

Linear Algebra for Graphics

Lecturer:

Carol O'Sullivan

Professor of Visual Computing

Carol.OSullivan@cs.tcd.ie

Course www:

<http://isg.cs.tcd.ie/cosulliv/>

Linear Algebra

- Linear algebra is the cornerstone of computer graphics.
- Fundamentally, we need to be able to manipulate *points* and *vectors*.
 - these form the basis of all geometric objects & operations
- Geometric operations (*scale, rotate, translate, perspective projection*) are defined using matrix transformations.
- Optical effects (*reflect, refract*) defined using vector algebra.

Conventions

- Vector quantities denoted as \mathbf{v} or \vec{v}
- We will use *column format* vectors:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \neq [v_1 \quad v_2 \quad v_3] \quad \left(= [v_1 \quad v_2 \quad v_3]^T \right)$$

- Each vector is defined with respect to a set of *basis vectors* (which define a co-ordinate system).

Row vs. Column Formats

- Both formats, though appearing equivalent, are in fact fundamentally different:
 - be wary of different formats used in textbooks

column format

row format

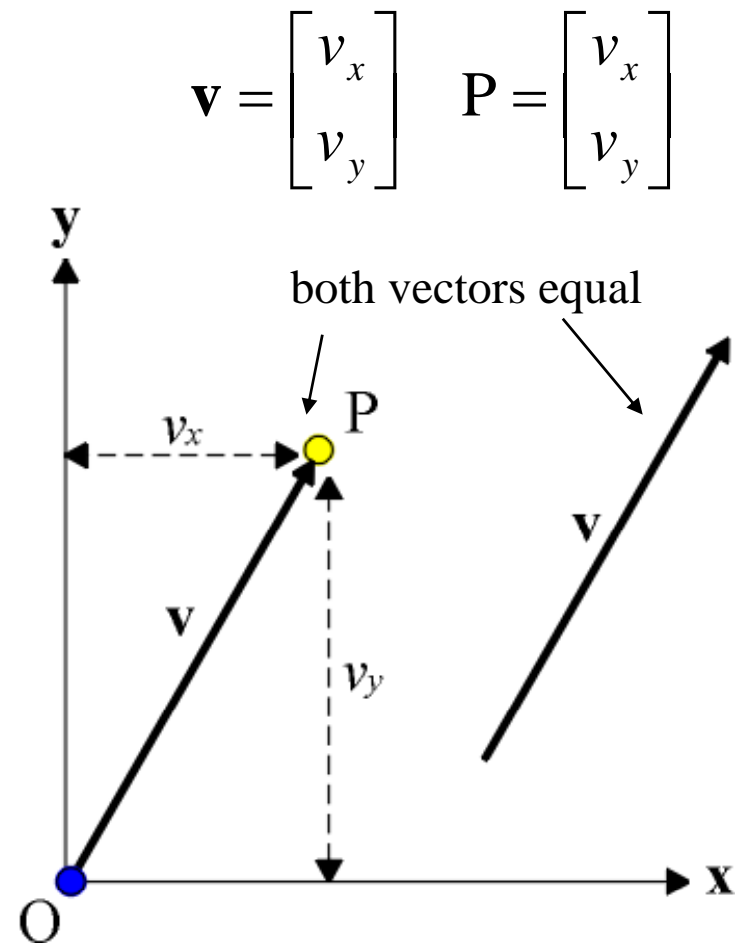
$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u & v & w \end{bmatrix} \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

$\mathbf{M}\mathbf{v} = \mathbf{v}^T\mathbf{M}^T$

transposed

Vectors & Points

- Although *vectors* and *points* are often used inter-changeably in graphics texts, it is important to distinguish between them.
 - vectors represent directions
 - points represent positions
- Both are meaningless without reference to a *coordinate system*
 - vectors require a set of *basis vectors*
 - points require an *origin* and a *vector space*

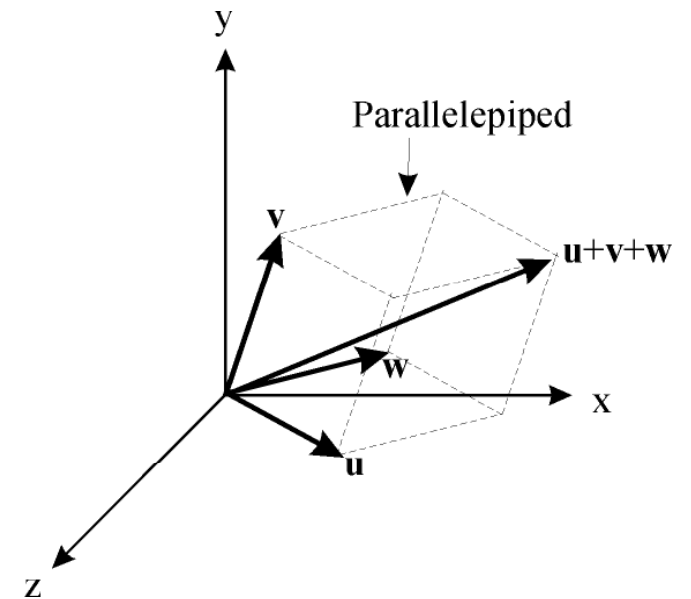
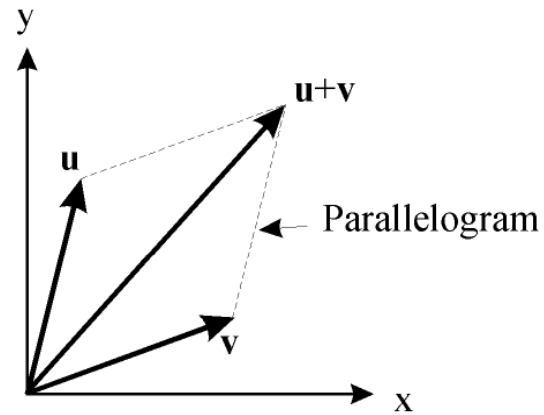


Vector Addition & Subtraction

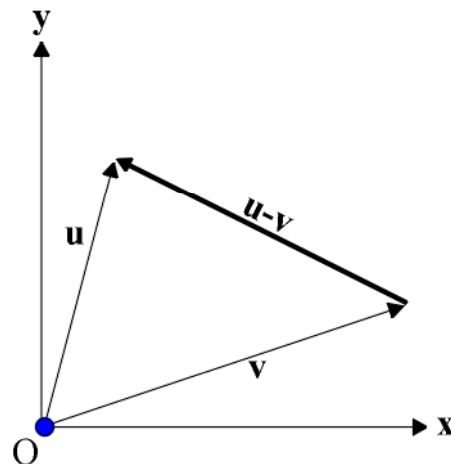
- Addition of vectors follows the *parallelogram* law in 2D and the *parallelepiped* law in higher dimensions:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix}$$

$\mathbf{u} \quad \mathbf{v} \quad \mathbf{u+v}$

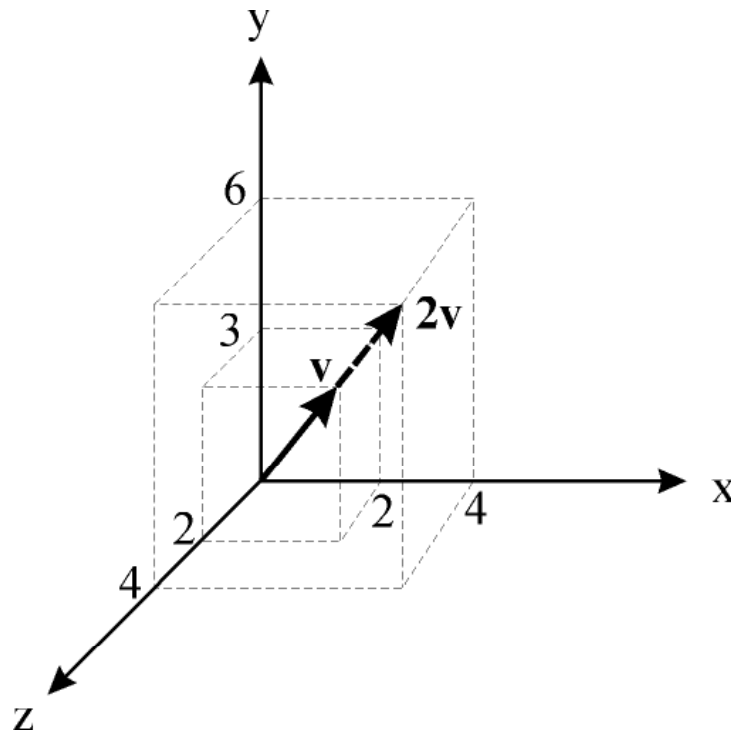


- Subtraction:



Vector Multiplication by a Scalar

- Each vector has an associated length
- Multiplication by a scalar scales the vectors length appropriately (but does not affect direction):



Linear Combinations

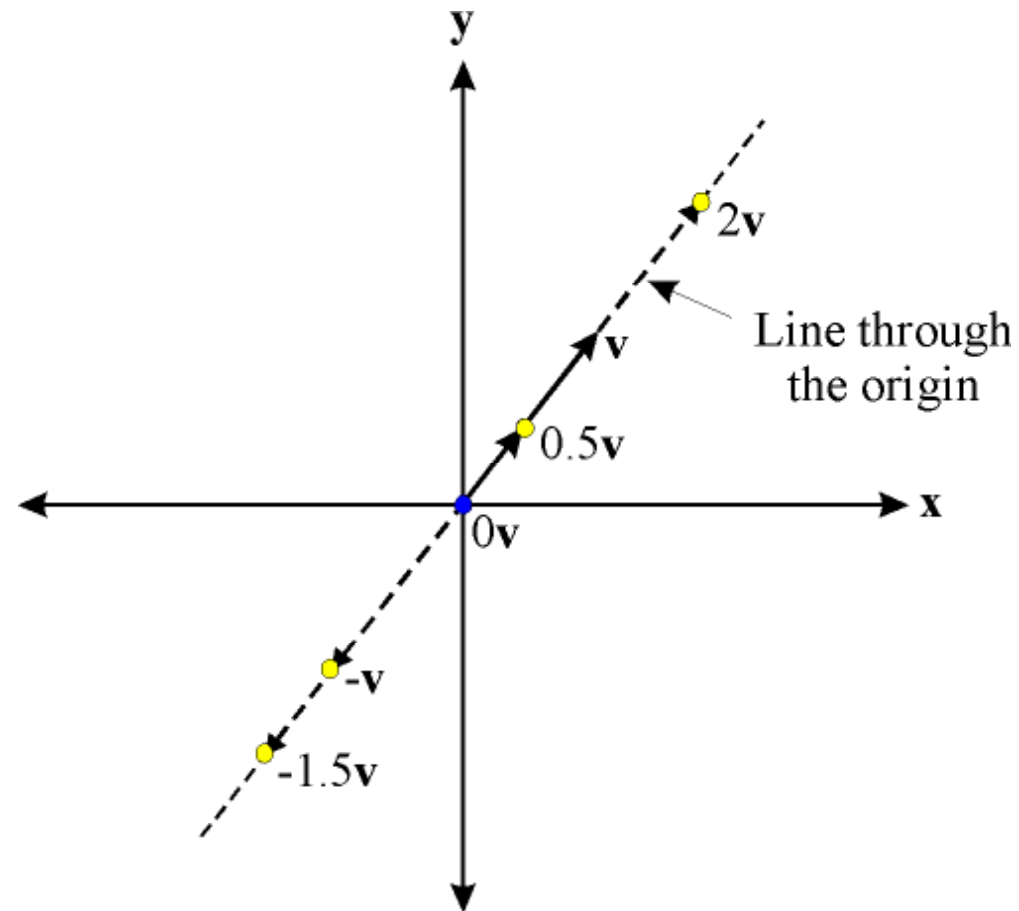
- The *linear combination* of a set of vectors is the sum of scalar multiples of those vectors:

$$\mathbf{u} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_n\mathbf{v}_n$$

- Fixing vectors \mathbf{v}_i yields an infinite number of \mathbf{u} depending on the scalars a_j .
- The set \mathbf{u} is called the *span* of the vectors \mathbf{v}_i .
- The vectors \mathbf{v}_i are termed *basis vectors* for the space.
- If none of the \mathbf{v}_i can be created as a linear combination of the others, the vectors \mathbf{v}_i are said to be *linearly independent*.
- All linear combinations contain the *zero vector*.

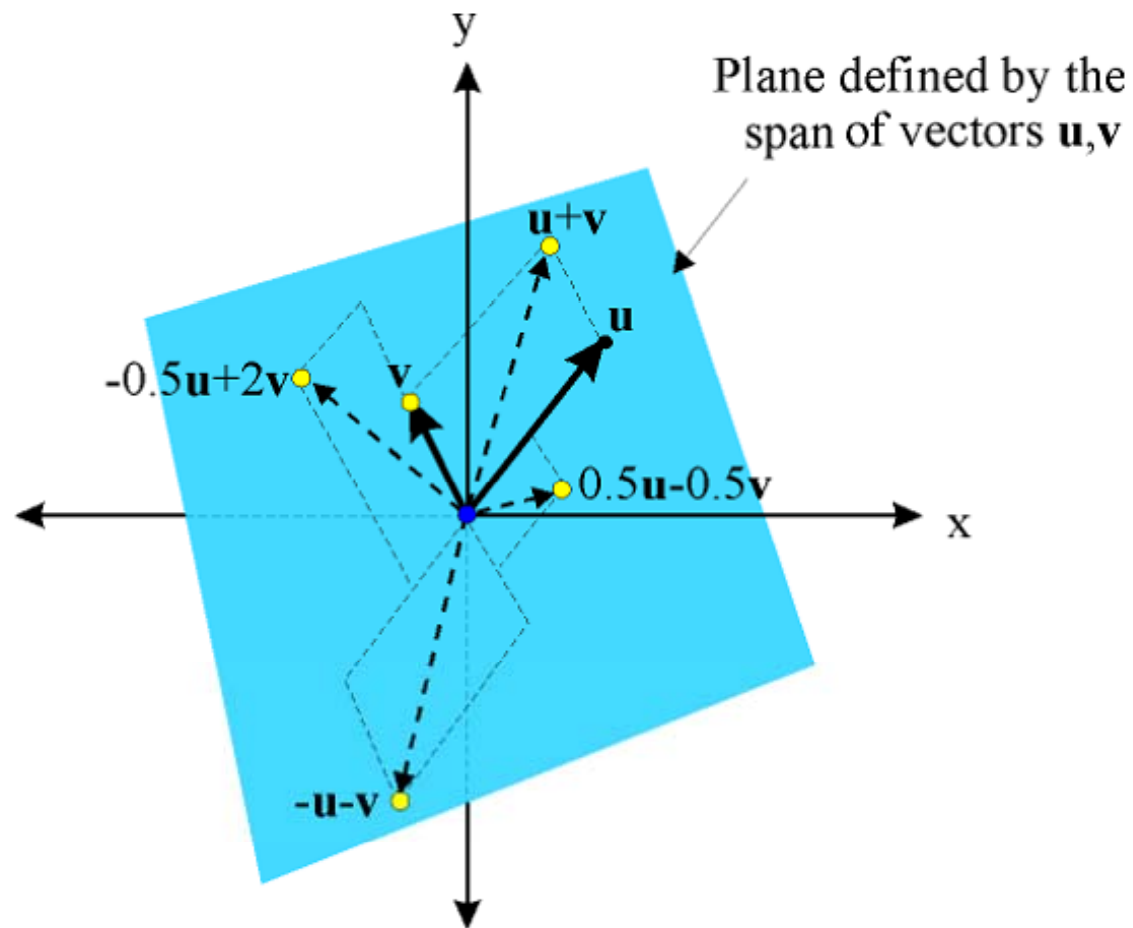
Linear Combinations

- Linear combinations of 1 vector = an *infinite line*:



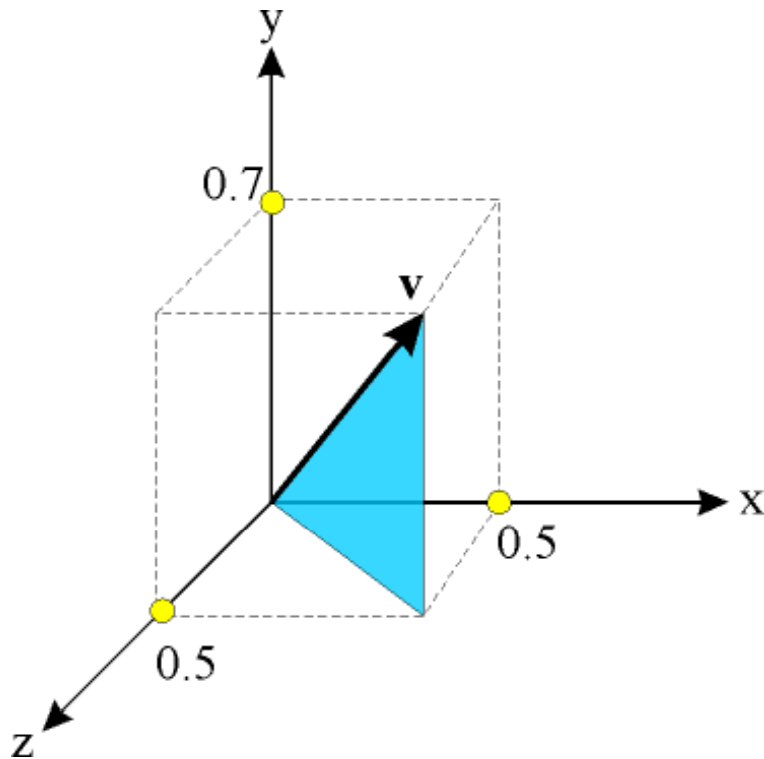
Linear Combinations

- Linear combinations of 2 vectors = a *plane*



Linear Combinations

- The linear combination of 3 vectors = a *3D volume*.
- The 3D Cartesian coordinate system employs the well-known 3D co-ordinate basis: **x**, **y** and **z**



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The vector **v** here is a *linear combination* of the basis vectors **x**, **y** and **z**:

$$\mathbf{v} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix} = 0.5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 0.7 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 0.5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Vector Magnitude

- The *magnitude* or *norm* of a vector of dimension n is given by the standard *Euclidean distance metric*:

- For example:
$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

$$\left\| \begin{array}{c} 1 \\ 3 \\ 1 \end{array} \right\| = \sqrt{1^2 + 3^2 + 1^2} = \sqrt{11}$$

- Vectors of length 1 (unit vectors) are often termed *normal* or *normalised vectors*.

Normalised Vectors

- When we wish to describe direction we use *normalised vectors*.
- We normalise a vector by dividing by its magnitude:

$$\mathbf{v}' = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{1}{\sqrt{v_1^2 + v_2^2 + \dots + v_n^2}} \mathbf{v}$$

Dot Product

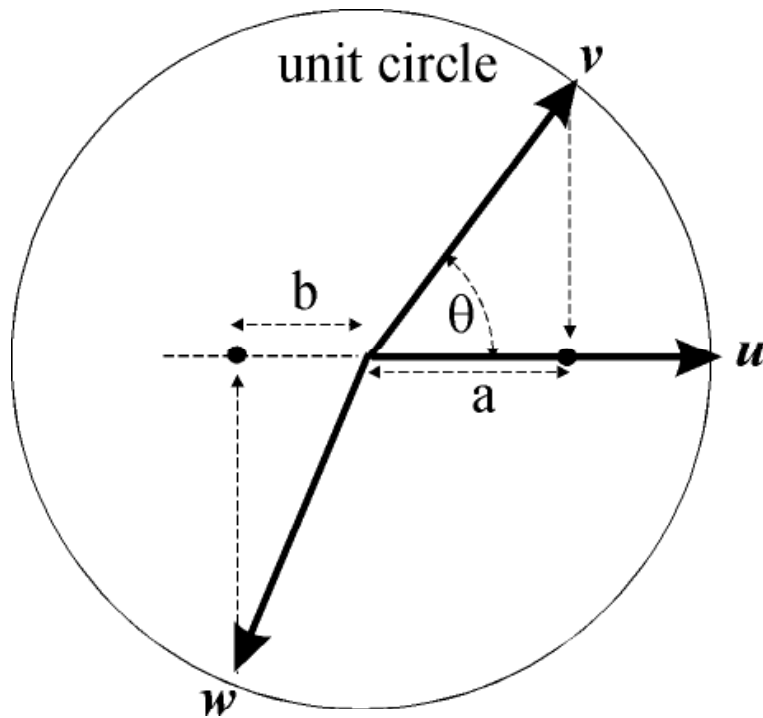
- Dot product (*inner product*) is defined as:

$$\mathbf{u} \cdot \mathbf{v} = \sum_i u_i v_i$$
$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

- Note: $\mathbf{u} \cdot \mathbf{u} = u_1^2 + u_2^2 + u_3^2 = \|\mathbf{u}\|^2$
- Therefore we can also define magnitude in terms of the dot-product operator: $\|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$
- Dot product operator is *commutative*.

Dot Product

- If both vectors are normalised, the dot product defines the cosine of the angle between the vectors:



$$\mathbf{u} \cdot \mathbf{v} = \cos \theta$$

In general:

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$$

$$\Rightarrow \theta = \cos^{-1} \left[\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right]$$

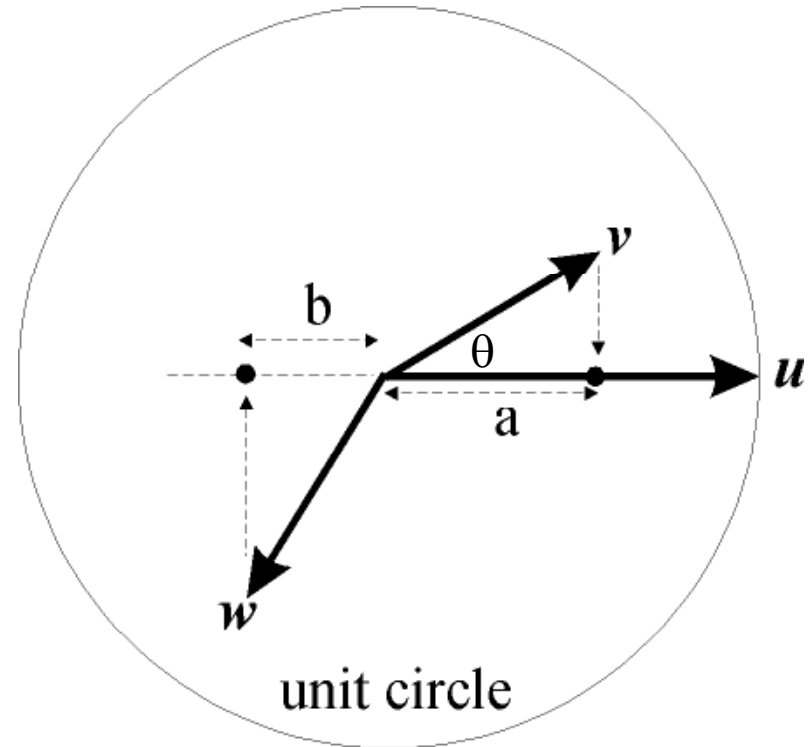
Dot Product

- If one of the vectors is normal, the dot product defines the *projection* of the other onto it (perpendicularly)
- In this example, a is positive and b is negative.
- Note that if both vectors are pointing in *same direction*, the dot-product is positive.

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$$

$$\Rightarrow a = \|\mathbf{v}\| \cos \theta$$

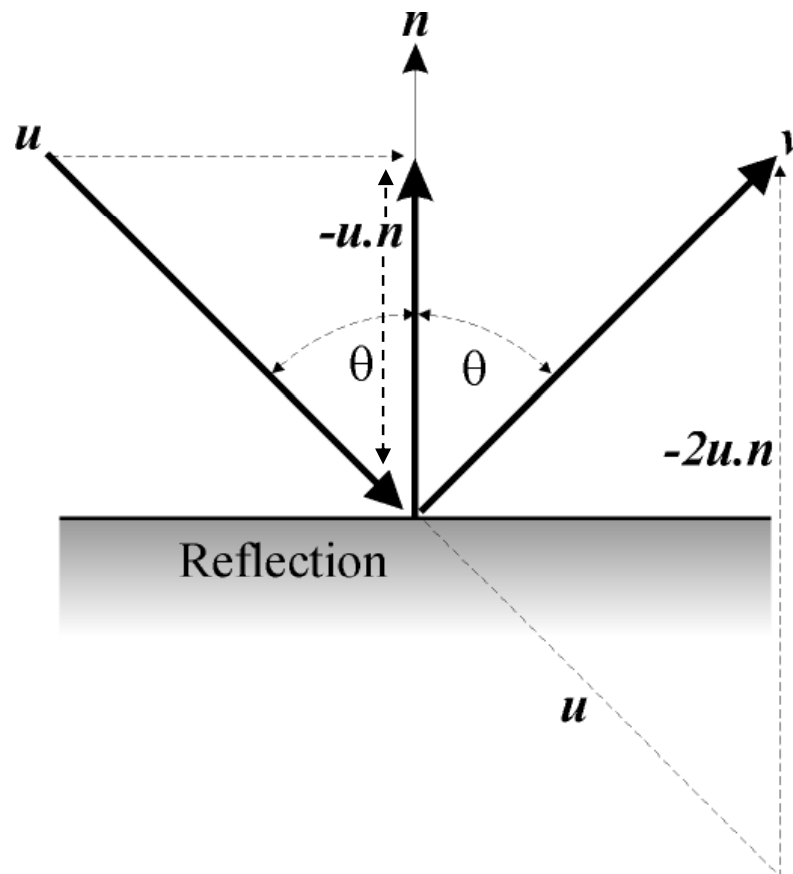
$$\therefore \cos \theta = \frac{a}{\|\mathbf{v}\|}$$



$$a = \mathbf{u} \cdot \mathbf{v} \quad b = \mathbf{u} \cdot \mathbf{w}$$

Dot Product

- Note that if $\theta = 90$ then the dot product = 0, i.e. the projection of one onto the other has zero length \Rightarrow vectors are *orthogonal*.
- Also, if $\theta > 90$ then the dot product is negative.
- Example:



$$\mathbf{v} = \mathbf{u} - 2\mathbf{n}(\mathbf{u} \cdot \mathbf{n})$$

Cross Product

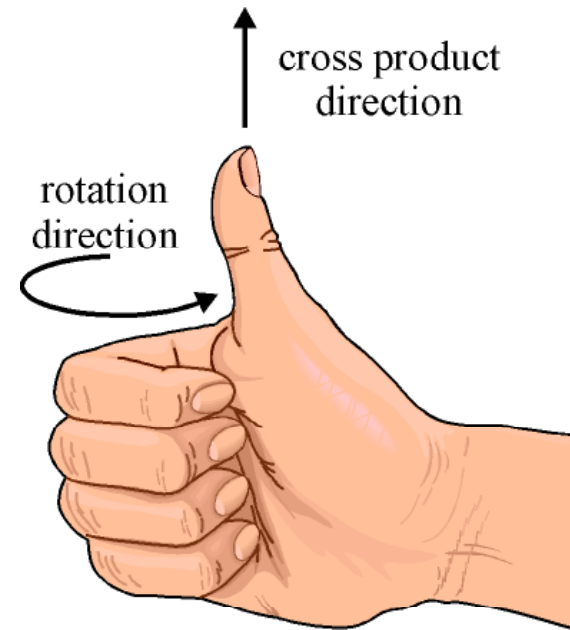
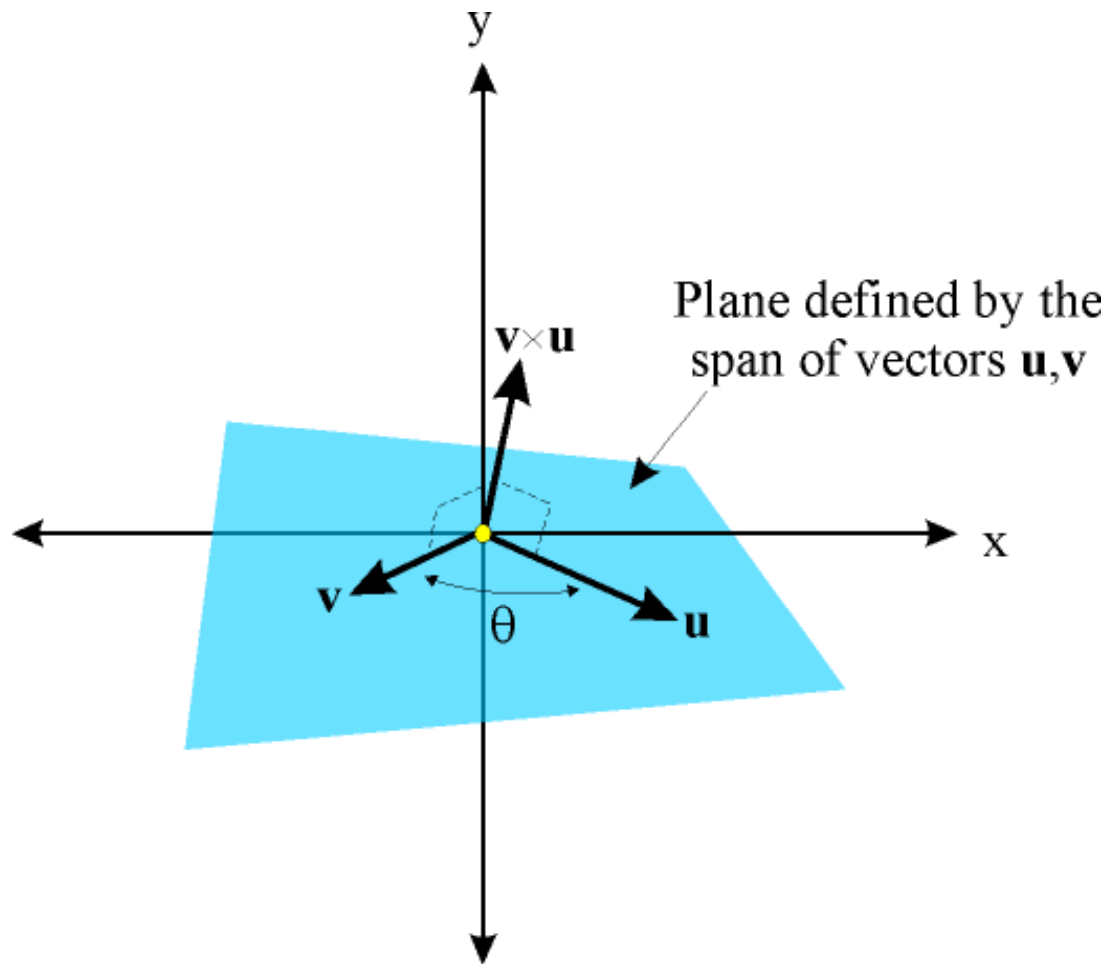
- Used for defining *orientation* and constructing *co-ordinate axes*.
- Cross product defined as:

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

- The result is a *vector*, perpendicular to the plane defined by \mathbf{u} and \mathbf{v} :

$$\mathbf{u} \times \mathbf{v} = \mathbf{w} \|\mathbf{u}\| \|\mathbf{v}\| \sin \theta$$

Cross Product



Right Handed Coordinate System

Cross Product

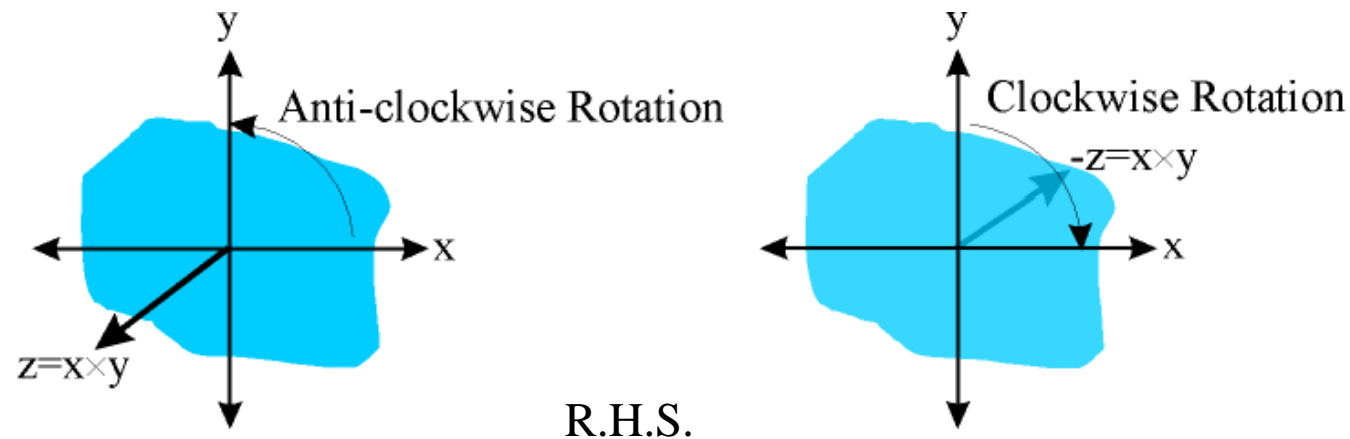
- Cross product is *anti-commutative*:

$$\mathbf{u} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{u})$$

- It is **not** *associative*:

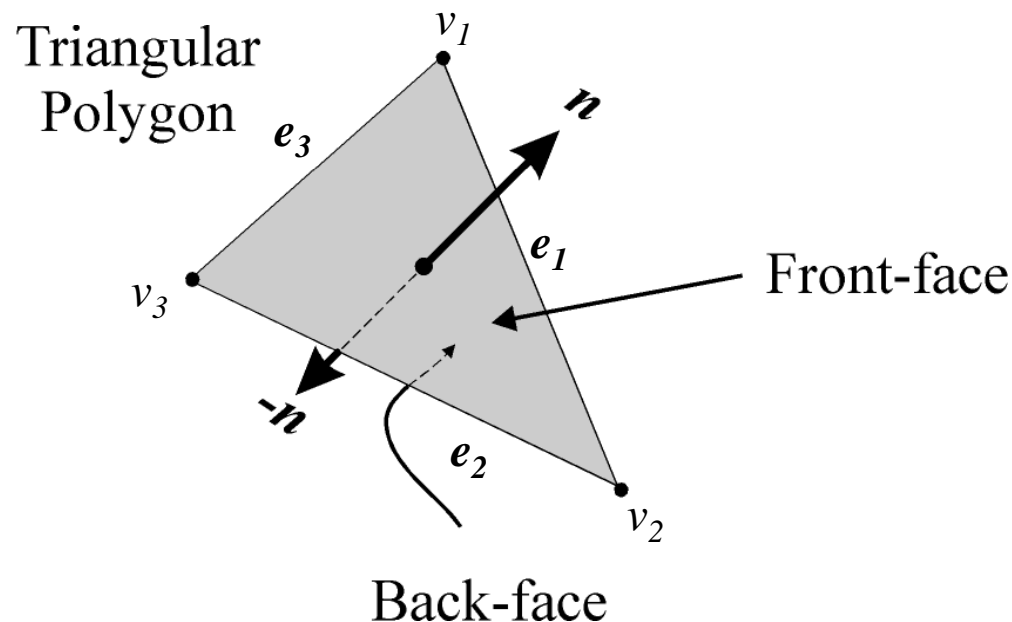
$$\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) \neq (\mathbf{u} \times \mathbf{v}) \times \mathbf{w}$$

- Direction of resulting vector defined by operand order:



Normals & Polygons

- Polygons are (usually) planar regions bounded by n edges connecting $n+1$ points or *vertices*.
- For lighting and viewing calculations we need to define the normal to a polygon:



- The normal distinguishes the *front-face* from the *back-face* of the polygon.

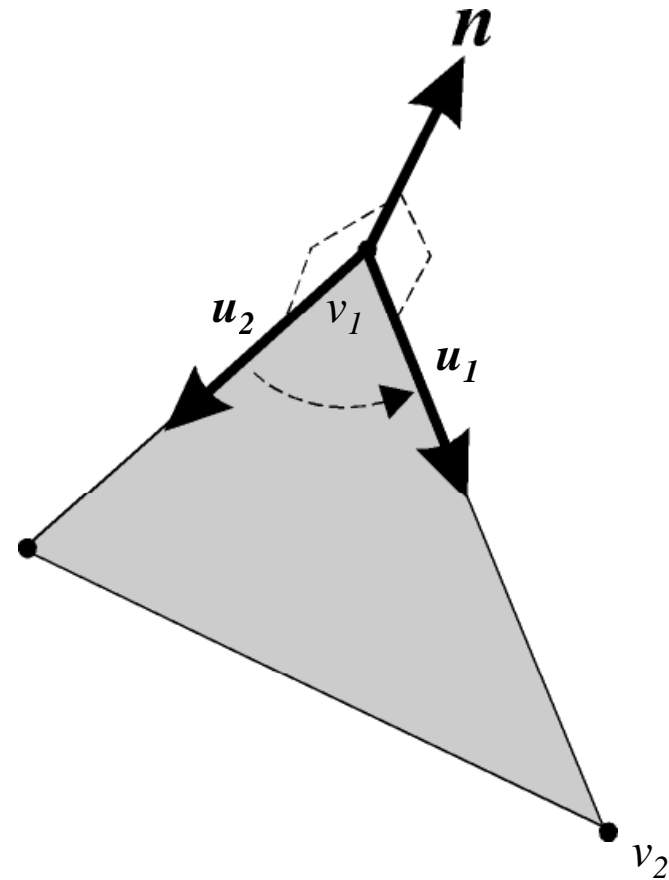
Normals & Polygons

- First determine the 2 *edge vectors* from the vertices:

$$\mathbf{u}_1 = \frac{v_2 - v_1}{\|v_2 - v_1\|} \quad \mathbf{u}_2 = \frac{v_3 - v_1}{\|v_3 - v_1\|}$$

- The polygon normal is given by:

$$\mathbf{n} = \frac{\mathbf{u}_2 \times \mathbf{u}_1}{\|\mathbf{u}_2 \times \mathbf{u}_1\|}$$

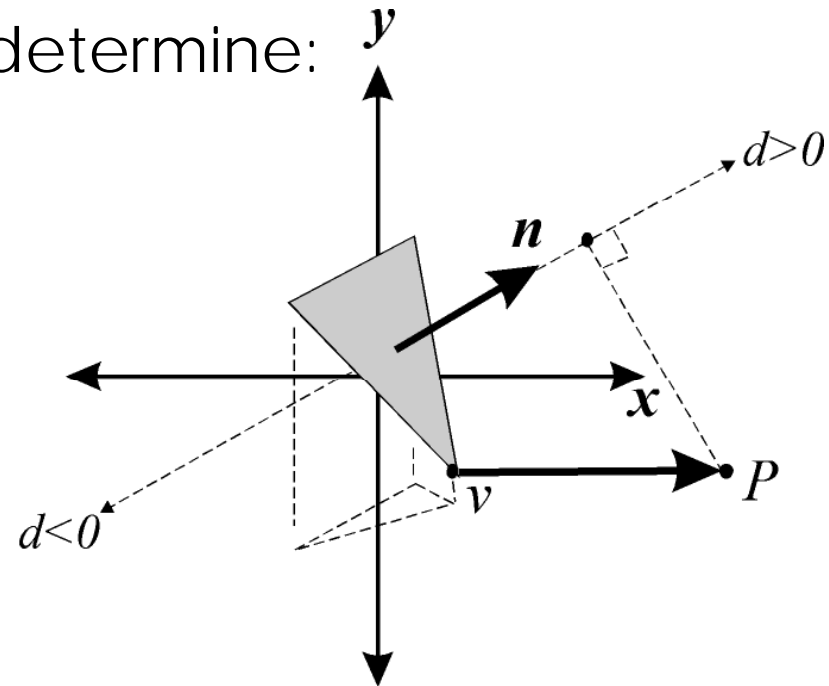


Normals & Polygons

- The plane of the polygon divides 3D space into 2 *half-spaces*
- All points P are either in front of or behind the polygon.
- To determine the side determine:

$$d = \mathbf{n} \cdot (P - v_i)$$

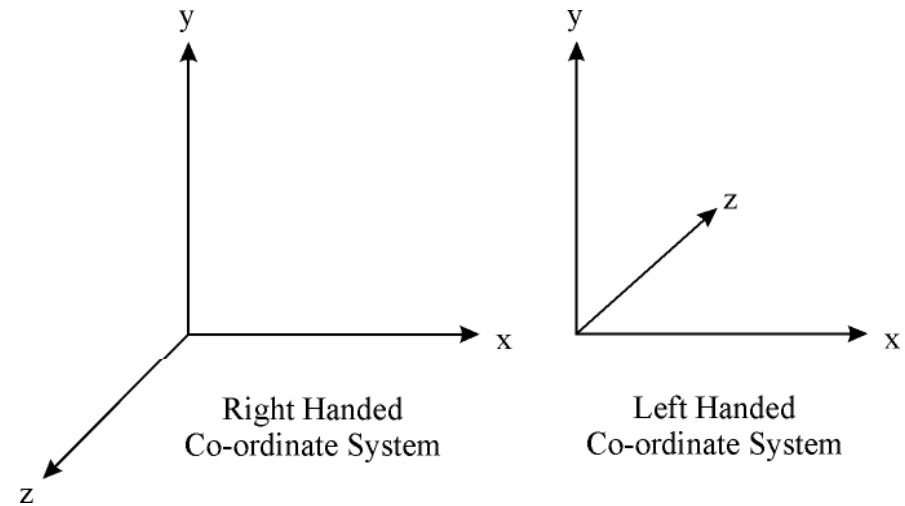
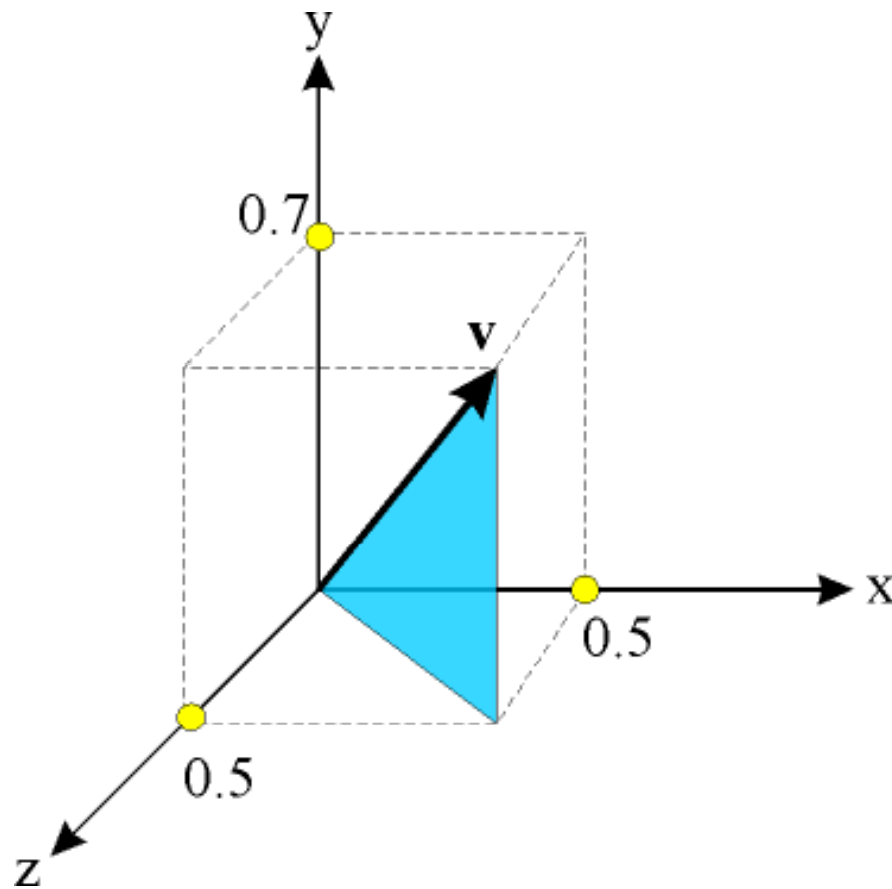
- $d < 0 \Rightarrow P$ behind
- $d = 0 \Rightarrow P$ on polygon
- $d > 0 \Rightarrow P$ in front



Co-ordinate Systems

- By convention we usually employ a *Cartesian basis*:
 - basis vectors are *mutually orthogonal* and *unit length*
 - basis vectors named **x**, **y** and **z**
- We need to define the relationship between the 3 vectors: there are 2 possibilities:
 - *right handed systems*: **z** comes out of page
 - *left handed systems*: **z** goes into page
- This affects direction of rotations and specification of *normal vectors*

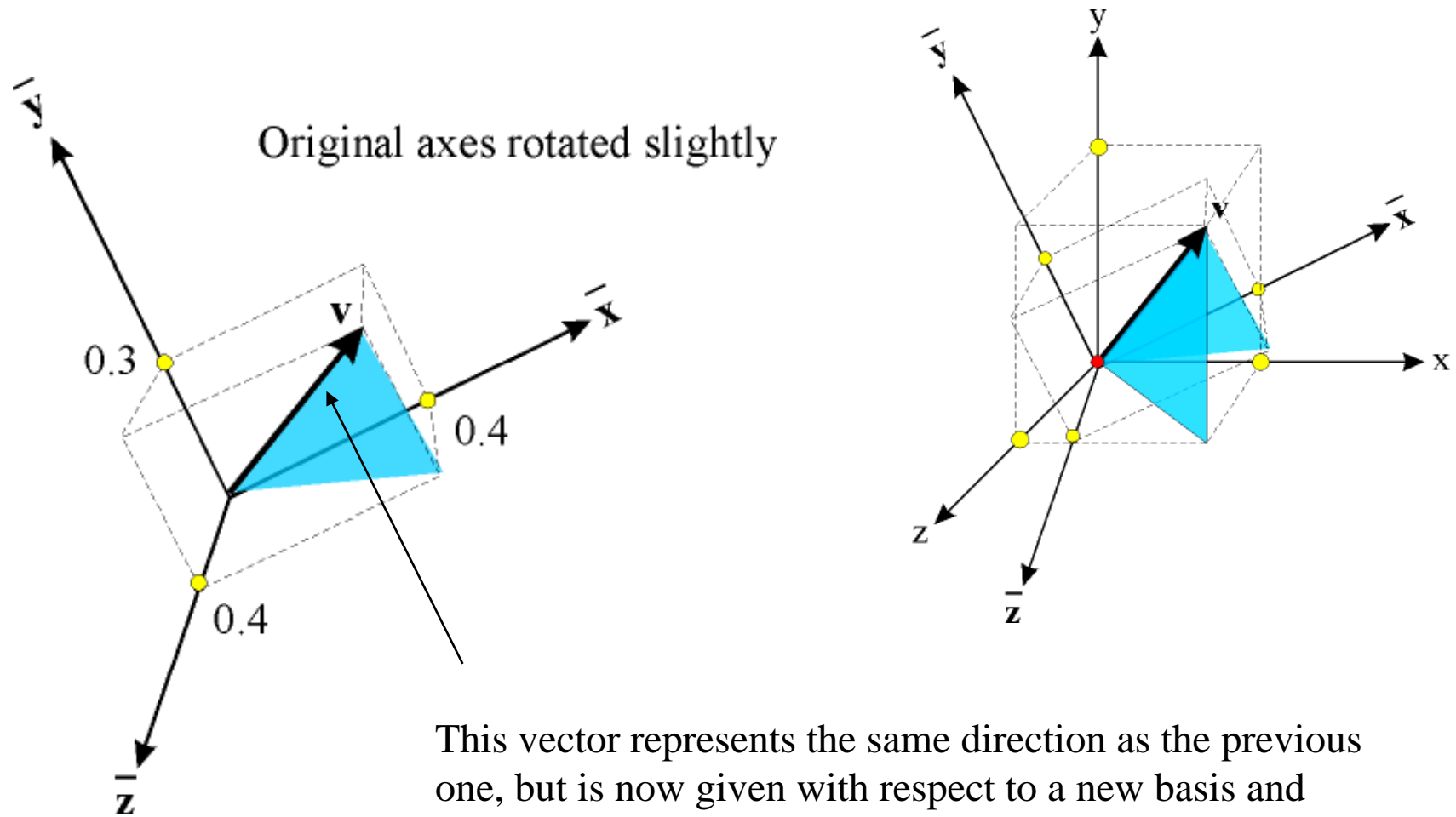
Cartesian co-ordinate System



$$\mathbf{v} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.7 \\ -0.5 \end{bmatrix}$$

RHS LHS

... same vector in a new co-ordinate system



This vector represents the same direction as the previous one, but is now given with respect to a new basis and therefore its value changes accordingly.

Change of Basis

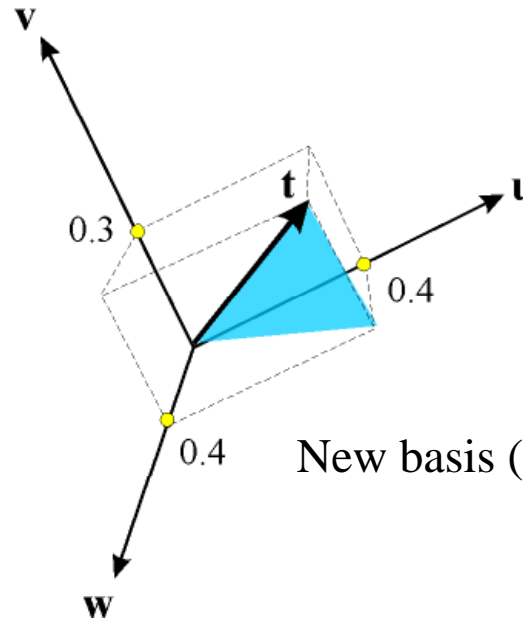
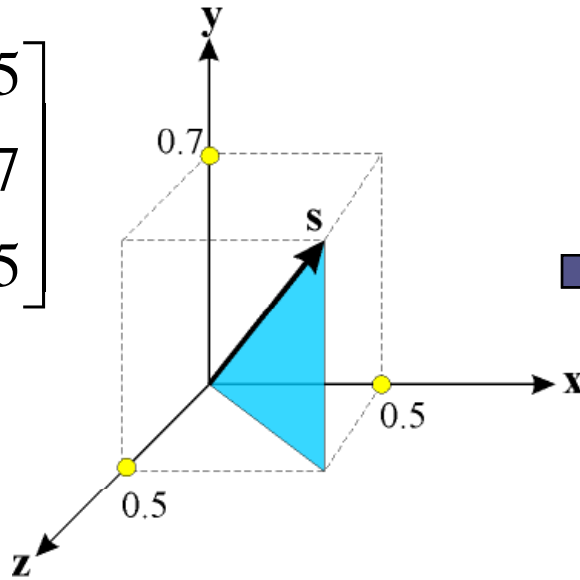
- If we know \mathbf{s} defined w.r.t. basis \mathbf{xyz} we can determine \mathbf{t} which is the same vector defined w.r.t. basis \mathbf{uvw} .
 - t_u is the projected distance of \mathbf{s} onto \mathbf{u}
 - t_v is the projected distance of \mathbf{s} onto \mathbf{v}
 - t_w is the projected distance of \mathbf{s} onto \mathbf{w}

$$\mathbf{t} = \begin{bmatrix} \mathbf{s} \cdot \mathbf{u} \\ \mathbf{s} \cdot \mathbf{v} \\ \mathbf{s} \cdot \mathbf{w} \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix} \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \mathbf{M}\mathbf{s} \begin{cases} t_u = u_x s_x + u_y s_y + u_z s_z = \mathbf{u} \cdot \mathbf{s} \\ t_v = v_x s_x + v_y s_y + v_z s_z = \mathbf{v} \cdot \mathbf{s} \\ t_w = w_x s_x + w_y s_y + w_z s_z = \mathbf{w} \cdot \mathbf{s} \end{cases}$$

- Matrix \mathbf{M} allows us to transform a vector from one basis to another $\Rightarrow \mathbf{M}$ is a *transformation matrix*.
- Many common geometric operations can be expressed as a transformation matrix.

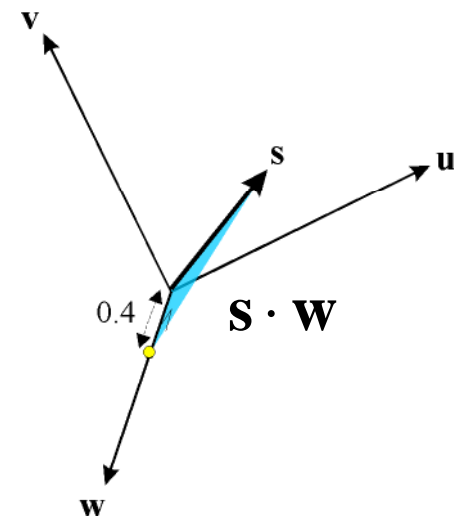
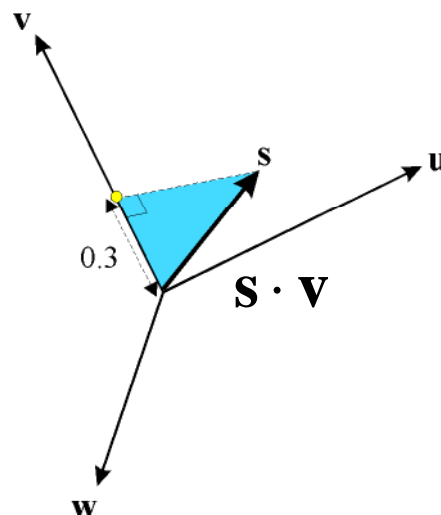
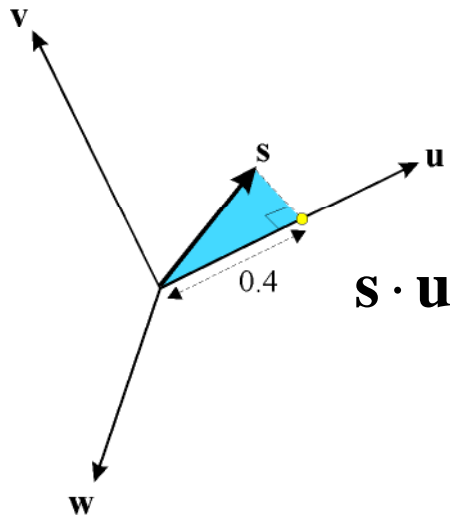
Change of Basis

$$\mathbf{s} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix}$$



$$\mathbf{t} = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.4 \end{bmatrix}$$

New basis (co-ordinate system)



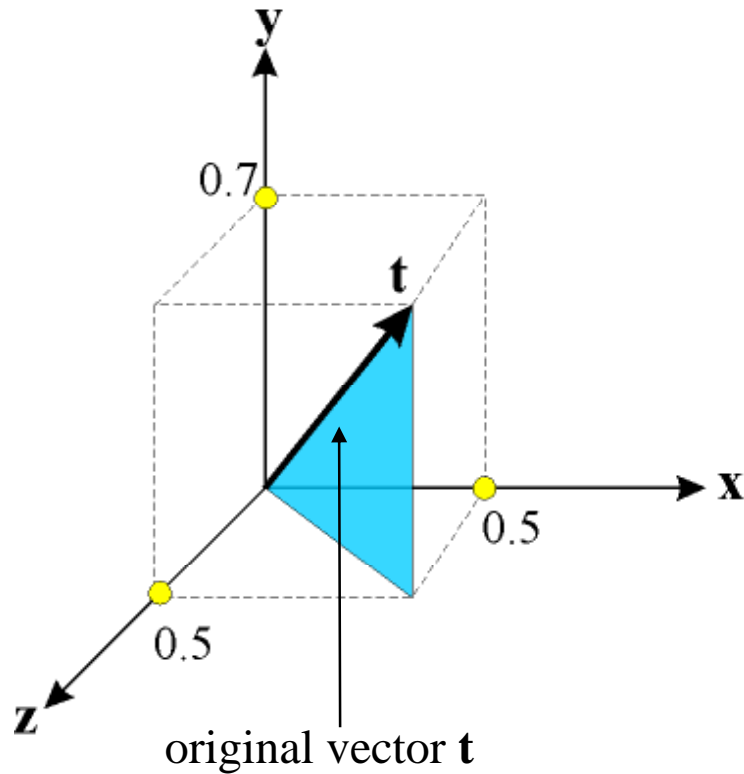
Change of Basis

- Normally the vectors forming the basis of a coordinate system are *unit length* and *mutually orthogonal*
 - basis is said to be *orthonormal*
- This leads to a useful property of the coordinate matrix: $\mathbf{M}^{-1} = \mathbf{M}^T$
 - a property shared by all rotation matrices
 - not true for scaling transformation
- Therefore if we have a vector \mathbf{t} defined w.r.t. basis \mathbf{uvw} then the vector w.r.t. basis \mathbf{xyz} is given by:

$$\mathbf{s} = t_u \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} + t_v \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + t_w \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \begin{bmatrix} t_u \\ t_v \\ t_w \end{bmatrix} = \mathbf{M}^{-1} \mathbf{t} = \mathbf{M}^T \mathbf{t}$$

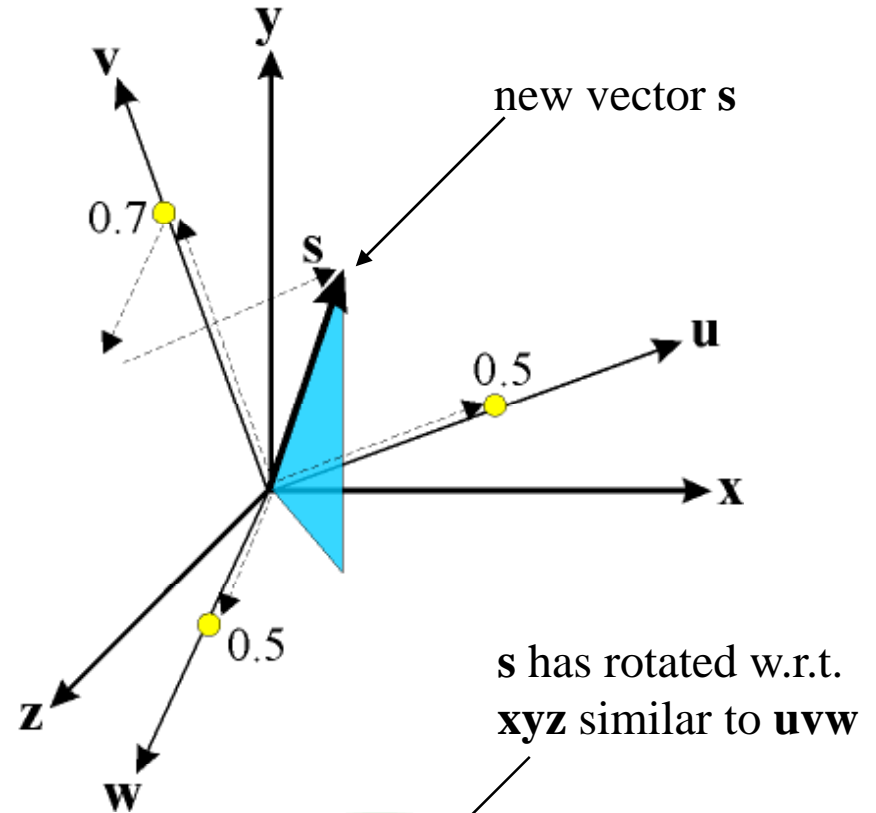
Change of Basis = Transformation

Changing basis is geometrically equivalent to transformation:



$$\mathbf{t} \text{ w.r.t. } \mathbf{xyz} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix}$$

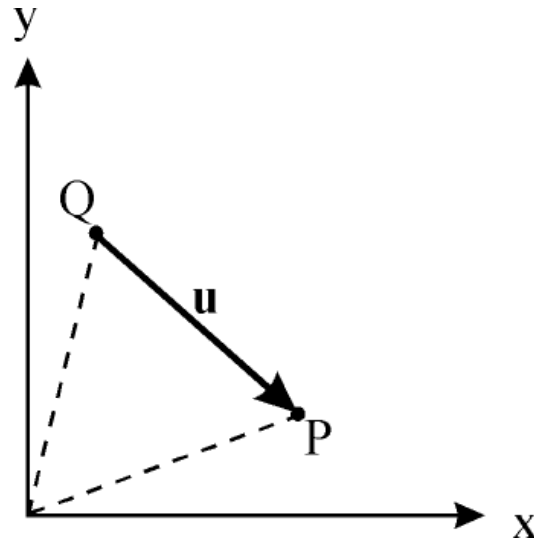
$$\mathbf{s} \text{ w.r.t. } \mathbf{uvw} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix}$$



$$\mathbf{s} \text{ w.r.t. } \mathbf{xyz} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.3 \end{bmatrix}$$

Affine Spaces

- Vectors define direction and magnitude only.
- To encode position we need to fix the *origin*.
- The origin is a *point*.
- *Affine space* = a set of points with an associated vector space with the operations difference and translate.
- Points are related by vectors: $\mathbf{u} = P - Q$ or $Q + \mathbf{u} = P$



Affine Combinations

- We never multiply a point by a scalar or another point or add a point to another point
 - undefined geometric operations
- Affine combination of points P and Q by scalar t is:

$$R = P + t(Q - P) = (1 - t)P + tQ$$

- If $0 \leq t \leq 1$ then the affine combination is *convex*

