Modelling Geometry

Modelling

- Generic definition – Model: an abstract or actual representation of system or an object.
- Models in Computer Graphics deal with data that eventually leads to graphical output (images + animation + video)

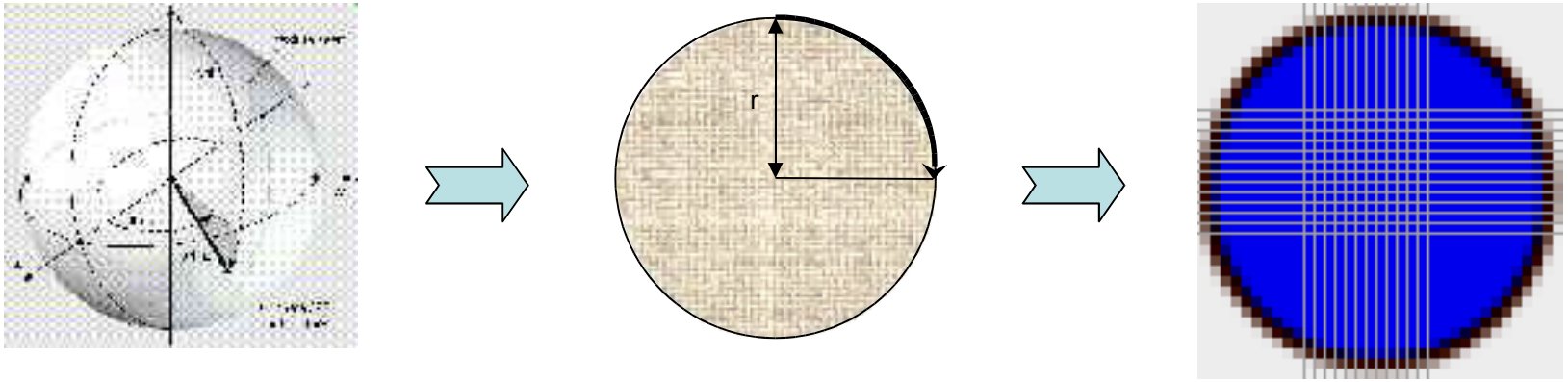


Modelling spectrum

- 3D Modelling
- 2D Modelling
- Painting / drawing

Each stage has it's own unique type of data and can accept user input.

The more abstract the representation then (arguably) the better our opportunities for re-usability and data/computational efficiency.



Modelling vs. Painting

- Models are usually:
  - More abstract
  - More compact
  - Reusable
  - Modifiable
- Models need to be rasterized/processed before drawing

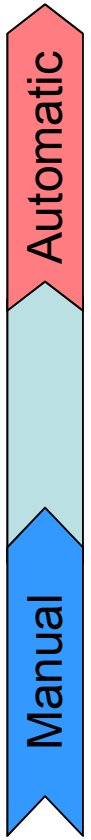
- Images:
  - Low level
  - Resolution bound
  - Less re-usable
  - More limited in how we can change them

Modelling Issues

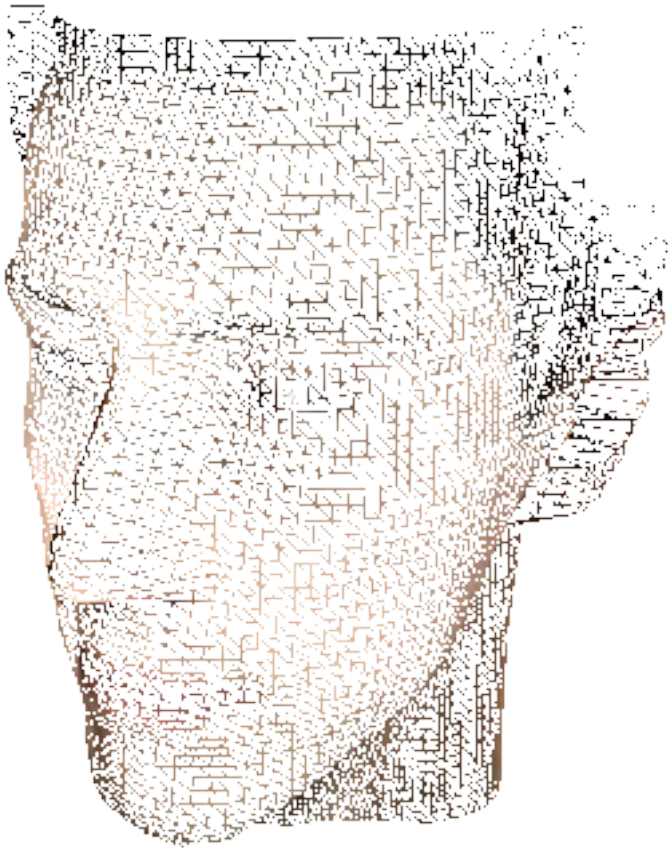
- A few questions that should concern us about models:
  - Where do models come from?
  - How are models stored?
  - How are models modified?
  - How are models *rendered*?

Modelling

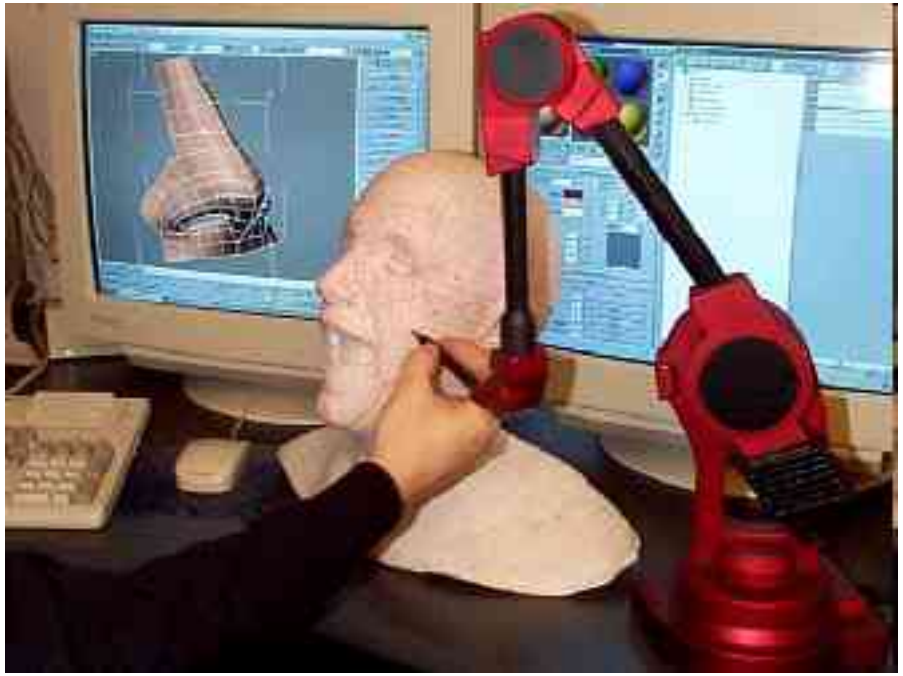
- Model Acquisition/ Creation
  - Image/3D Scanners
  - 3D Model-extraction
    - from images
    - from video
  - Mechanical Digitizers
  - Modelling Tools
  - From scratch
- Model Manipulation
  - Transformation / deformation operations
  - Procedural animation
  - Modelling Tools
  - User manipulation



3D Scanning



3D mechanical digitizers



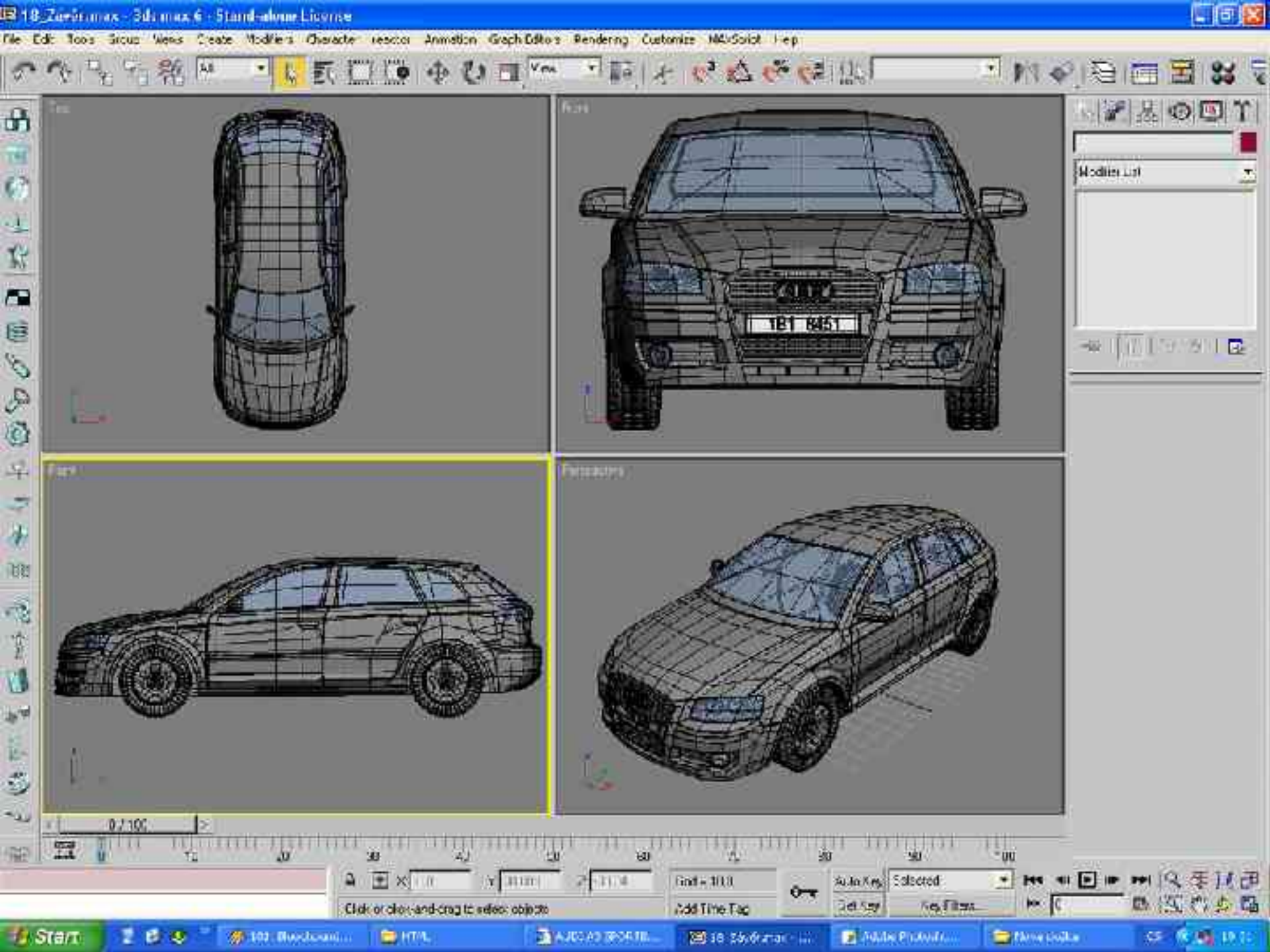
Ghost 3D Microscribe

- Co-ordinate measurement machine (CMM) record 3D points
  - Similar to “tracing” in 3D

Ghost 3D Microscribe Laser

Some Modelling tools

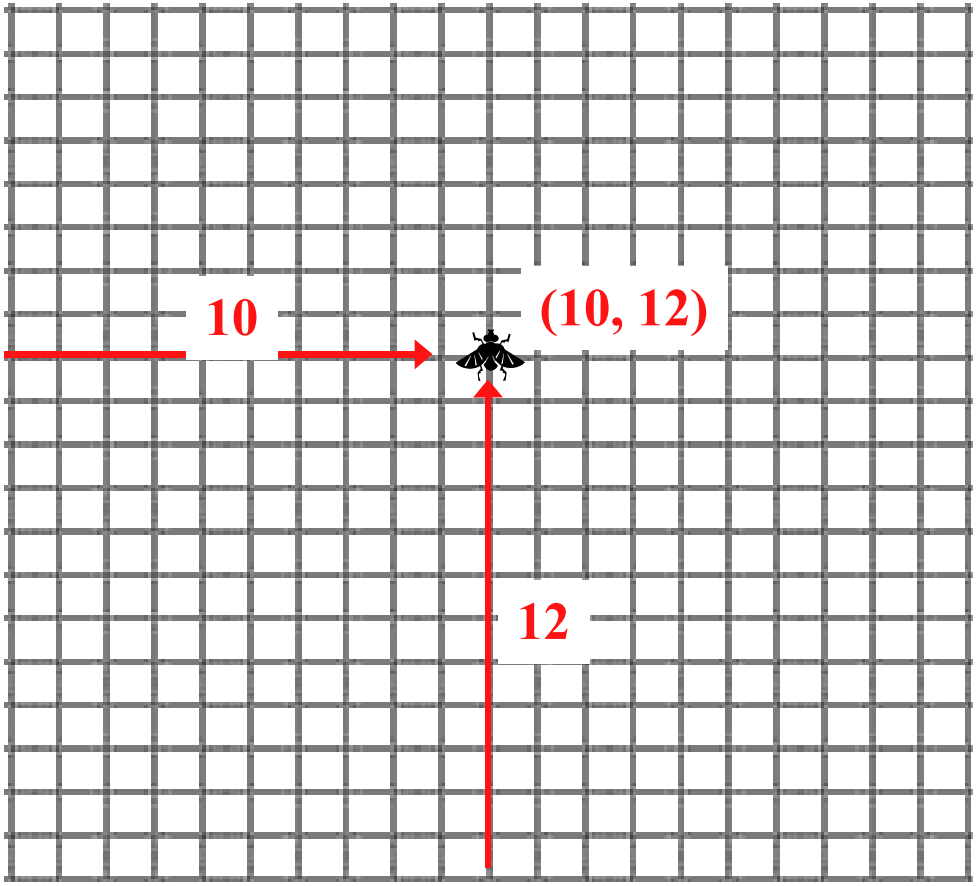
- Graphics APIs for 2D/3D programming
- 3D Animation Tools
- Video editors
- 3D Rendering Tools
- 3D Scene Modelling (world modelling)
- 3D Object Modelling Applications
- 2D Painting, sketching curve drawing applications
- Image/photo editors


File Edit Tools Group Views Create Modifiers Character reactor Animation Graph Editors Rendering Customize MAXScript Help
Modifier List
Click or click-and-drag to select objects
Add Time Tag
Auto Key
Set Key
Key Filters...
Start

The building blocks of a model

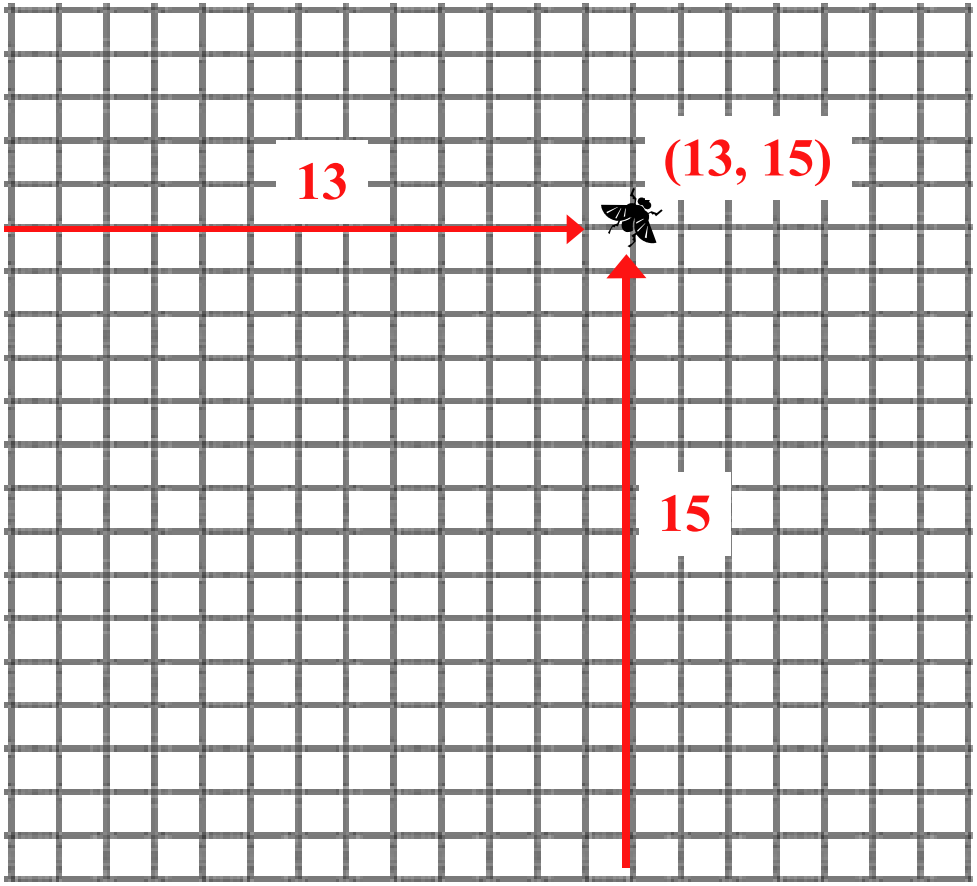
- Geometry is often represented in terms of very basic numerical data
  - Scalars: single values
  - Vectors: a couple or triple (or n-tuple) containing scalars – the number of scalars is the dimensionality.
    - Vectors usually denote a value and a direction.
- In Graphics, more complex geometry e.g. Curves and Surfaces can also be defined based on scalars and vectors

Cartesian Coordinates



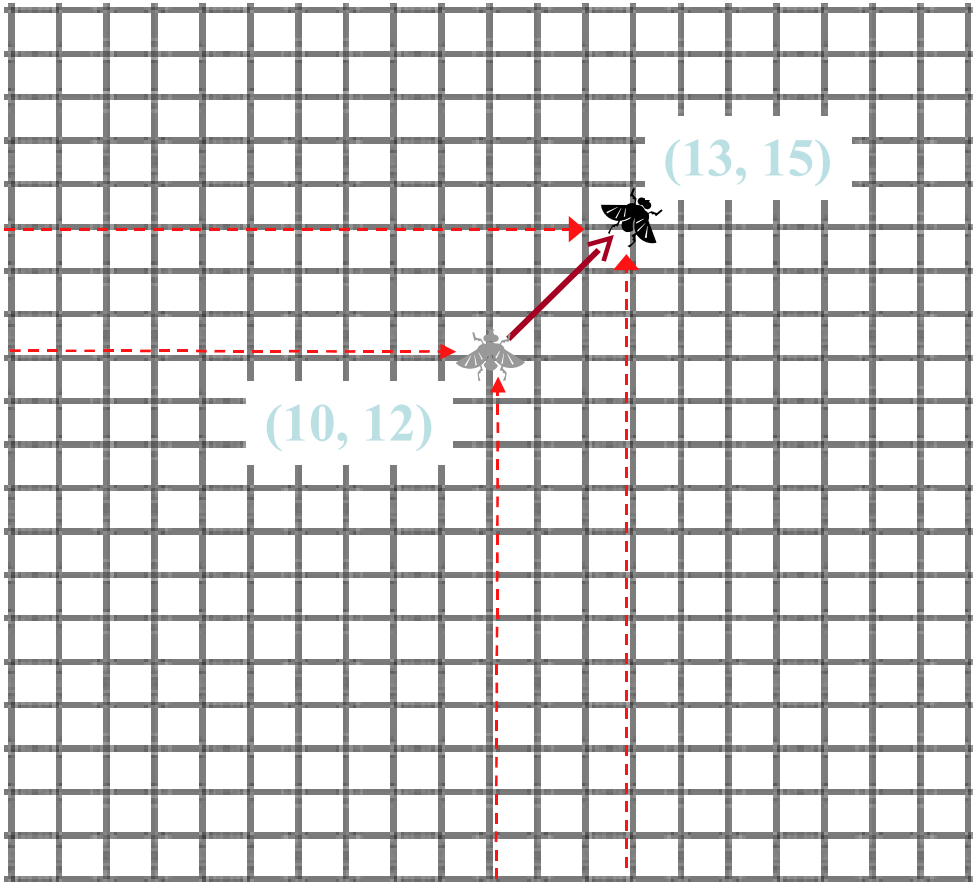
Rene Descartes - 1637

Cartesian Coordinates



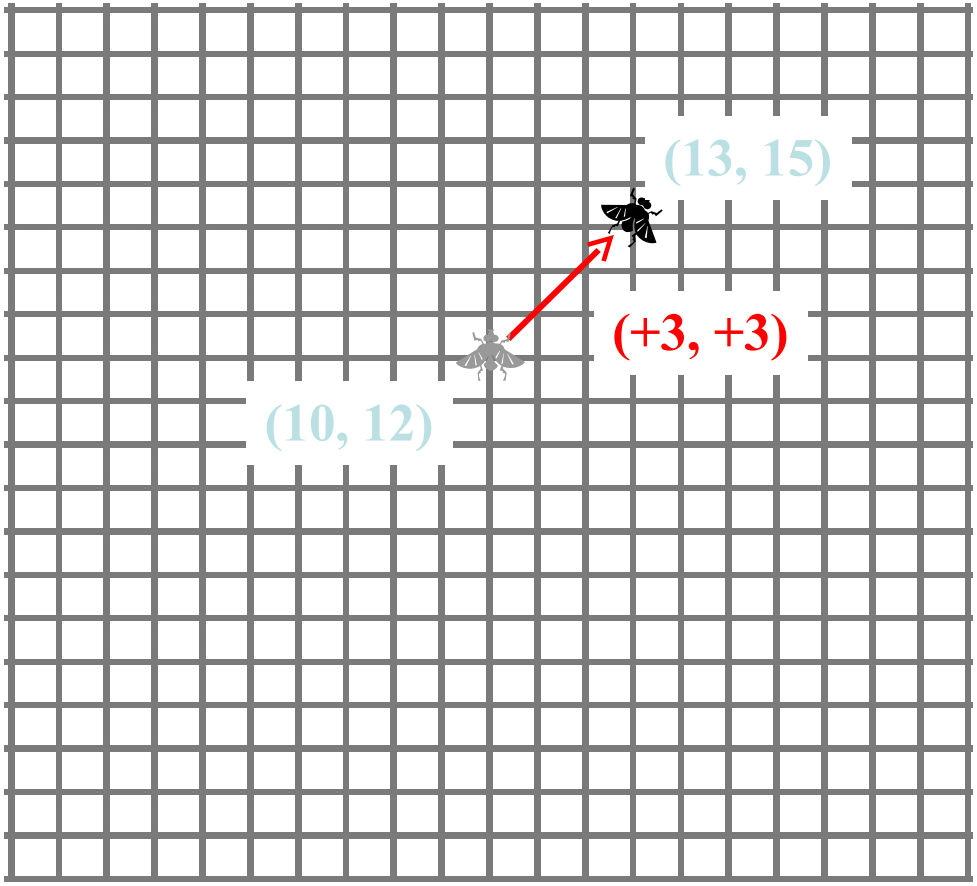
- Two distances used to denote unique **position** in 2D

Cartesian Coordinates



- Two distances used to denote unique **position** in 2D
- We can also represent a line with two **endpoints**, each donated by vector.

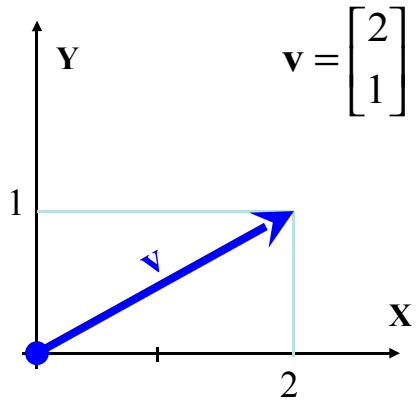
Cartesian Coordinates



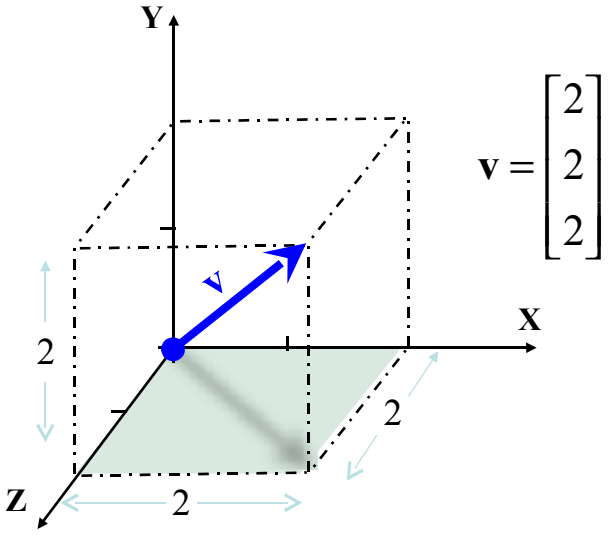
- Two distances used to denote unique **position** in 2D
- We can also represent a line with two **endpoints**, each donated by a vector.
- Finally we can denote ***translation*** (or change in position) with two values.

Vector

- A quantity characterized by a **magnitude** and **direction**
- Can be represented by an arrow, where magnitude is the *length* of the arrow and the direction is given by *slope* of the line

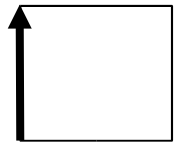


A vector in 2D

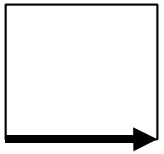


A vector in 3D

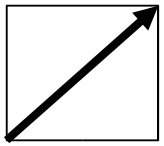
Examples in 2D



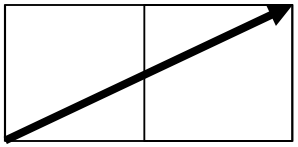
$$"up" = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



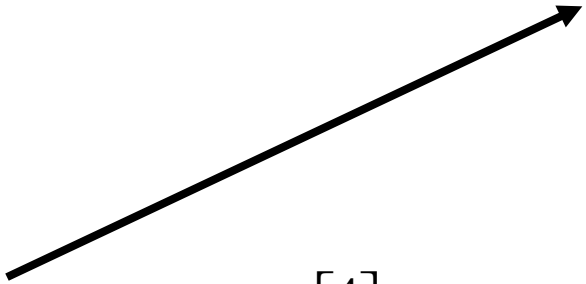
$$"right" = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



$$"north-east" = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Conventions

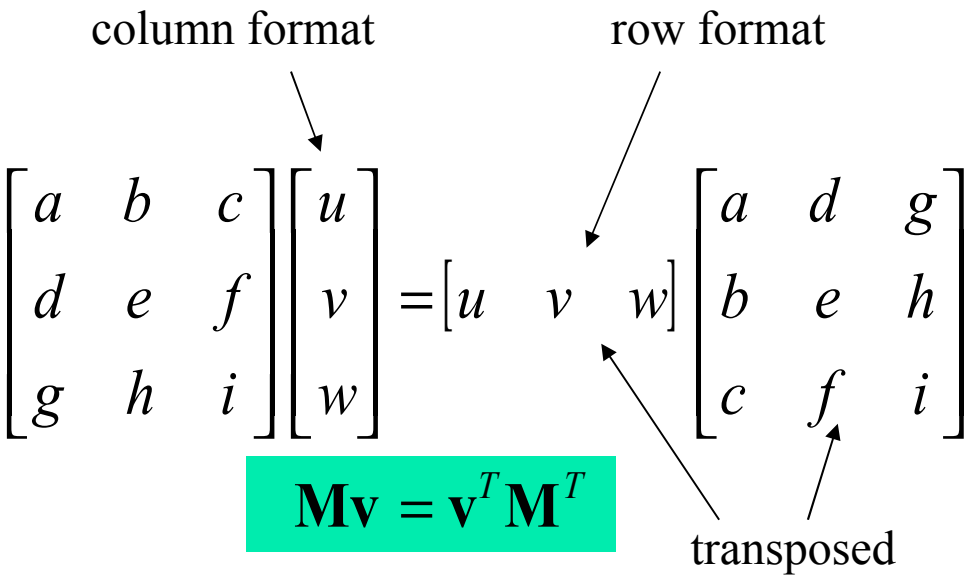
- Vector quantities denoted as v or $\vec{v}$
- We will use column format vectors:

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

In code, we may have to revert to { v1, v2, v3 }

Row vs. Column Formats

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



Linear Combinations

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

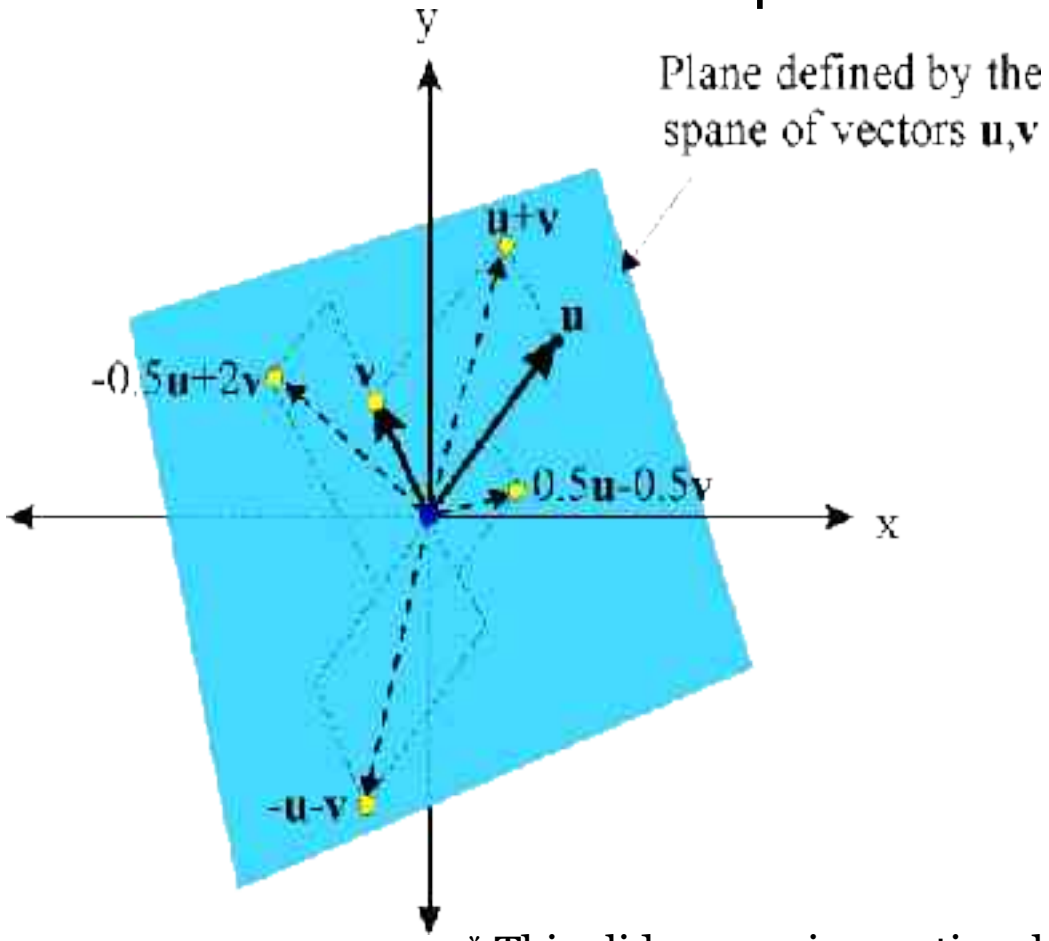
$$\mathbf{u} = a_1\mathbf{v_1} + a_2\mathbf{v_2} + \cdots + a_n\mathbf{v_n}$$

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

* This slide comprises optional background info.

Linear Combinations

- Linear combinations of 2 vectors = a plane

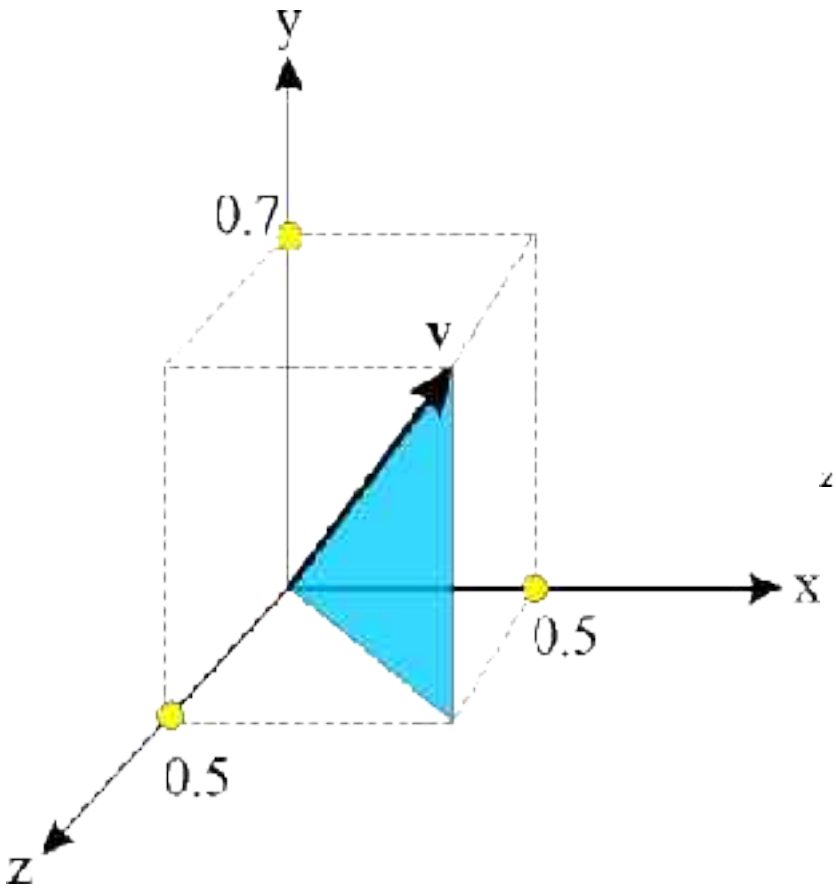


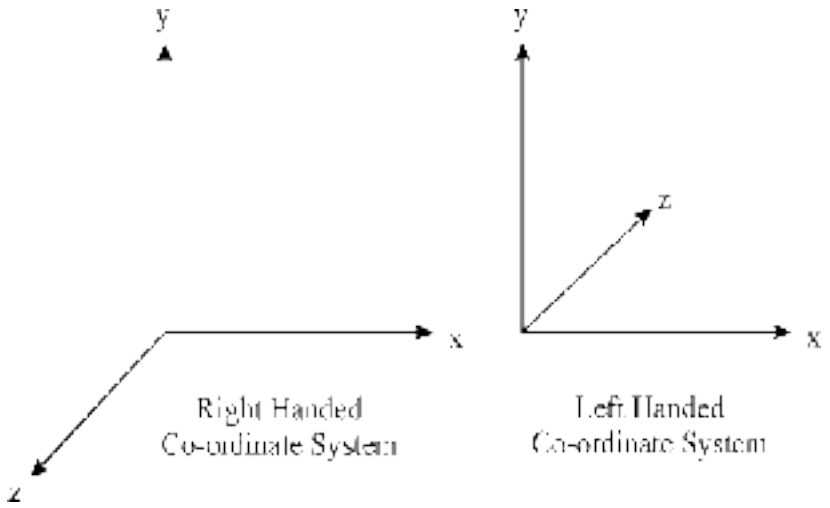
* This slide comprises optional background info.

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

3D Cartesian co-ordinates





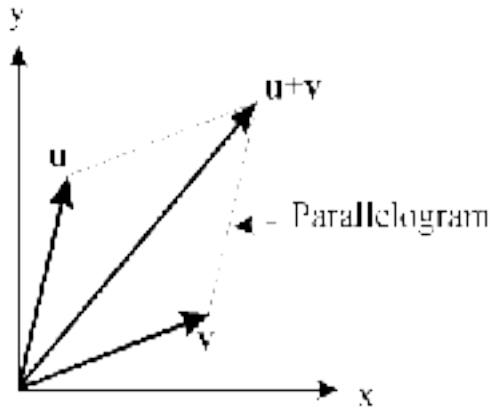
$$\mathbf{v} = \begin{bmatrix} 0.5 \\ 0.7 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.7 \\ -0.5 \end{bmatrix}$$

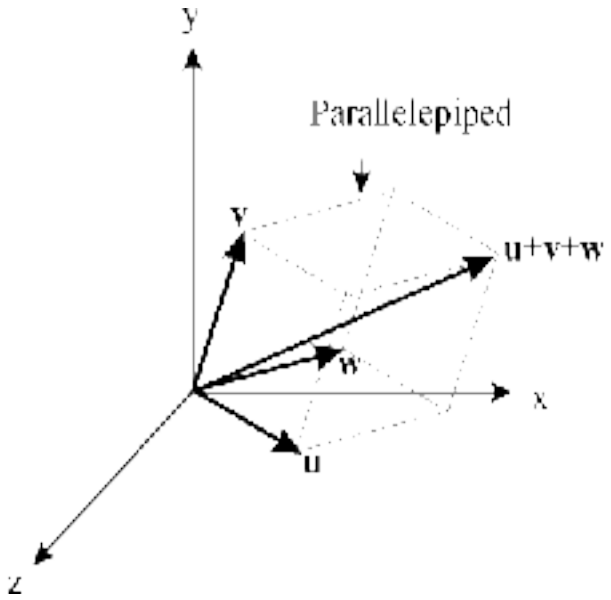
RHS LHS

Vector Addition

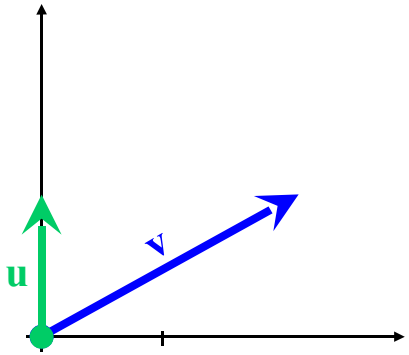
- Addition of vectors follows the *parallelogram* law in 2D and the *parallelepiped* law in higher dimensions:

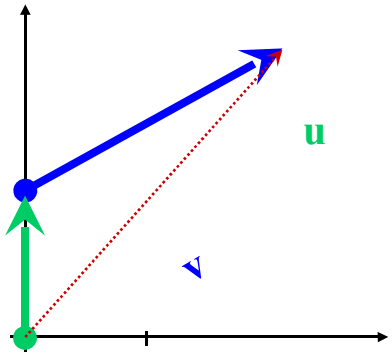
$$\underbrace{\begin{bmatrix}1\\2\\3\end{bmatrix}}_{\mathbf{u}} + \underbrace{\begin{bmatrix}4\\5\\6\end{bmatrix}}_{\mathbf{v}} = \underbrace{\begin{bmatrix}5\\7\\9\end{bmatrix}}_{\mathbf{u+v}}$$



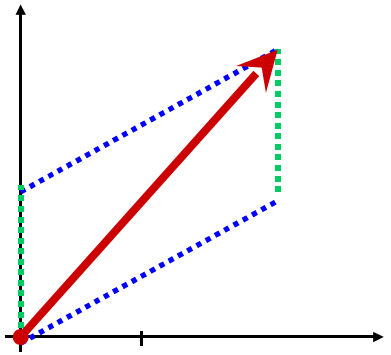


Vector Addition





OR

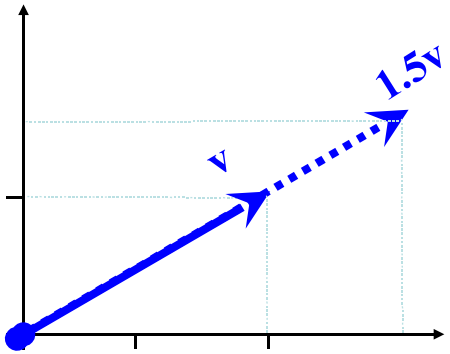


$$\mathbf{u} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

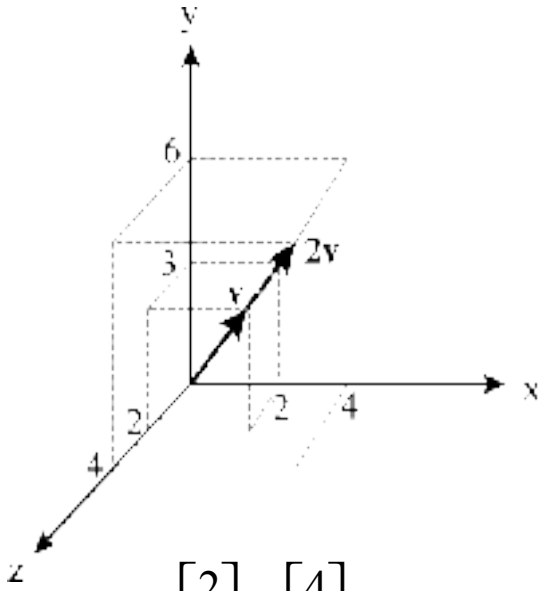
$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Vector Multiplication by a Scalar

- Multiplication by a scalar *scales* the vectors length appropriately (but does not affect direction):

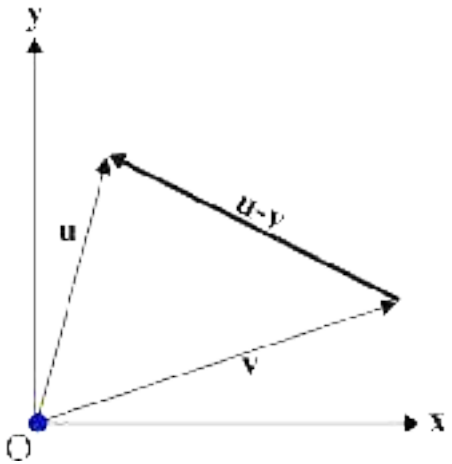


$$1.5\begin{bmatrix}2\\1\end{bmatrix}=\begin{bmatrix}1.5\times 2\\1.5\times 1\end{bmatrix}=\begin{bmatrix}3\\1.5\end{bmatrix}$$

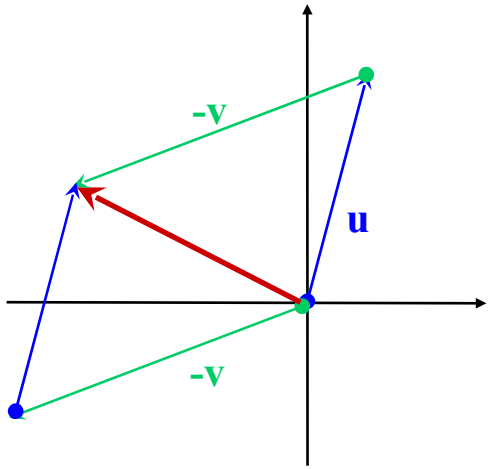


$$2\begin{bmatrix}2\\3\\2\end{bmatrix}=\begin{bmatrix}4\\6\\4\end{bmatrix}$$

Subtraction

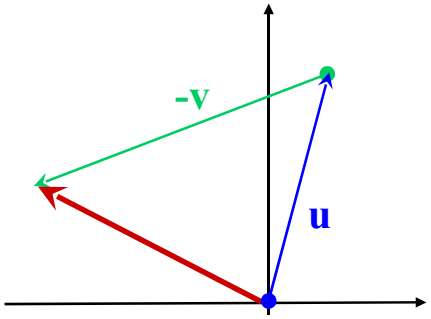


$$\underset{\mathbf{u}}{\begin{bmatrix}1\\4\end{bmatrix}}-\underset{\mathbf{v}}{\begin{bmatrix}5\\2\end{bmatrix}}=\underset{\mathbf{u+v}}{\begin{bmatrix}-4\\2\end{bmatrix}}$$



Can be seen as an addition of

u + (-1v)

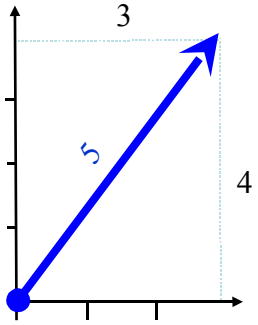


Vector Magnitude

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

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

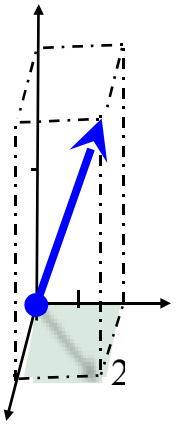
2D example



$$\mathbf{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\|\mathbf{v}\| = \left\| \begin{bmatrix} 3 \\ 4 \end{bmatrix} \right\| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$$

3D example

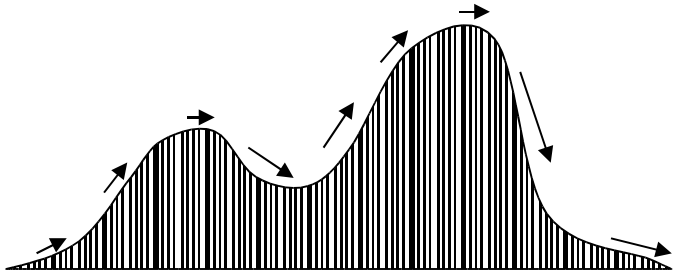


$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

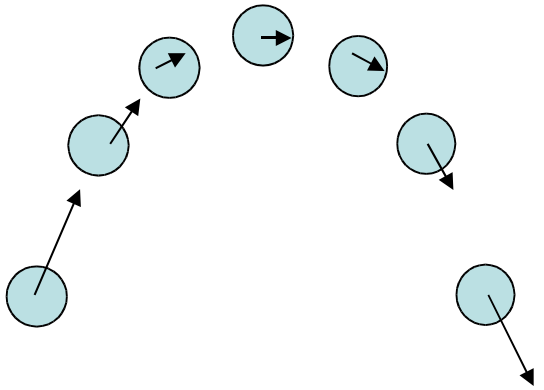
$$\|\mathbf{v}\| = \left\| \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} \right\| = \sqrt{1^2 + 3^2 + 1^2} = \sqrt{11}$$

Vectors for direction

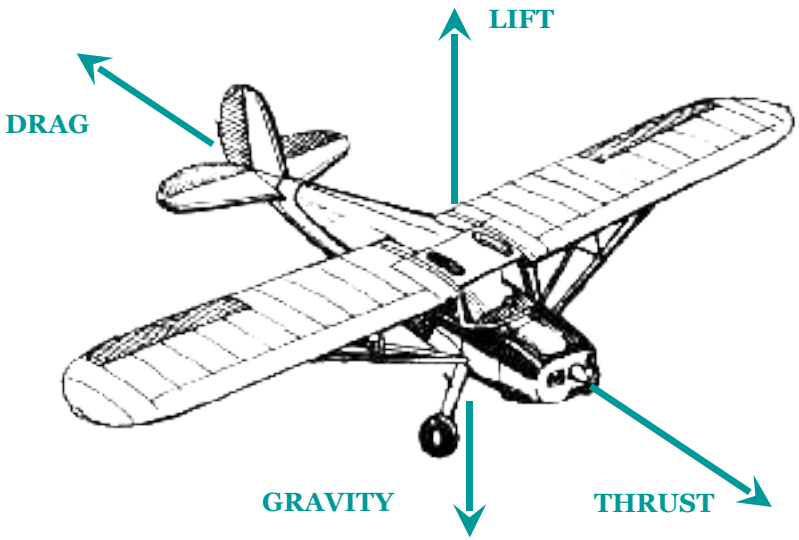
- Vectors represent
  - a direction
  - and a magnitude
- Can be used for representing e.g.
  - Position
  - Velocity
  - Forces / impulses



Describing velocity (direction and speed) of roller coaster and ball at different points in time

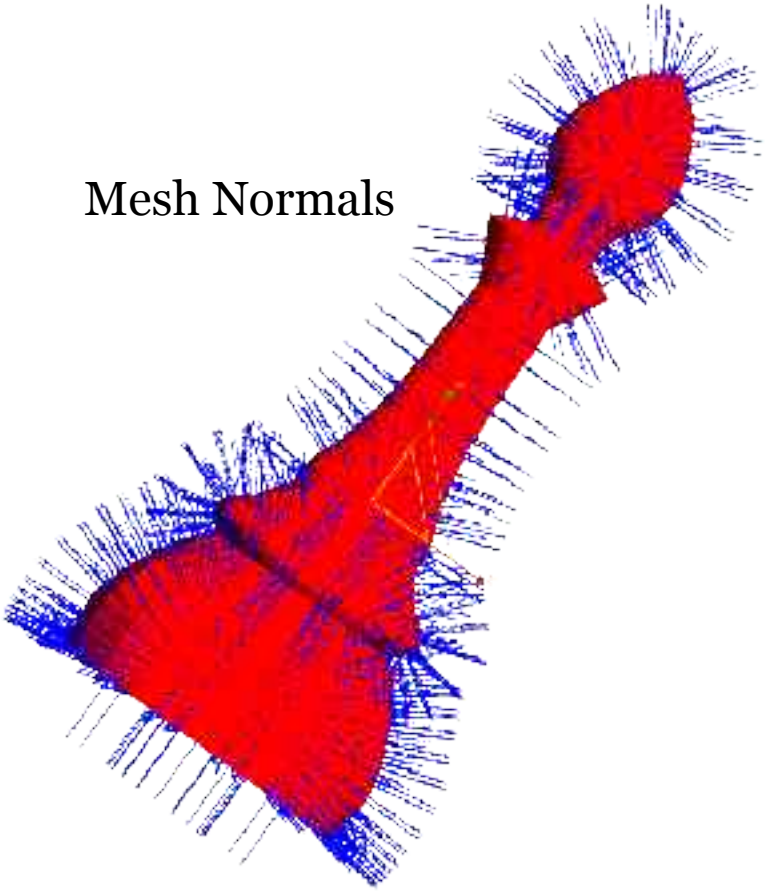


Other Vectors



The **forces** of flight.

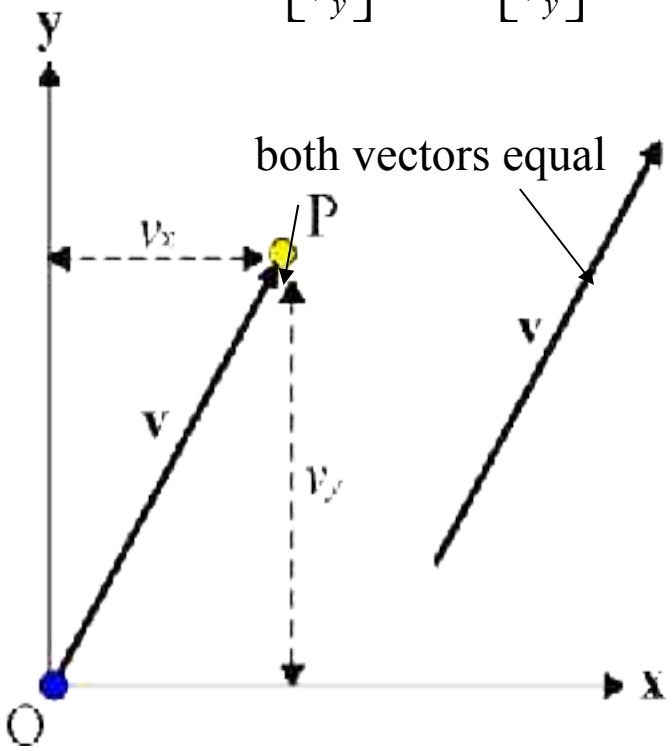
Mesh Normals



Vertices/Points

- Vectors can however communicate a position – referred to as a point or vertex
- A vertex is actually represented by its *displacement* from the **origin** { 0, 0, 0 }

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad \mathrm{P} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$



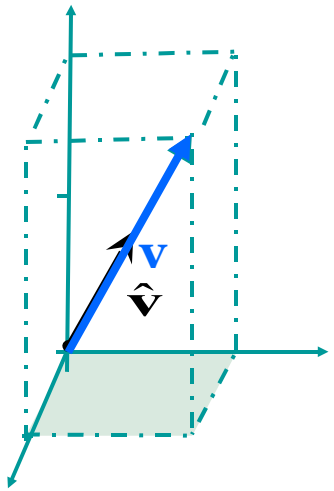
With the origin O, we can use this to represent a unique position in space.

Unit Vectors

- Vectors of length 1 are often termed unit vectors (a.k.a. normalised vectors).
- When we only wish to describe direction we use normalised vectors – often to avoid redundancy
- For this and other reasons, we often need to normalise a vector:

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

- e.g.



$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} \qquad \hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \left(\frac{1}{\sqrt{11}}\right)\mathbf{v} = \left(\frac{1}{\sqrt{11}}\right)\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = \left(\frac{1}{\sqrt{11}}\right)\begin{bmatrix} \frac{1}{\sqrt{11}} \\ \frac{3}{\sqrt{11}} \\ \frac{1}{\sqrt{11}} \end{bmatrix} = \begin{bmatrix} .3015 \\ .9045 \\ .3015 \end{bmatrix}$$

Dot Product

- Dot product (inner product) is defined as: $\mathbf{u} \cdot \mathbf{v} = \sum_i u_i v_i$

$$\mathbf{u} \cdot \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = u_1 v_1 + u_2 v_2 \qquad \mathbf{u} \cdot \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

- Note:
  - Therefore we can redefine magnitude in terms of the dot-product operator:

$$\mathbf{u} \cdot \mathbf{u} = u_1^2 + u_2^2 + u_3^2 = \|\mathbf{u}\|^2 \quad \Rightarrow \quad \|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$$

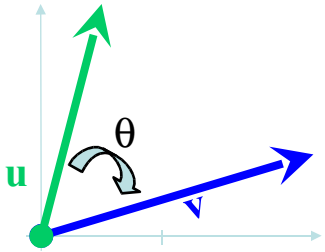
- The dot product operator is commutative and associative.

Dot Product

- The Dot Product can also be obtained from the following equation:

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

where θ is the angle between the two vectors



- So, if we know the vectors **u** and **v**, then the dot product is useful for finding the angle between two vectors.

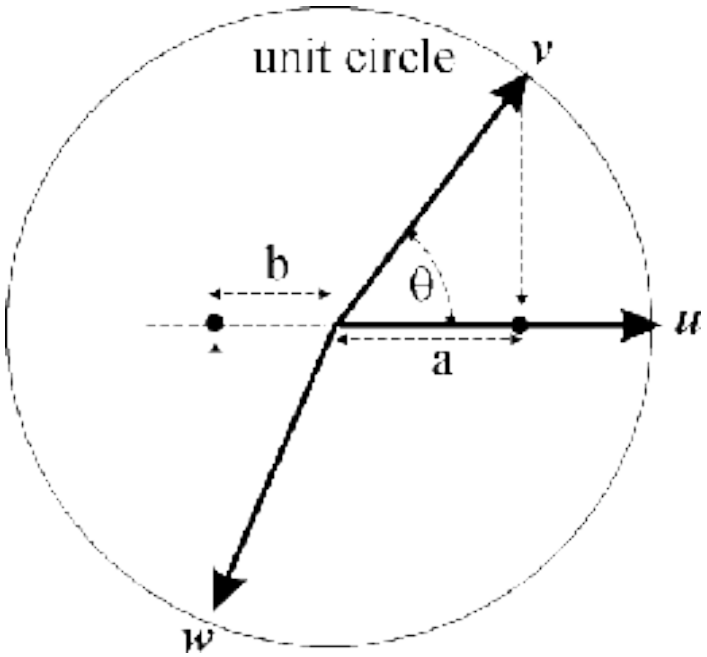
$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos\theta \quad \Rightarrow \quad \theta = \cos^{-1}\left[\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}\right]$$

  - Note that if we had already normalised the vectors **u** and **v** then it would simply be:

$$\theta = \cos^{-1}\left[\hat{\mathbf{u}} \bullet \hat{\mathbf{v}}\right]$$

Dot Product

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



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

In general:

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

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

if $\theta > 90$ then the dot product is negative.

Dot Product Examples

- Find the angle between vectors { 1, 1, 0} and {0, 1, 0}?

Cross Product

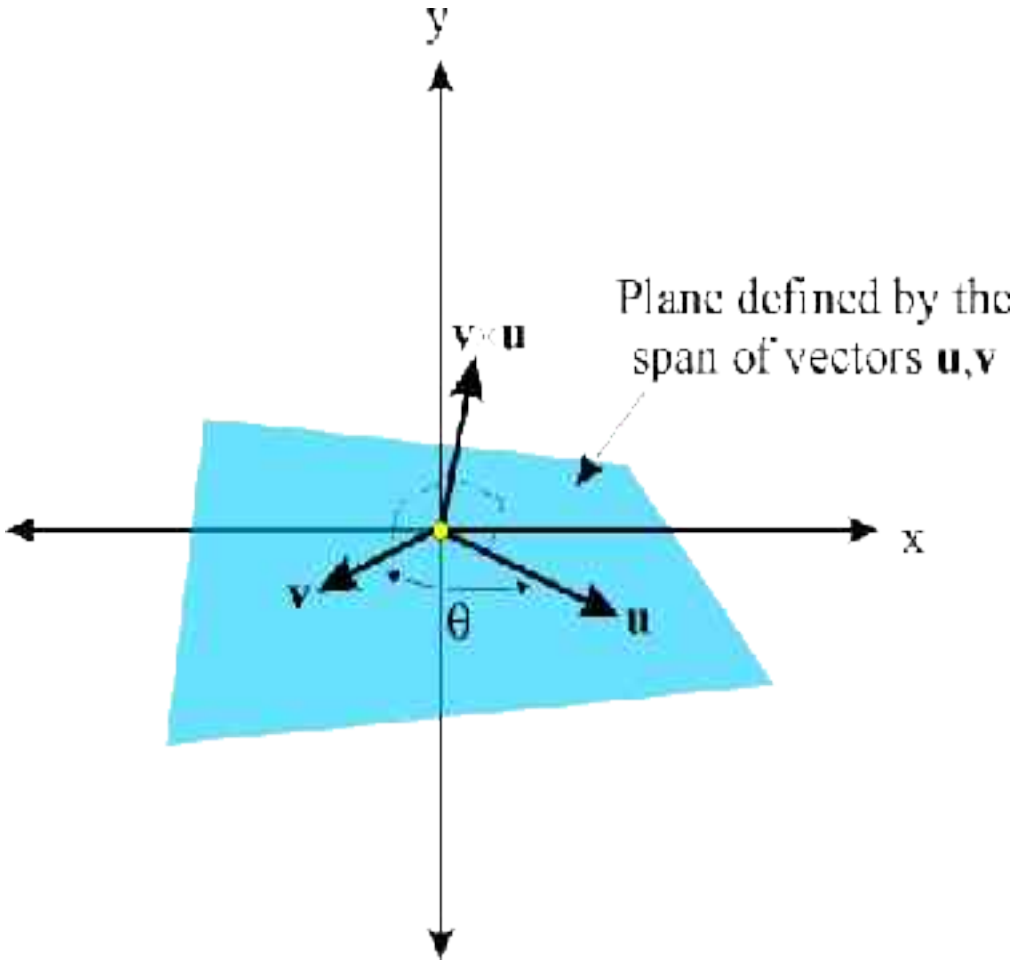
- Used for defining *orientation* and constructing *co-ordinate axes*.
- Cross product defined as:

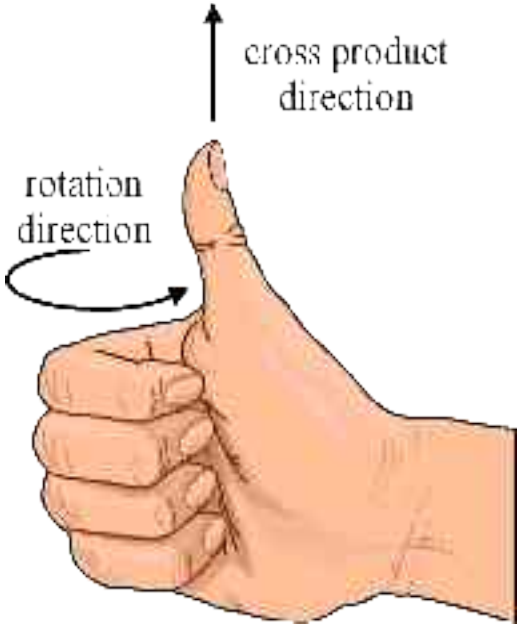
$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix}$$

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

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

Cross Product

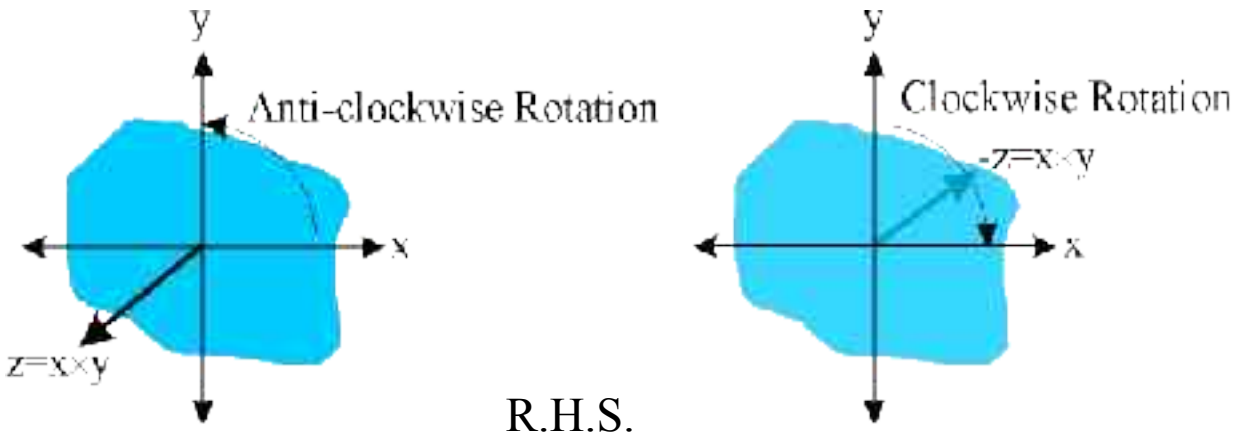




Right Handed Coordinate System

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:



R.H.S.

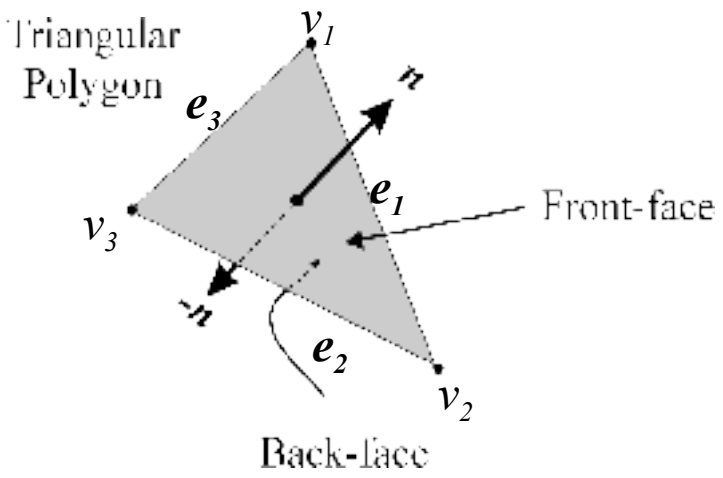
Cross Product Examples

- Find the line that is orthogonal to a polygon defined by the vertices { 1, 1, 0} {0, -1, 0} and {-2, 0, 0}.

- If a person stands upright (direction {0, 1, 0}) at the origin {0, 0, 0}, facing a direction { 1, 0, 1 },
  - Then what vector describes the direction that the person considers “LEFT”?

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.

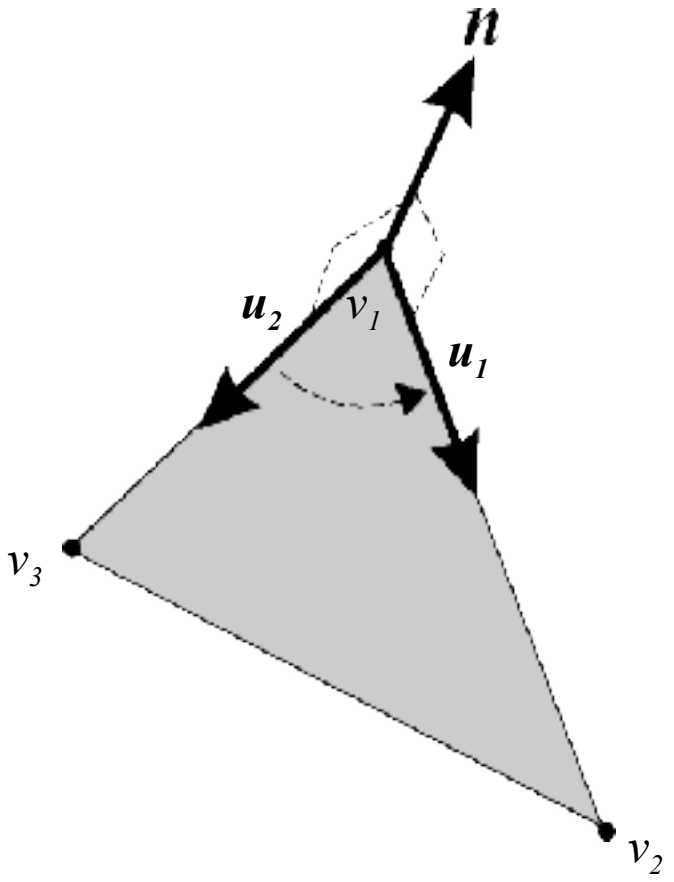
Normals & Polygons

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

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

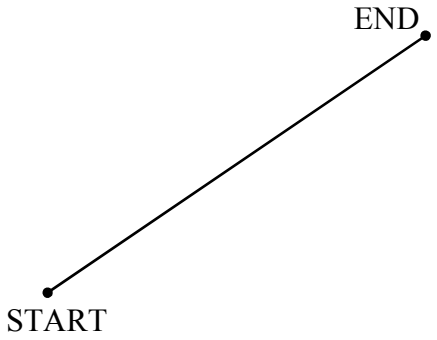
- The polygon normal is given by:

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

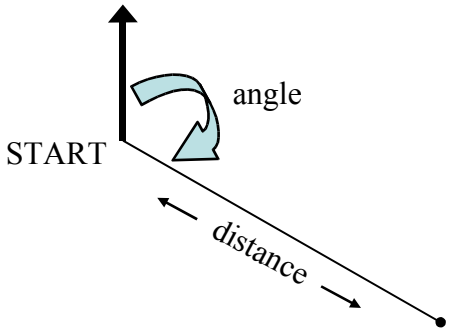


Describing a Line

- **Line Segment:** Start and Endpoint
  - START at { 0, 0 }
  - END at { 200, 160 }

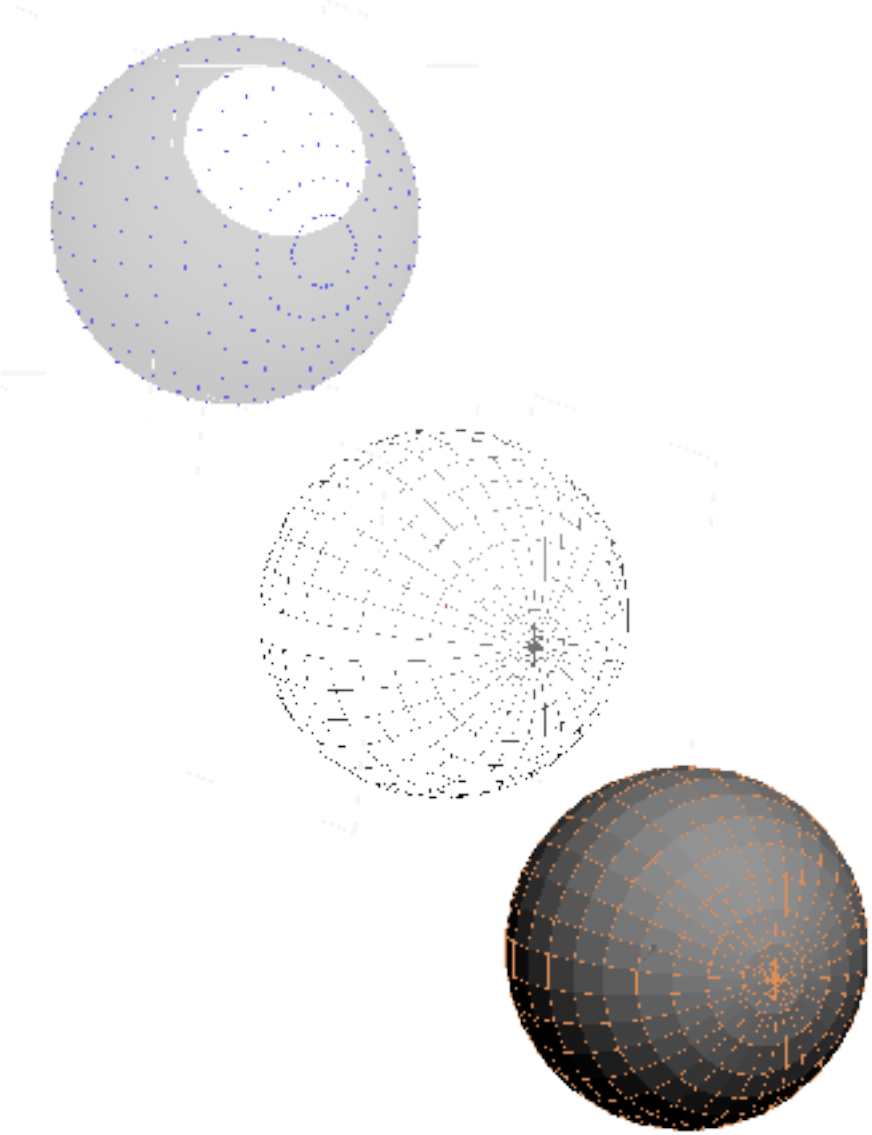


- **Ray:** Start, Direction, Length
  - START at { 0, 0 }
  - Turn 115° from due north
  - Move forward by 160



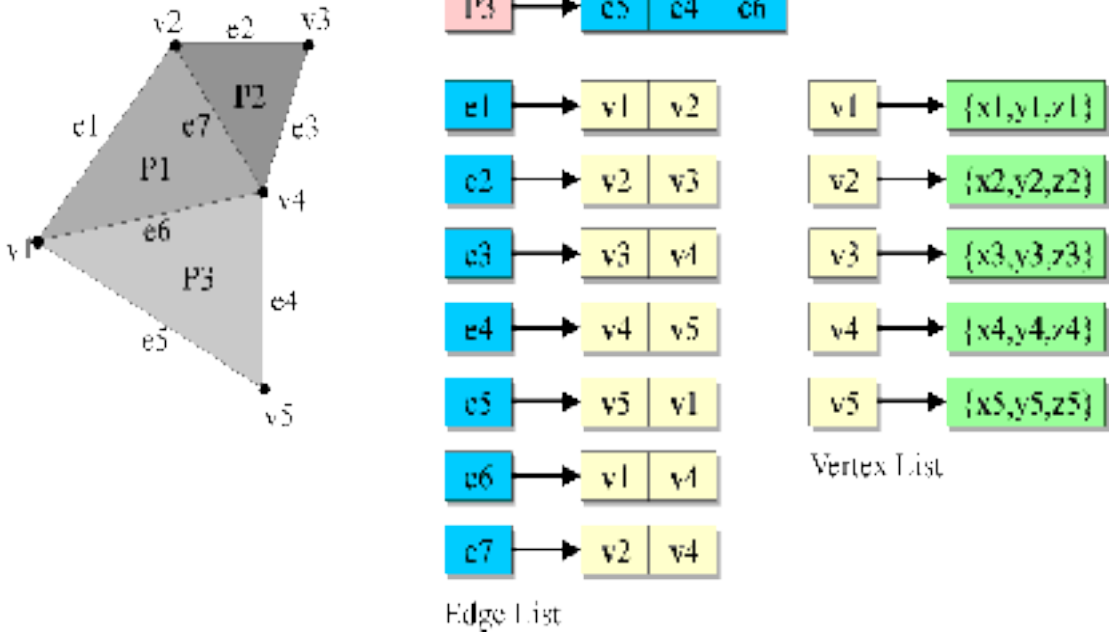
Representing objects with lines

- Start with **Points/Vertices** on the surface of an object
  - Defined as *displacement vectors* $\{ \mathbf{x}, \mathbf{y}, \mathbf{z} \}$ from the origin
- **Edges** are line segments on the surface, defined by pairs of points.
- Closed **polygons** are made up of a number of co-planar edges.



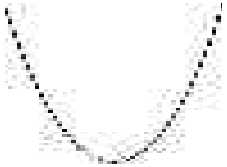
Surface Representations

- Polygon Mesh a.k.a. Wire frame

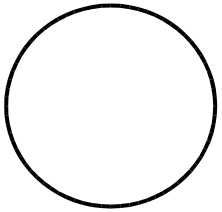


Curves and Splines

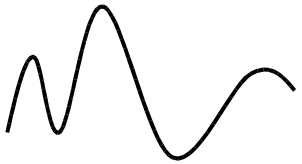
- A larege number of smooth curves can be defined by mathematical functions. However finding a mathematical function that defines every real-world object is difficult



$y = ax^2$

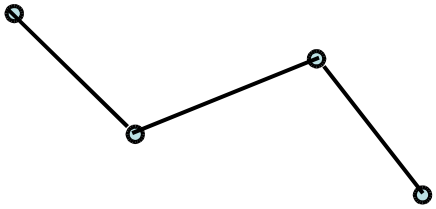


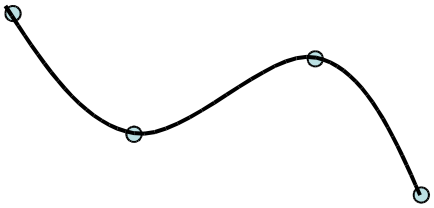
$x^2 + y^2 = r^2$

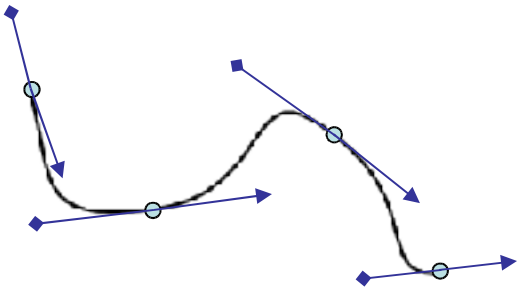


???

- Splines are curves fitted to a set of control points, and are useful for more generic modelling
- Valid in 2D and 3D

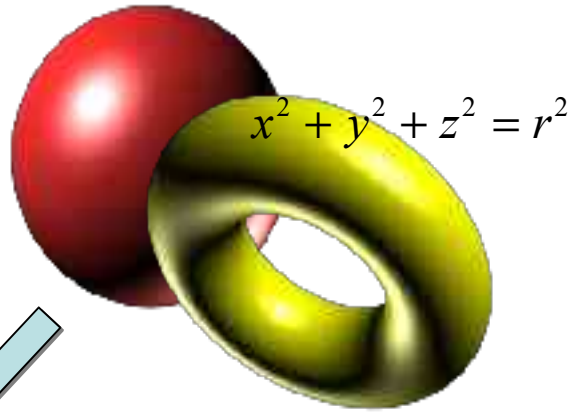






Modelling Techniques

Polygonal

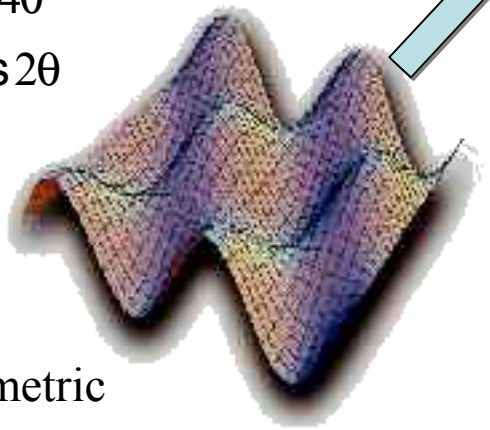


Implicit

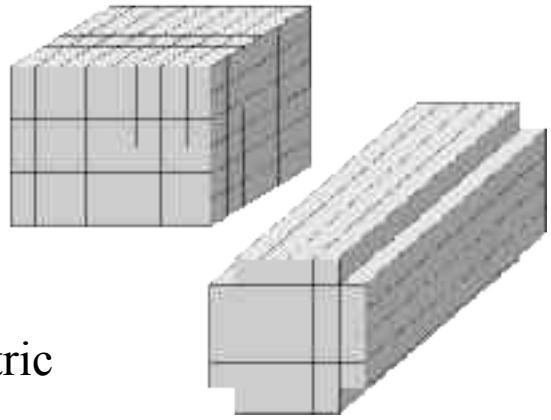
3D Modelling

$x = \sin 4\theta$

$y = \cos 2\theta$



Parametric



Volumetric

Modelling 3D Shape

- Boundary Representation – Surface Models

  vs

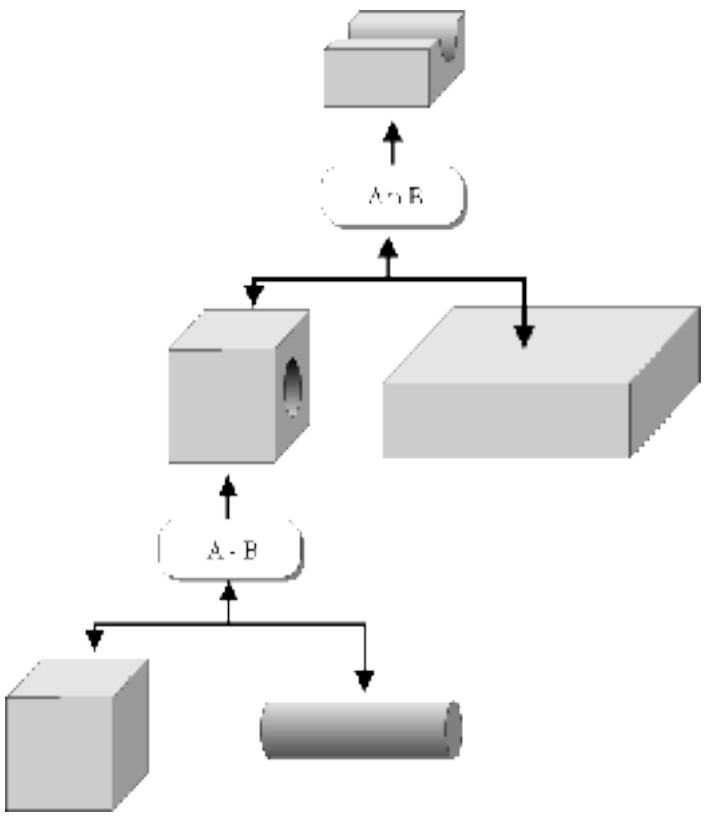
- Spatial Partitioning – Volume Models

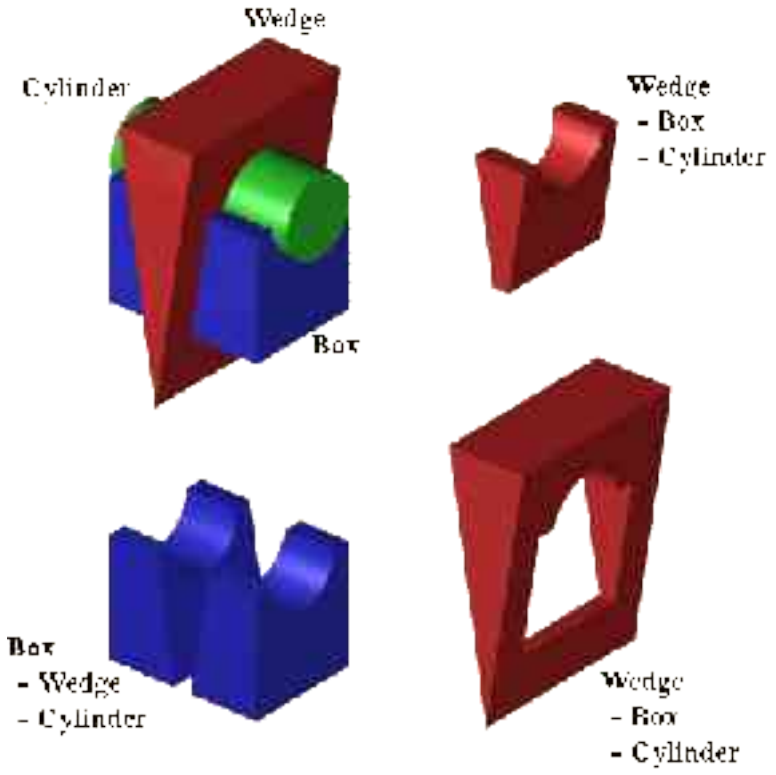
- Mathematical Models

  vs

- Discrete Models

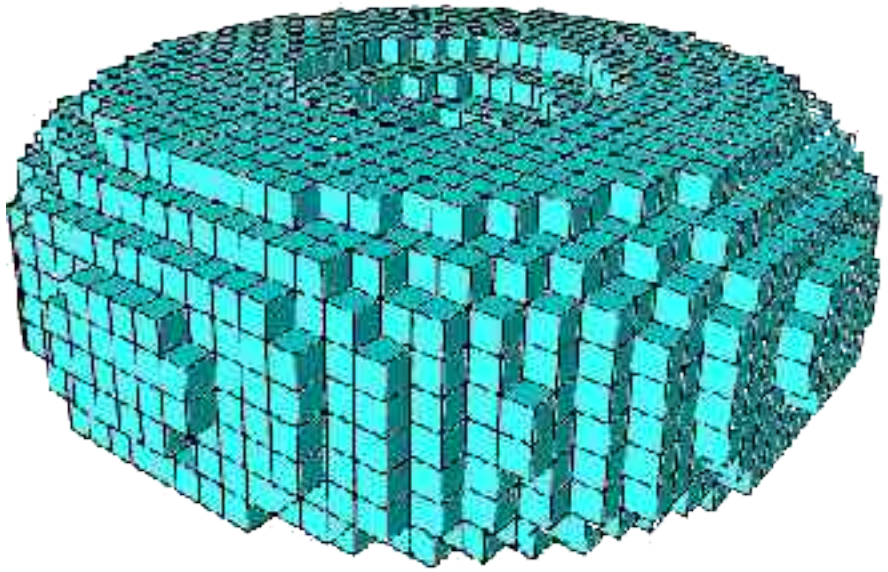
Volumetric Modules





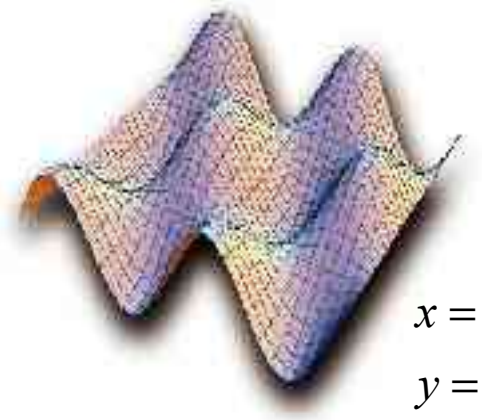
Constructive Solid Geometry

Volumetric Models



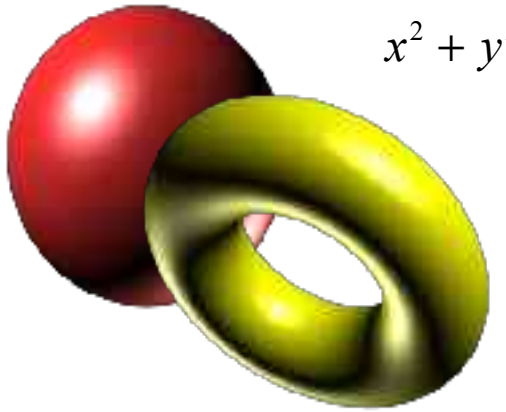
Voxels (Volume elements/Volumetric pixels)

Mathematical Models



$$x = \sin 4t$$
$$y = \cos 2t$$

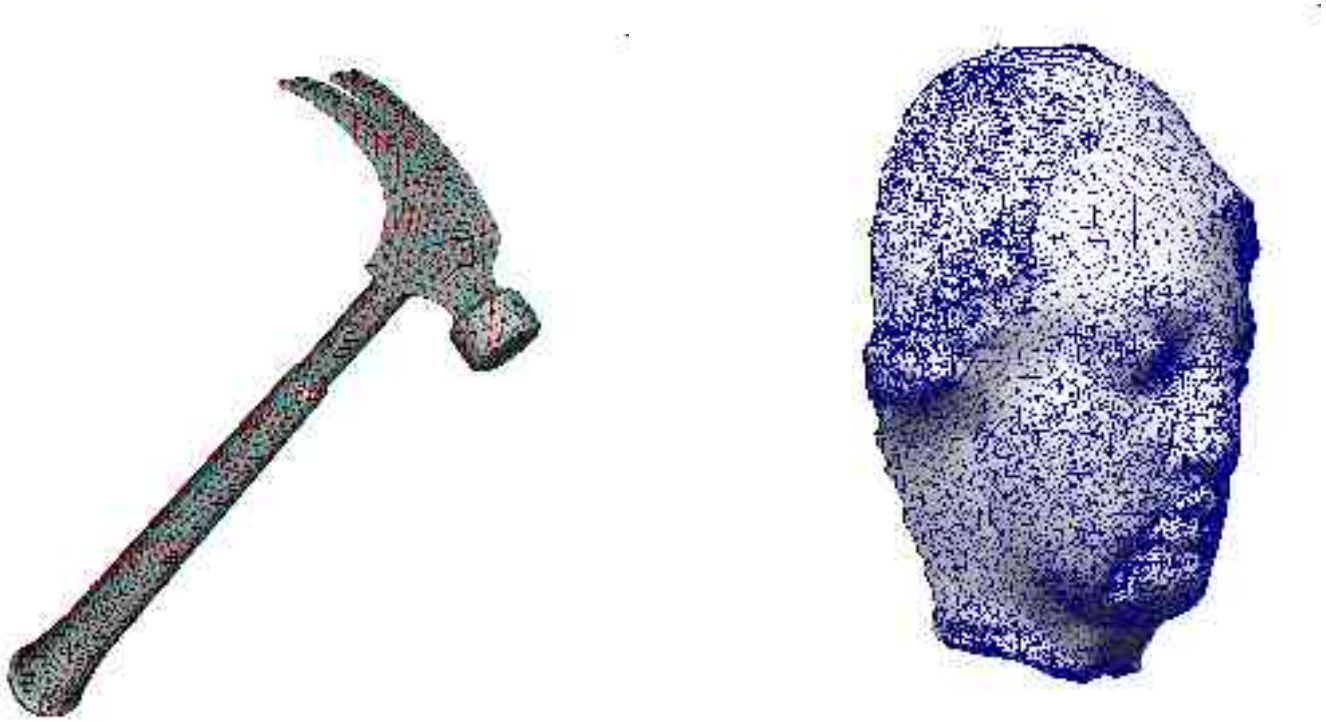
Parametric



$$x^2 + y^2 + z^2 = r^2$$

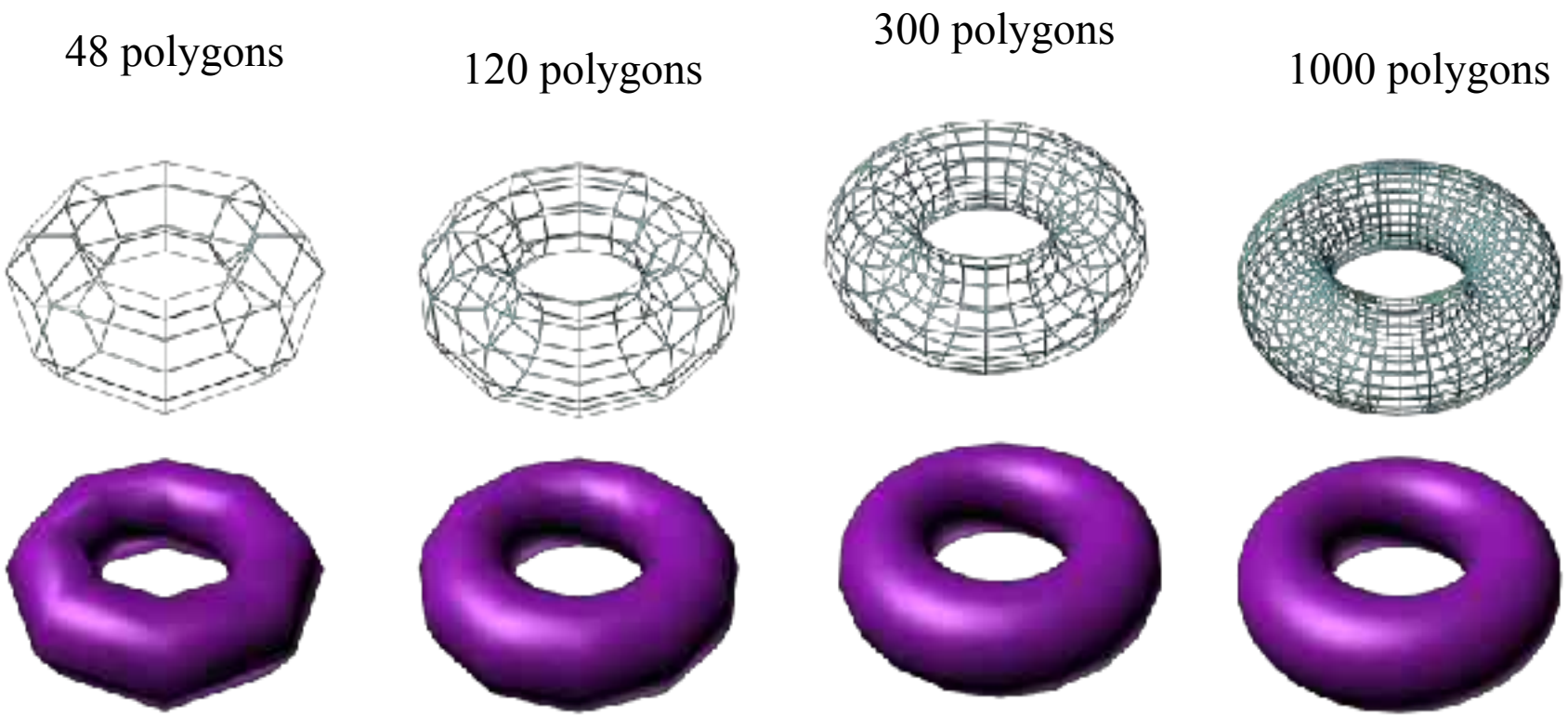
Implicit

Polygon Mesh



- This is the basis for modelling a large range of complex shapes in 3D applications

Discretization



In most cases, mathematical models need to be converted to discrete representations before we use them in scenes. We need to find an appropriate ***resolution*** for this.