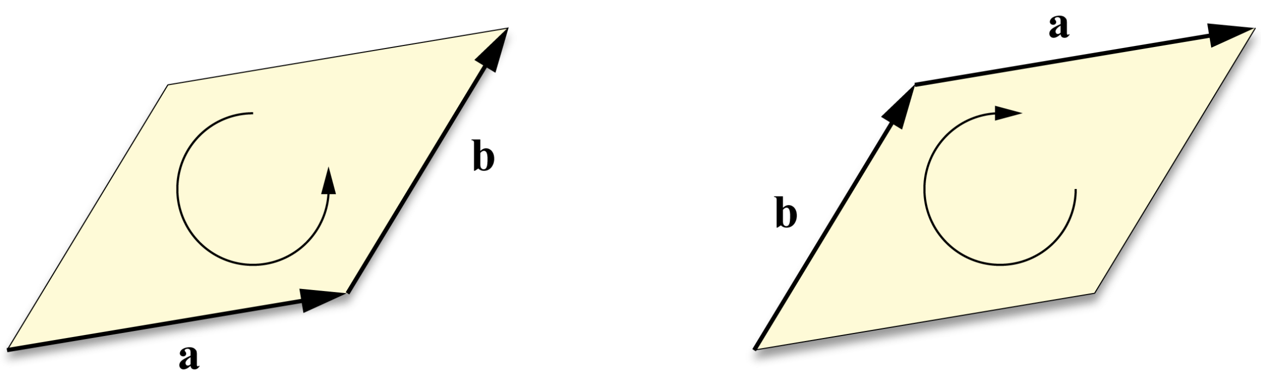
Bivectors

- Bivector is two directions and magnitude



Bivectors

- Order of multiplication matters



$$\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a}$$

Bivectors in 3D

- Start with 3 orthonormal basis vectors:

$$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$$

- Then a 3D vector **a** can be expressed as

$$a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3$$

Bivectors in 3D

$$\mathbf{a} \wedge \mathbf{b} = (a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3) \wedge (b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3)$$

$$\begin{aligned}\mathbf{a} \wedge \mathbf{b} = {} & a_1b_2(\mathbf{e}_1 \wedge \mathbf{e}_2) + a_1b_3(\mathbf{e}_1 \wedge \mathbf{e}_3) + a_2b_1(\mathbf{e}_2 \wedge \mathbf{e}_1) \\ & + a_2b_3(\mathbf{e}_2 \wedge \mathbf{e}_3) + a_3b_1(\mathbf{e}_3 \wedge \mathbf{e}_1) + a_3b_2(\mathbf{e}_3 \wedge \mathbf{e}_2)\end{aligned}$$

$$\begin{aligned}\mathbf{a} \wedge \mathbf{b} = {} & (a_2b_3 - a_3b_2)(\mathbf{e}_2 \wedge \mathbf{e}_3) + (a_3b_1 - a_1b_3)(\mathbf{e}_3 \wedge \mathbf{e}_1) \\ & + (a_1b_2 - a_2b_1)(\mathbf{e}_1 \wedge \mathbf{e}_2)\end{aligned}$$

Bivectors in 3D

- The result of the wedge product has three components on the basis

$$\mathbf{e}_2 \wedge \mathbf{e}_3, \quad \mathbf{e}_3 \wedge \mathbf{e}_1, \quad \mathbf{e}_1 \wedge \mathbf{e}_2$$

- Written in order of which basis *vector* is missing from the basis *bivector*

Bivectors in 3D

- Do the components look familiar?

$$\mathbf{a} \wedge \mathbf{b} = (a_2 b_3 - a_3 b_2)(\mathbf{e}_2 \wedge \mathbf{e}_3) + (a_3 b_1 - a_1 b_3)(\mathbf{e}_3 \wedge \mathbf{e}_1) + (a_1 b_2 - a_2 b_1)(\mathbf{e}_1 \wedge \mathbf{e}_2)$$

- These are identical to the components produced by the cross product **a** × **b**

Shorthand notation

$$\mathbf{e}_{12} = \mathbf{e}_1 \wedge \mathbf{e}_2$$

$$\mathbf{e}_{23} = \mathbf{e}_2 \wedge \mathbf{e}_3$$

$$\mathbf{e}_{31} = \mathbf{e}_3 \wedge \mathbf{e}_1$$

$$\mathbf{e}_{123} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$$

Bivectors in 3D

$$\mathbf{a} \wedge \mathbf{b} = (a_2 b_3 - a_3 b_2)\mathbf{e}_{23} + (a_3 b_1 - a_1 b_3)\mathbf{e}_{31} + (a_1 b_2 - a_2 b_1)\mathbf{e}_{12}$$

Comparison with cross product

- The cross product is not associative:

$$(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} \neq \mathbf{a} \times (\mathbf{b} \times \mathbf{c})$$

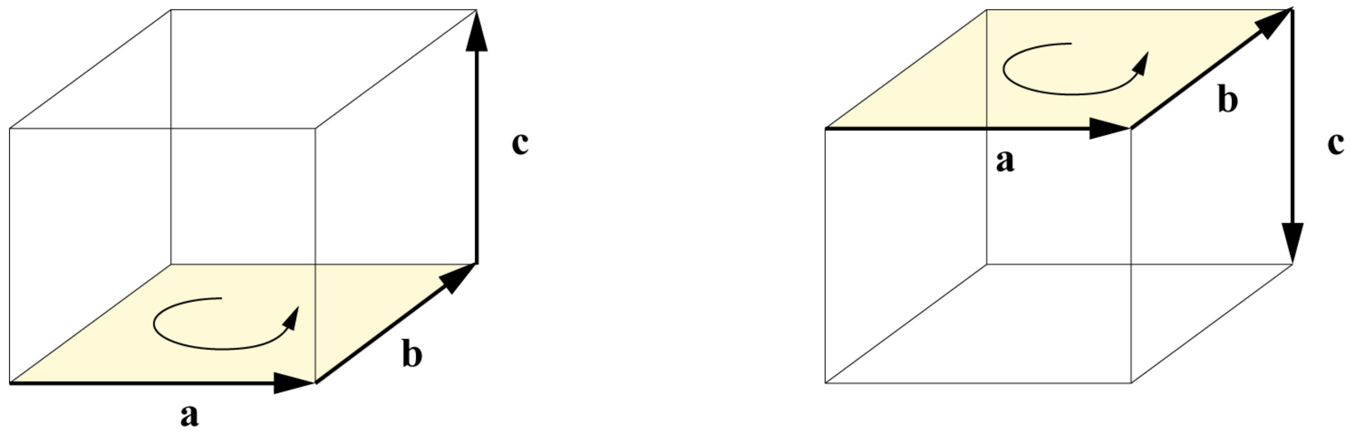
- The cross product is only defined in 3D

- The wedge product is associative,
and it's defined in all dimensions

Trivectors

- Wedge product among three vectors produces a “trivector”
  - Another new mathematical entity
  - Distinct from scalars, vectors, and bivectors
  - Represents a 3D oriented volume

Trivectors



$$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$$

Trivectors in 3D

- A 3D trivector has one component:

$$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} =$$
$$(a_1 b_2 c_3 + a_2 b_3 c_1 + a_3 b_1 c_2 - a_1 b_3 c_2 - a_2 b_1 c_3 - a_3 b_2 c_1) \cdot$$
$$(\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3)$$

- The magnitude is $\det([\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}])$

Trivectors in 3D

- 3D trivector also called *pseudoscalar* or *antiscalar*
- Only one component, so looks like a scalar
- But flips sign under reflection

Scalar Triple Product

- The product

$$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$$

produces the same magnitude as

$$(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

but also extends to higher dimensions

Grading

- The *grade* of an entity is the number of vectors wedged together to make it

- Scalars have grade 0
- Vectors have grade 1
- Bivectors have grade 2
- Trivectors have grade 3
- Etc.

3D multivector algebra

- 1 scalar element
- 3 vector elements
- 3 bivector elements
- 1 trivector element
- No higher-grade elements
- Total of 8 *multivector* basis elements

Multivectors in general dimension

- In n dimensions, the number of basis k-vector elements is

$$\binom{n}{k}$$

- This produces a nice symmetry
- Total number of basis elements always 2^n

Multivectors in general dimension

Dimension	Graded elements
1	1 1
2	1 2 1
3	1 3 3 1
4	1 4 6 4 1
5	1 5 10 10 5 1

Four dimensions

- Four basis vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$
- Number of basis bivectors is

$$\binom{4}{2} = 6$$

- There are 4 basis trivectors

Vector / bivector confusion

- In 3D, vectors have three components
- In 3D, bivectors have three components

- Thus, vectors and bivectors *look like* the same thing!
- This is a big reason why knowledge of the difference is not widespread

Cross product peculiarities

- Physicists noticed a long time ago that the cross product produces a *different* kind of vector
- They call it an “axial vector”, “pseudovector”, “covector”, or “covariant vector”
- It transforms differently than ordinary “polar vectors” or “contravariant vectors”

Cross product transform

- Simplest example is a reflection:

$$\mathbf{M} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Cross product transform

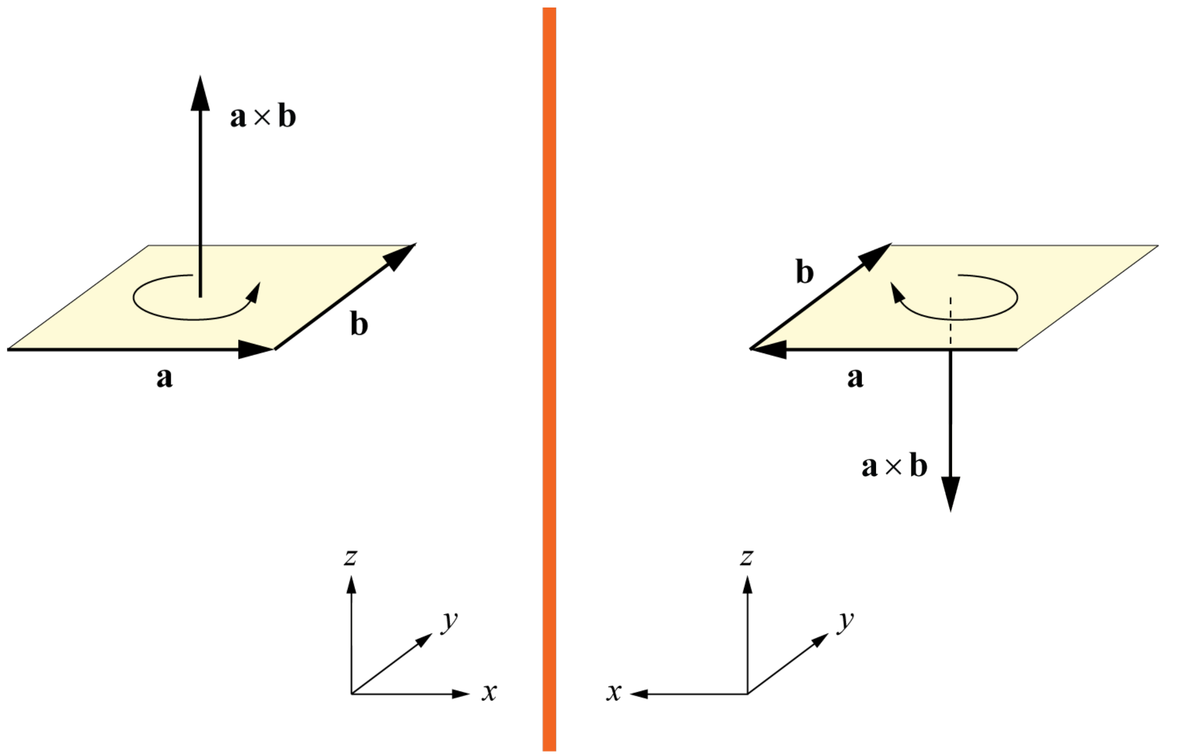
$$(1,0,0)\times(0,1,0)=(0,0,1)$$

$$\mathbf{M}(1,0,0)\times\mathbf{M}(0,1,0)$$
$$=(-1,0,0)\times(0,1,0)=(0,0,-1)$$

- Not the same as

$$\mathbf{M}(0,0,1)=(0,0,1)$$

Cross product transform



Cross product transform

- In general, for 3 x 3 matrix **M**,

$$\mathbf{M}\left(a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3\right) = a_1\mathbf{M}_1 + a_2\mathbf{M}_2 + a_3\mathbf{M}_3$$

$$\mathbf{Ma} \times \mathbf{Mb} =$$

$$\left(a_1\mathbf{M}_1 + a_2\mathbf{M}_2 + a_3\mathbf{M}_3\right) \times \left(b_1\mathbf{M}_1 + b_2\mathbf{M}_2 + b_3\mathbf{M}_3\right)$$

Cross product transform

$$
\begin{aligned}
&\mathbf{Ma} \times \mathbf{Mb} = \\
&\quad (a_2 b_3 - a_3 b_2)(\mathbf{M}_2 \times \mathbf{M}_3) \\
&\quad + (a_3 b_1 - a_1 b_3)(\mathbf{M}_3 \times \mathbf{M}_1) \\
&\quad + (a_1 b_2 - a_2 b_1)(\mathbf{M}_1 \times \mathbf{M}_2)
\end{aligned}
$$

Products of matrix columns

- What are these cross products?

$$(\mathbf{M}_2 \times \mathbf{M}_3) \cdot \mathbf{M}_1 = \det \mathbf{M}$$
$$(\mathbf{M}_3 \times \mathbf{M}_1) \cdot \mathbf{M}_2 = \det \mathbf{M}$$
$$(\mathbf{M}_1 \times \mathbf{M}_2) \cdot \mathbf{M}_3 = \det \mathbf{M}$$

- They are complements of the columns of **M**

Matrix inversion

- Cross products as rows of matrix:

$$\begin{bmatrix} \mathbf{M}_2 \times \mathbf{M}_3 \\ \mathbf{M}_3 \times \mathbf{M}_1 \\ \mathbf{M}_1 \times \mathbf{M}_2 \end{bmatrix} \mathbf{M} = \begin{bmatrix} \det \mathbf{M} & 0 & 0 \\ 0 & \det \mathbf{M} & 0 \\ 0 & 0 & \det \mathbf{M} \end{bmatrix}$$

- This forms inverse of **M** times det **M**

Cross product transform

- Transforming the cross product requires the inverse matrix:

$$\begin{bmatrix} \mathbf{M}_2 \times \mathbf{M}_3 \\ \mathbf{M}_3 \times \mathbf{M}_1 \\ \mathbf{M}_1 \times \mathbf{M}_2 \end{bmatrix} = (\det \mathbf{M})\mathbf{M}^{-1}$$

Cross product transform

- Inverse transpose correctly transforms result of cross product:

$$(\det \mathbf{M})\mathbf{M}^{-T}\begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} =$$

$$(a_2b_3 - a_3b_2)(\mathbf{M}_2 \times \mathbf{M}_3) + (a_3b_1 - a_1b_3)(\mathbf{M}_3 \times \mathbf{M}_1)$$
$$+(a_1b_2 - a_2b_1)(\mathbf{M}_1 \times \mathbf{M}_2)$$

Cross product transform

- Transformation formula:

$$\mathbf{Ma} \times \mathbf{Mb} = (\det \mathbf{M})\, \mathbf{M}^{-T} (\mathbf{a} \times \mathbf{b})$$

- Result of cross product must be transformed by inverse transpose times determinant

Cross product transform

- If **M** is orthogonal, then inverse transpose is the same as **M**
- If the determinant is positive, then it can be left out if you don’t care about length
- Determinant times inverse transpose is called *adjugate transpose*

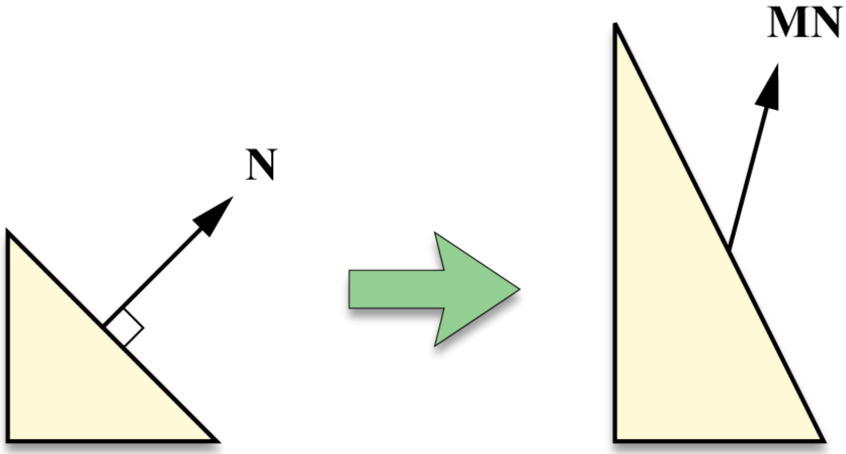
Cross product transform

- What’s really going on here?
- When we take a cross product, we are really creating a bivector
- Bivectors are not vectors, and they don’t behave like vectors

Normal “vectors”

- A triangle normal is created by taking the cross product between two tangent vectors
- A normal is really a bivector, and it transforms as such

Normal "vector" transformation



Classical derivation

- Standard proof for inverse transpose for transforming normals:
  - Preserve zero dot product with tangent
  - Misses extra factor of det **M**

$$\mathbf{N} \cdot \mathbf{T} = 0$$

$$\mathbf{UN} \cdot \mathbf{MT} = 0$$

$$\mathbf{N}^T \mathbf{U}^T \mathbf{MT} = 0$$

$$\mathbf{U}^T = \mathbf{M}^{-1}$$

$$\mathbf{U} = \mathbf{M}^{-T}$$

Higher dimensions

- In n dimensions, the $(n - 1)$-vectors have n components, just as 1-vectors do
- Each 1-vector basis element uses exactly one of the spatial directions $\mathbf{e}_1 \ldots \mathbf{e}_n$
- Each $(n - 1)$-vector basis element uses all *except* one of the spatial directions $\mathbf{e}_1 \ldots \mathbf{e}_n$

Symmetry in three dimensions

- Vector basis and bivector (n − 1) basis

$\mathbf{e}_1$	$\mathbf{e}_2 \wedge \mathbf{e}_3$
$\mathbf{e}_2$	$\mathbf{e}_3 \wedge \mathbf{e}_1$
$\mathbf{e}_3$	$\mathbf{e}_1 \wedge \mathbf{e}_2$

Symmetry in four dimensions

- Vector basis and trivector (n − 1) basis

$$\mathbf{e}_1 \qquad \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$$
$$\mathbf{e}_2 \qquad \mathbf{e}_1 \wedge \mathbf{e}_4 \wedge \mathbf{e}_3$$
$$\mathbf{e}_3 \qquad \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4$$
$$\mathbf{e}_4 \qquad \mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_2$$

Dual basis

- Use special notation for wedge product of all but one basis vector:

$$\overline{\mathbf{e}}_1 = \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$$

$$\overline{\mathbf{e}}_2 = \mathbf{e}_1 \wedge \mathbf{e}_4 \wedge \mathbf{e}_3$$

$$\overline{\mathbf{e}}_3 = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_4$$

$$\overline{\mathbf{e}}_4 = \mathbf{e}_1 \wedge \mathbf{e}_3 \wedge \mathbf{e}_2$$

Dual basis

- Order of wedged basis vectors chosen so that

$$\mathbf{e}_i \overline{\mathbf{e}}_i = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4$$

Dual basis

- Instead of saying ($n - 1$)-vector, we call these “antivectors”
- In n dimensions, antivector always means a quantity expressed on the basis elements having grade $n - 1$

Vector / antivector product

- Wedge product between vector and antivector is the origin of the dot product:

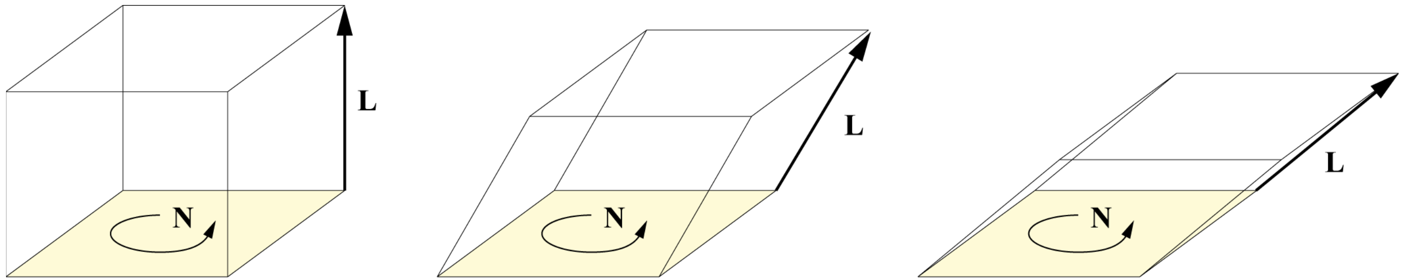
$$(a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3) \wedge (b_1\overline{\mathbf{e}}_1 + b_2\overline{\mathbf{e}}_2 + b_3\overline{\mathbf{e}}_3)$$
$$= (a_1b_1 + a_2b_2 + a_3b_3)(\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3)$$

- They complement each other, and “fill in” the volume element

Vector / antivector product

- Many of the dot products you take are actually vector / antivector wedge products
- For instance, **N** • **L** in diffuse lighting
- **N** is an antivector
- Calculating volume of extruded bivector

Diffuse lighting

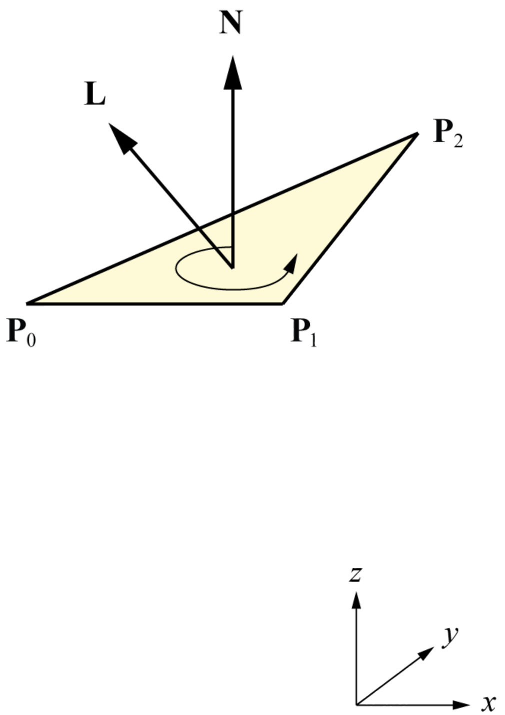


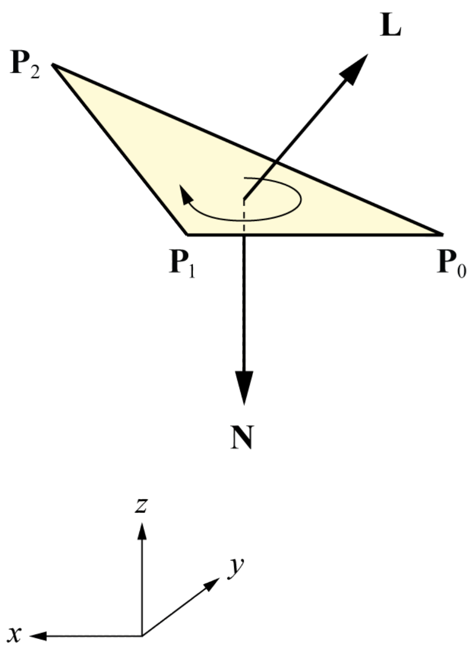
Diffuse lighting

- **N • L** is really the antiscalar produced by $\mathbf{N} \wedge \mathbf{L}$
- **N** transforms with $(\det \mathbf{M})\mathbf{M}^{-T}$
- **N • L** transforms as

$$
\begin{aligned}
&(\det \mathbf{M})\mathbf{M}^{-T}\mathbf{N} \cdot \mathbf{ML} \\
&= \mathbf{N}^{T}(\det \mathbf{M})\mathbf{M}^{-1}\mathbf{ML} \\
&= (\det \mathbf{M})\mathbf{N} \cdot \mathbf{L}
\end{aligned}
$$

Diffuse lighting





The regressive product

- Grassmann realized there is another product symmetric to the wedge product
- Not well-known at all
  - Most books on geometric algebra leave it out completely
- Very important product, though!

The regressive product

- Operates on antivectors in a manner symmetric to how the wedge product operates on vectors

- Uses an upside-down wedge:

$$\overline{\mathbf{e}}_1 \vee \overline{\mathbf{e}}_2$$

- We call it the “antiwedge” product

The antiwedge product

- Has same properties as wedge product, but for antivectors
- Operates in complementary space on dual basis or “antibasis”

The antiwedge product

- Whereas the wedge product increases grade, the antiwedge product decreases it
- Suppose, in *n*-dimensional Grassmann algebra, **A** has grade *r* and **B** has grade *s*
- Then $\mathbf{A} \wedge \mathbf{B}$ has grade $r + s$
- And $\mathbf{A} \vee \mathbf{B}$ has grade

$$n - (n - r) - (n - s) = r + s - n$$

Antiwedge product in 3D

$$\overline{\mathbf{e}}_1 \vee \overline{\mathbf{e}}_2 = (\mathbf{e}_2 \wedge \mathbf{e}_3) \vee (\mathbf{e}_3 \wedge \mathbf{e}_1) = \mathbf{e}_3$$

$$\overline{\mathbf{e}}_2 \vee \overline{\mathbf{e}}_3 = (\mathbf{e}_3 \wedge \mathbf{e}_1) \vee (\mathbf{e}_1 \wedge \mathbf{e}_2) = \mathbf{e}_1$$

$$\overline{\mathbf{e}}_3 \vee \overline{\mathbf{e}}_1 = (\mathbf{e}_1 \wedge \mathbf{e}_2) \vee (\mathbf{e}_2 \wedge \mathbf{e}_3) = \mathbf{e}_2$$

Similar shorthand notation

$$\overline{\mathbf{e}}_{12} = \overline{\mathbf{e}}_1 \vee \overline{\mathbf{e}}_2$$

$$\overline{\mathbf{e}}_{23} = \overline{\mathbf{e}}_2 \vee \overline{\mathbf{e}}_3$$

$$\overline{\mathbf{e}}_{31} = \overline{\mathbf{e}}_3 \vee \overline{\mathbf{e}}_1$$

$$\overline{\mathbf{e}}_{123} = \overline{\mathbf{e}}_1 \vee \overline{\mathbf{e}}_2 \vee \overline{\mathbf{e}}_3$$

Join and meet

- Wedge product joins *vectors* together
  - Analogous to union
- Antiwedge product joins *antivectors*
  - Antivectors represent absence of geometry
  - Joining antivectors is like removing vectors
  - Analogous to intersection
  - Called a meet operation

Homogeneous coordinates

- Points have a 4D representation:

$$\mathbf{P} = (x, y, z, w)$$

- Conveniently allows affine transformation through 4 x 4 matrix

- Used throughout 3D graphics

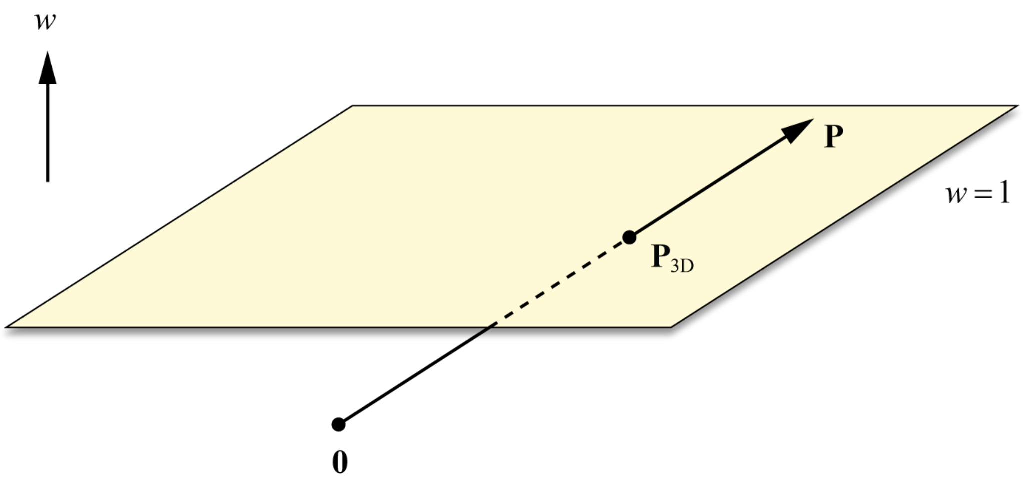
Homogeneous points

- To project onto 3D space, find where 4D vector intersects subspace where *w* = 1

$$\mathbf{P} = (x, y, z, w)$$

$$\mathbf{P}_{\mathrm{3D}} = \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)$$

Homogeneous points



Homogeneous model

- With Grassmann algebra, homogeneous model can be extended to include 3D points, lines, and planes
- Wedge and antiwedge products naturally perform union and intersection operations among all of these

4D Grassmann algebra

- Scalar unit
- Four vectors: $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$
- Six bivectors: $\mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{41}, \mathbf{e}_{42}, \mathbf{e}_{43}$
- Four antivectors: $\overline{\mathbf{e}}_1, \overline{\mathbf{e}}_2, \overline{\mathbf{e}}_3, \overline{\mathbf{e}}_4$
- Antiscalar unit (quadvector)

Homogeneous lines

- Take wedge product of two 4D points

$$\mathbf{P} = \left(P_x, P_y, P_z, 1\right) = P_x\mathbf{e}_1 + P_y\mathbf{e}_2 + P_z\mathbf{e}_3 + \mathbf{e}_4$$

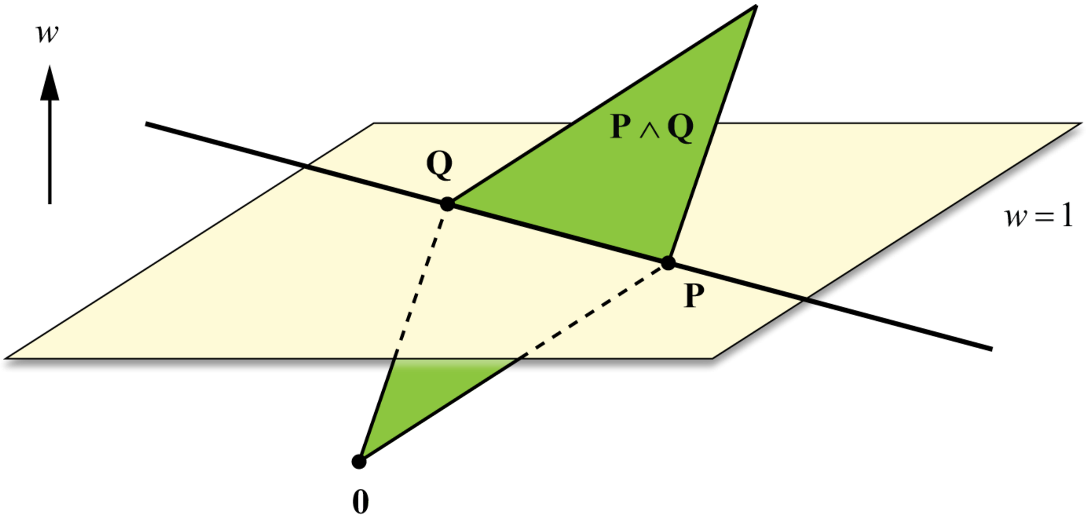
$$\mathbf{Q} = \left(Q_x, Q_y, Q_z, 1\right) = Q_x\mathbf{e}_1 + Q_y\mathbf{e}_2 + Q_z\mathbf{e}_3 + \mathbf{e}_4$$

Homogeneous Lines

$$\mathbf{P} \wedge \mathbf{Q} = (Q_x - P_x)\mathbf{e}_{41} + (Q_y - P_y)\mathbf{e}_{42} + (Q_z - P_z)\mathbf{e}_{43}$$
$$+ (P_y Q_z - P_z Q_y)\mathbf{e}_{23} + (P_z Q_x - P_x Q_z)\mathbf{e}_{31} + (P_x Q_y - P_y Q_x)\mathbf{e}_{12}$$

- This bivector spans a 2D plane in 4D
- In subspace where *w* = 1, this is a 3D line

Homogeneous lines



Homogeneous lines

- The 4D bivector can be decomposed into two 3D components:
  - A tangent vector and a moment bivector
  - These are perpendicular

$$\mathbf{P} \wedge \mathbf{Q} = (Q_x - P_x)\mathbf{e}_{41} + (Q_y - P_y)\mathbf{e}_{42} + (Q_z - P_z)\mathbf{e}_{43}$$
$$+ (P_y Q_z - P_z Q_y)\mathbf{e}_{23} + (P_z Q_x - P_x Q_z)\mathbf{e}_{31} + (P_x Q_y - P_y Q_x)\mathbf{e}_{12}$$

Homogeneous lines

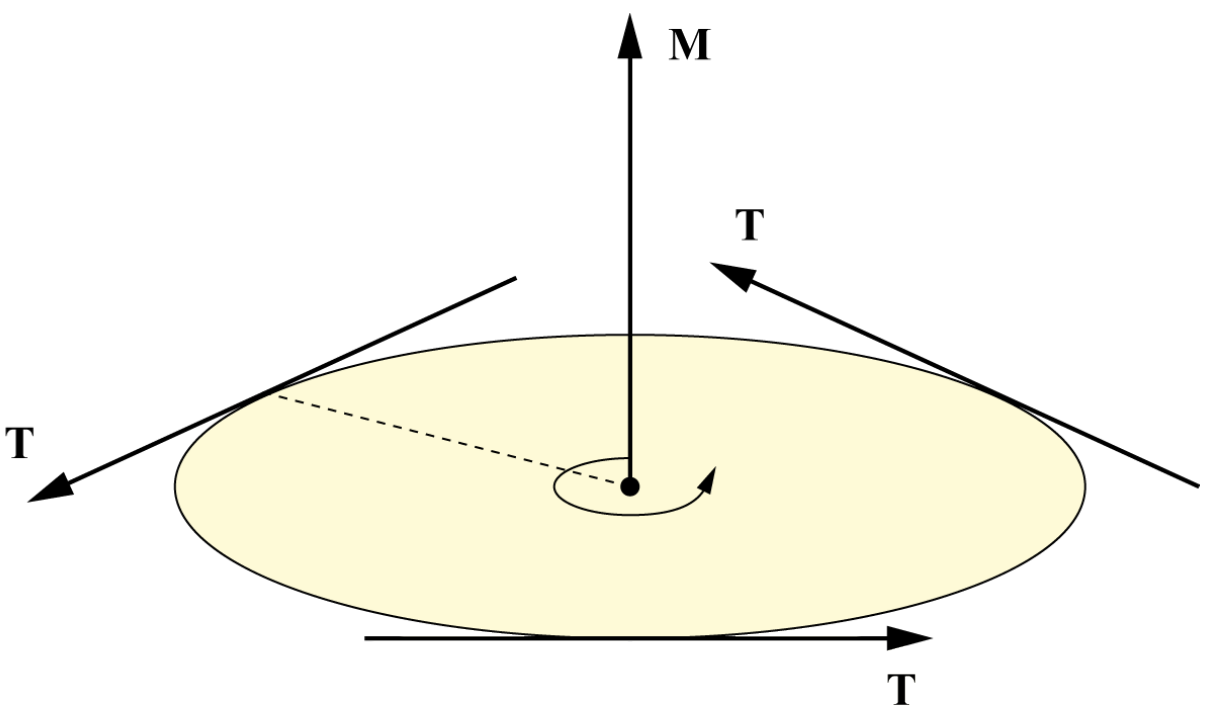
- The 4D bivector no longer contains any information about the two points used to create it
- Contrary to parametric origin / direction representation

Homogeneous lines

- Tangent **T** vector is $\mathbf{Q}_{3\mathrm{D}} - \mathbf{P}_{3\mathrm{D}}$
- Moment **M** bivector is $\mathbf{P}_{3\mathrm{D}} \wedge \mathbf{Q}_{3\mathrm{D}}$

$$\mathbf{P} \wedge \mathbf{Q} = (Q_x - P_x)\mathbf{e}_{41} + (Q_y - P_y)\mathbf{e}_{42} + (Q_z - P_z)\mathbf{e}_{43}$$
$$+ (P_y Q_z - P_z Q_y)\mathbf{e}_{23} + (P_z Q_x - P_x Q_z)\mathbf{e}_{31} + (P_x Q_y - P_y Q_x)\mathbf{e}_{12}$$

Moment bivector



Plücker coordinates

- Origin of Plücker coordinates revealed!
- They are the coefficients of a 4D bivector

- A line **L** in Plücker coordinates is

$$\mathbf{L} = \{\mathbf{Q} - \mathbf{P} : \mathbf{P} \times \mathbf{Q}\}$$

- A bunch of seemingly arbitrary formulas in Plücker coordinates demystified

Homogeneous planes

- Take wedge product of three 4D points

$$\mathbf{P} = (P_x, P_y, P_z, 1) = P_x\mathbf{e}_1 + P_y\mathbf{e}_2 + P_z\mathbf{e}_3 + \mathbf{e}_4$$

$$\mathbf{Q} = (Q_x, Q_y, Q_z, 1) = Q_x\mathbf{e}_1 + Q_y\mathbf{e}_2 + Q_z\mathbf{e}_3 + \mathbf{e}_4$$

$$\mathbf{R} = (R_x, R_y, R_z, 1) = R_x\mathbf{e}_1 + R_y\mathbf{e}_2 + R_z\mathbf{e}_3 + \mathbf{e}_4$$

Homogeneous planes

$$\mathbf{P} \wedge \mathbf{Q} \wedge \mathbf{R} = N_x \overline{\mathbf{e}}_1 + N_y \overline{\mathbf{e}}_2 + N_z \overline{\mathbf{e}}_3 + D \overline{\mathbf{e}}_4$$

- **N** is the 3D normal bivector
- *D* is the offset from origin in units of **N**

$$\mathbf{N} = \mathbf{P}_{3D} \wedge \mathbf{Q}_{3D} + \mathbf{Q}_{3D} \wedge \mathbf{R}_{3D} + \mathbf{R}_{3D} \wedge \mathbf{P}_{3D}$$

$$D = -\mathbf{P}_{3D} \wedge \mathbf{Q}_{3D} \wedge \mathbf{R}_{3D}$$

Plane transformation

- A homogeneous plane is a 4D antivector
- It transforms by the inverse of a 4 x 4 matrix
  - Just like a 3D antivector transforms by the inverse of a 3 x 3 matrix
  - Orthogonality not common here due to translation in the matrix

Projective geometry

4D Entity	3D Geometry
Vector (1-space)	Point (0-space)
Bivector (2-space)	Line (1-space)
Trivector (3-space)	Plane (2-space)

- We always project onto the 3D subspace where $w = 1$

Geometric computation in 4D

- Wedge product
  - Multiply **two points** to get the line containing both points
  - Multiply **three points** to get the plane containing all three points
  - Multiply **a line and a point** to get the plane containing the line and the point

Geometric computation in 4D

- Antiwedge product
  - Multiply **two planes** to get the line where they intersect
  - Multiply **three planes** to get the point common to all three planes
  - Multiply **a line and a plane** to get the point where the line intersects the plane

Geometric computation in 4D

- Wedge or antiwedge product
  - Multiply **a point and a plane** to get the signed minimum distance between them in units of the normal magnitude
  - Multiply **two lines** to get a special signed crossing value

Product of two lines

- Wedge product gives an antiscalar (quadvector or 4D volume element)
- Antiwedge product gives a scalar
- Both have same sign and magnitude
- Grassmann treated scalars and antiscalars as the same thing

Product of two lines

- Let $\mathbf{L}_1$ have tangent $\mathbf{T}_1$ and moment $\mathbf{M}_1$
- Let $\mathbf{L}_2$ have tangent $\mathbf{T}_2$ and moment $\mathbf{M}_2$

- Then,

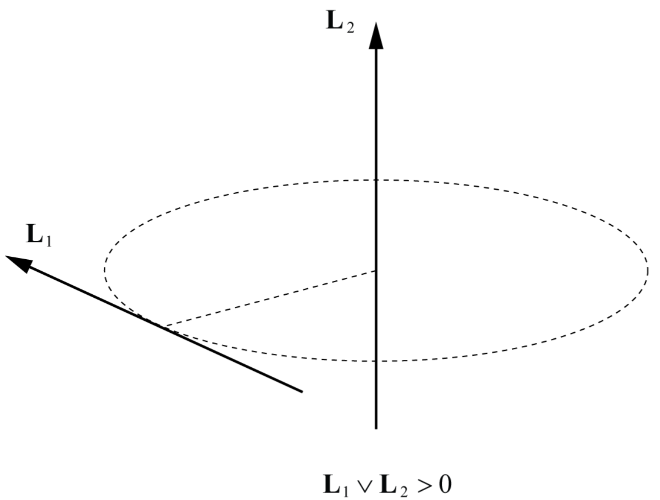
$$\mathbf{L}_1 \wedge \mathbf{L}_2 = -(\mathbf{T}_1 \wedge \mathbf{M}_2 + \mathbf{T}_2 \wedge \mathbf{M}_1)$$

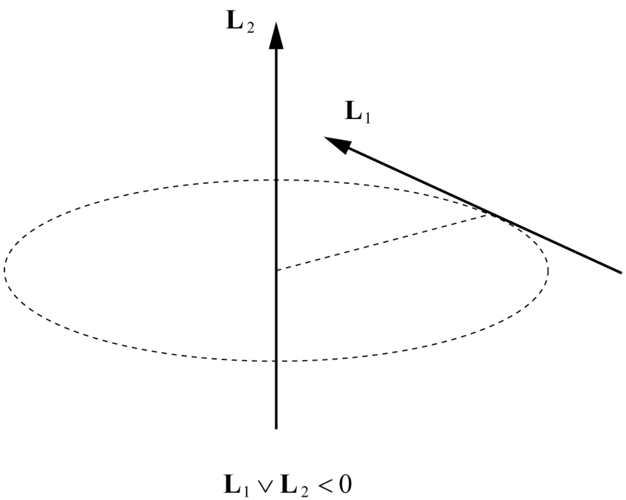
$$\mathbf{L}_1 \vee \mathbf{L}_2 = -(\mathbf{T}_1 \vee \mathbf{M}_2 + \mathbf{T}_2 \vee \mathbf{M}_1)$$

Product of two lines

- The product of two lines gives a "crossing" relation
  - Positive value means clockwise crossing
  - Negative value means counterclockwise
  - Zero if lines intersect

Crossing relation





Distance between lines

- Product of two lines also relates to signed minimum distance between them

$$d = \frac{\mathbf{L}_1 \wedge \mathbf{L}_2}{\left\| \mathbf{T}_1 \wedge \mathbf{T}_2 \right\|}$$

- (Here, numerator is 4D wedge product, and denominator is 3D wedge product)

Ray-triangle intersection

- Application of line-line product
- Classic barycentric calculation difficult due to floating-point round-off error
  - Along edge between two triangles, ray can miss both or hit both
  - Typical solution involves use of ugly epsilons

Ray-triangle intersection

- Calculate 4D bivectors for triangle edges and ray
  - Take wedge products between ray and three edges
  - Same sign for all three edges is a hit
  - Impossible to hit or miss both triangles sharing edge unless exact intersection
  - Need to handle zero in consistent way

Weighting

- Points, lines, and planes have “weights” in homogeneous coordinates

Entity	Weight
Point	*w* coordinate
Line	Tangent component **T**
Plane	*x*, *y*, *z* component

Weighting

- Mathematically, the weight components can be found by taking the antiwedge product with the antivector (0,0,0,1)
- We would never really do that, though, because we can just look at the right coefficients

Normalized lines

- Tangent component has unit length
- Magnitude of moment component is perpendicular distance to the origin

Normalized planes

- (x,y,z) component has unit length
- Wedge product with (normalized) point is perpendicular distance to plane

Programming considerations

- Convenient to create classes to represent entities of each grade
  - Vector4D
  - Bivector4D
  - Antivector4D

Programming considerations

- Fortunate happenstance that C++ has an overloadable operator ^ that looks like a wedge
- But be careful with operator precedence if you overload ^ to perform wedge product
  - Has lowest operator precedence, so get used to enclosing wedge products in parentheses
- E.g., x ^ y > 0 compiles as x ^ (y > 0)

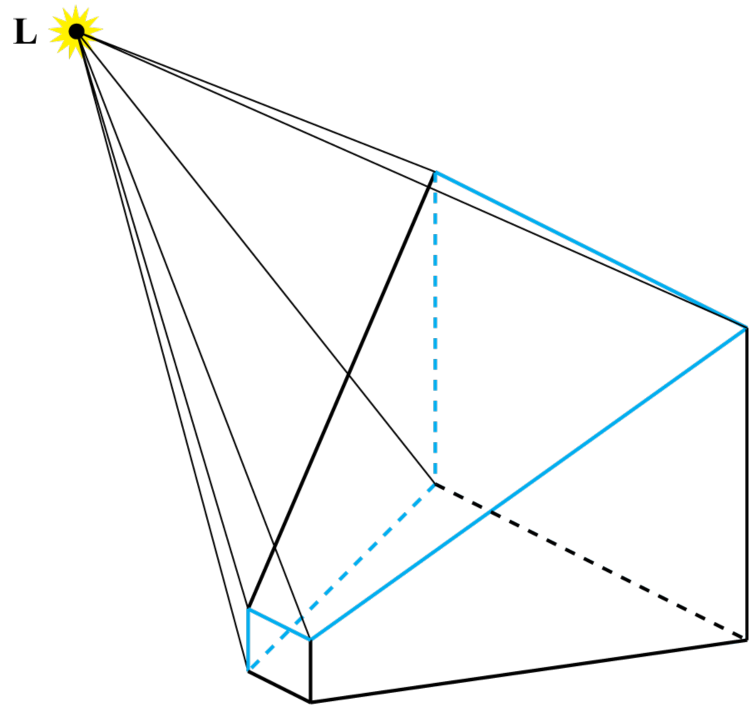
Combining wedge and antiwedge

- The same operator can be used for wedge product and antiwedge product
- Either they both produce the same scalar and antiscalar magnitudes with the same sign
- Or one of the products is identically zero
- For example, you would always want the antiwedge product for two planes because the wedge product is zero for all inputs

Example application

- Calculation of shadow region planes from light position and frustum edges

- Simply a wedge product between edge line and light position **L**



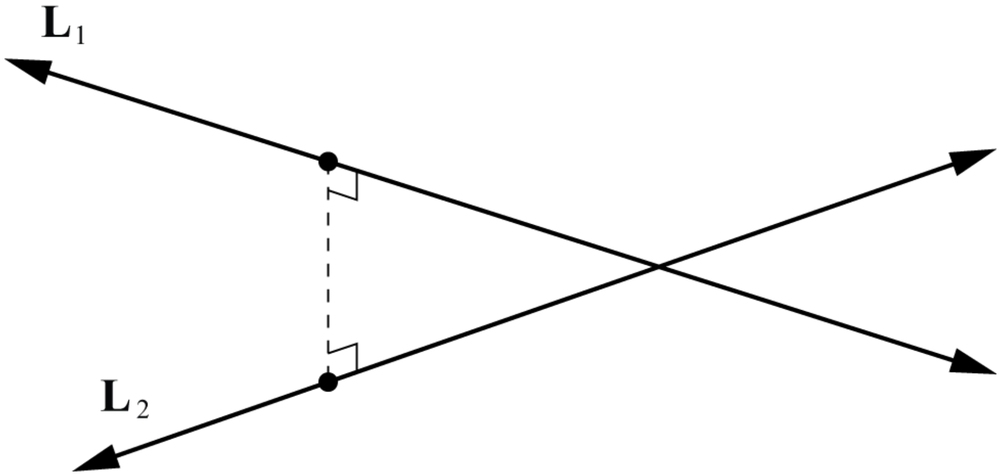
Summary

Old school	New school
Cross product → axial vector	Wedge product → bivector
Dot product	Antiwedge vector / antivector
Scalar triple product	Triple wedge product
Plücker coordinates	4D bivectors
Operations in Plücker coordinates	4D wedge / antiwedge products
Transform normals with inverse transpose	Transform antivectors with adjugate transpose

Supplemental Slides

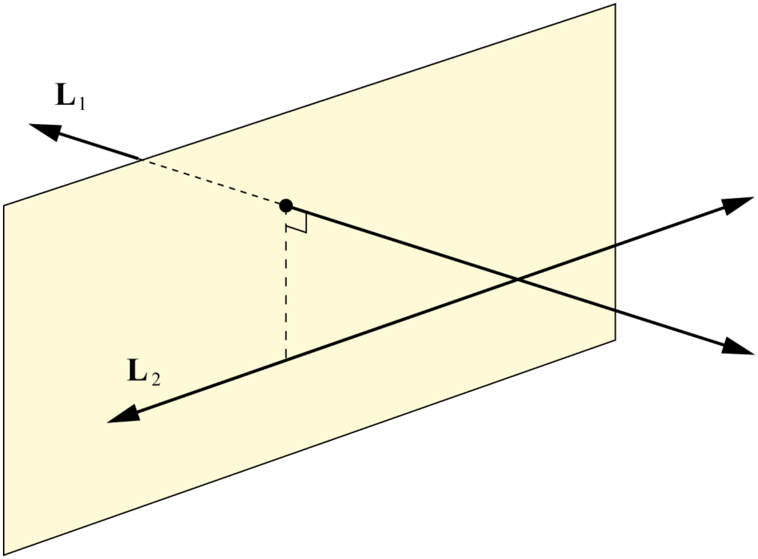
Points of closest approach

- Wedge product of line tangents gives complement of direction between closest points



Points of closest approach

- Plane containing this direction and first line also contains closest point on second line



Two dimensions

- 1 scalar unit
- 2 basis vectors
- 1 bivector / antiscalar unit
- No cross product
- All rotations occur in plane of 1 bivector

One dimension

- 1 scalar unit
- 1 single-component basis vector
  - Also the antiscalar unit
- Equivalent to “dual numbers”
- All numbers have form $a + b\mathbf{e}$
  - Where $e^2 = 0$

Matrix inverses

- The *i*-th row of the inverse of **M** is 1/(det **M**) times the wedge product of all columns of **M** except column *i*.

Explicit formulas

- Define points **P**, **Q** and planes **E**, **F**, and line **L**

$$\mathbf{P} = (P_x, P_y, P_z, 1) = P_x\mathbf{e}_1 + P_y\mathbf{e}_2 + P_z\mathbf{e}_3 + \mathbf{e}_4$$

$$\mathbf{Q} = (Q_x, Q_y, Q_z, 1) = Q_x\mathbf{e}_1 + Q_y\mathbf{e}_2 + Q_z\mathbf{e}_3 + \mathbf{e}_4$$

$$\mathbf{E} = (E_x, E_y, E_z, E_w) = E_x\overline{\mathbf{e}}_1 + E_y\overline{\mathbf{e}}_2 + E_z\overline{\mathbf{e}}_3 + E_w\overline{\mathbf{e}}_4$$

$$\mathbf{F} = (F_x, F_y, F_z, F_w) = F_x\overline{\mathbf{e}}_1 + F_y\overline{\mathbf{e}}_2 + F_z\overline{\mathbf{e}}_3 + F_w\overline{\mathbf{e}}_4$$

$$\mathbf{L} = T_x\mathbf{e}_{41} + T_y\mathbf{e}_{42} + T_z\mathbf{e}_{43} + M_x\mathbf{e}_{23} + M_y\mathbf{e}_{31} + M_z\mathbf{e}_{12}$$

Explicit formulas

- Product of two points

$$\mathbf{P} \wedge \mathbf{Q} = (Q_x - P_x)\mathbf{e}_{41} + (Q_y - P_y)\mathbf{e}_{42} + (Q_z - P_z)\mathbf{e}_{43}$$
$$+ (P_y Q_z - P_z Q_y)\mathbf{e}_{23} + (P_z Q_x - P_x Q_z)\mathbf{e}_{31} + (P_x Q_y - P_y Q_x)\mathbf{e}_{12}$$

Explicit formulas

- Product of two planes

$$\mathbf{E} \vee \mathbf{F} = (E_z F_y - E_y F_z)\mathbf{e}_{41} + (E_x F_z - E_z F_x)\mathbf{e}_{42} + (E_y F_x - E_x F_y)\mathbf{e}_{43}$$
$$+ (E_x F_w - E_w F_x)\mathbf{e}_{23} + (E_y F_w - E_w F_y)\mathbf{e}_{31} + (E_z F_w - E_w F_z)\mathbf{e}_{12}$$

Explicit formulas

- Product of line and point

$$\mathbf{L} \wedge \mathbf{P} = \left(T_y P_z - T_z P_y + M_x\right)\overline{\mathbf{e}}_1 + \left(T_z P_x - T_x P_z + M_y\right)\overline{\mathbf{e}}_2 + \left(T_x P_y - T_y P_x + M_z\right)\overline{\mathbf{e}}_3 + \left(-P_x M_x - P_y M_y - P_z M_z\right)\overline{\mathbf{e}}_4$$

Explicit formulas

- Product of line and plane

$$\mathbf{L} \vee \mathbf{E} = \left(M_z E_y - M_y E_z - T_x E_w \right)\mathbf{e}_1 + \left(M_x E_z - M_z E_x - T_y E_w \right)\mathbf{e}_2 + \left(M_y E_x - M_x E_y - T_z E_w \right)\mathbf{e}_3 + \left(E_x T_x + E_y T_y + E_z T_z \right)\mathbf{e}_4$$