# SOME 024:
# Computer Aided Design

E. Rozos

# Introduction to CAD theory part 1

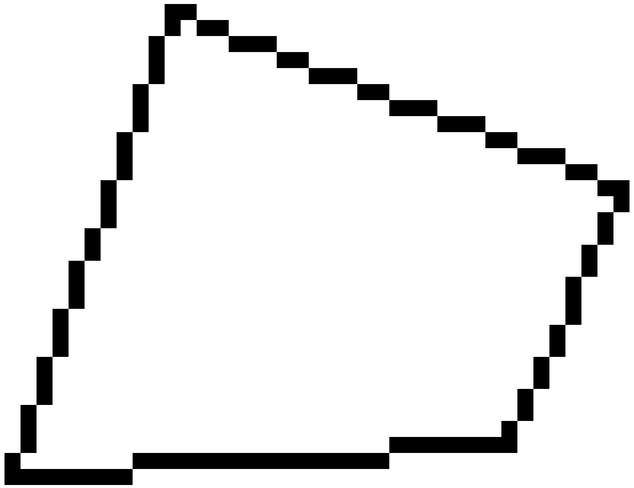
# Lesson structure

- Definitions (raster-vector, topology-geometry, coordinates, projections)
- Basic 2D shapes
- 3D geometric modelling
- Parametric shapes
- Surface modelling

# Definitions-type of graphics

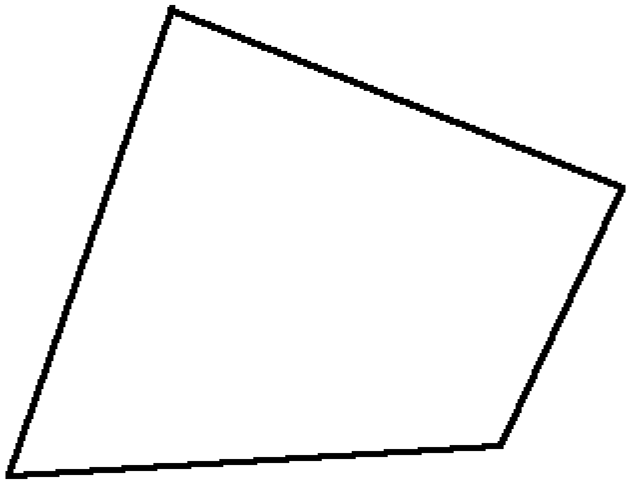
## Raster graphics

The objects are represented with
a set of pixels-cells laid on a grid.



## Vector graphics

The objects are represented
with geometrical primitives.



# Definitions-type of graphics

## Raster graphics

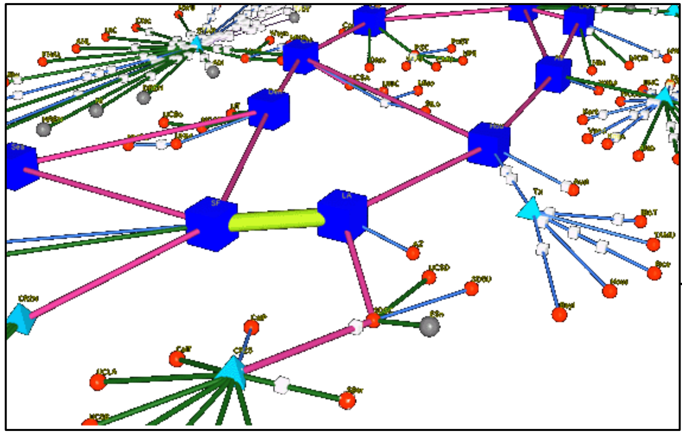
**Applications:** link the spatial data with other kind of information (e.g. height, temperature, mean annual rainfall), spatial related operations (aggregation, interpolation), spatial queries, spatial statistics, 2D numerical models.
**Data storage:** binary with the form of a 2D matrix

## Vector graphics

**Applications:** Geometrical operations (intersections, union), objects are handled as entities, representation accuracy.
**Data storage:** parametrically using analytical forms

# Definitions-topology geometry

**Topology**
The branch of mathematics dealing with the Connectivity of geometric elements not including specific information of those elements.

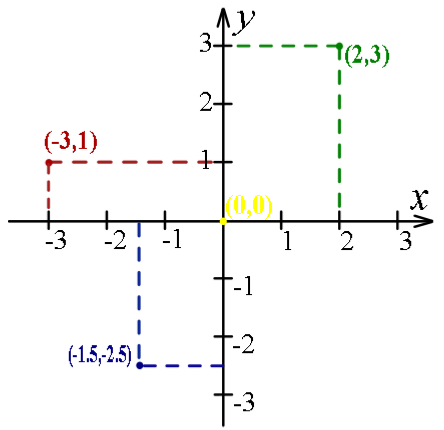


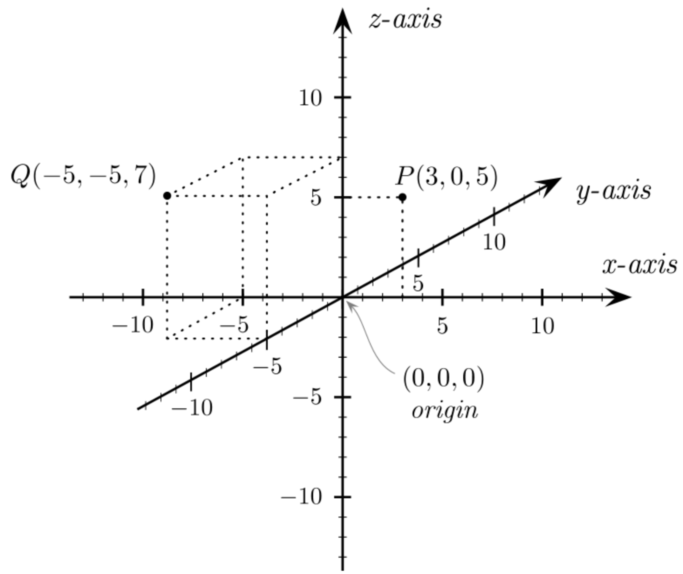
Topological representation of the Internet

**Geometry**
The branch of mathematics dealing with the measurements of lines, angles, surfaces and solids.

# Definitions-coordinate types

**Cartesian coordinates**




2D cartesian coordinates (x,y)

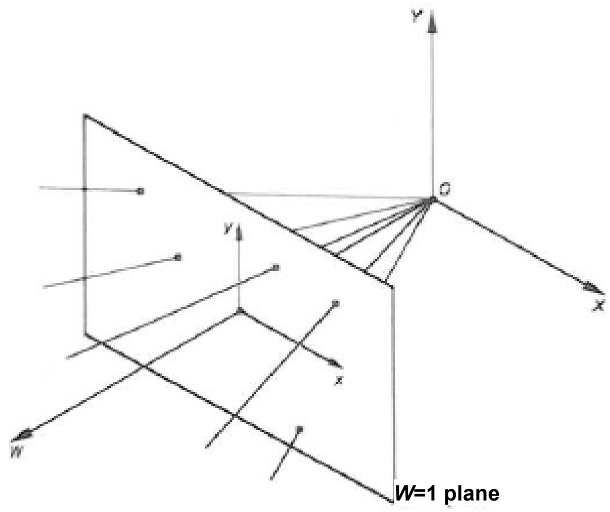
3D cartesian coordinates (x,y,z)

Suitable for spaces described by Euclidian geometry.

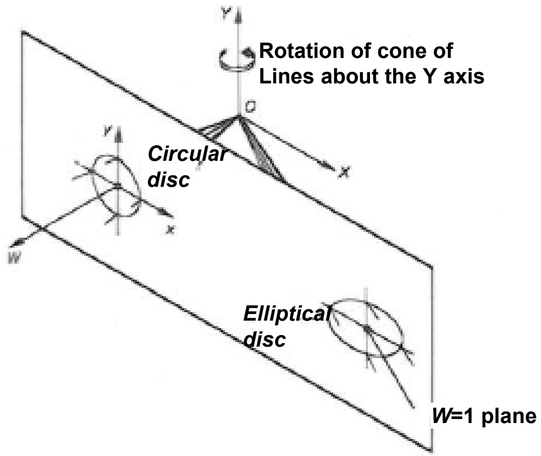
# Definitions-coordinate types

**Homogeneous coordinates**

2D Homogeneous coordinates (x,y,w). Each point on plain W=1 is associated with a line joining it with the origin of the 3D space.




Transformation of a circle disk to elliptical disk.

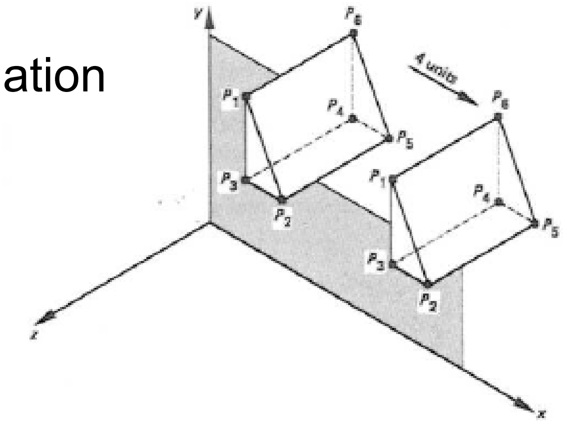


# Definitions-coordinate types

**Transformations**

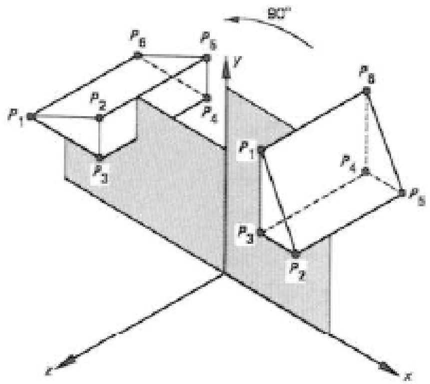
Translation, rotation, reflection, scaling, inversion, shearing.

Translation



| Corner points | Original coordinates | New coordinates |
|---|---|---|
| P1 | (2,4.5,1) | (6,4.5,1) |
| P2 | (3,2.5,1) | (7,2.5,1) |
| P3 | (2,2.5,1) | (6,2.5,1) |
| P4 | (2,2.5,-2) | (6,2.5,-2) |
| P5 | (3,2.5,-2) | (7,2.5,-2) |
| P6 | (2,4.5,-2) | (6,4.5,-2) |

Rotation



| Corner points | Original coordinates | New coordinates |
|---|---|---|
| P1 | (2,4.5,1) | (-4.5,2,1) |
| P2 | (3,2.5,1) | (-2.5,3,1) |
| P3 | (2,2.5,1) | (-2.5,2,1) |
| P4 | (2,2.5,-2) | (-2.5,2,-2) |
| P5 | (3,2.5,-2) | (-2.5,3,-2) |
| P6 | (2,4.5,-2) | (-4.5,2,-2) |

# Definitions-coordinate types

**Mathematic implementation of the transformations using 3D homogeneous coordinates.**

Rotation around x $\mathbf{R}_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Translation $\mathbf{T} = \begin{vmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Rotation around y $\mathbf{R}_y = \begin{vmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Scaling $\mathbf{S} = \begin{vmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Rotation around z $\mathbf{R}_z = \begin{vmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Apply translation, rotation around x, rotation around y to $\mathbf{P_1}$:

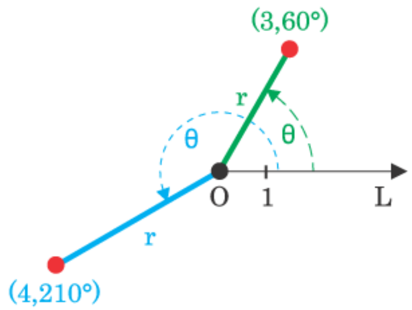
$\mathbf{P_2} = \mathbf{R}_y \mathbf{R}_x \mathbf{T}\ \mathbf{P_1}$

Undo:

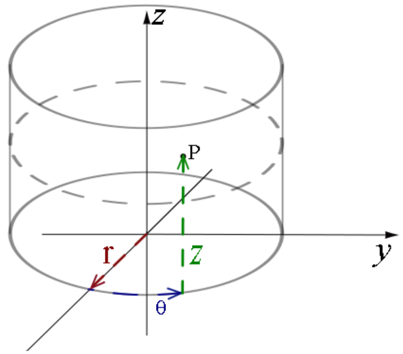
$\mathbf{P_1} = \mathbf{T^{-1}}\ \mathbf{R_x^{-1}}\ \mathbf{R_y^{-1}}\ \mathbf{P_2}$

# Definitions-coordinate types

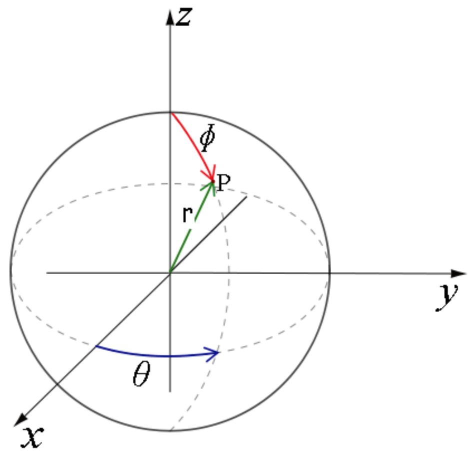
**2D polar coordinates (*r*, *θ*)**
$x = r\cos\theta,\ y = r\sin\theta$
Applications in problems with circular symmetry (point heat source in an infinite plate)




**3D cylindrical coordinates (r, θ, Z)**
$x = r\cos\theta,\ y = r\sin\theta,\ z = Z$

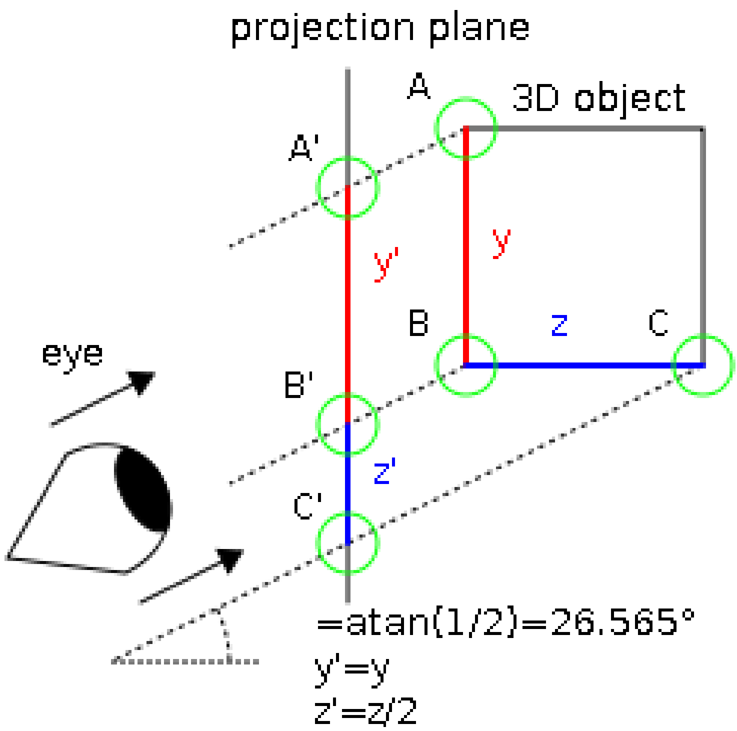


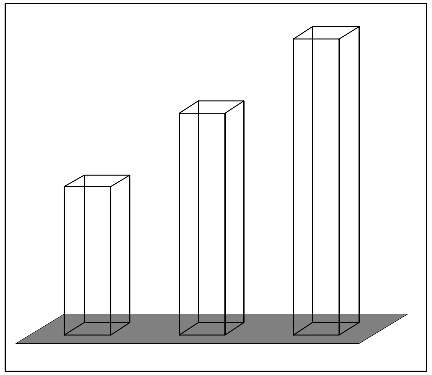

**3D spherical coordinates (r, θ, φ)**
$x = r\sin\varphi\cos\theta,\ y = r\sin\varphi\sin\theta,\ z = r\cos\varphi$
Applied in non Euclidean spaces.



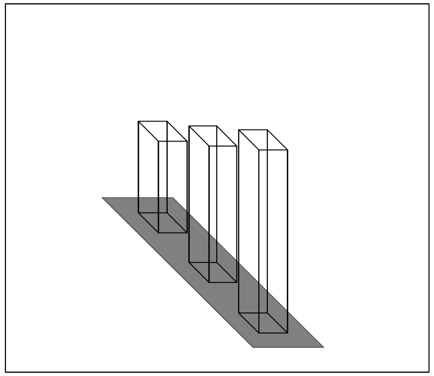
# Definitions-projection types

## Parallel projection



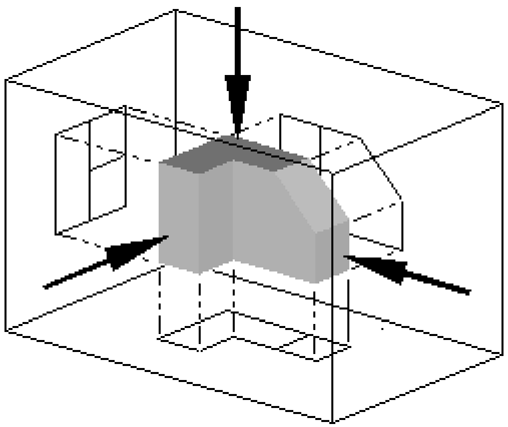
Oblique, only sizes parallel to projection plane are preserved



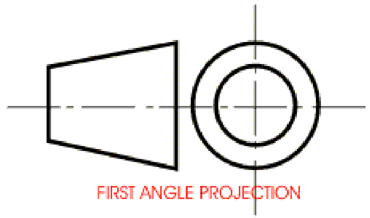
Isometric, axes equally foreshortened

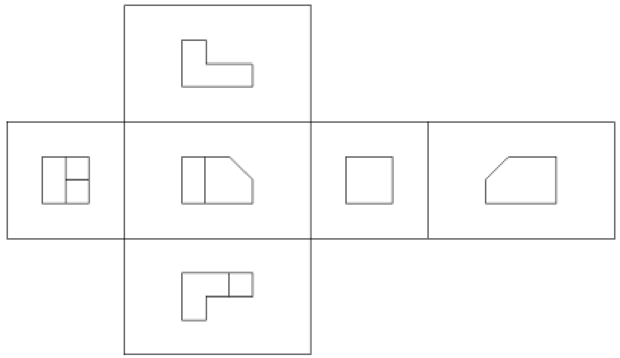
# Definitions-projection types

## Orthographic Multi-views projection

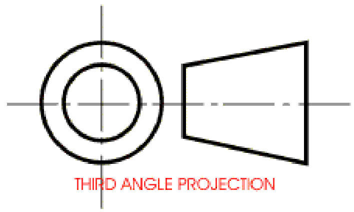


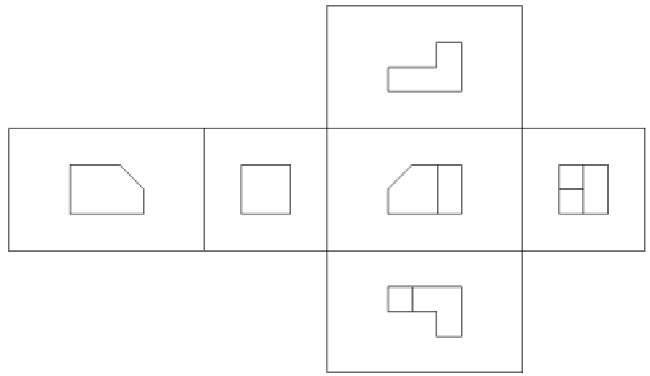
1st angle projection
(viewer inside box)



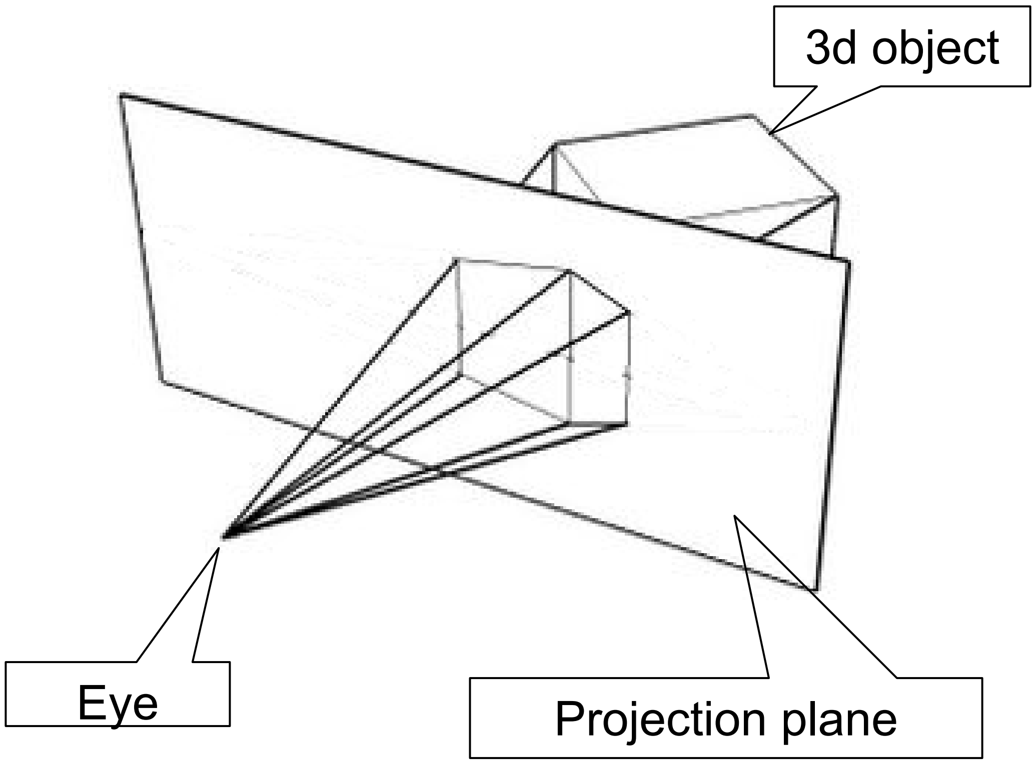
3rd angle projection
(viewer outside box)

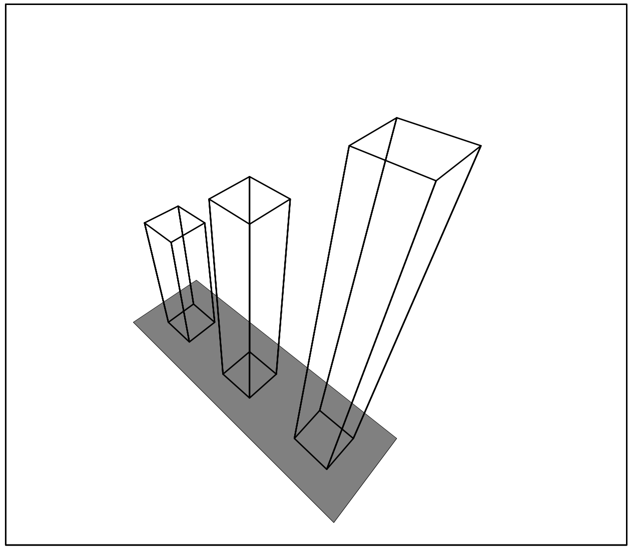


# Definitions-projection types

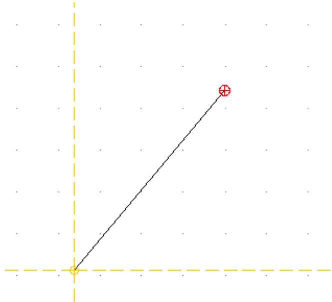
## Perspective projection



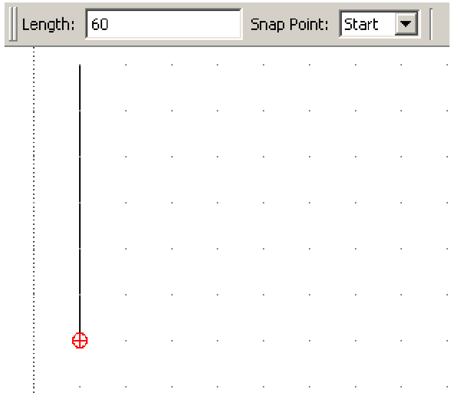



Easy to conceive the 3D form, shape and size to scale are not preserved

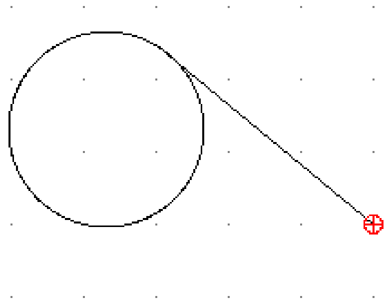
# Basic 2D shapes-line segment



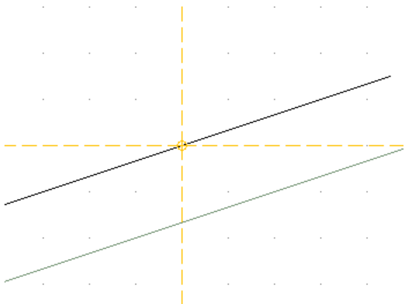
Two points



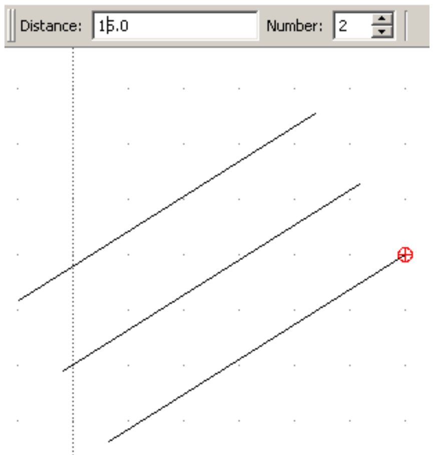

Parallel to an axis



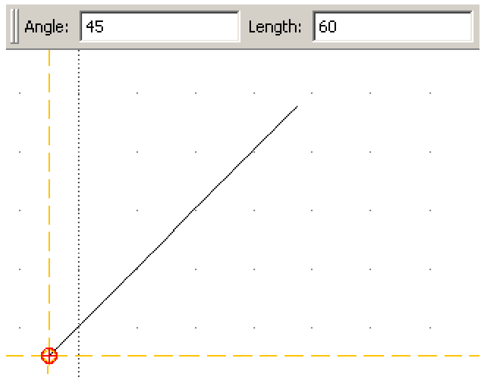
Tangent to a circle



Parallel to lin. seg.
through specified point

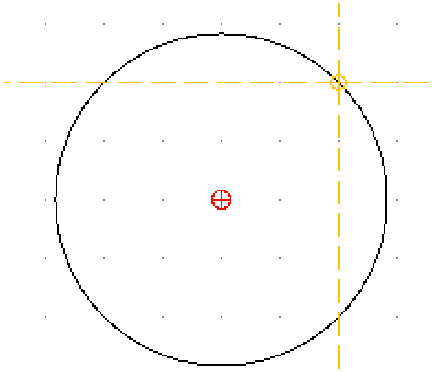



Parallel to a lin. seg.
at specified distance

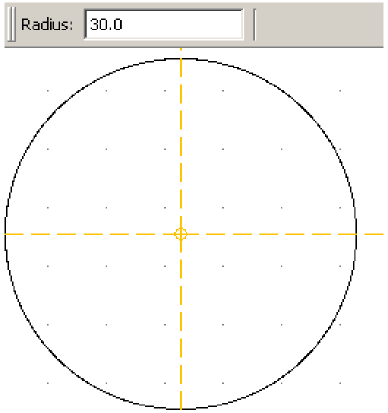



From a point with
specified length and angle

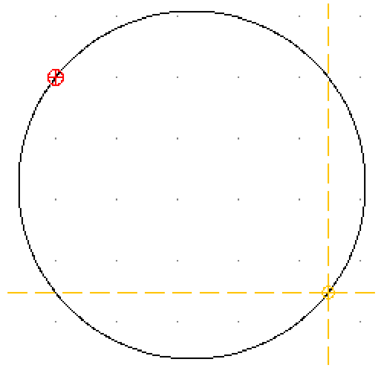
# Basic 2D shapes-circle, arc



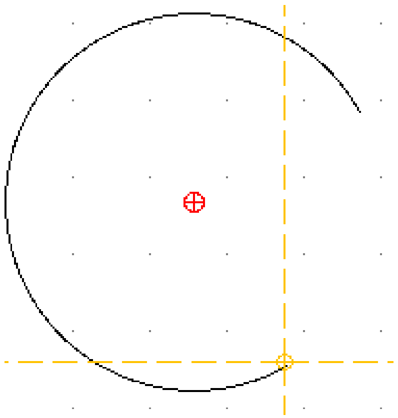
Centre, peripheral point



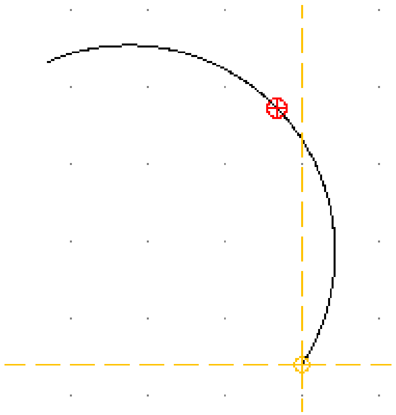

Centre, radius



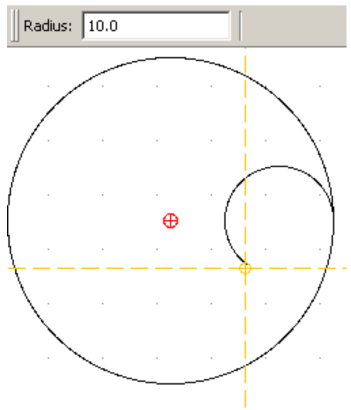
Two points



Centre, start, end point



Three points




Tangent to figure given radius + end point.

# Wire frame modelling

A wire frame representation of a 3D object consist of a finite set of points (vertices) and connecting edges. The representation is a trade of between fidelity and easiness of representation.

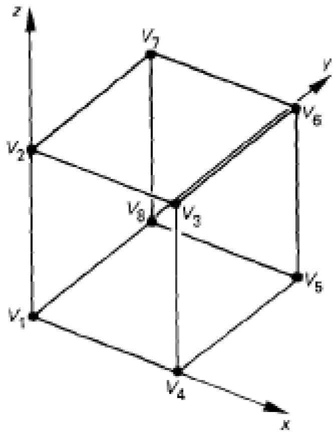
The wire frame representation consist of two types of information concerning:

1. The geometric data (coordinate positions of the vertices)
2. The topological data (relate pairs of vertices together as edges)

# 3D geometric modelling

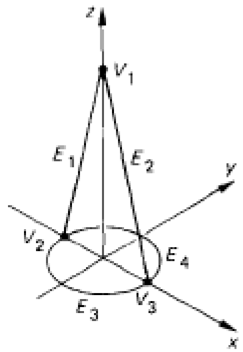
| Vertex list | Edge list | | Face list |
|---|---|---|---|
| $V_1$ (0,0,0) | $E_1$ $\langle V_1, V_2 \rangle$ | $E_7$ $\langle V_7, V_8 \rangle$ | $F_1$ $\langle E_1, E_2, E_3, E_{11} \rangle$ |
| $V_2$ (0,0,1) | $E_2$ $\langle V_2, V_3 \rangle$ | $E_8$ $\langle V_8, V_1 \rangle$ | $F_2$ $\langle E_7, E_6, E_5, E_{12} \rangle$ |
| $V_3$ (1,0,1) | $E_3$ $\langle V_3, V_4 \rangle$ | $E_9$ $\langle V_2, V_7 \rangle$ | $F_3$ $\langle E_1, E_9, E_7, E_8 \rangle$ |
| $V_4$ (1,0,0) | $E_4$ $\langle V_4, V_5 \rangle$ | $E_{10}$ $\langle V_3, V_6 \rangle$ | $F_4$ $\langle E_3, E_{10}, E_5, E_4 \rangle$ |
| $V_5$ (1,1,0) | $E_5$ $\langle V_5, V_6 \rangle$ | $E_{11}$ $\langle V_1, V_4 \rangle$ | $F_5$ $\langle E_8, E_{12}, E_4, E_{11} \rangle$ |
| $V_6$ (1,1,1) | $E_6$ $\langle V_6, V_7 \rangle$ | $E_{12}$ $\langle V_8, V_5 \rangle$ | $F_6$ $\langle E_9, E_6, E_{10}, E_2 \rangle$ |
| $V_7$ (0,1,1) | | | |
| $V_8$ (0,1,0) | | | |

Geometric data                Topological data




The simplest representation of a cone with only 3 vertices, 1 vertex for the apex, 2 vertices defining the diameter of the base. The fidelity may be improved by adding more vertices to the base.

| Vertex list | Edge list | Edge type |
|---|---|---|
| $V_1$ (0,0,3) | $E_1$ $\langle V_1, V_2 \rangle$ | Linear |
| $V_2$ (−1,0,0) | $E_2$ $\langle V_1, V_3 \rangle$ | Linear |
| $V_3$ (1,0,0) | $E_3$ $\langle V_2, V_3 \rangle$ | Semi-circular |
| | $E_4$ $\langle V_2, V_3 \rangle$ | Semi-circular |

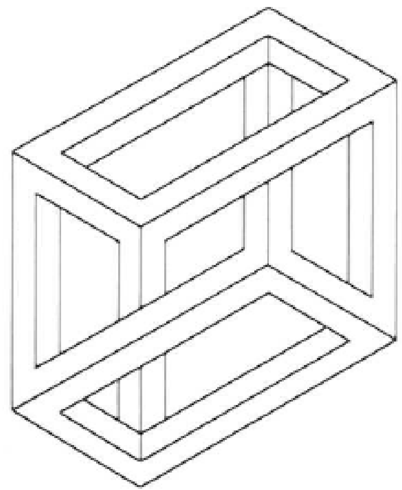


# 3D geometric modelling

**Low level constrains**

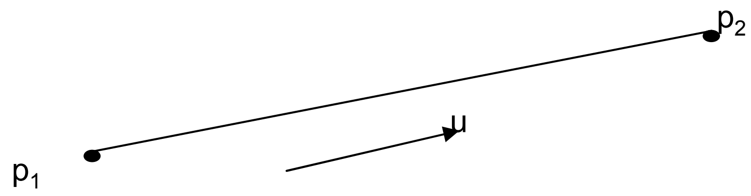
- Each vertex described by 3 coordinate values
- Each edge associated with only 2 vertices
- Edges must form close loops
- Edges should not intersect

**Object validity is not guaranteed**

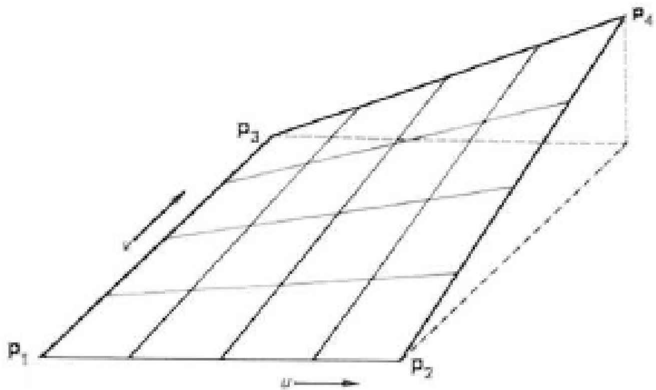


# Parametric curve and surface geometry

Parametric form of a straight line with 2 control points ($\mathbf{p}_1$, $\mathbf{p}_2$)
$\mathbf{r}(u) = (1-u)\,\mathbf{p}_1 + u\,\mathbf{p}_2$



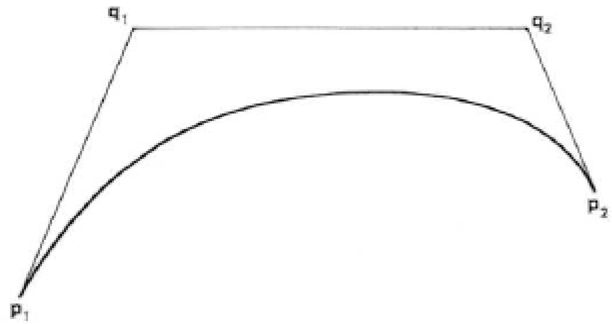

Parametric bilinear surface with 4 control points ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$)
$\mathbf{r}(u,v) = (1-v)\,(1-u)\,\mathbf{p}_1 + (1-v)\,u\,\mathbf{p}_2 + v\,(1-u)\,\mathbf{p}_3 + v\,u\,\mathbf{p}_4$



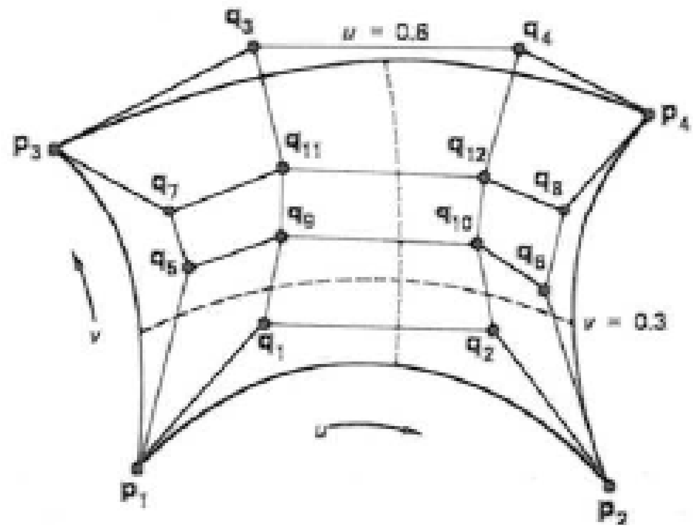
# Parametric curve and surface geometry

Bezier cubic parametric curve 4 control points ($\mathbf{p}_1$, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{p}_2$)

$\mathbf{r}(u) = (1-u)^3\ \mathbf{p}_1 + 3u(1-u)^2\ \mathbf{q}_1 + 3u^2\ (1-u)\ \mathbf{q}_2 + u^3$

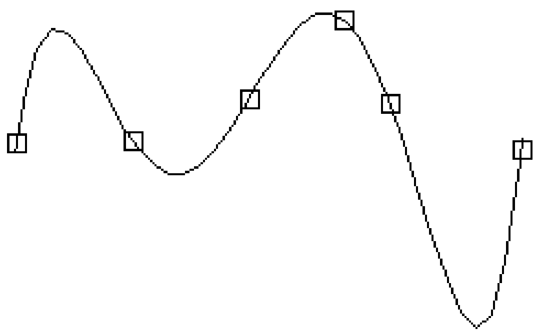



Bezier bicubic parametric patch with 16 control points ($\mathbf{p}_1,\ldots,\ \mathbf{p}_4$, $\mathbf{q}_1,\ldots,\ \mathbf{q}_{12}$)

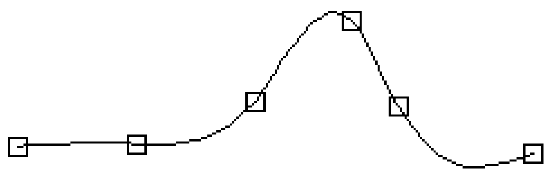


# Parametric curve and surface geometry

## Splines



For n given points there exists a unique polynomial of degree n-1 or less which passes through these points.



A spline is a piecewise polynomial such that the function (G0), its derivative (G1) and its second derivative (G2) are continuous at the interpolation nodes.

# Parametric curve and surface geometry

## Non uniform rational Basis-Splines (NURBS)

A NURBS curve is defined by:

- its order i.e. the maximum degree of the polynomial basis functions.
- a set of control points i.e. the points from which the curve passes.
- a knot vector that determines where and how the control points affect the NURBS curve; the number of knots is always equal to the number of control points plus curve degree plus one.

# Parametric curve and surface geometry

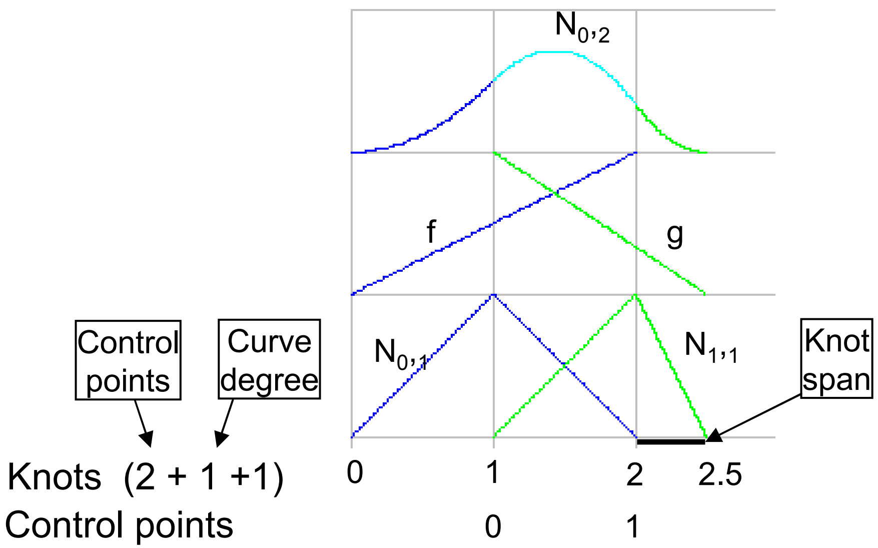
## Non uniform rational Basis-Splines (NURBS)

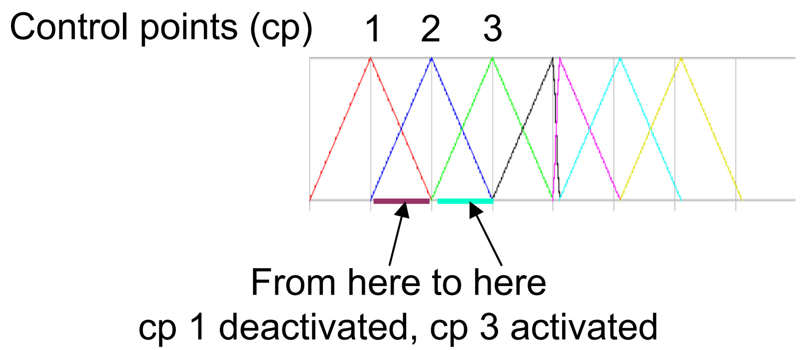
Basis functions used in NURBS.

$N_{i,n} = f_{i,n}N_{i,n\text{-}1} + g_{i+1,n}N_{i+1,n-1}$

*where i* the ith control point, *n* degree of basis function, f and g

weigthing functions depending on knots.



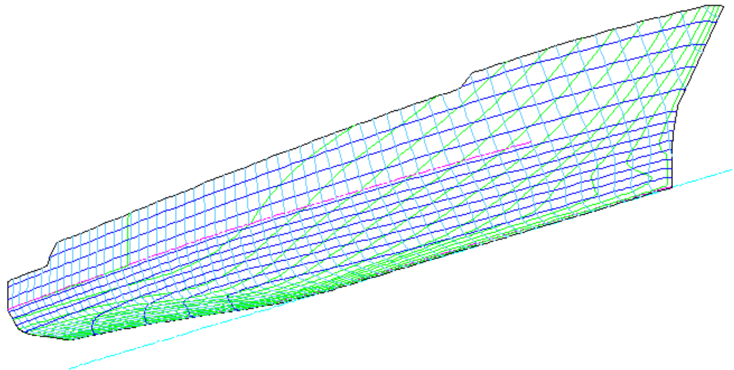
# Surface modelling

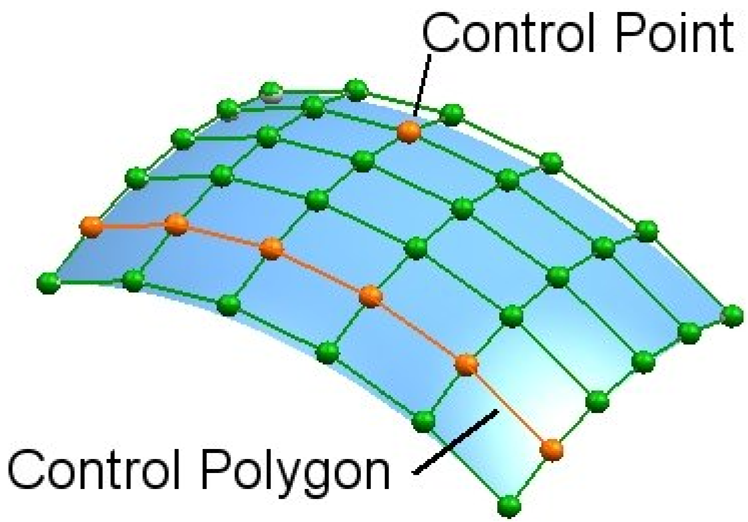
Test the aesthetic and functionality of surfaces in components such as car bodies or simulated the fluid dynamics (turbine blades, boat hulls, etc.)

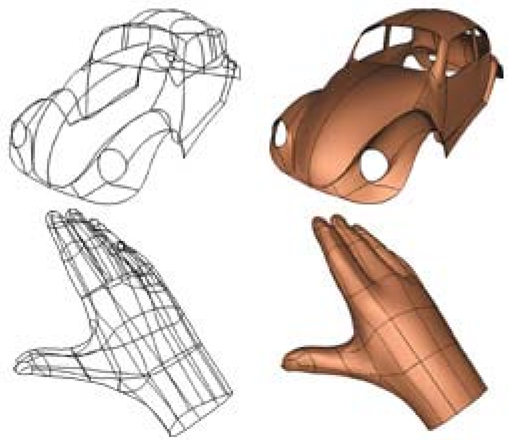
Creation methods

1. The user supplies an array of control points. The surface modeller fits a Bezier bicubic parametric patch to the control features.
2. The 3D surface is created either by sweeping a curve along a guide rail or by lofting a mesh of curves.

# Surface modelling





Lofting