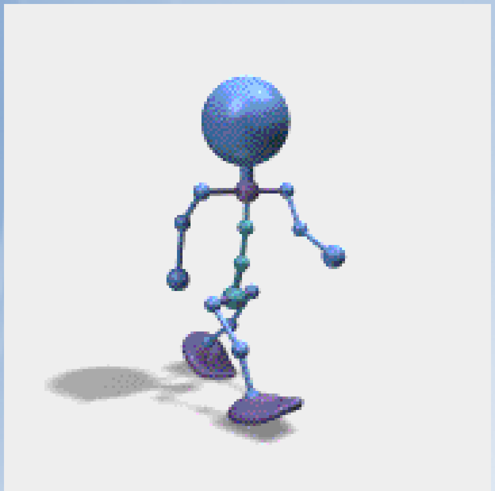
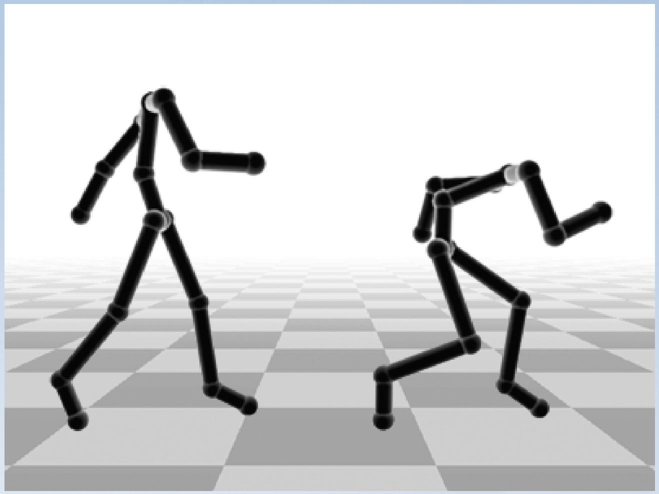



Advanced CAD
2. Geometric Modeling,
3. Transformations

Hikmet Kocabaş, Prof., PhD.
Istanbul Technical University

Lectures, Outline of the course

1 Advanced CAD Technologies, Hardwares, Softwares
2 **Geometric Modeling, 2D Drawing**
3 **Transformations, 3D**
4 Parametric Curves
5 Splines, NURBS
6 Parametric Surfaces
7 Solid Modeling
8 API programming

Why Study Geometric Modeling

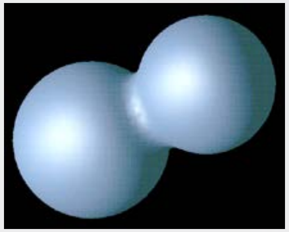
The knowledge of the geometric modeling entities increase your productivity.

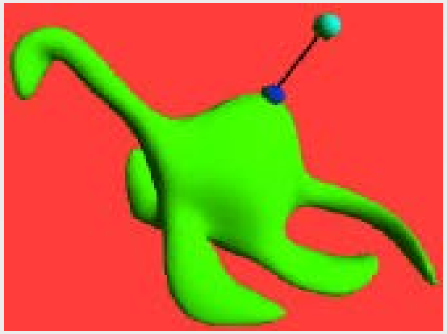
Understand how the math presentation of various entities relates to a user interface.

Understand what is impossible and which way can be more efficient when creating or modifying an entity. Control the shape of an existing object in design.

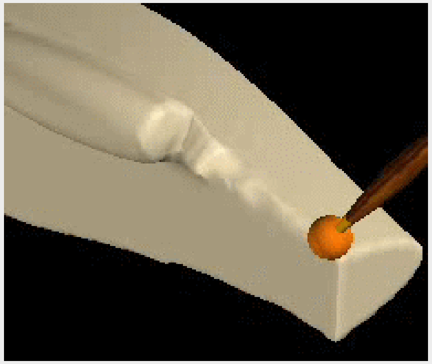
The storage, computation and transformation of objects. Calculate the intersections and physical properties of objects.

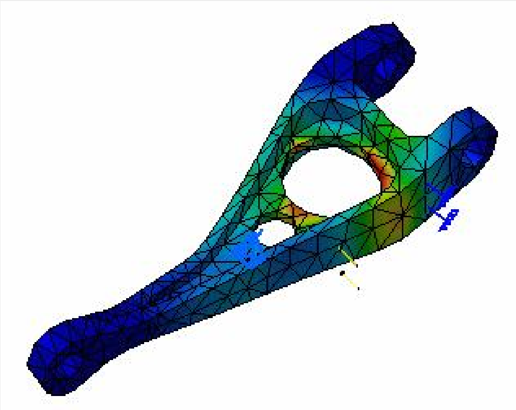
Geometric Modeling is important

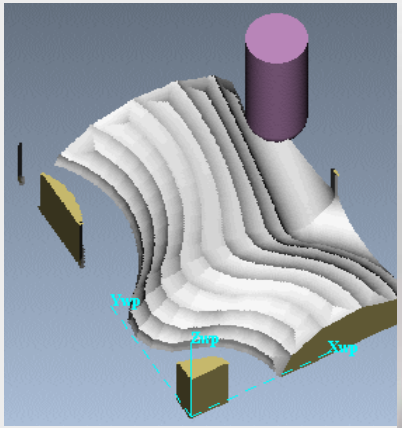




- to meet certain geometric requirements
- such as slopes and/or curvatures in model
- interpretation of unexpected results
- evaluations, simulations of CAD/CAM systems cutting
- use of the tools in particular (robotic) applications
- creation of new attributes
- modify the obtained models

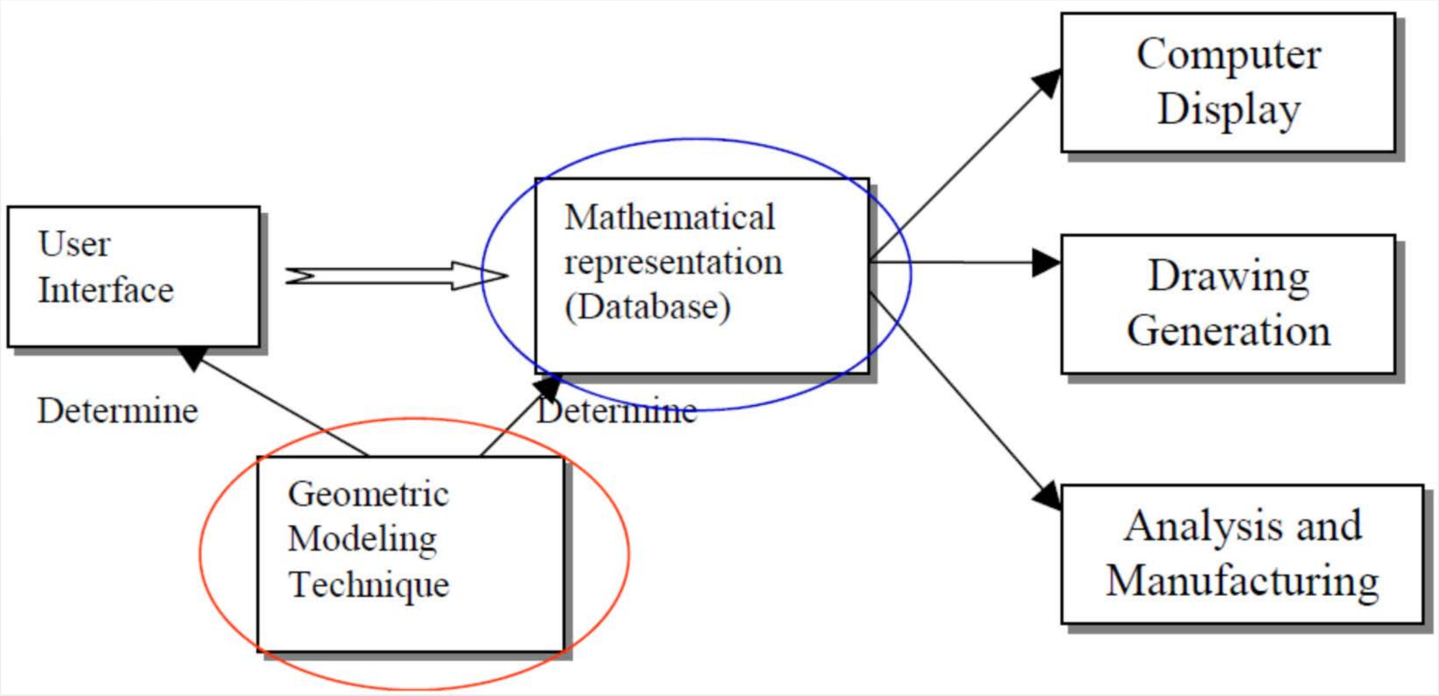




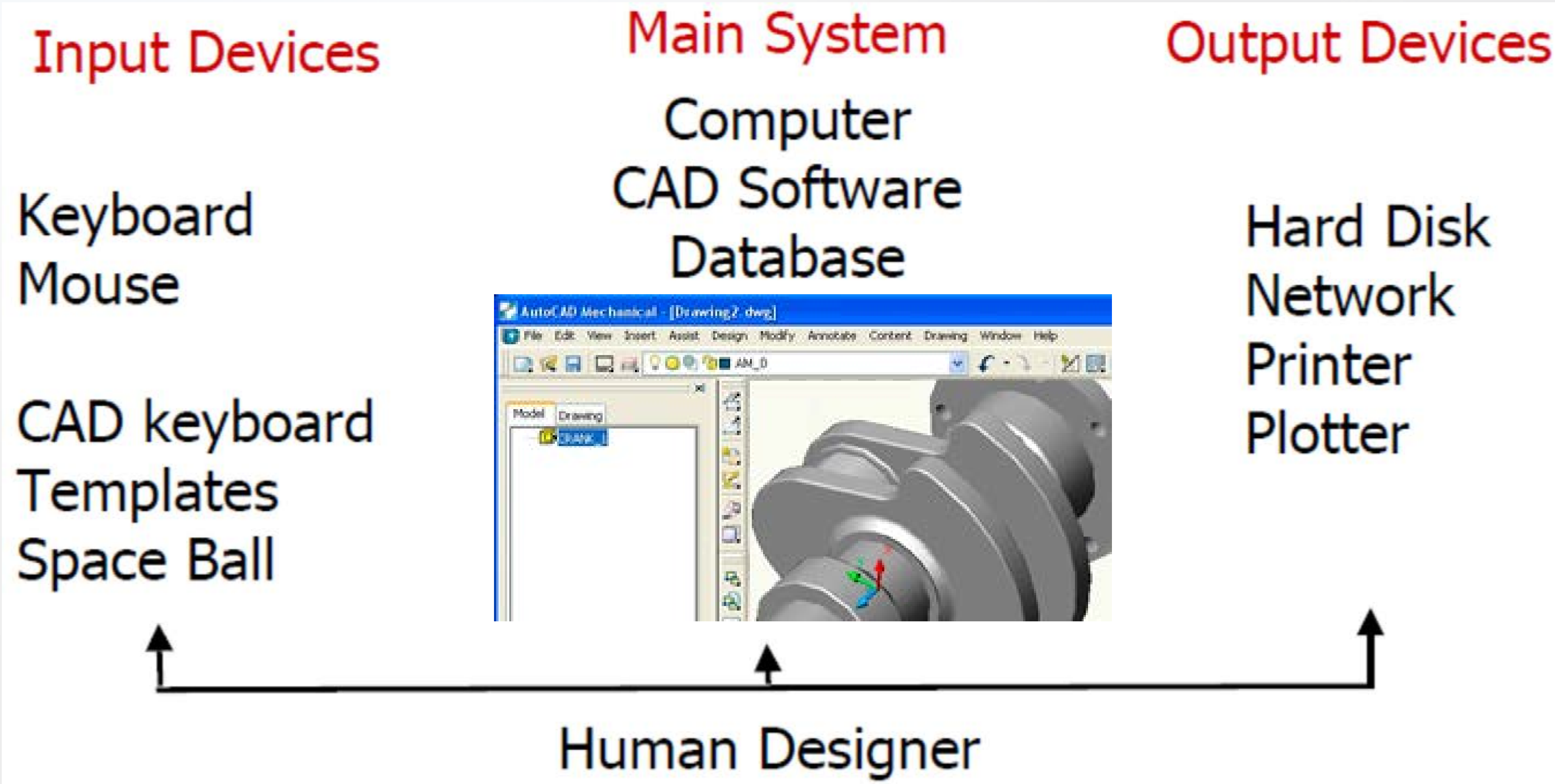


Geometric Modeling in CAD

Geometric modeling is only a means not the goal in engineering. Engineering analysis needs product geometry; the degree of detail depends on the analysis procedure that uses the geometry.



Basic Elements of a CAD System



Fundamental Features

- **Geometry**: Position, direction, length, area, normal, tangent, etc.
- **Interaction**: Size, continuity, collision, intersection
- **Topology**
- **Differential properties**: Curvature, arc-length
- **Physical attributes**
- Computer representation & data structure
- Others…

Professional CAD/CAE/CAM products

Unigraphics (UGS), **NX** (EDS)
I-DEAS (SDRC)
Pro/Engineer, **Pro/Mechanica**, Pro/E, Creo (PTC)
AutoCAD (AutoDesk, Inventor)
ANSYS (ANSYS Inc.)
CATIA, Delmia, **SolidWorks** (Dassault Systemes - IBM)
Nastran, Patran (MacNeal-Schwendler)
...
SurfCam, Solid Edge (EDS), MicroStation, Intergraph,
CADKey, DesignCAD, ThinkDesign,
3DStudio MAX, Rhinoceros, ...

AutoCAD

A world's leading PC-based 3D mechanical design package, from AutoDesk Inc.

Used to be the primary PC drafting package (dealer, PC)

The world's most popular CAD software due to its lower cost and PC platform

New features:

- ACIS 3.0 Advanced Solid Modeling Engine
- NURBS Surface Modeling
- Robust Assembly Modeling and Automated Associative Drafting

Flexible programming tools, **AutoLISP**, **ADS** and **ARX**

Integrated CAD/CAM Tools

ANSYS (from ANSYS Inc.)

- A growth leader in CAE and integrated design analysis and optimization (DAO) software
- Covering solid mechanics, kinematics, dynamics, and multi-physics (CFD, EMAG, HT, Acoustics)
- Interfacing with key CAD systems

NASTRAN (from MacNeal-Schwendler): **PATRAN** provides an open flexible MCAE environment for multidisciplinary design analysis.

Pro/MECHANICA (integrated with Pro/E)

Integrated CAD/CAM Tools

SURFCAM (from Surfware Inc. CA)

- An outgrowth of the Diehl family's machine shop
- A system for generating 2~5- axis milling, turning, drilling, and wire EDM.
- Toolpath verification (MachineWorks Ltd.)

Rhinoceros (NURBS modeling)

– Industrial, marine, and jewelry designs; cad/cam; rapid prototyping; and reverse engineering

Applications of CAD

Geometric modeling, visual computing

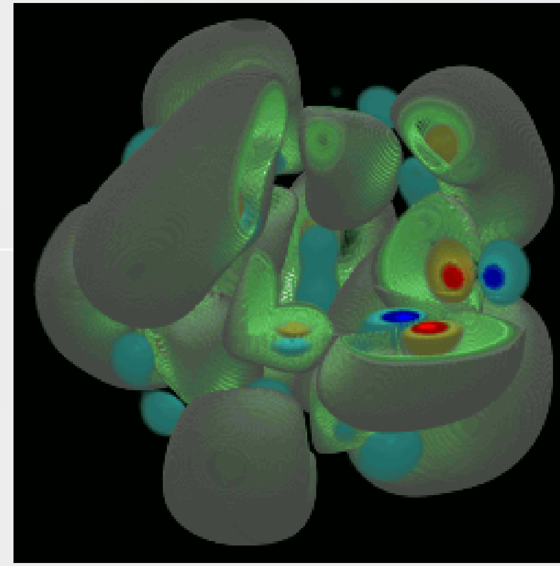
- Computer graphics

Visualization, animation, virtual reality

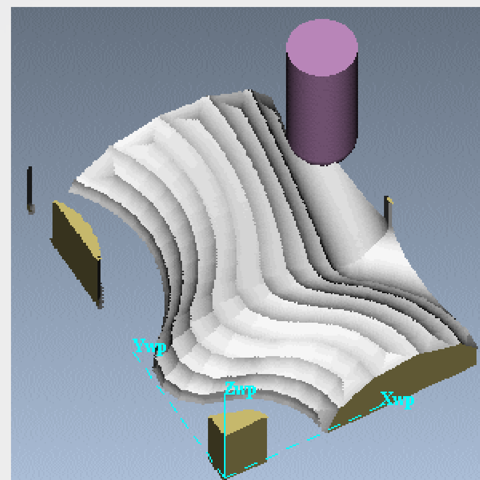
- CAD/CAM
- Virtual Prototyping

Engineering, manufacturing

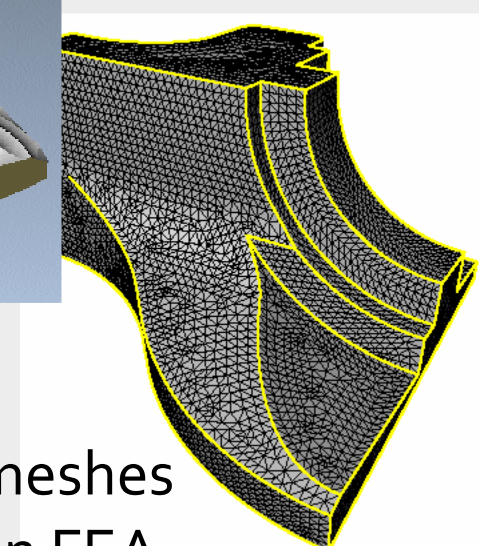
- Computer vision
- Mesh generation
- Physical simulation
- Design optimization
- Reverse engineering, Prototyping



visualization

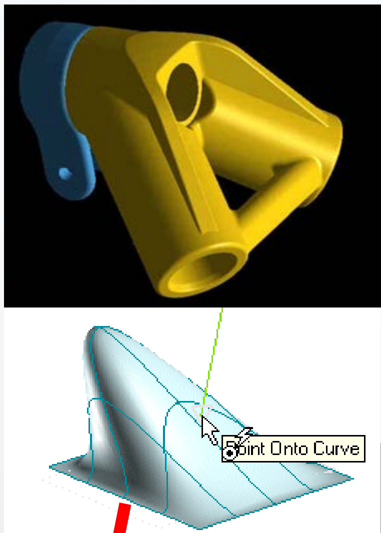


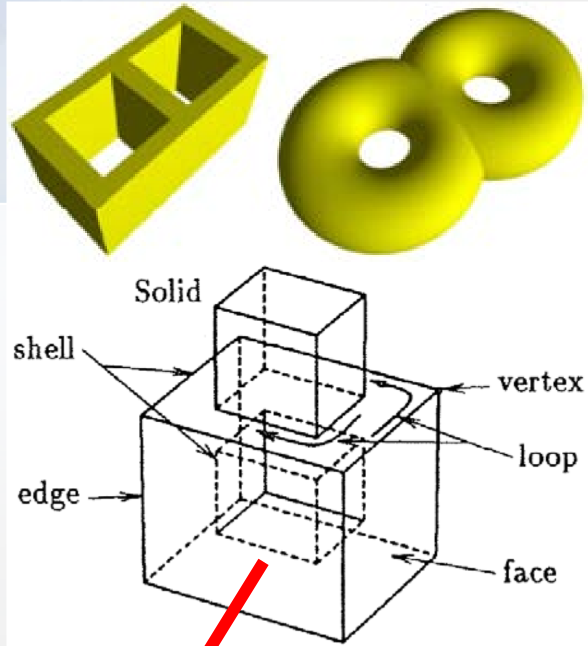
tool path

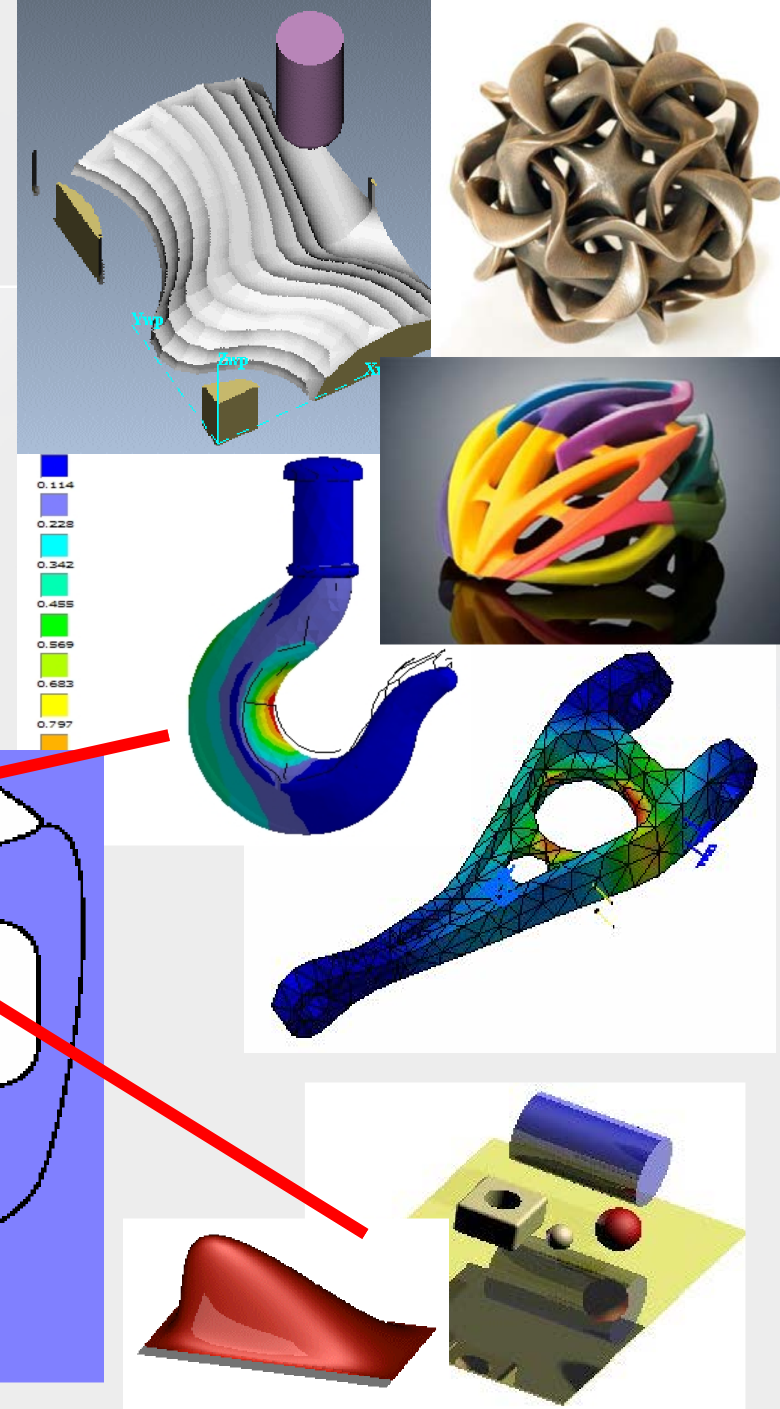


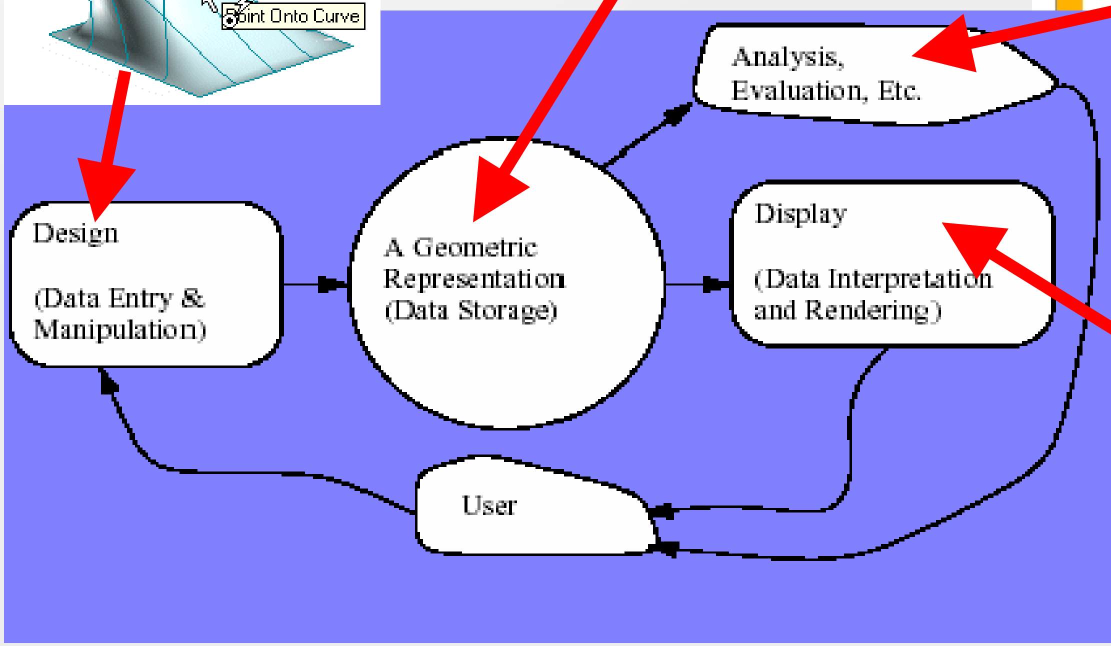
meshes in FEA

CAD Software



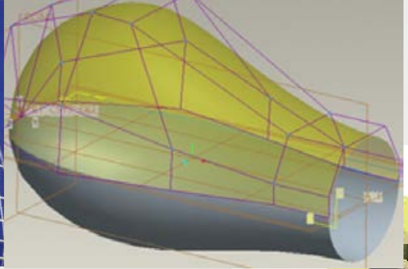
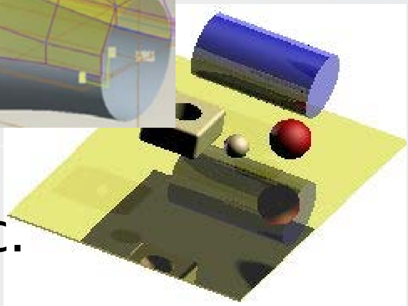
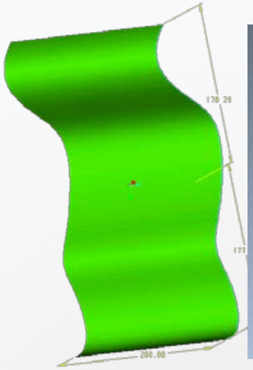
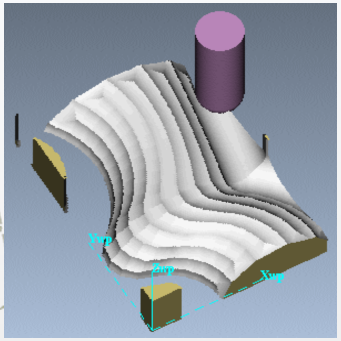
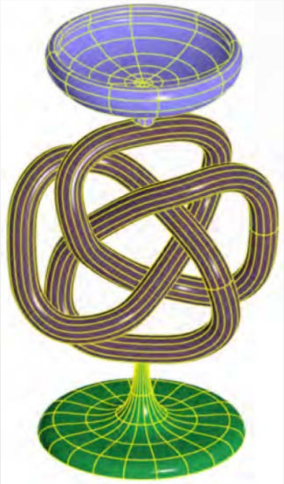




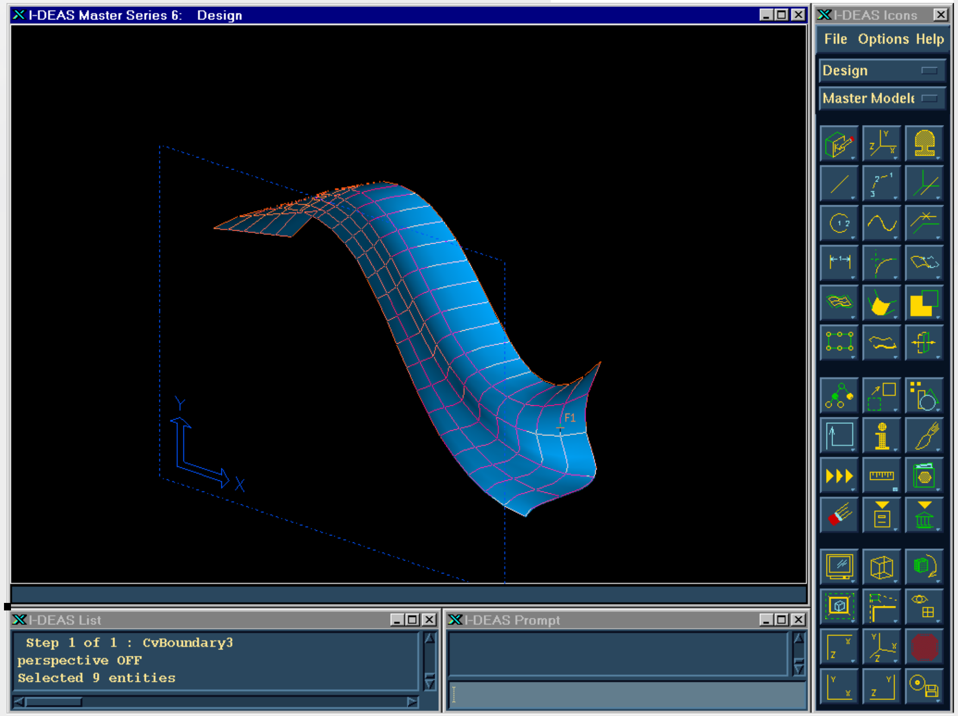


Surface Modeling

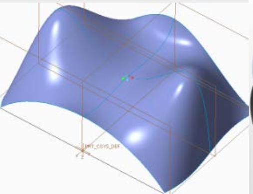
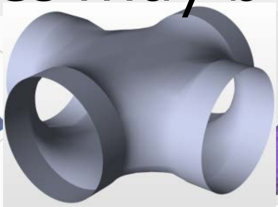
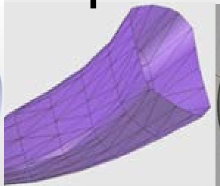
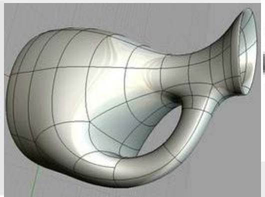
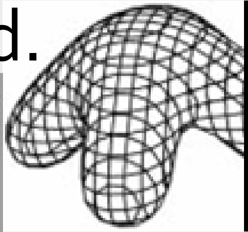
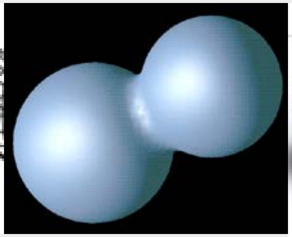
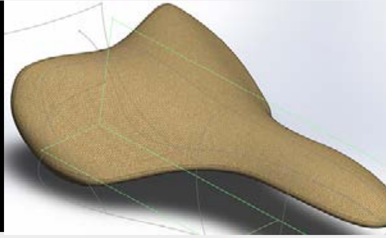
- Models 2D surfaces in 3D space
- All points on surface are defined
- useful for machining, 3d printing, visualization, etc.

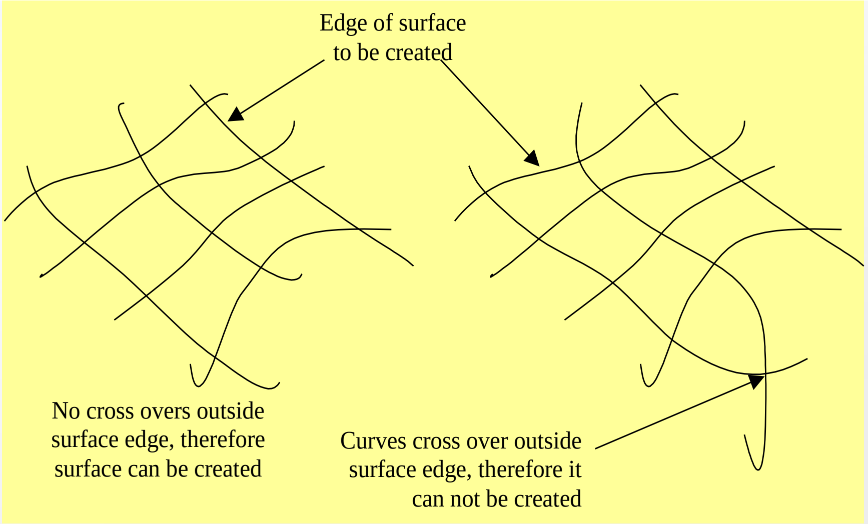
Surfaces
have no thickness,
no volume or solid properties.

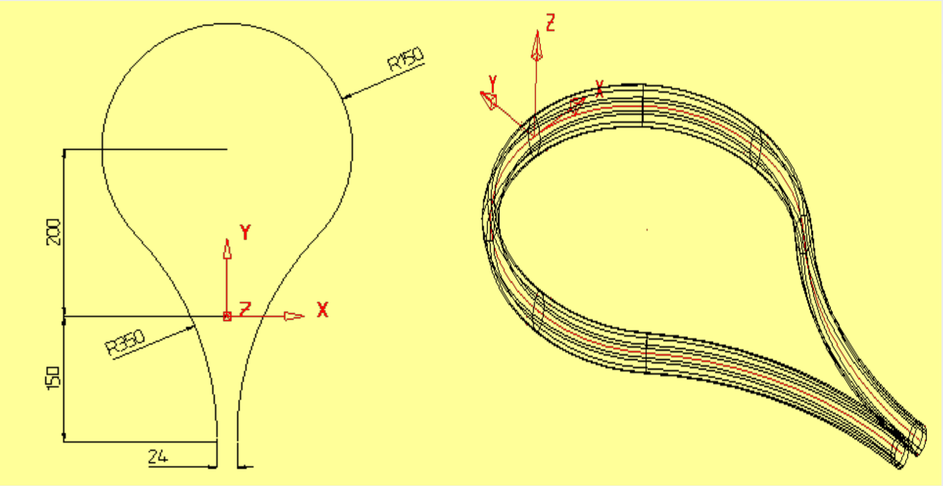


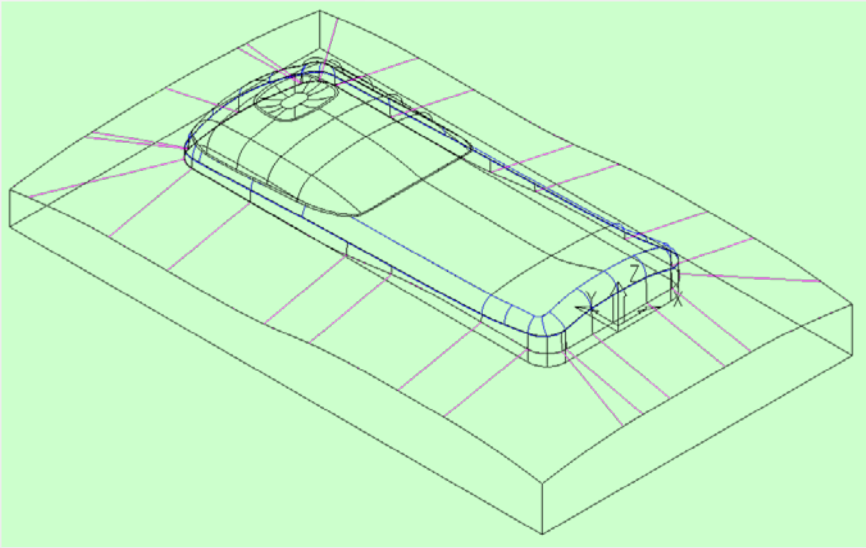
Surfaces may be open or closed.

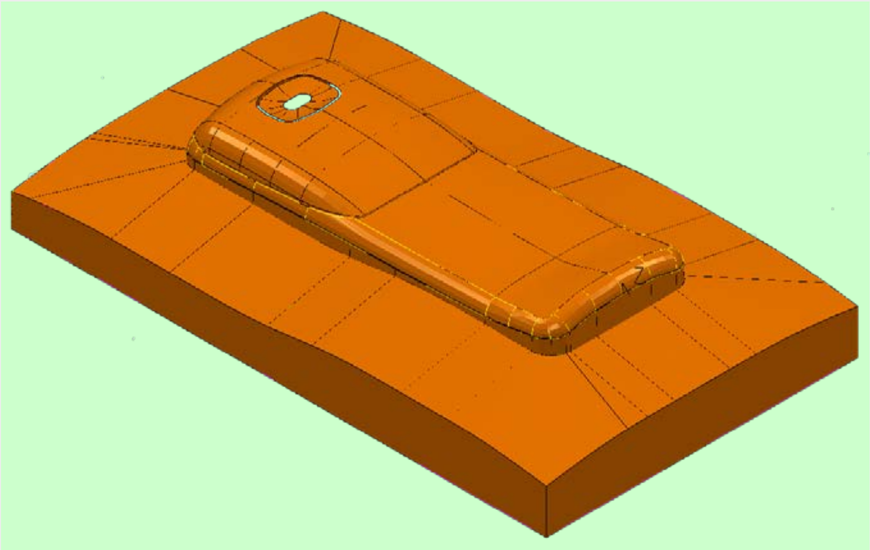








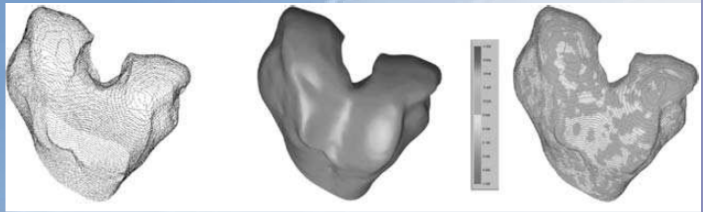
Surfaces from Curves

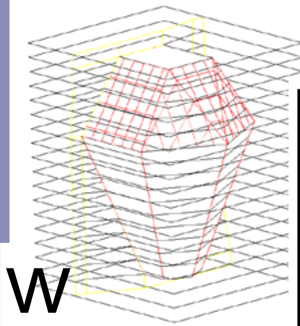






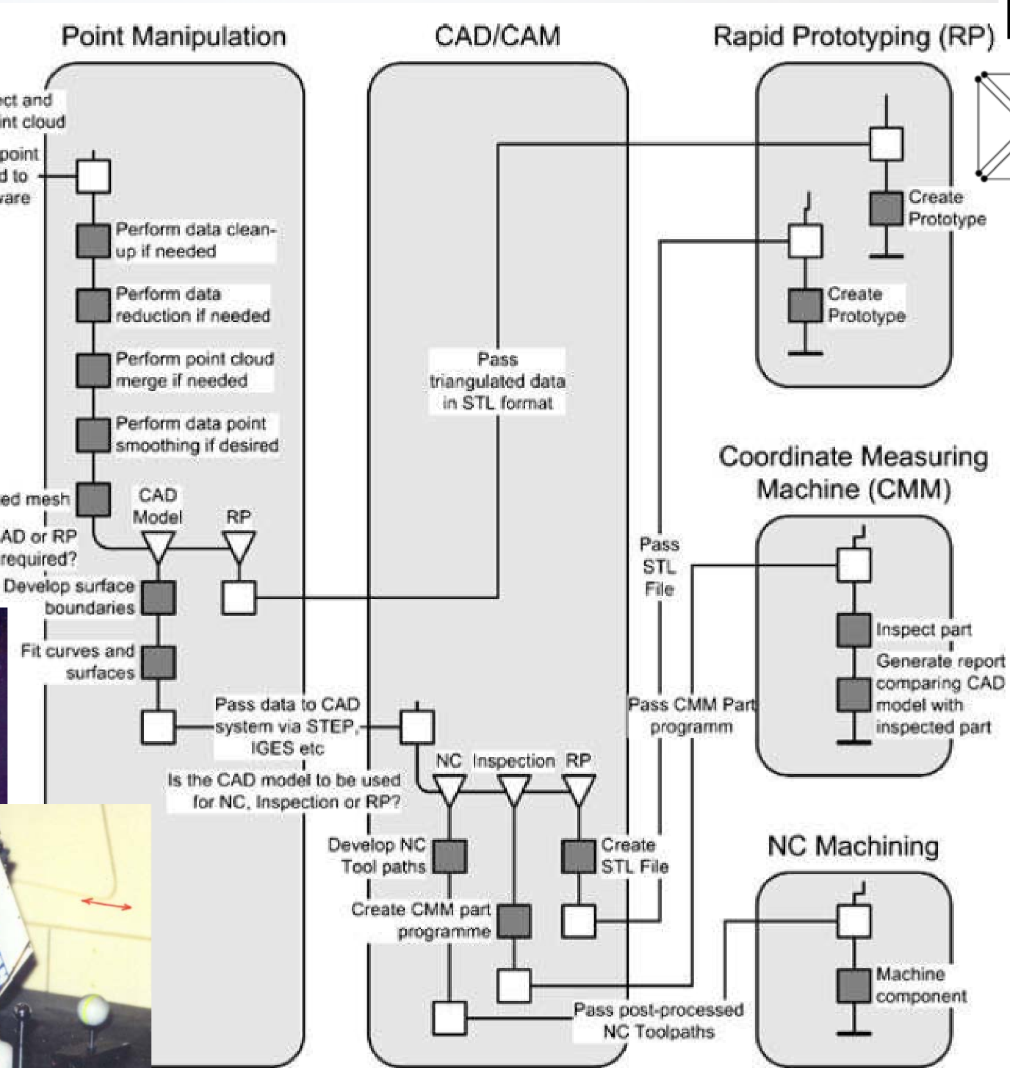


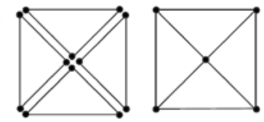




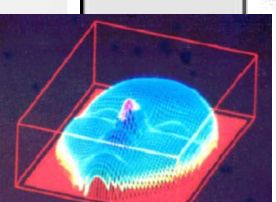
Reverse Engg workflow

Point Number	X	Y	Z	Resultant planar polygon
P1	2.3564	4.5673	7.3428	P2 E1 E2 P3 E3 P1
P2	3.5674	7.6784	7.3428	
P3	7.4536	5.9876	7.3428	
Edge	Start vertex	End vertex		
E1	P3	P2		
E2	P2	P1		
E3	P1	P3		

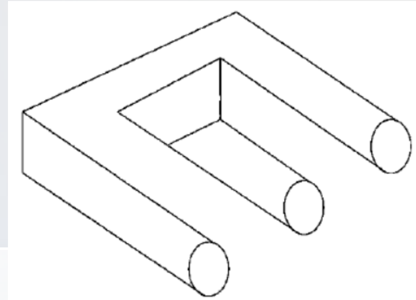


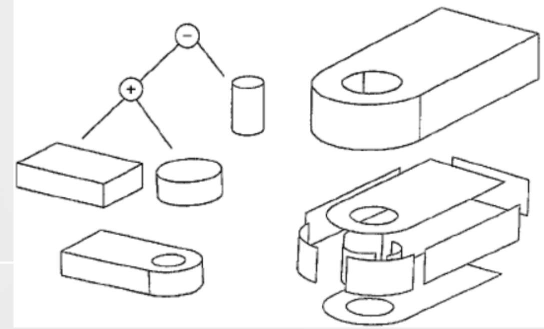


Topology of a triangle

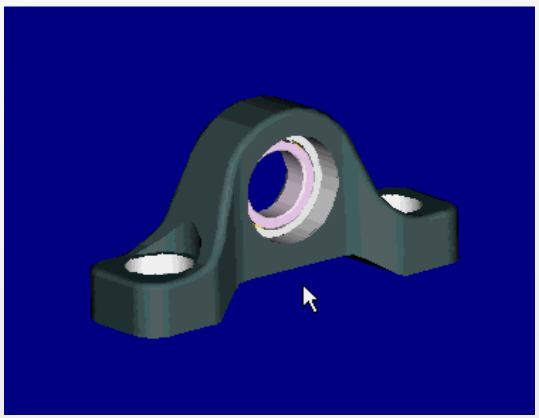


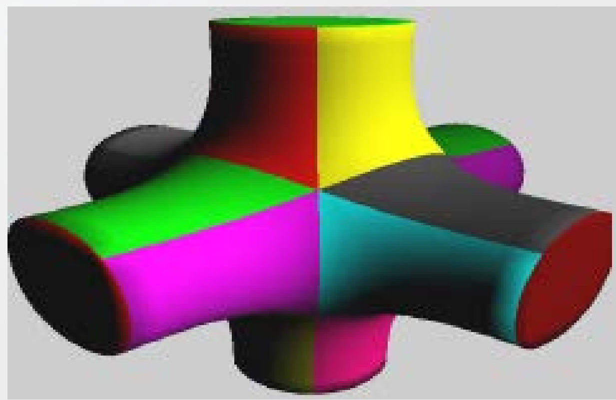
Solid Modeling

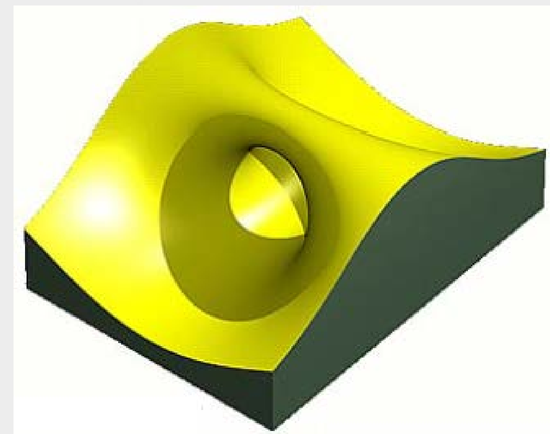


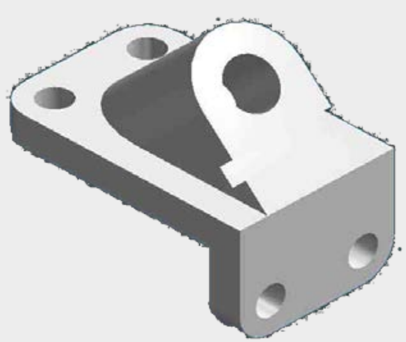


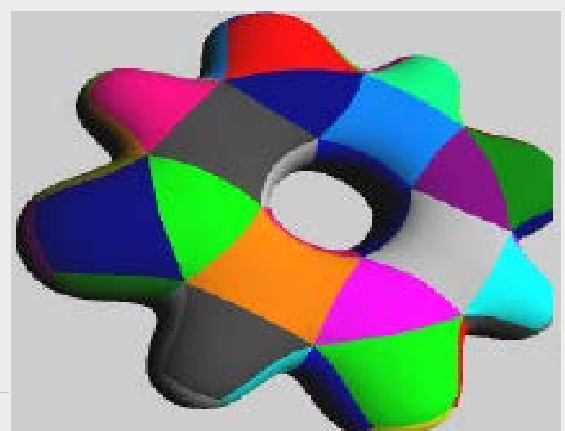
- Complete and unambiguous (clear, exact)
- Models have volume, and mass properties

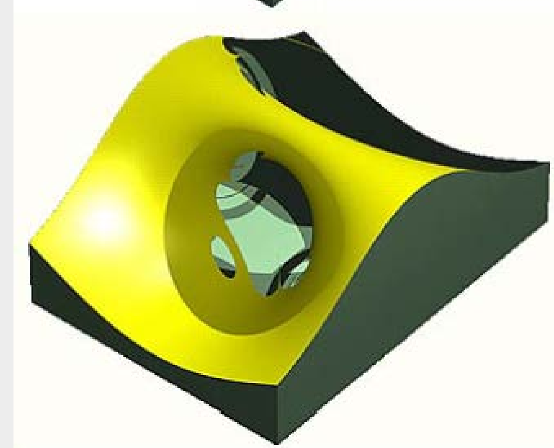






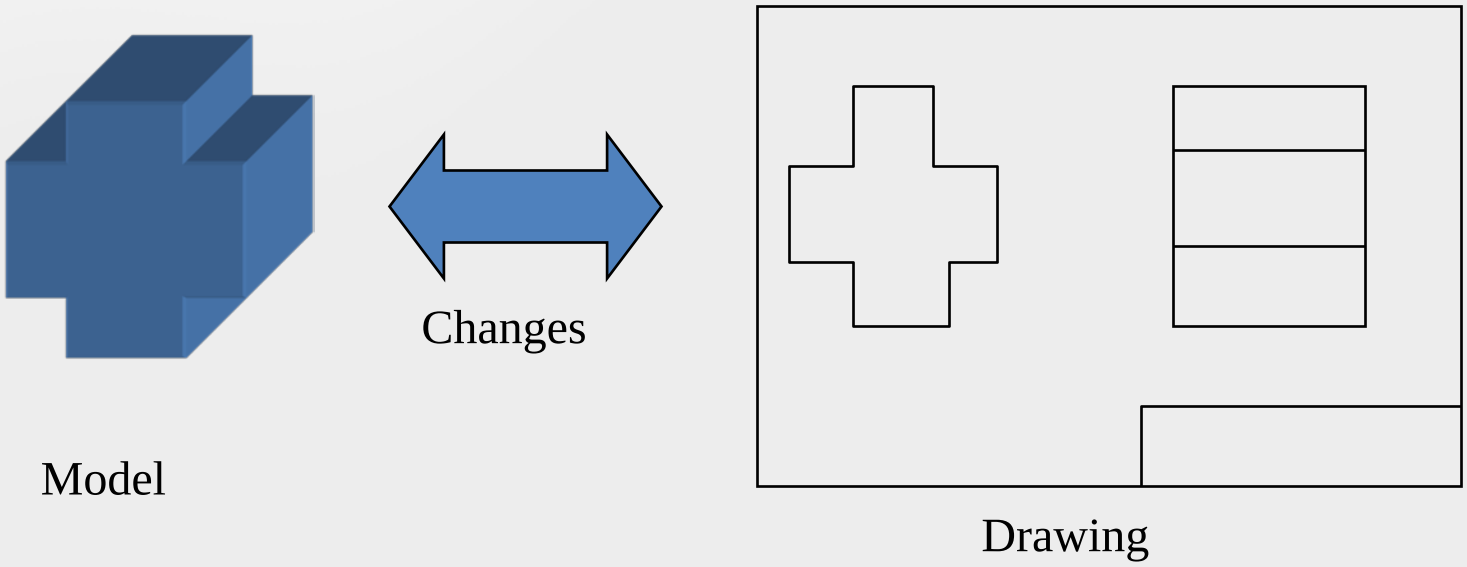






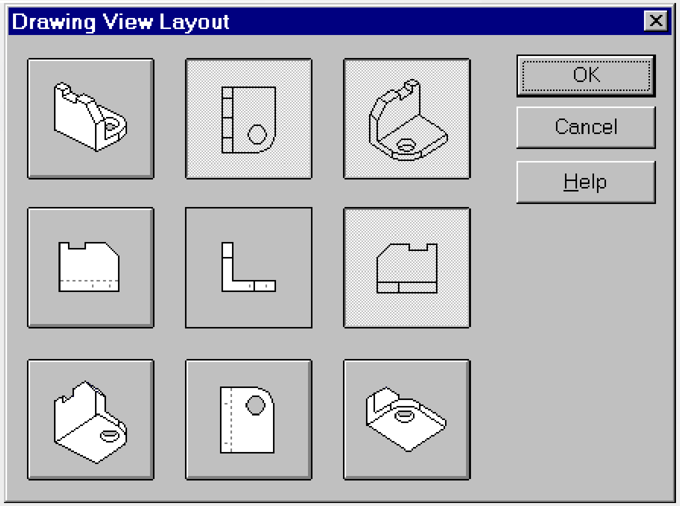
Associativity

- In modern CAD packages, drawings are associated with the underlying model, so that changes to the model cause drawings to be updated
- A CAD package has **bi-directional associativity** if:
  - A change to the model automatically updates the drawing <u>AND</u>
  - A change to the drawing automatically updates the model



Drawing Set Up and Layout

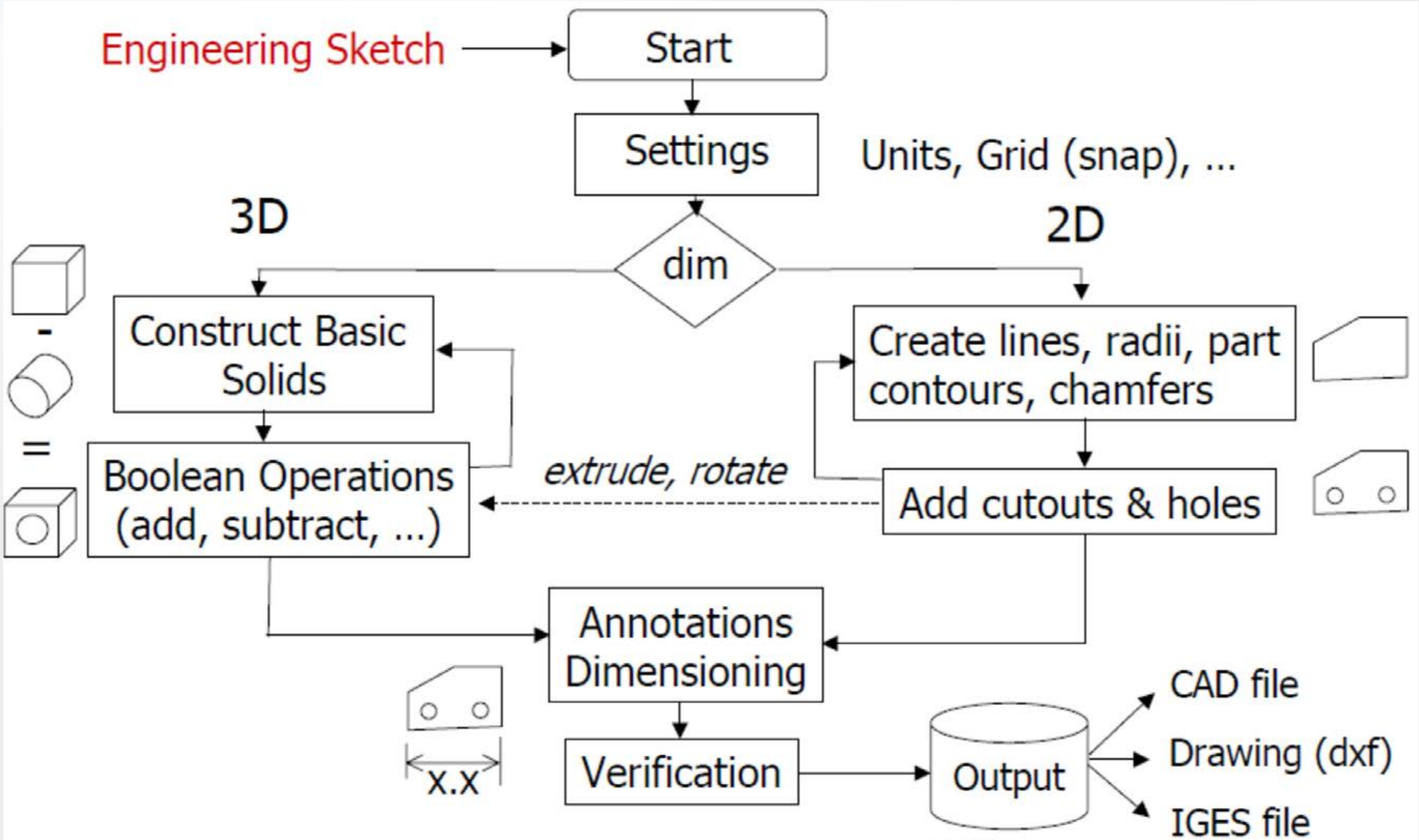
- Drawing Size
- Drawing Projection Angle



Selected views

- Front
- Top
- Right
- Isometric

Generic CAD Process



CAD Software, Graphic User Interface

Geometrical model

- 2D/3D
- Exact or faceted with planar polygons
- Mass properties

Editing

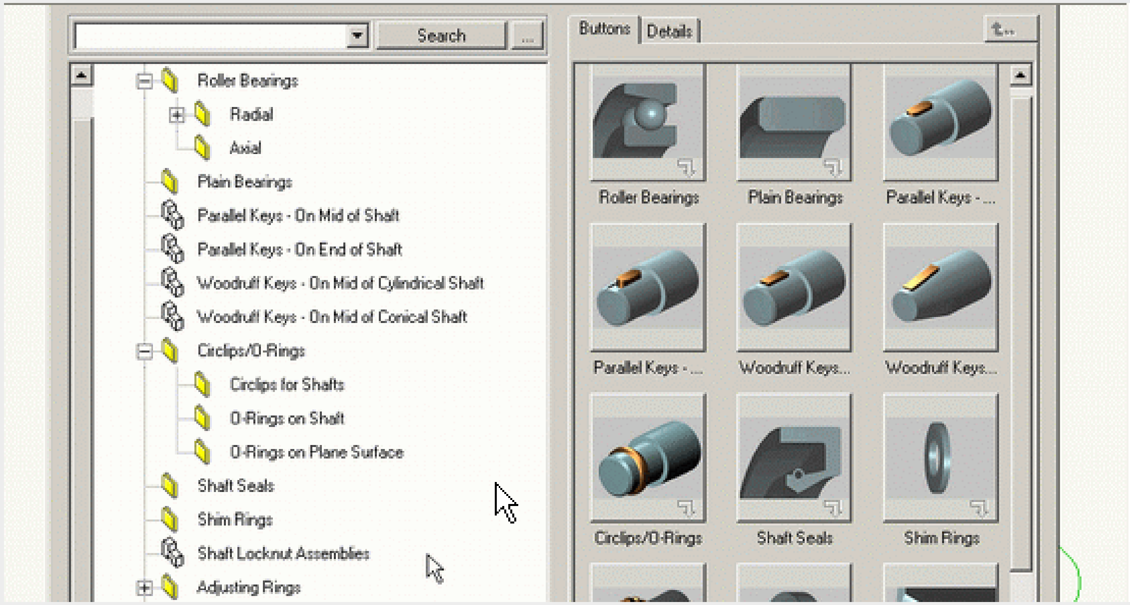
- Parametric

Object Organization

- Named Objects
- Layers
- Part libraries

Drawing Output

- Drafting module



CAD Software, Graphic User Interface

Analysis Module

- Finite Elements
- Plastic Flow
- Kinematics/Collisions
- Dynamics

Importing/Exporting

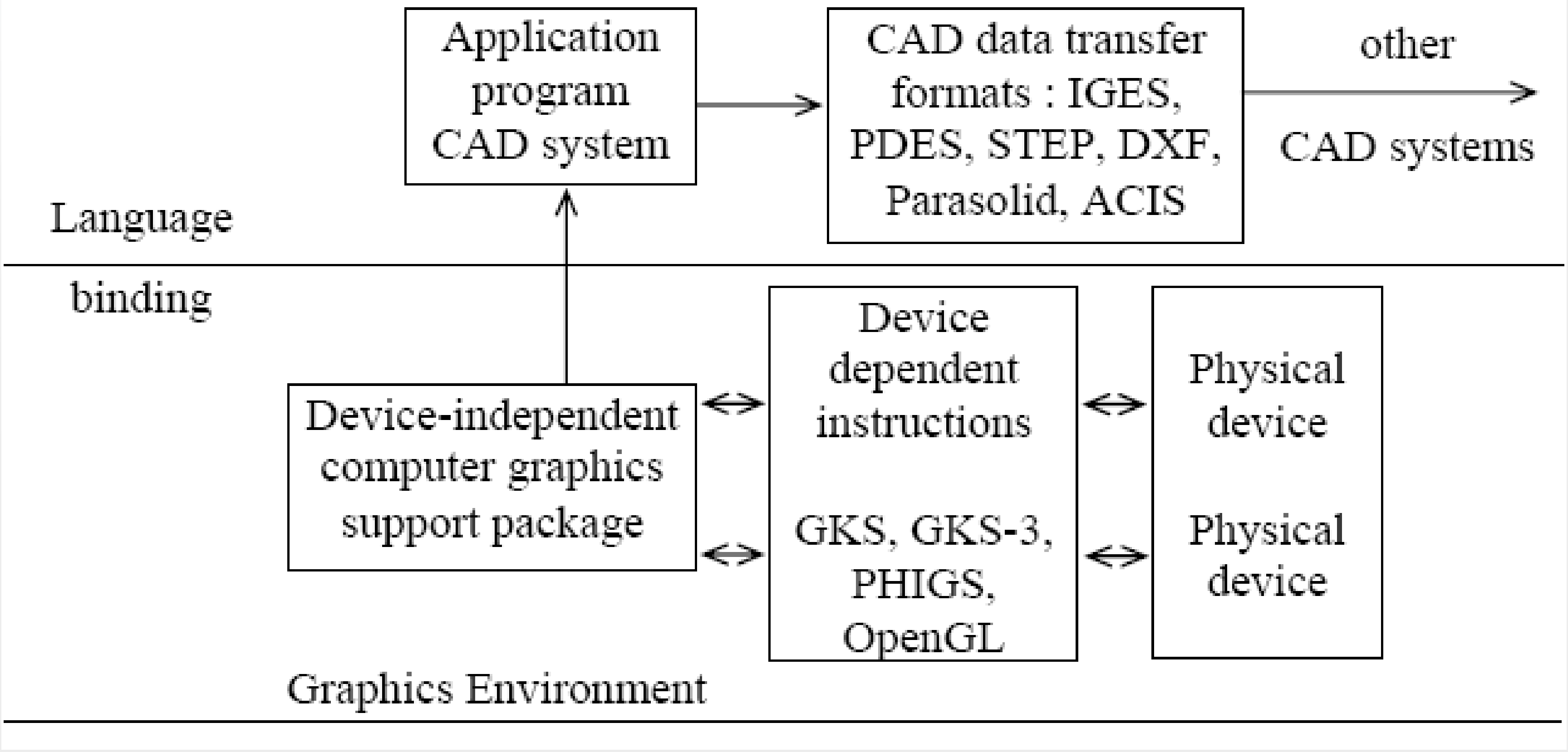
- Surface formats: IGES, DXF, CDL
- Solid Formats: PDES/STEP, ACIS, SAT
- Files for systems such as NASTRAN
- Can be linked to a user written program

Rendering

- Hidden line
- Shaded Image
- Ray Tracing
- Real Time Rotations

Graphics Standards

(GKS, PHIGS, OpenGL, IGES, PDES, STEP, DWG, DXF, Parasolid, ACIS,...)



Graphics Standards

Several graphics standards have been developed over the years, including **CORE** (1977-1979),
GKS (Graphical Kernel System, 1984-1985),
GKS-3D (added 3D capabilities),
PHIGS (Programmer's Hierarchical Graphics sys.1984),
PHIGS+ include more powerful 3D graphics functions,
X-Windows system (1987), and
OpenGL is adapted from Unix system.
DirectX (1994) API developed by Windows for 3D animation.

IGES, STEP, ACIS data exchange formats

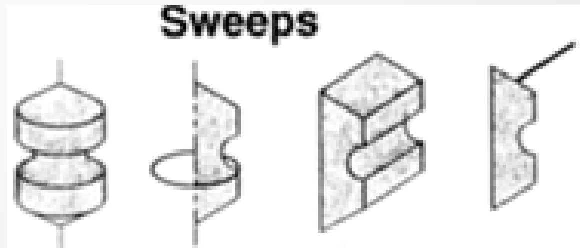
Import Formats	Export Formats
SolidWorks .sldprt, .sldasm	CATIA V4 .model
ACIS .sat	CATIA V5 .CATPRODUCT, .CATPART
Inventor .ipt, .iam	ACIS .sat
CATIA V5 (visualization data) .CATPRODUCT, .CATPART, .CGR	VDA-FS .vda
CATIA V4/V5 .model, .session, .exp, .CATPRODUCT, .CATPART, .CATSHAPE	Parasolid .x_t, .x_b
Pro/Engineer .prt, .asm, .xpr, .xas	STEP .step, .stp
NX (formerly Unigraphics) .prt	IGES .iges, .igs
VDA-FS .vda	COLLADA .dae
Parasolid .x_t, .x_b	VRML .wrl
STEP	X3D .x3d
IGES	DWF .dwf
	DWG .dwg
	OpenFlight .flt

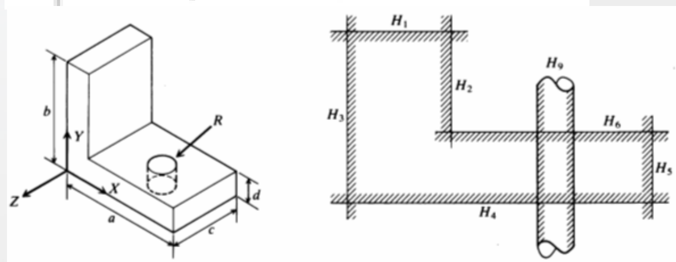
IGES, STEP, PDES, Parasolid formats

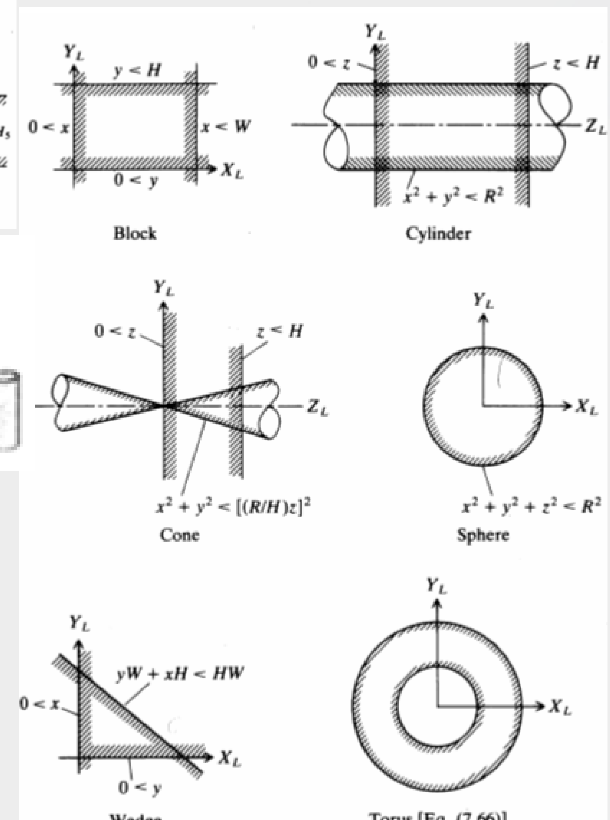
IGES (Initial Graphics Exchange Specification) initially published by ANSI in 1980. Version 5.3 (1996) is the last.
STEP (STandard for the Exchange of Product model data) (ISO 10303) released in 1994. A neutral representation of product data. Every year new parts are added or new revisions of older parts are released. This makes STEP the biggest standard within ISO.
PDES (Product Data Exchange Specification, PDDI) originated in 1988 by McDonnell Aircraft Corporation.
Parasolid (owned by Siemens) can represent wireframe, surface, solid, cellular and general non-manifold models. It stores topological and geometric information defining the shape of models in transmitting files.

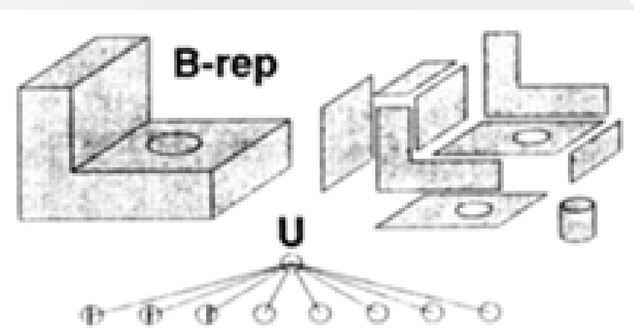
Solid modeling techniques

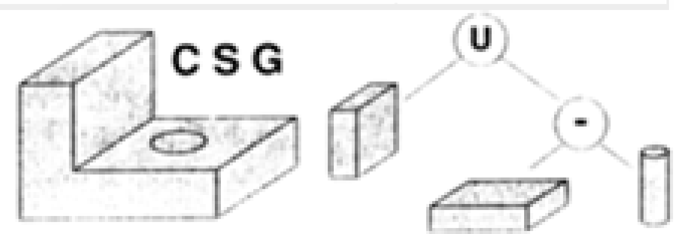
Sweeping,
Half Spaces,
CSG,
B-rep

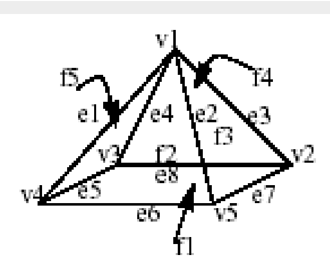


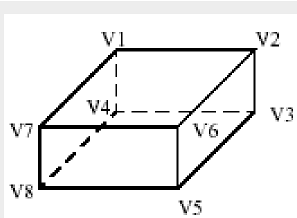




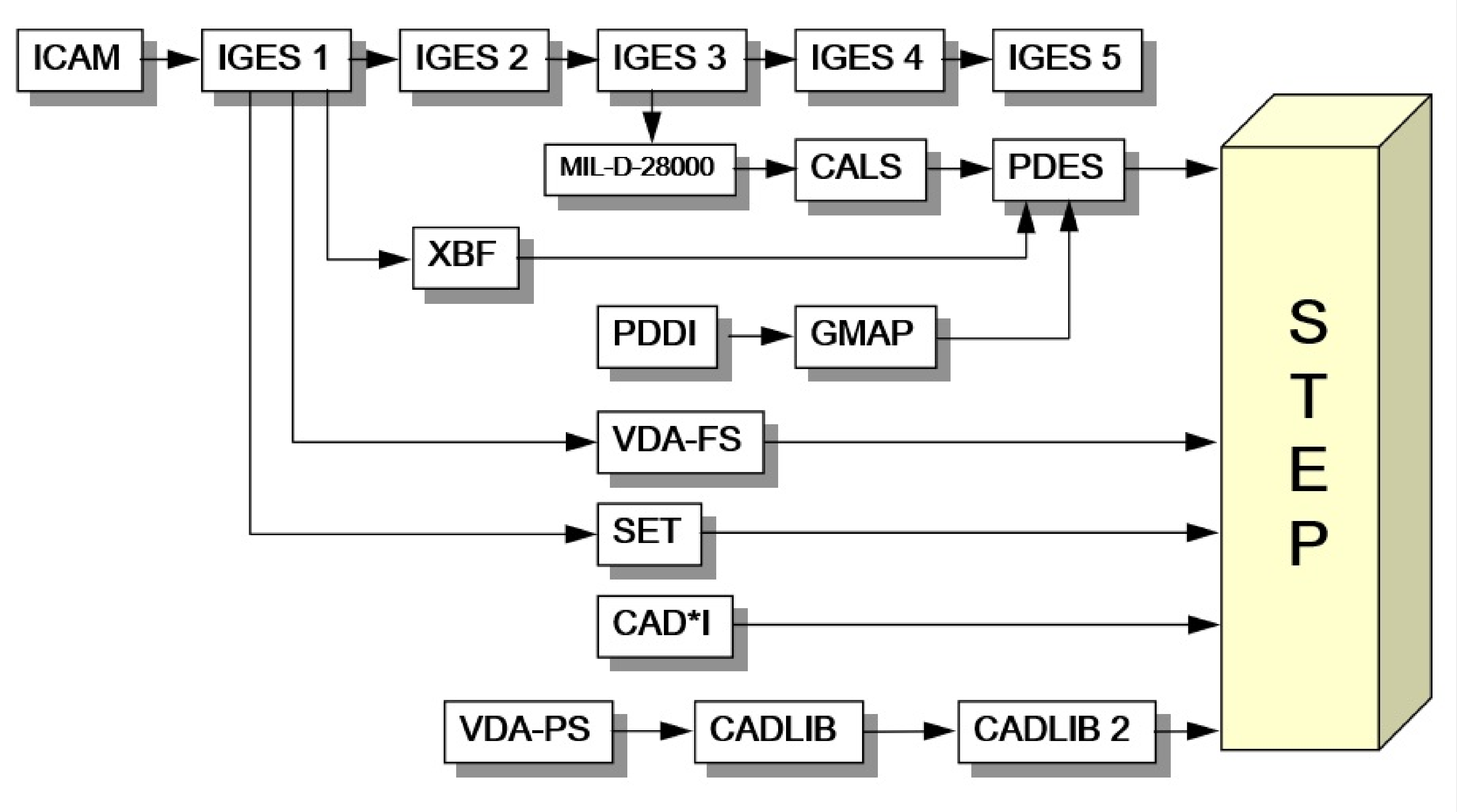






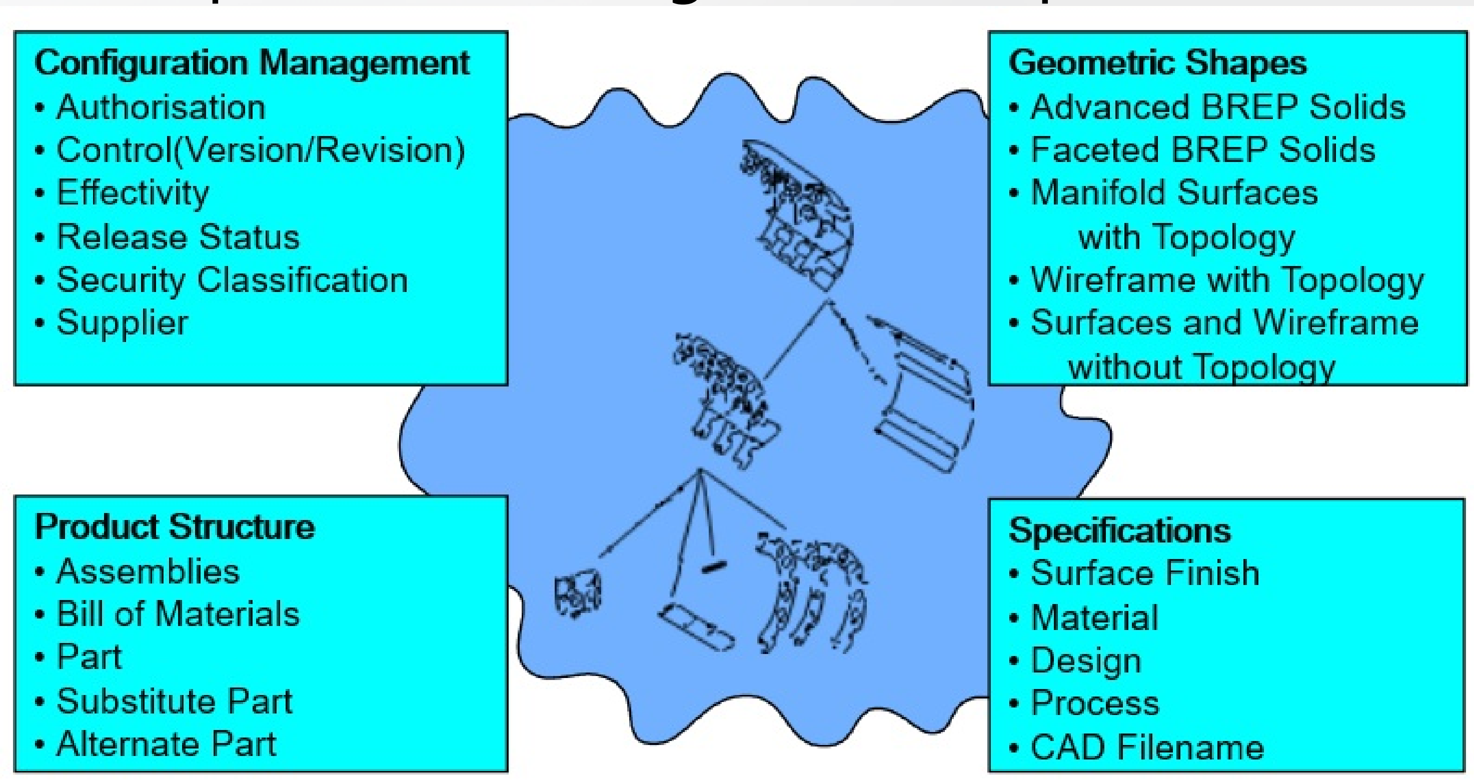


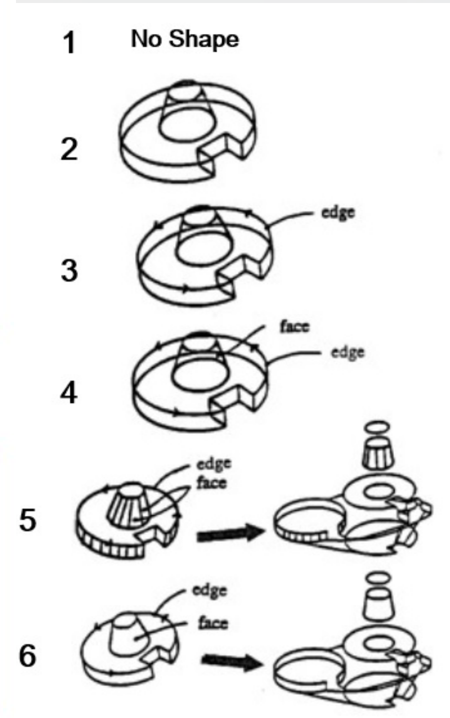
Migration of standards towards STEP



STEP configuration controlled 3D Design

(Standard for the Exchange of Product model data)
STEP is also referred as ISO 10303. (start.1984..1994...)
https://cadexchanger.com/step

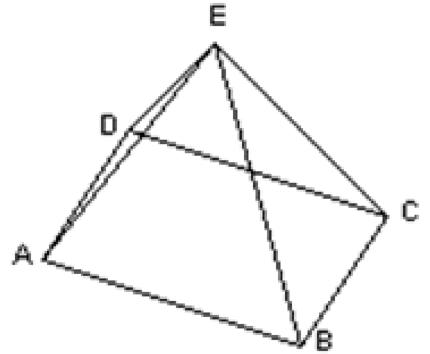




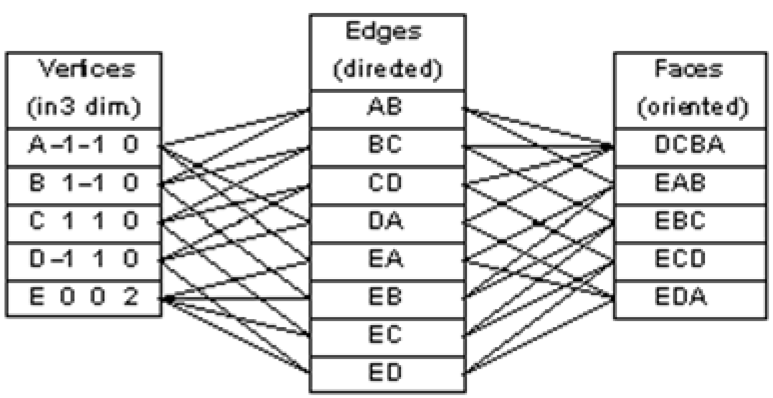
STEP, BREP: Boundary Representation

B-rep represent solids by their surfaces

Define vertices in space (-> exact geometry)
Define edges, faces in terms of vertices (-> structure)



Winged-Edge Data Structure



CLASS 1
CONFIGURATION MANAGEMENT INFORMATION WITHOUT SHAPE

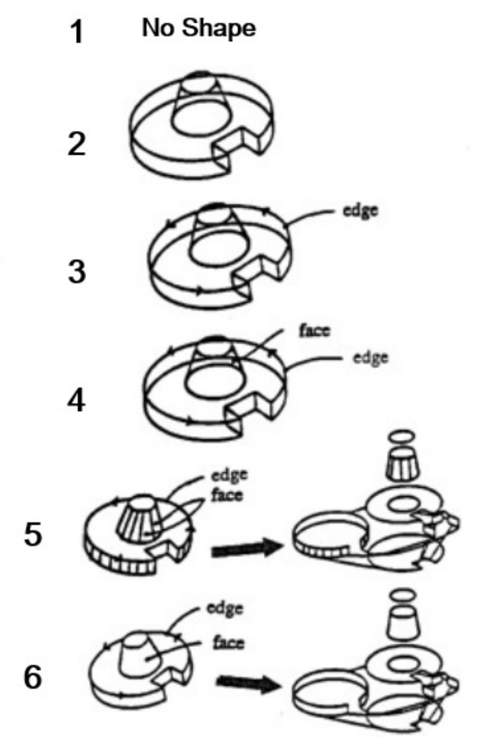
CLASS 2
CLASS 1 + SURFACE & WIREFRAME W/O TOPOLOGY

CLASS 3
CLASS 1 + WIREFRAME WITH TOPOLOGY

CLASS 4
CLASS 1 + MANIFOLD SURFACES WITH TOPOLOGY

CLASS 5
CLASS 1 + FACETED BOUNDARY REPRESENTATION

CLASS 6
CLASS 1 + ADVANCED BOUNDARY REPRESENTATION



Vector versus Raster Graphics

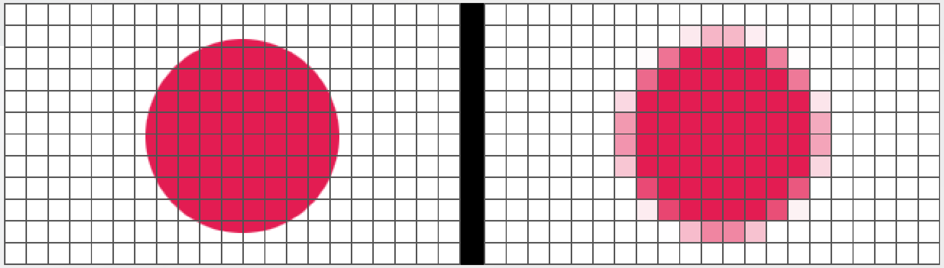
Raster Graphics

- Grid of pixels
  - No relationships between pixels
  - Resolution, e.g. 72 dpi (dots per inch)
  - Each pixel has color, e.g. 8-bit image has 256 colors

.bmp - raw data format

42	4D	BC	02	00	00	00	00	00	00	3E	00	00	00	28	00	00	00	42	00	00	00	35	00	00	00	01
00	01	00	00	00	00	00	00	00	00	00	12	0B	00	00	12	0B	00	00	00	00	00	00	00	00	00	00
FF	FF	FF	00	00	00	00	00	00	00	15	FD	00	00	00	00	00	00	00	00	00	00	FF	EF	F8	00	00
00	00	00	00	00	00	01	D0	00	5C	00	00	00	00	00	00	00	00	0F	80	00	0F	80	00	00	00	00
00	00	00	1C	00	00	01	40	00	00	00	00	00	00	00	38	00	00	00	E0	00	00	00	00	00	00	00
70	00	00	00	70	00	00	00	00	00	00	00	E0	00	00	00	38	00	00	00	00	00	00	01	C0	00	00
00	1C	00	00	00	00	00	00	07	80	00	00	00	0E	00	00	00	00	00	00	07	00	00	00	00	07	00
00	00	00	00	00	0E	00	00	00	03	BB	BB	BB	80	00	00	00	1C	00	00	00	03	FF	FF	FF	C0	00
00	00	18	00	00	00	03	00	C0	00	40	00	00	00	10	00	00	00	03	00	40	00	40	00	00	00	30
00	00	00	02	00	60	00	40	00	00	00	70	00	00	00	03	00	50	00	40	00	00	00	60	00	00	00
02	00	70	00	40	00	00	00	40	00	00	00	03	00	10	00	40	00	00	00	E0	00	00	00	03	00	30
00	40	00	00	00	40	00	00	00	03	00	10	00	40	00	00	00	C0	00	00	00	03	00	18	00	40	00
00	00	40	00	00	00	03	00	10	00	40	00	00	00	C0	00	00	00	02	00	18	00	40	00	00	00	C0
00	00	00	03	00	18	00	40	00	00	00	C0	00	00	00	02	00	08	00	40	00	00	00	C0	00	00	00
03	00	18	00	40	00	00	00	80	00	00	00	03	00	18	00	40	00	00	00	C0	00	00	00	03	00	10
00	40	00	00	00	80	00	00	00	03	00	18	00	40	00	00	00	40	00	00	00	03	00	10	00	40	00
00	00	C0	00	00	00	02	00	18	00	40	00	00	00	40	00	00	00	03	00	10	00	40	00	00	00	E0
00	00	00	02	00	38	00	40	00	00	00	40	00	00	00	03	00	10	00	40	00	00	00	60	00	00	00
03	00	30	00	40	00	00	00	70	00	00	00	03	00	70	00	40	00	00	00	30	00	00	00	03	00	60
00	40	00	00	00	10	00	00	00	03	77	77	77	40	00	00	00	18	00	00	00	03	FF	FF	FF	C0	00
00	00	1C	00	00	00	00	01	C0	00	00	00	00	00	0E	00	00	00	00	03	80	00	00	00	00	00	07
00	00	00	00	07	00	00	00	00	00	00	03	00	00	00	00	0E	00	00	00	00	00	00	01	00	00	00
00	14	00	00	00	00	00	00	01	E0	00	00	00	38	00	00	00	00	00	00	00	70	00	00	00	70	00
00	00	00	00	00	00	38	00	00	00	E0	00	00	00	00	00	00	00	1C	00	00	01	C0	00	00	00	00
00	00	00	0F	80	00	0F	80	00	00	00	00	00	00	00	01	D0	00	5C	00	00	00	00	00	00	00	00
00	FF	BB	F8	00	00	00	00	00	00	00	00	00	17	FF	40	00	00	00	00	00	00	00	00	00		

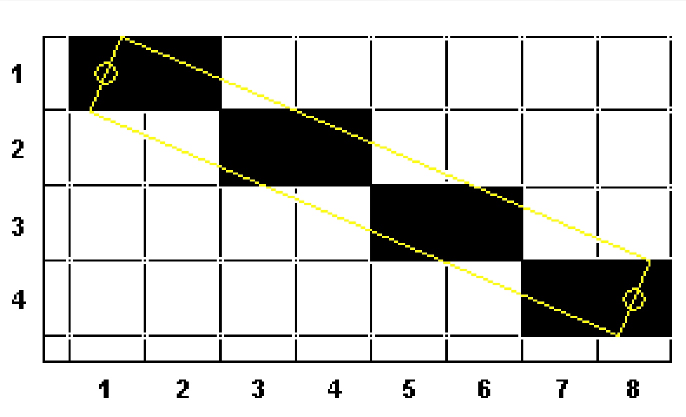
Raster Graphics

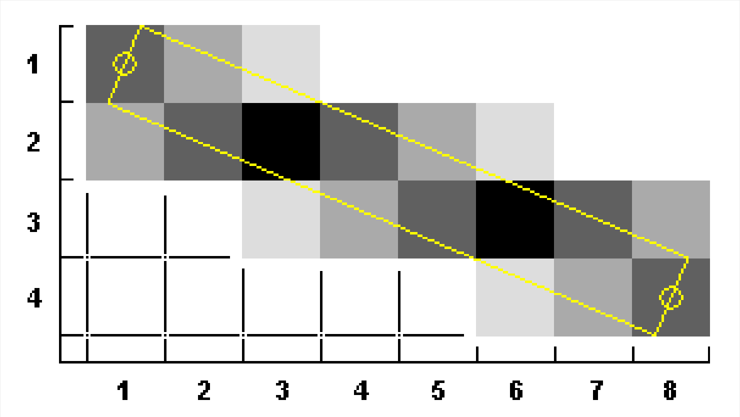


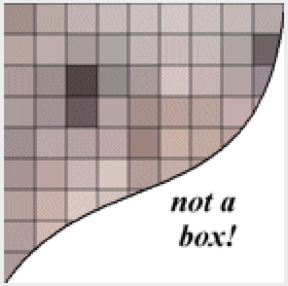
Tessellation

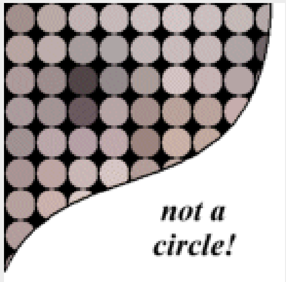
Sampling & Antialiasing

It is easy to rasterize mathematical line segments into pixels, but polynomials and other parametric functions are harder.

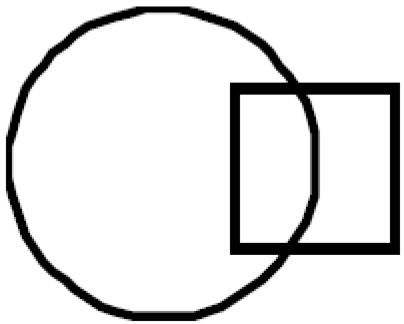








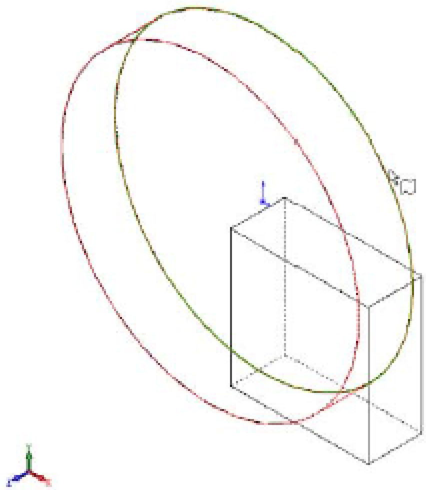
Vector Graphics



.emf format

CAD Systems use vector graphics

Most common interface file: IGES



- Object Oriented
  - relationship between pixels captured
  - describes both (anchor/control) points and lines between them
  - Easier scaling & editing

Curve representation equations

A **line** can be defined using either **parametric** equation or implicit, explicit **nonparametric** equations.

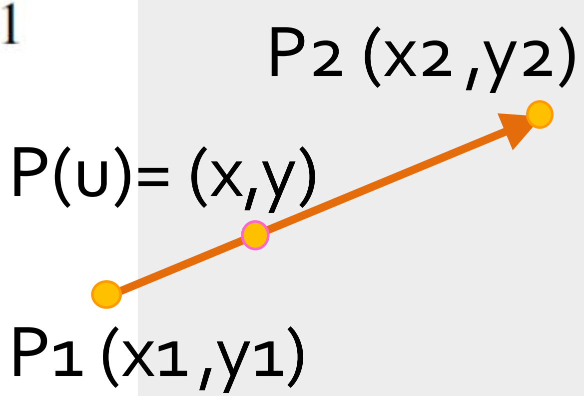
Given two points (x1, y1) and (x2, y2)

Implicit: $(x2 - x1)(y - y1) - (y2 - y1)(x - x1) = 0$

Explicit: $y = (y2 - y1)(x - x1)/(x2 - x1) + y1$

Parametric $\quad Let \quad u = \dfrac{x - x1}{x2 - x1} = \dfrac{y - y1}{y2 - y1}$

$$\begin{cases} x = (1-u)x1 + ux2 \\ y = (1-u)y1 + uy2 \end{cases} \qquad 0 \leqslant u \leqslant 1$$



Curve representation equations

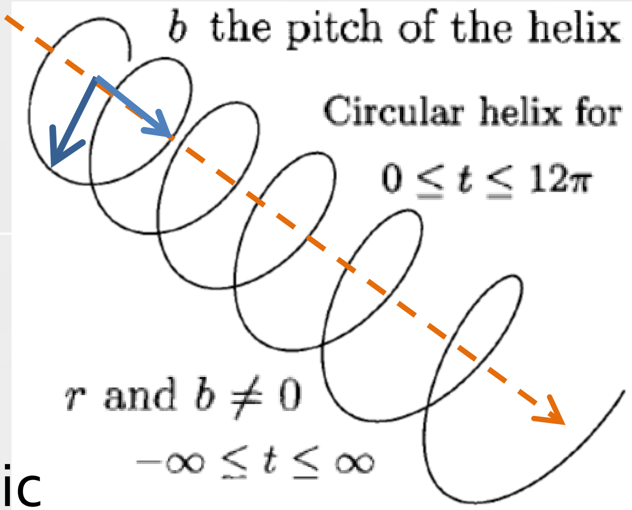
There are two types of curve equations

(1) Parametric equation

x, y, z coordinates are related by a parametric and independent variable (u, θ or t)

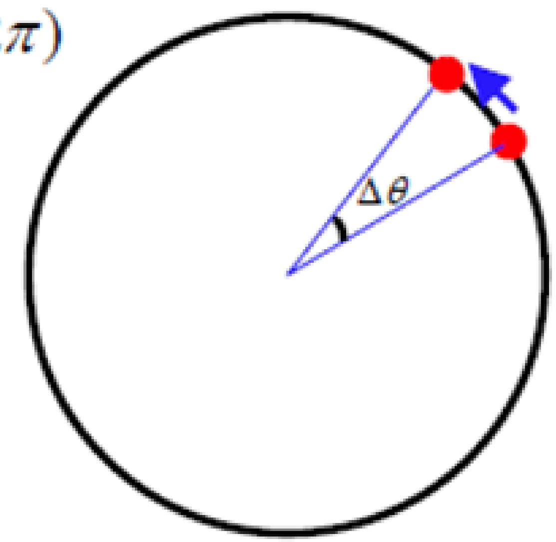
Point on 3-D curve: $\mathbf{p} = [x(u) \quad y(u) \quad z(u)]$

Point on 2-D curve: $\mathbf{p} = [x(u) \quad y(u)]$



$$P(t) = [x(t) \quad y(t) \quad z(t)]$$
$$= [r\cos t \quad r\sin t \quad bt]$$

$$x = R\cos\theta, \qquad y = R\sin\theta \qquad (0 \le \theta \le 2\pi)$$



(2) Nonparametric equation

x, y, z coordinates are related by a function

Implicit: $x^2 + y^2 - R^2 = 0$

Explicit: $y = \pm\sqrt{R^2 - x^2}$

Curve representation equations

Which is better for CAD/CAE ? : Parametric equation
It is good for calculating the points at a certain interval along a curve.

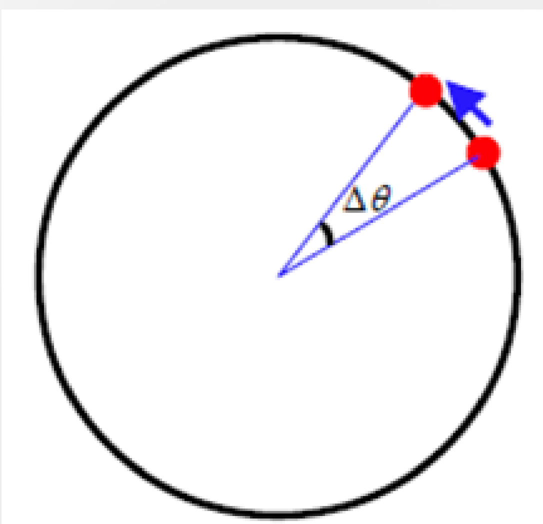
Example:
Circle

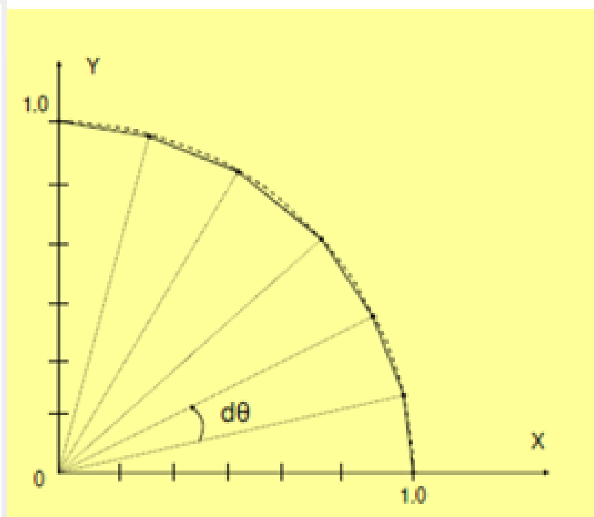
Parametric:

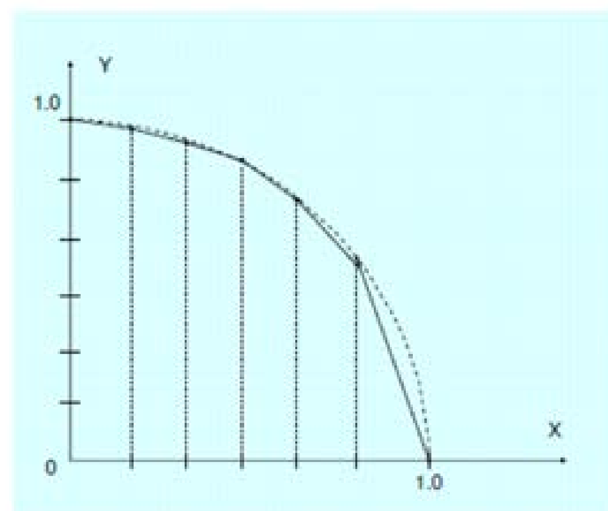
$$x = \cos\theta \quad 0 \le \theta \le 2\pi$$
$$y = \sin\theta$$

Implicit: $x^2 + y^2 = 1 \quad 0 \le x \le 1$

Explicit: $y = \sqrt{1 - x^2} \quad 0 \le x \le 1$







Comparison

Explicit Form

- Easy to render
- Unique representation
- Difficult to represent all tangents

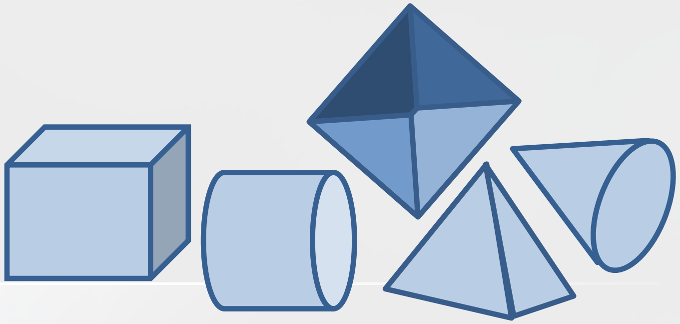
Implicit Form

- Easy to determine if a point lies on, inside, or outside a curve or surface
- Unique representation
- Difficult to render

Parametric Representation

- Easy to render and common in modeling
- Representation is not unique

Geometric Modeling



solids

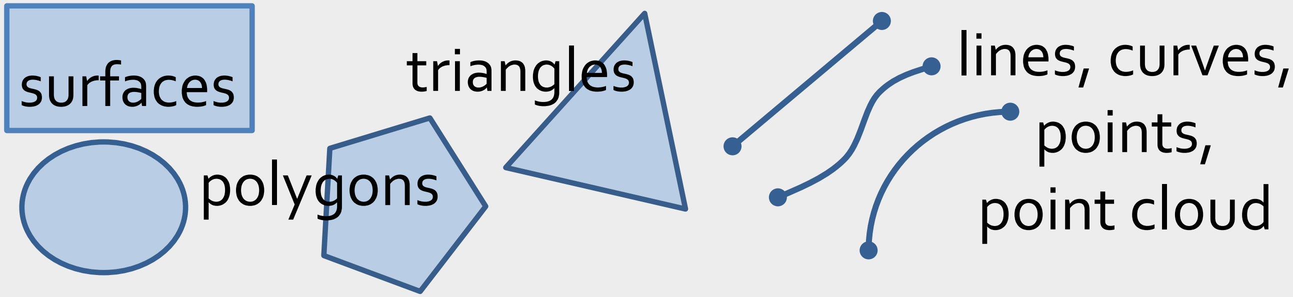
A typical solid model is defined by solids, surfaces, curves, and points.

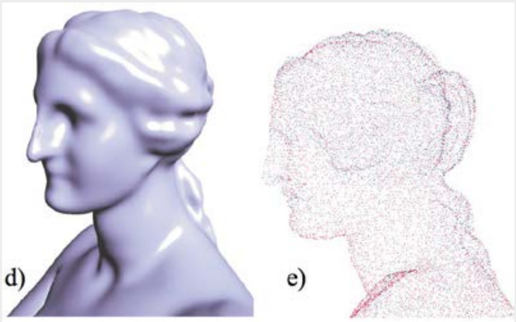
Solids are bounded by surfaces. They represent solid objects. Analytic shape.

Surfaces are bounded by lines. They represent surfaces of solid objects, or planar or shell objects. Quadric surfaces, sphere, ellipsoid, torus.

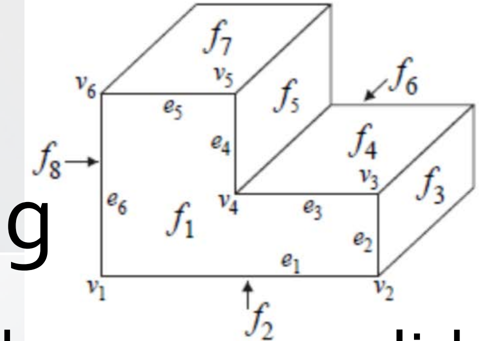
Curves are bounded by points. They represent edges of objects. Lines, polylines, curve.

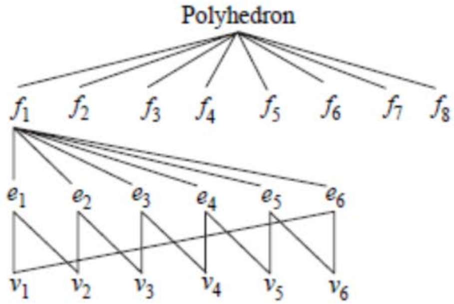
Points are locations in 3-D space. They represent vertices of objects. A set of points.





Geometric Modeling





There is a built-in hierarchy among solid model entities.

Points are the foundation entities.

Curves are built from the points,

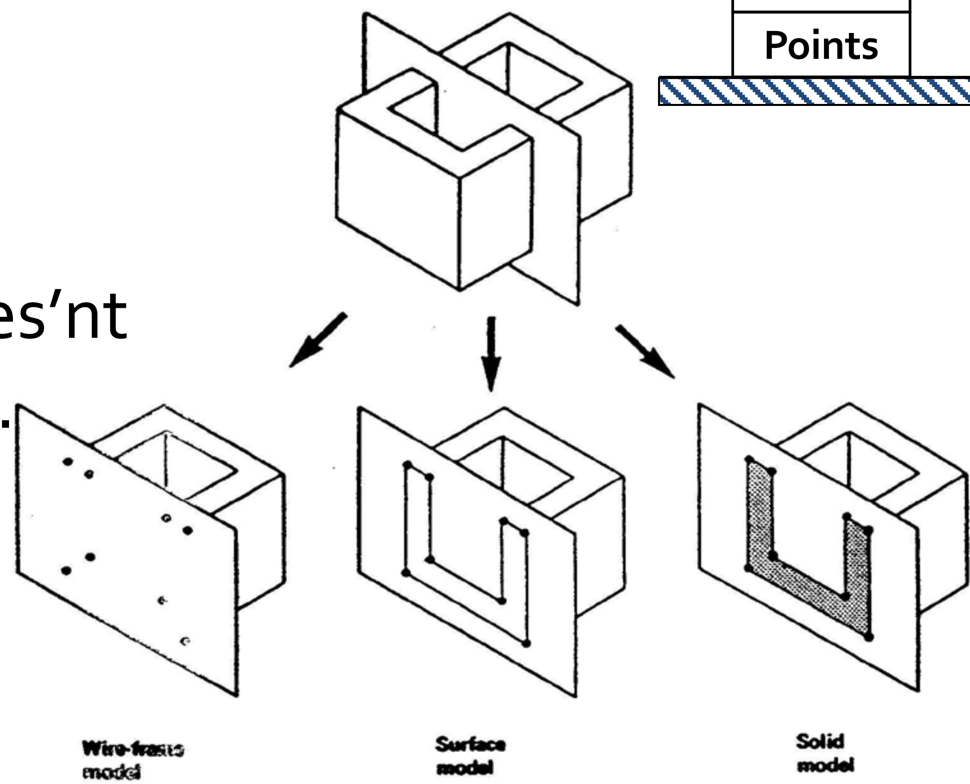
Surfaces from curves,

Solids from surfaces.

Solids
Surfaces
Curves
Points

The wire frame models does'nt have the surface definition.

Difference between wire, surface and solid model

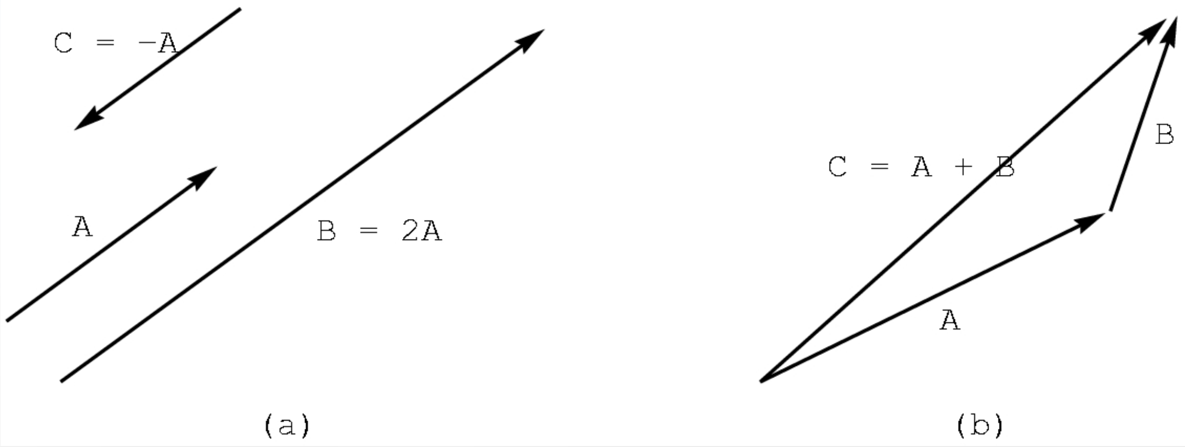


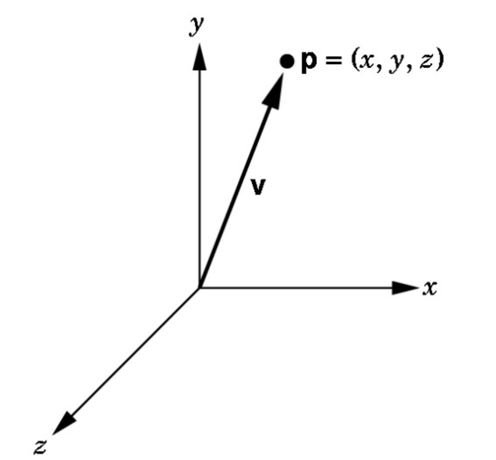
Vector Algebra and Transformations

Source books:
Computer Aided Geometric Design, Thomas W. Sederberg, 2003.
CAD/CAM Theory and Practice , Ibrahim Zeid, McGraw Hill , 1991, Mastering CAD/CAM, ed. 2004

Points and Vectors
Motions and Projections
Homogeneous matrix algebra

Geometric View of Points & Vectors





- vectors have no fixed position
- had-to-tail rule – useful to express functionality
  C* = *A* + *B
- points & vectors – distinct geometric types!
- a given vector can be defined as from a fixed reference point (origin) to the given point ***p***

Vectors (Lines) in Affine Space

Symbols:

α, β, γ - scalars

P, *Q*, *R* – points

u, *v*, *w* – vectors

Typical geometrical operations:

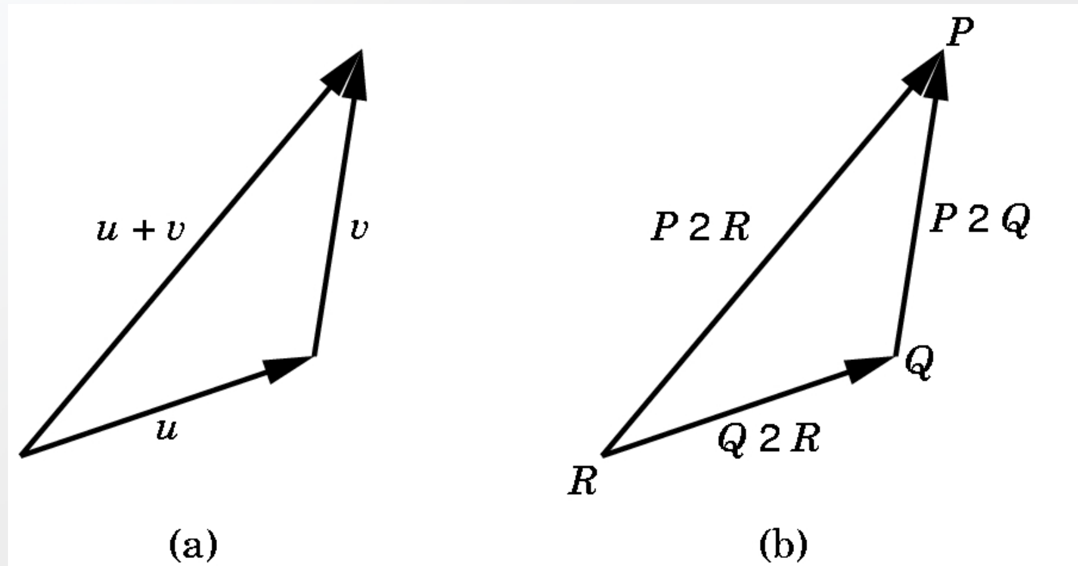
$|\, \alpha v| = |\, \alpha \,|\, |\, v \,|$

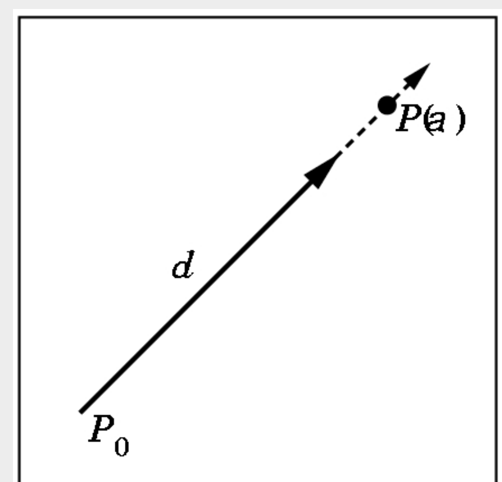
$v = P - Q \;\; => P = v + Q$

$(\, P - Q\,) + (\, Q - R\,) = P - R$

$P(\alpha) = P_0 + \alpha\, d$

(a line in an affine space – param.form)





Vector Sums in Affine Space

new point P can be defined as

$$P = Q + \alpha v$$

Point *R*

$$v = R - Q$$

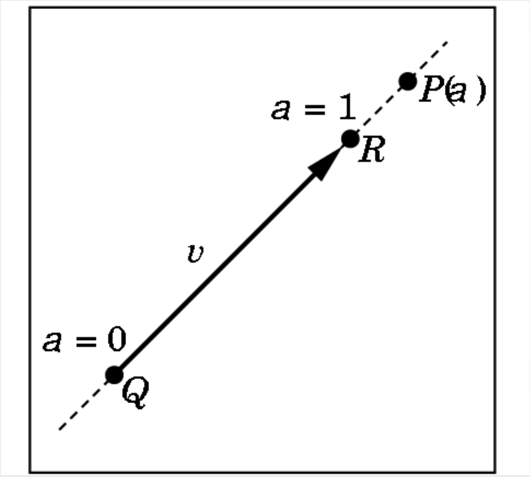
and

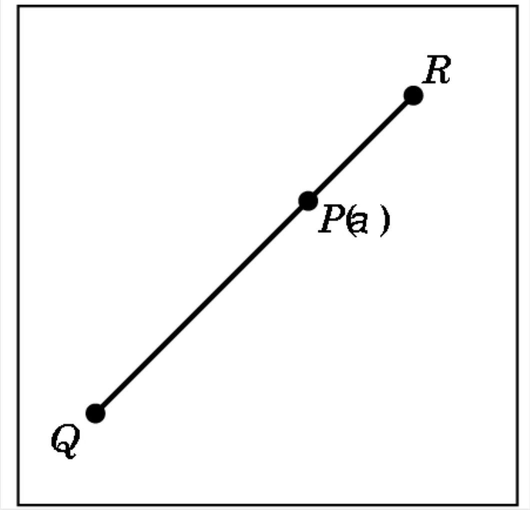
$$P = Q + \alpha(R - Q) = \alpha R + (1 - \alpha)Q$$

$$P = \alpha_1 R + \alpha_2 Q$$

where

$$\alpha_1 + \alpha_2 = 1$$





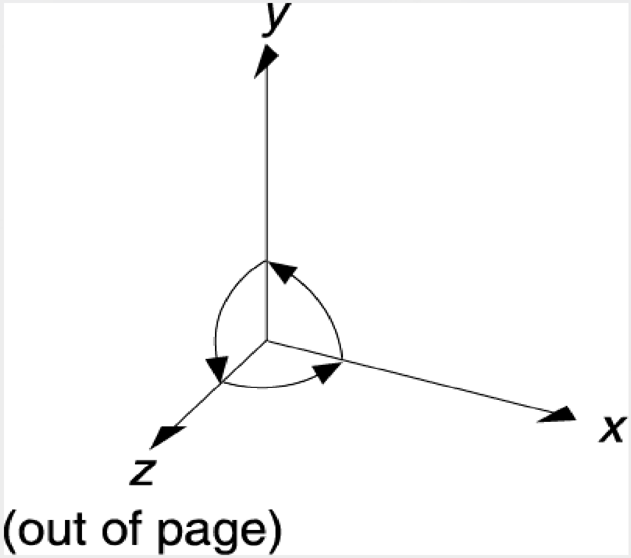
Representation of 3D Transformations

Z axis represents depth

Right Handed System

When looking "down" at the origin,

Positive rotation is CCW.



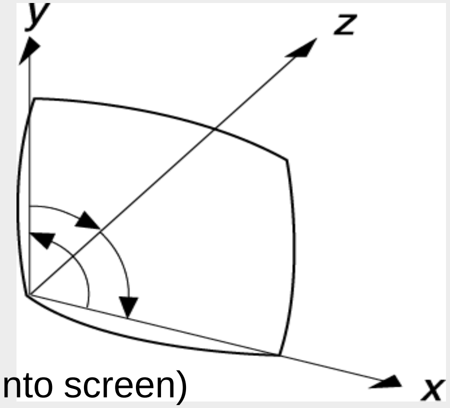
Left Handed System

When looking "down",

positive rotation is in **CW**.

More natural interpretation

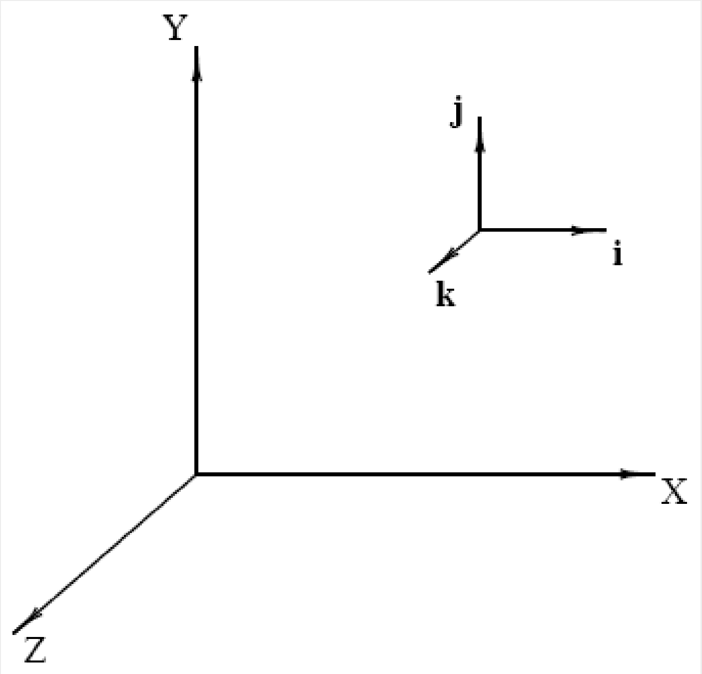
for displays, big *z* means "far"



(into screen)

Points, Vectors and Coordinate Systems

The Cartesian coordinates (x, y, z) are the distances of the vertex with respect to the coordinate system we defined.

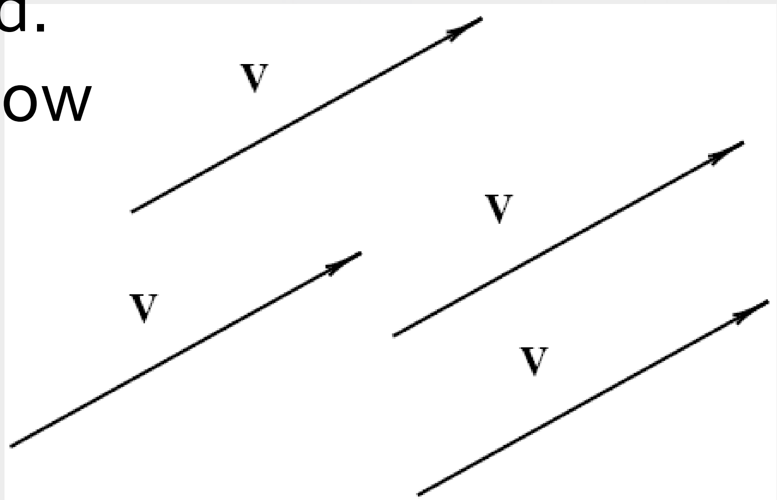


Unit Vectors

A *unit vector is a vector whose length equals unity.*

Vectors

A vector can be pictured as a line segment of definite length with an arrow on one end.
We will call the end with the arrow the tip or head and the other end the tail.



Equivalent Vectors
Two vectors are equivalent if they have the same length, are parallel, and point in the same direction (have the same sense) as shown in Figure.

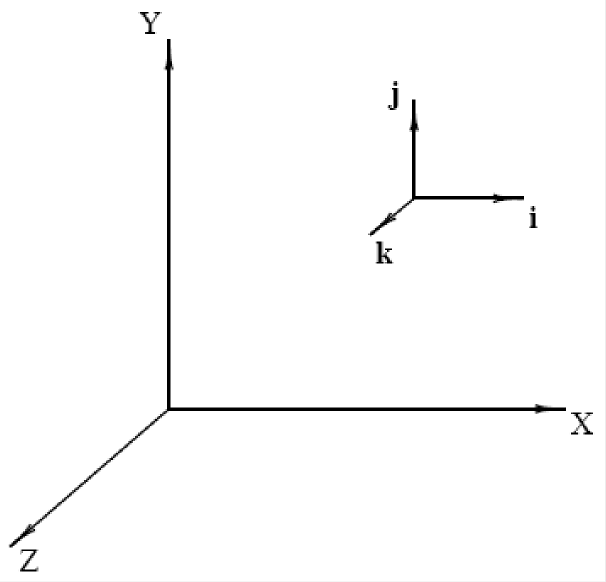
Unit vectors

The symbols i, j, and k denote vectors of “unit length” (based on the unit of measurement of the coordinate system) which point in the positive x, y, and z directions respectively (see Figure). Unit vectors allow us to express a vector in component form

P = (a, b, c) = ai + bj + ck

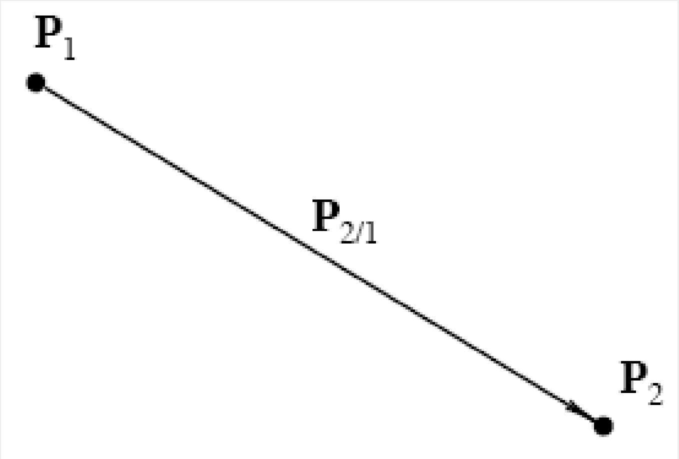
Unit Vectors

A *unit vector is a vector whose length equals unity.*



Points and Vectors

An expression such as (x, y, z) can be called a triple of numbers. A triple can signify either a point or a vector. Relative Position Vectors Given two points P_1 and P_2, we can define $P_{2/1} = P_2 - P_1$
as the vector pointing from P_1 to P_2. This notation $P_{2/1}$ is widely used in engineering mechanics, and can be read "the position of point P_2 relative to P_1"
(see Figure).



The distance between two points

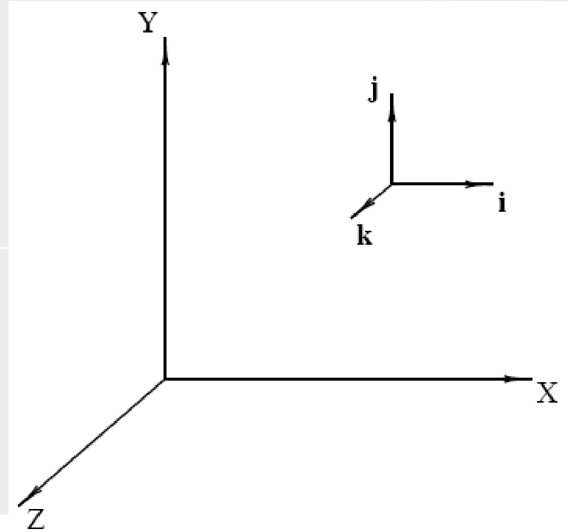
In a Euclidean space we define the distance between two points p and q as the norm of the vector p – q.

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

Because points correspond to vectors, for a fixed origin, and vectors correspond to column matrices, for a fixed basis, there is also a one-to-one correspondence between points and column matrices. A pair (origin, basis) is called a *frame or coordinate system. For a fixed* frame, points correspond to column matrices.

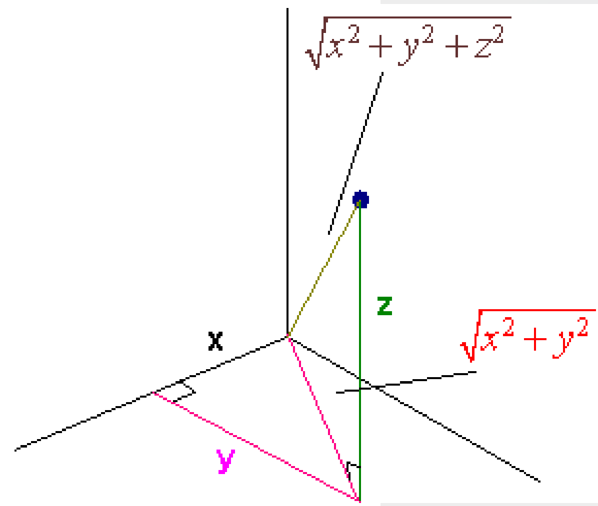
Vector algebra

Let **A**, **B**, and **C** be independent vectors, $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ be unit vectors in the X, Y, and Z directions respectively.



1. Magnitude of a vector is $|\mathbf{A}| = \sqrt{A_x^2 + A_y^2 + A_z^2}$

   where A_x, A_y, and A_z are the cartesian components of the vector **A**.



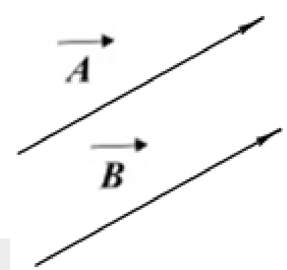
2. The unit vector in the direction of **A** is

$$\hat{\mathbf{n}}_A = \frac{\mathbf{A}}{|\mathbf{A}|} = n_{Ax}\hat{\mathbf{i}} + n_{Ay}\hat{\mathbf{j}} + n_{Az}\hat{\mathbf{k}}$$

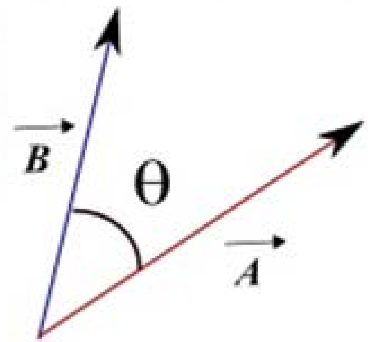
   The components of $\hat{\mathbf{n}}_A$ are also the direction cosines of the vector **A**.

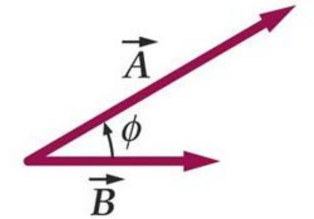
3. If two vectors **A** and **B** are equal, then

$$A_x = B_x \qquad A_y = B_y \quad \text{and} \quad A_z = B_z$$



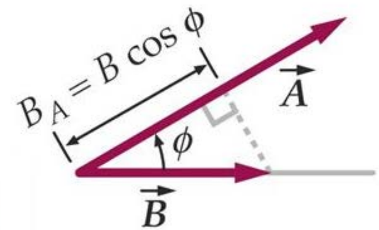
Vector algebra





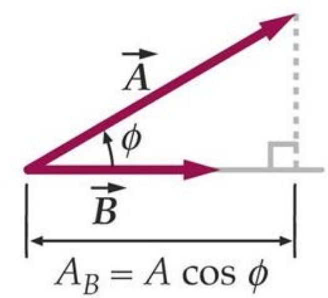
4. The scalar (dot or inner) product of two vectors **A** and **B** is a scalar value

$$\mathbf{A}\cdot\mathbf{B} = \mathbf{B}\cdot\mathbf{A} = A_xB_x + A_yB_y + A_zB_z = |\mathbf{A}|\,|\mathbf{B}|\cos\theta$$



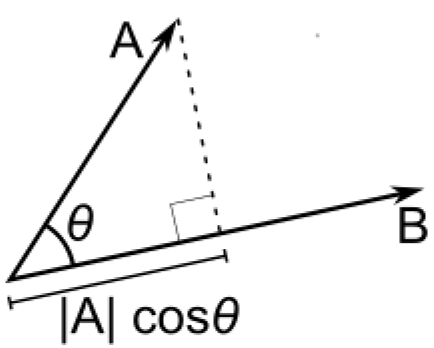
where θ is the angle between **A** and **B**.
Therefore the angle θ between two vectors is given by

$$\cos\theta = \frac{\mathbf{A}\cdot\mathbf{B}}{|\mathbf{A}|\,|\mathbf{B}|}$$



The scalar product can give the component of a vector **A** in the direction of another vector **B** as

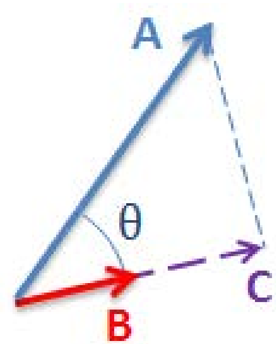
$$\mathbf{A}\cdot\hat{\mathbf{n}}_B = |\mathbf{A}|\cos\theta$$

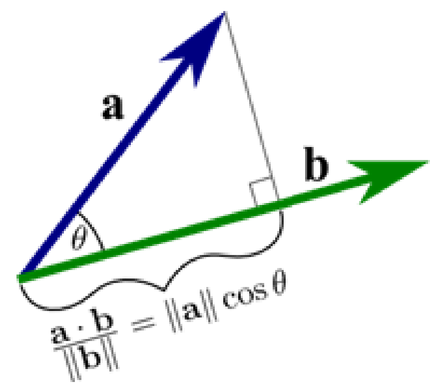


$A \cdot B = |A||B|\cos(\theta)$

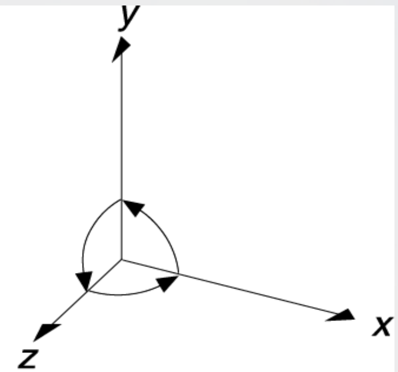
if the magnitude of B is 1, then

$C = A \cdot B = |A|\cos(\theta)$





Vector algebra

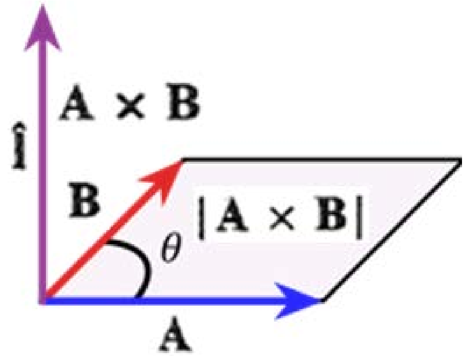


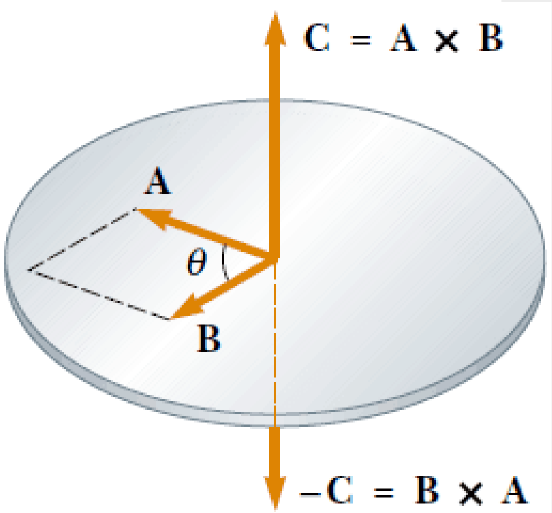
5. The vector (cross) product of two vectors **A** and **B** is a vector perpendicular to the plane formed by **A** and **B** and is given by

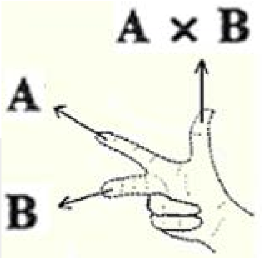
$$\mathbf{A} \times \mathbf{B} = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = (A_y B_z - A_z B_y)\hat{\mathbf{i}} + (A_z B_x - A_x B_z)\hat{\mathbf{j}} + (A_x B_y - A_y B_x)\hat{\mathbf{k}}$$

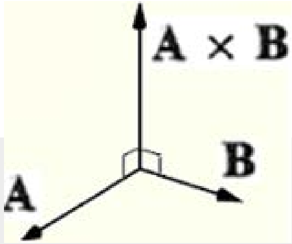
$$\mathbf{A} \times \mathbf{B} = (|\mathbf{A}|\,|\mathbf{B}| \sin\theta)\,\hat{\mathbf{l}}$$

where $\hat{\mathbf{l}}$ is a unit vector in a direction perpendicular to the plane of **A** and **B** when it is rotated from **A** to **B** (the right-hand rule).









Vector algebra

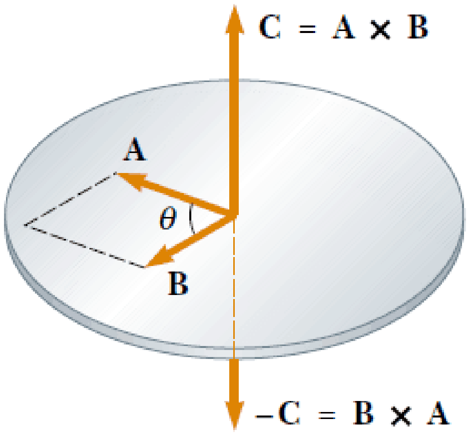
$$\mathbf{A} \times \mathbf{B} = (|\mathbf{A}|\,|\mathbf{B}| \sin\theta)\,\hat{\mathbf{i}}$$

$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}|\,|\mathbf{B}| \cos\theta$$

$$\sin\theta = \frac{|\mathbf{A} \times \mathbf{B}|}{|\mathbf{A}|\,|\mathbf{B}|} \qquad \cos\theta = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}|\,|\mathbf{B}|}$$

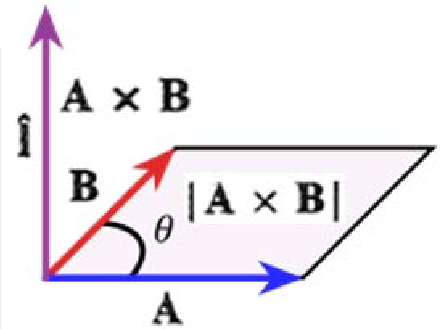
and vector products, the angle θ between two vectors

$$\tan\theta = \frac{|\mathbf{A} \times \mathbf{B}|}{\mathbf{A} \cdot \mathbf{B}}$$



The vector product can give the component of a vector **A** in a direction perpendicular to another vector **B** as

$$|\mathbf{A} \times \hat{\mathbf{n}}_B| = |\mathbf{A}| \sin\theta$$



6. Two vectors **A** and **B** are parallel if and only if

$$\hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B = 1 \qquad \text{or} \qquad |\hat{\mathbf{n}}_A \times \hat{\mathbf{n}}_B| = 0$$

7. Two vectors **A** and **B** are perpendicular if and only if

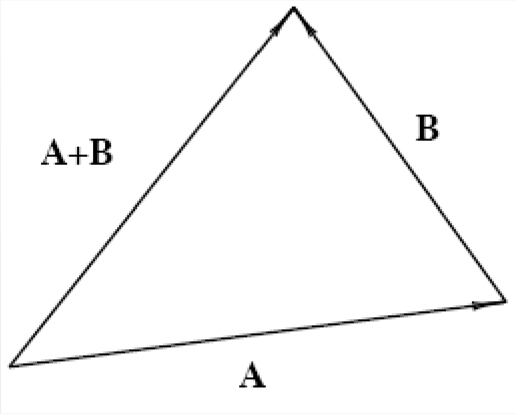
$$\hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B = 0 \qquad \text{or} \qquad |\hat{\mathbf{n}}_A \times \hat{\mathbf{n}}_B| = 1$$

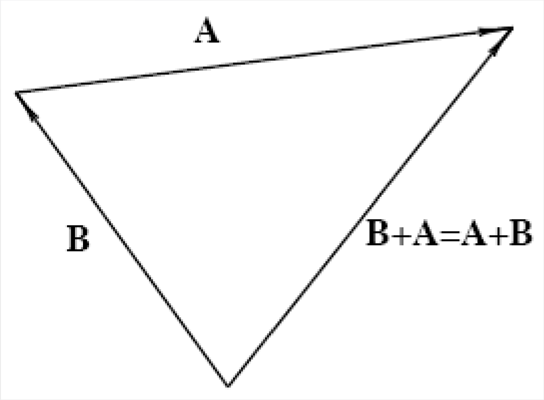
Vector Algebra

Given two vectors $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$, the following operations are defined:

Addition:

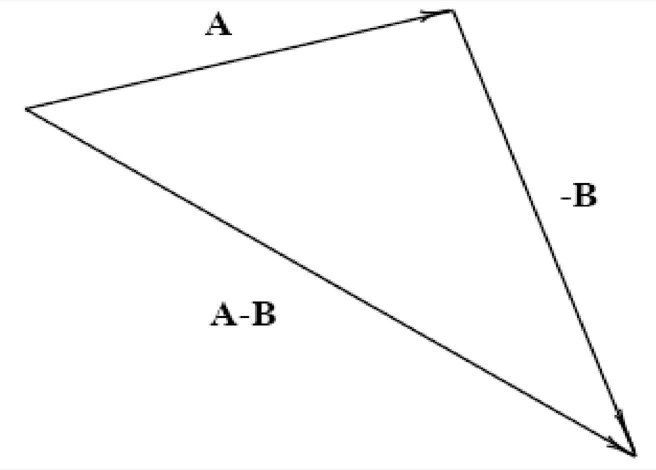
$P_1 + P_2 = P_2 + P_1 = (x_1 + x_2, y_1 + y_2, z_1 + z_2)$



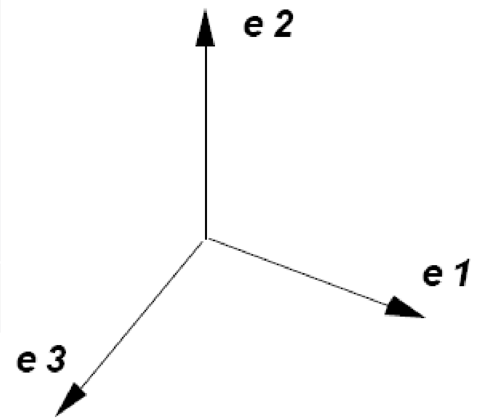


Subtraction:

$P_1 - P_2 = (x_1 - x_2, y_1 - y_2, z_1 - z_2)$



Vector Algebra



Using matrix notation a Vector can be written as

$x = x_1e_1 + x_2e_2 + \ldots + x_ne_n$

$x = EX.$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad E = \begin{bmatrix} e_1 & e_2 & \ldots & e_n \end{bmatrix}.$$

The correspondence between vectors and matrices preserves addition and multiplication by a scalar.

The matrix *Z that* corresponds to the sum of two vectors *$z = x + y$ is the sum*

$$X + Y = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}.$$

Vector Algebra

For multiplication by a scalar, *Z = a X, or*

$cP1 = c\,(x_1, y_1, z_1) = (cx_1, cy_1, cz_1)$

$$Z = \begin{bmatrix} ax_1 \\ ax_2 \\ \vdots \\ ax_n \end{bmatrix}.$$

The inner or dot product, denoted **x . y** ,
is another operation defined on vectors. It produces a scalar given two vector arguments. The square root of the inner product of a vector with itself is the norm or length of the vector, denoted $\|x\| = \sqrt{x.x}\,.$

The length of **x** in an orthonormal basis becomes

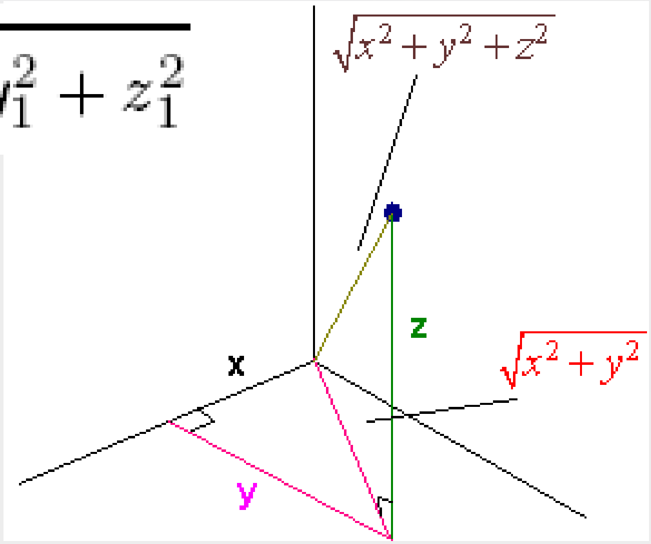
$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Vector Algebra, Dot (Scaler) Product

Length of a vector: $$|\mathbf{P}_1| = \sqrt{x_1^2 + y_1^2 + z_1^2}$$

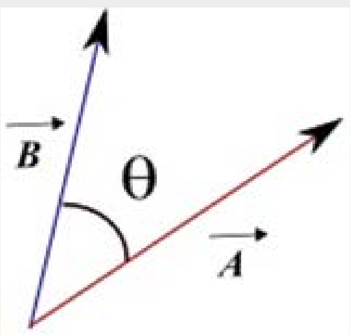
Magnitude of a vector

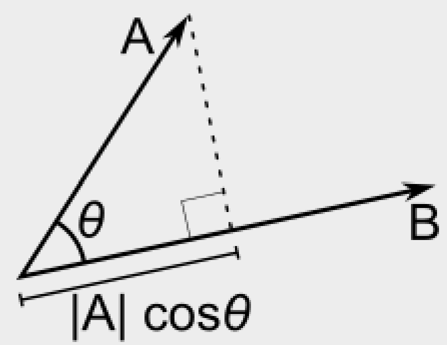
$$|\mathbf{A}| = \sqrt{A_x^2 + A_y^2 + A_z^2}$$



Dot Product: The dot product of two vectors is defined

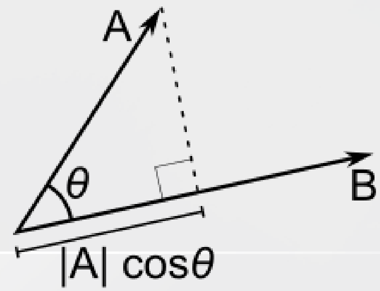
P1 · P2 = |P1||P2| cosθ

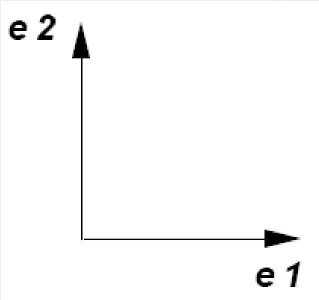




where **θ** is the angle between the two vectors.

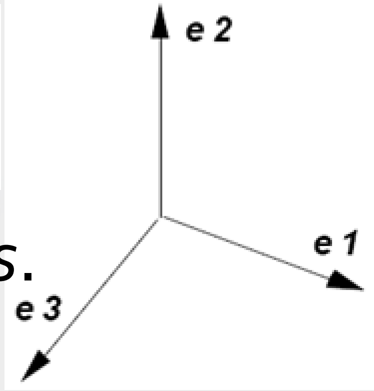
Dot (Scaler) Product





Two vectors are *orthogonal if their dot* product is zero.

The cosine of the angle between two vectors is given by $\cos\theta = \dfrac{x.y}{\|x\|.\|y\|}$



The most convenient bases are the *orthonormal bases, composed of unit vectors*.

In an orthonormal basis the inner product of two vectors is $\boldsymbol{x.y = X^T Y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n}$

where the superscript (T) denotes matrix transposition, obtained by interchanging rows with columns.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix} . \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Vector Algebra, Dot (Scaler) Product

Since the unit vectors i, j, k are mutually perpendicular,

$i \cdot i = j \cdot j = k \cdot k = 1$

$i \cdot j = i \cdot k = j \cdot k = 0.$

Since the dot product obeys the distributive law

$P1 \cdot (P2 + P3) = P1 \cdot P2 + P1 \cdot P3,$

we can easily derive the very useful equation

$$P1 \cdot P2 = (x_1 i + y_1 j + z_1 k) \cdot (x_2 i + y_2 j + z_2 k)$$
$$= (x_1 * x_2 + y_1 * y_2 + z_1 * z_2)$$

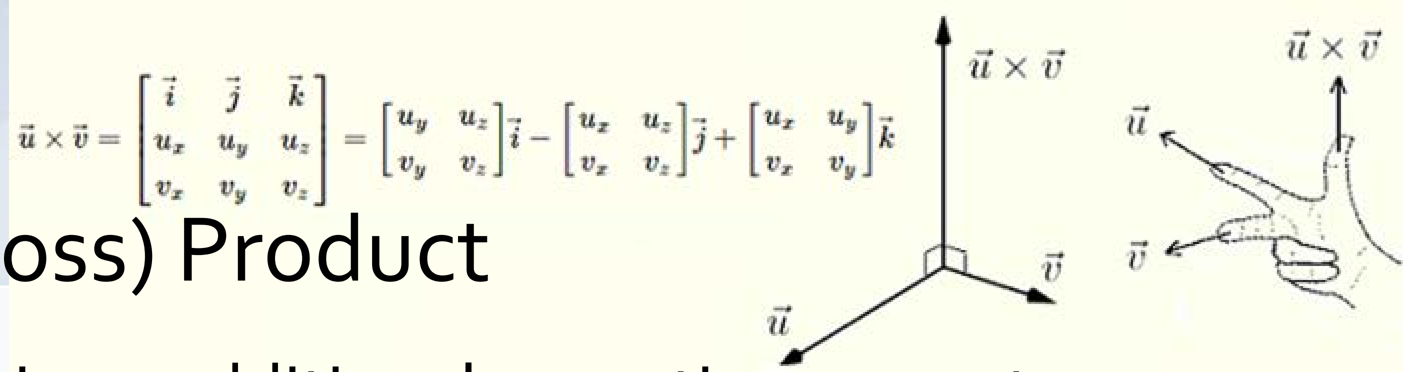
Vector Algebra, Angle between Vectors

The dot product allows us to easily compute the angle between any two vectors. From the dot product equation

$$\theta = \cos^{-1}\left(\frac{\mathbf{P}_1 \cdot \mathbf{P}_2}{|\mathbf{P}_1||\mathbf{P}_2|}\right)$$

Example. Find the angle between vectors (1, 2, 4) and (3,−4, 2).

$$\begin{aligned}
\theta &= \cos^{-1}\left(\frac{\mathbf{P}_1 \cdot \mathbf{P}_2}{|\mathbf{P}_1||\mathbf{P}_2|}\right) \\
&= \cos^{-1}\left(\frac{(1,2,4)\cdot(3,-4,2)}{|(1,2,4)||(3,-4,2)|}\right) \\
&= \cos^{-1}\left(\frac{3}{\sqrt{21}\sqrt{29}}\right) \\
&\approx 83.02^\circ
\end{aligned}$$

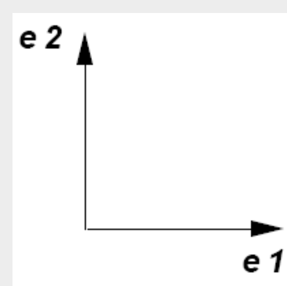


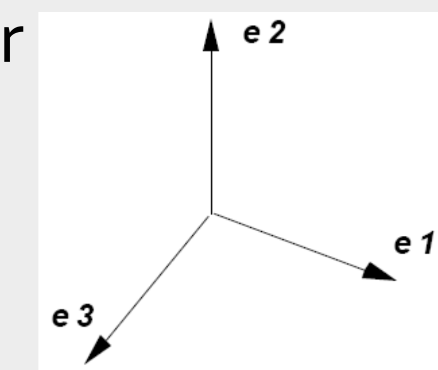
Vector (Cross) Product

Finally, there is an additional operation on vectors, called the *vector product (also known* as *cross, or exterior product), that is very useful, especially in 3-D. Here we define it in* terms of components in a right-handed, orthonormal, 3-D basis:

$$x \times y = (x_2y_3 - x_3y_2)e_1 + (x_3y_1 - x_1y_3)e_2 + (x_1y_2 - x_2y_1)e_3$$

The result of a cross product is not truly a vector, and its definition depends on the orientation or handedness of a basis.

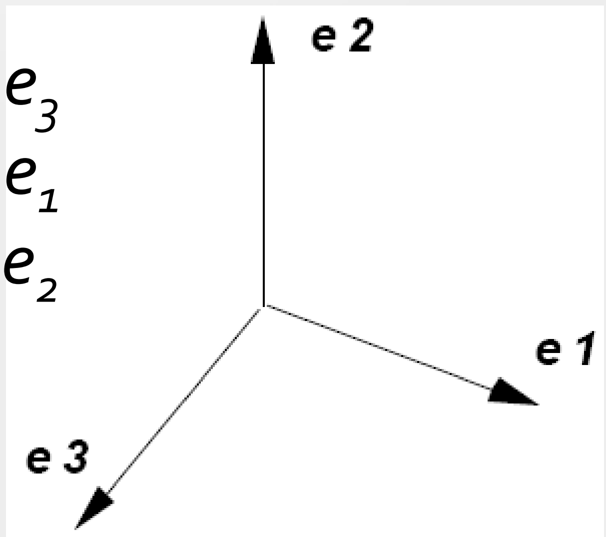


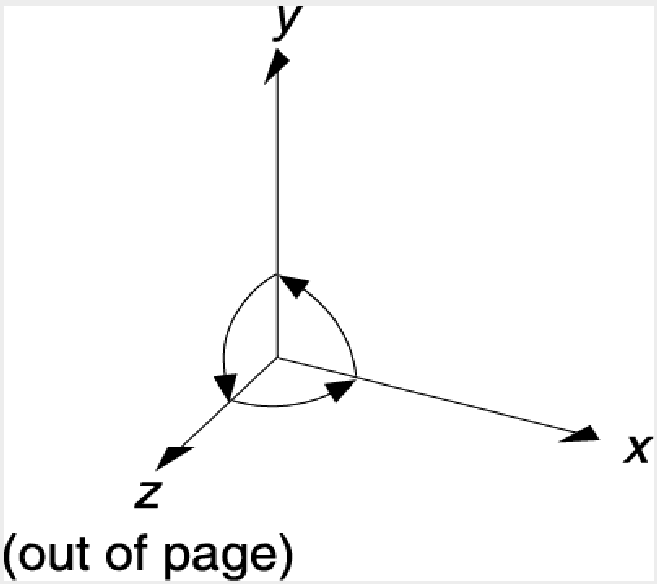


Vector (Cross) Product

The cross product of two parallel vectors is zero. For two non-parallel vectors, *x and y* , the cross-product *x × y is perpendicular to both x and y . In particular,* if *E is a righthanded* orthonormal basis in 3-D, then

$$e_1 \times e_2 = e_3$$
$$e_2 \times e_3 = e_1$$
$$e_3 \times e_1 = e_2$$





Vector (Cross) Product

Cross Product: The cross product $P_1 \times P_2$ is a vector whose magnitude is

$|P_1 \times P_2| = |P_1||P_2| \sin\theta$

(where again θ is the angle between P_1 and P_2), and whose direction is mutually perpendicular to P_1 and P_2 with a sense defined by the right hand rule as follows. Point your fingers in the direction of P_1 and orient your hand such that when you close your fist your fingers pass through the direction of P_2. Then your right thumb points in the sense of $P_1 \times P_2$.

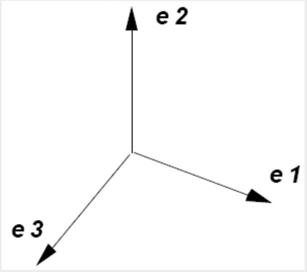
Vector (Cross) Product

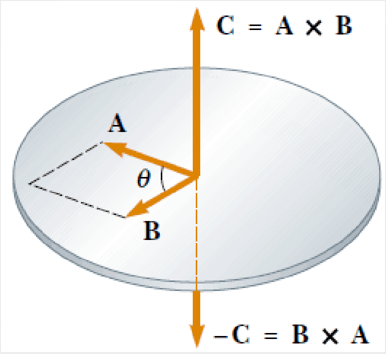
From this basic definition, one can verify that

$P_1 \times P_2 = -P_2 \times P_1$,

$i \times j = k$, $j \times k = i$, $k \times i = j$

$j \times i = -k$, $k \times j = -i$, $i \times k = -j$.



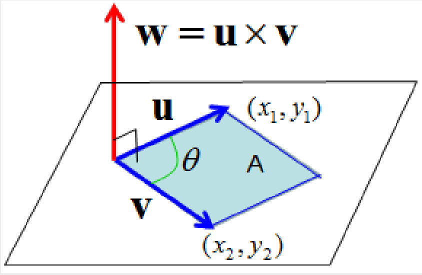


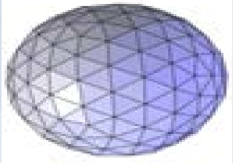
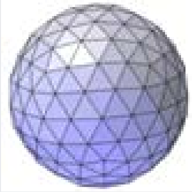
Since the cross product obeys the distributive law

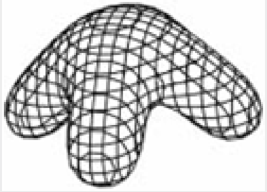
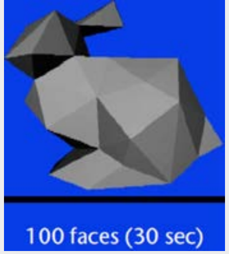
$P_1 \times (P_2 + P_3) = P_1 \times P_2 + P_1 \times P_3$,

we can derive the important relation

$$\begin{aligned} \mathbf{P}_1 \times \mathbf{P}_2 &= (x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}) \times (x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}) \\ &= (y_1 z_2 - y_2 z_1, x_2 z_1 - x_1 z_2, x_1 y_2 - x_2 y_1) \\ &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} \end{aligned}$$



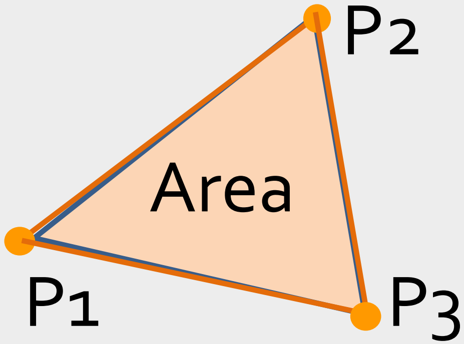



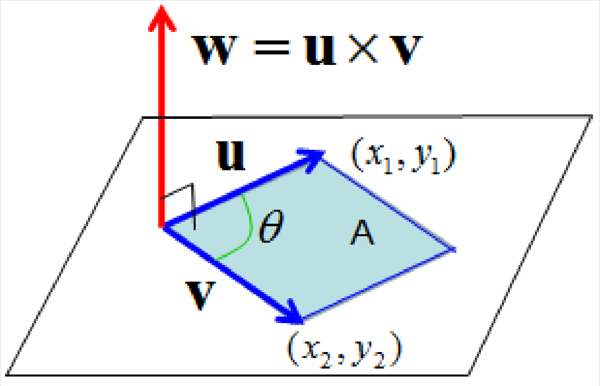



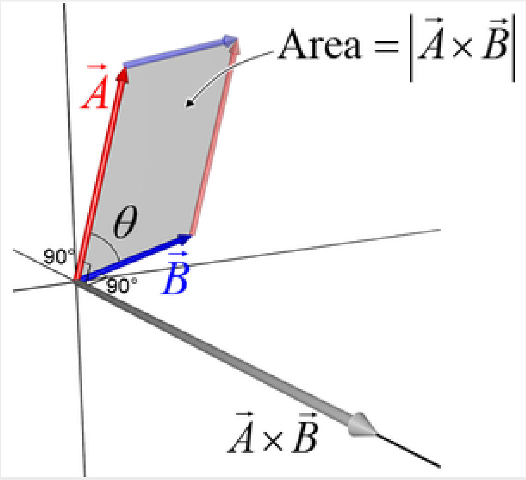
Cross Product, Area of a Triangle

Cross products have many important uses. For example, finding a vector which is perpendicular to two other vectors. Also, the cross product provides a method for finding the area of a triangle which is defined by three points P1, P2, P3 in space.

$$Area = \frac{1}{2}|\mathbf{P}_{1/2}||\mathbf{P}_{1/3}|\sin\theta_1 = \frac{1}{2}|\mathbf{P}_{1/2} \times \mathbf{P}_{1/3}|$$



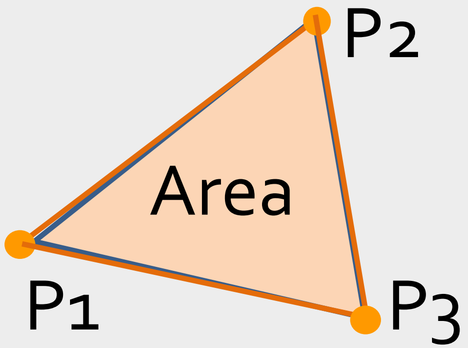


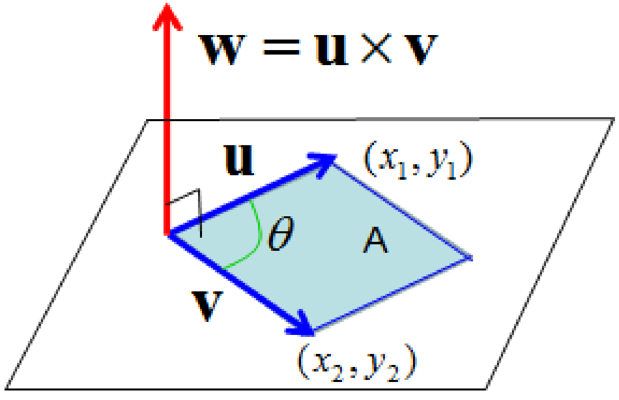


Cross Product, Area of a Triangle

For example, the area of a triangle with vertices P1 = (1, 1, 1), P2 = (2, 4, 5), P3 = (3, 2, 6) is

$$\begin{aligned} Area &= \frac{1}{2}|\mathbf{P}_{1/2} \times \mathbf{P}_{1/3}| \\ &= \frac{1}{2}|(1,3,4) \times (2,1,5)| \\ &= \frac{1}{2}|(11,3,-5)| = \frac{1}{2}\sqrt{11^2 + 3^2 + (-5)^2} \\ &\approx 6.225 \end{aligned}$$





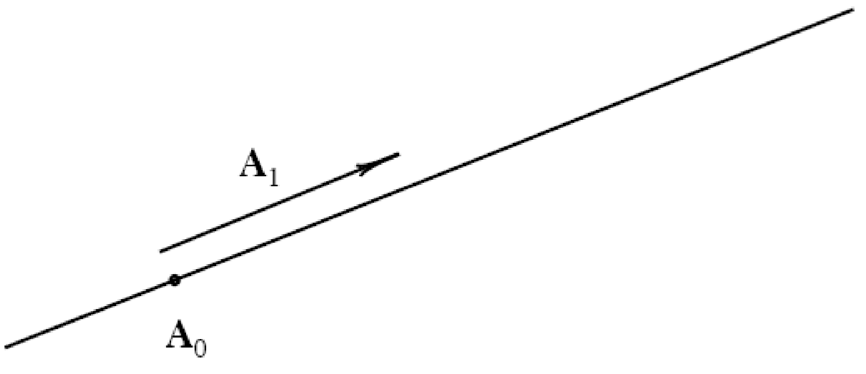
Parametric equation of Line

A line can be defined using either a parametric equation or an implicit equation.

Parametric equations of lines

Linear parametric equation. A line can be written in parametric form as follows:

$x = a_0 + a_1 t; \quad y = b_0 + b_1 t$



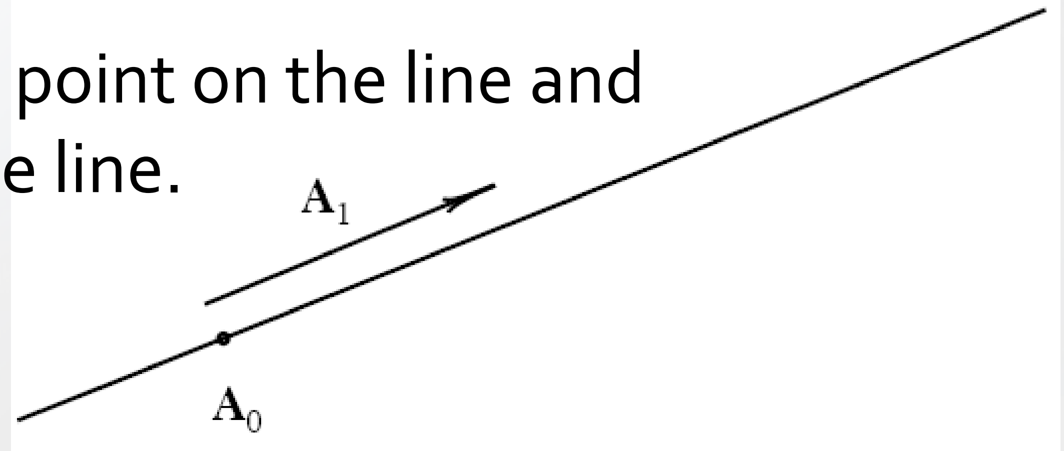
In vector form,

$$\mathbf{P}(t) = \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix} = \begin{Bmatrix} a_0 + a_1 t \\ b_0 + b_1 t \end{Bmatrix} = \mathbf{A}_0 + \mathbf{A}_1 t.$$

Parametric equation of Line

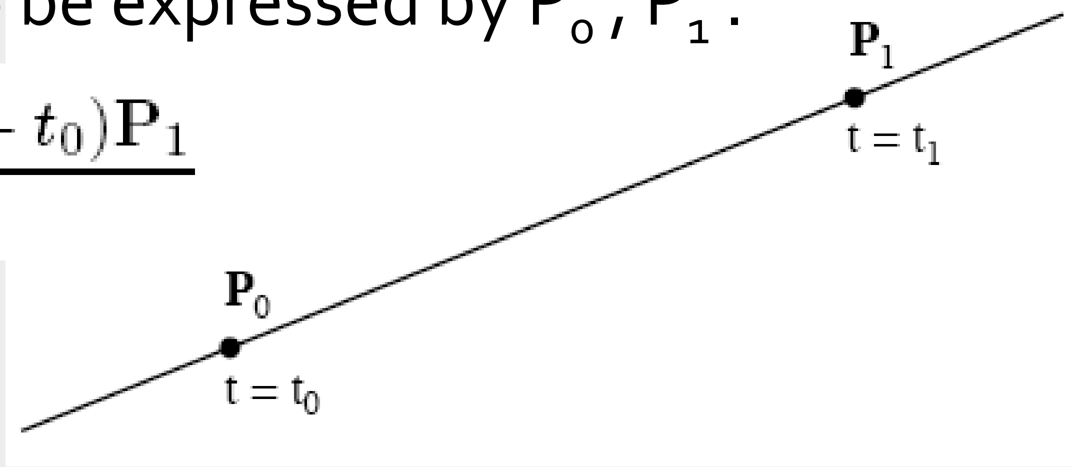
In this equation, A_0 is a point on the line and A_1 is the direction of the line.

Line given by $A_0 + A_1 t$



Affine parametric equation of a line (between P_0, P_1).
A straight line can also be expressed by P_0, P_1.

$$\mathbf{P}(t) = \frac{(t_1 - t)\mathbf{P}_0 + (t - t_0)\mathbf{P}_1}{t_1 - t_0}$$



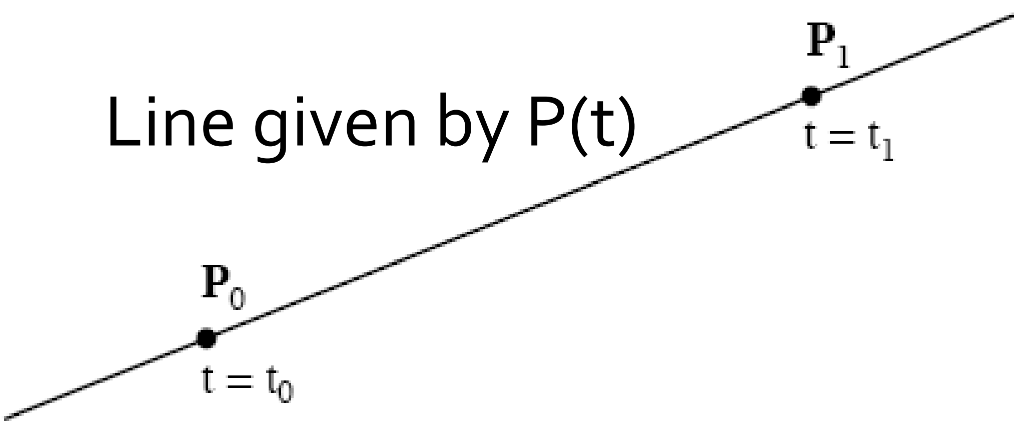
Parametric equation of Line

where P_0 and P_1 are two points on the line and t_0 and t_1 are any parameter values. Note that

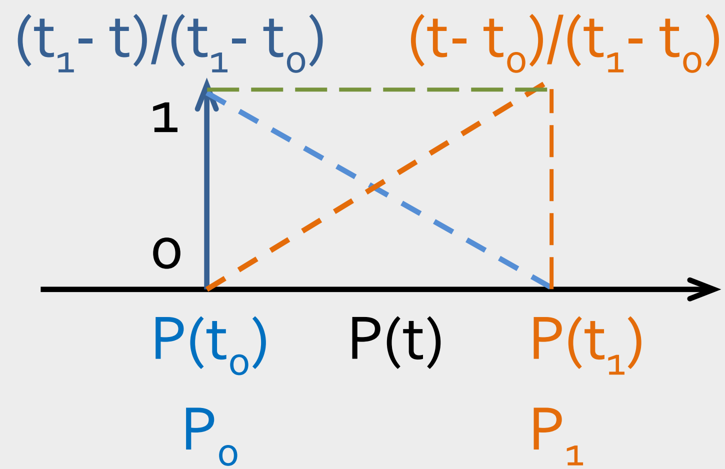
$P(t_0) = P_0$ and $P(t_1) = P_1$.

Note in Figure that the line segment P_0–P_1 is defined by restricting the parameter: $t_0 \le t \le t_1$.

$$\mathbf{P}(t) = \frac{(t_1 - t)\mathbf{P}_0 + (t - t_0)\mathbf{P}_1}{t_1 - t_0}$$



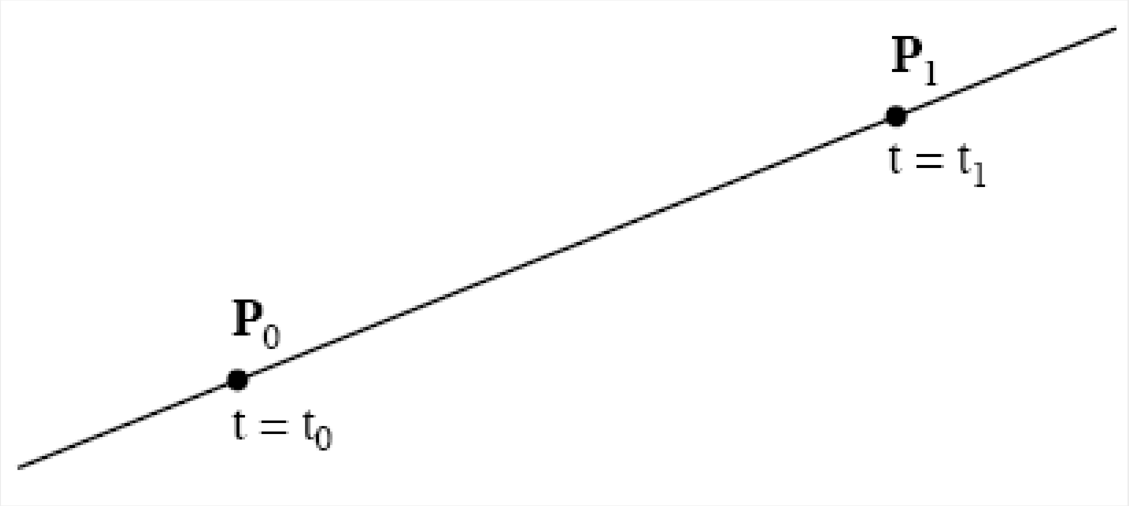
Line given by P(t)

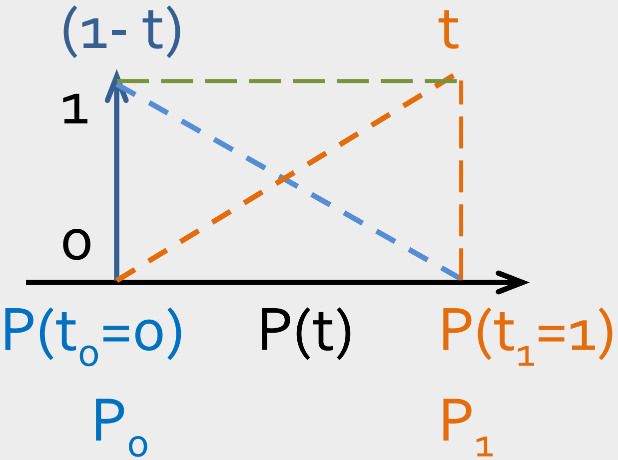


Linear interpolation

Parametric equation of Line

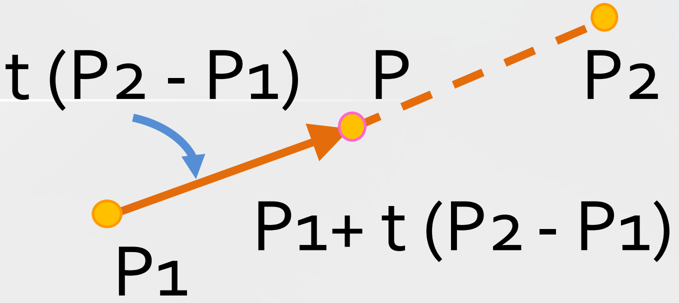
Sometimes this is expressed by saying that the line segment is the portion of the line in the parameter interval or domain $[t_0, t_1]$. We will soon see that the line in Figure is actually a degree one Bezier curve. Most commonly, we have $t_0 = 0$ and $t_1 = 1$ in which case $P(t) = (1 - t)P_0 + tP_1$.



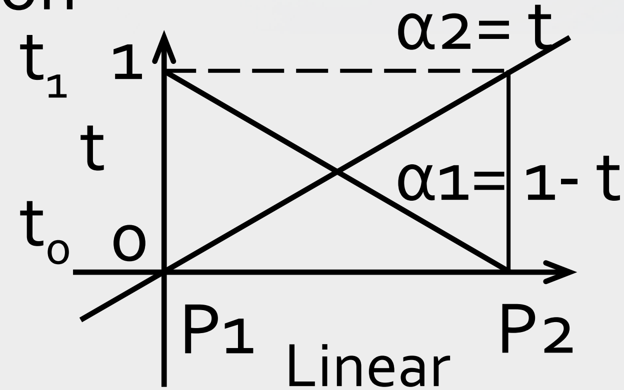


Linear interpolation

Line
(Combinations of Points)



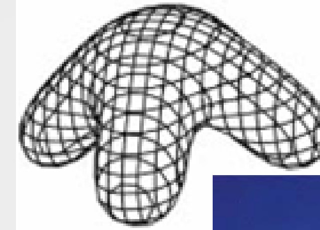
- Let P_1 and P_2 be points in space.
- if $0 \le t \le 1$ then P is somewhere on the line segment joining P_1 and P_2 .
- We may utilize the following notation
  $P = P(t) = (1 - t)\, P_1 + t\, P_2$



Linear interpolation

- We can then define a combination of two points P_1 and P_2 to be
  $P = \alpha_1 P_1 + \alpha_2 P_2$ where $\alpha_1 + \alpha_2 = 1$
- derive the transformation by setting $\alpha_2 = t$

Linear Parametric Plane Surface



We can generalize the line to define
a combination of an arbitrary number of points.

$\mathbf{P} = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$

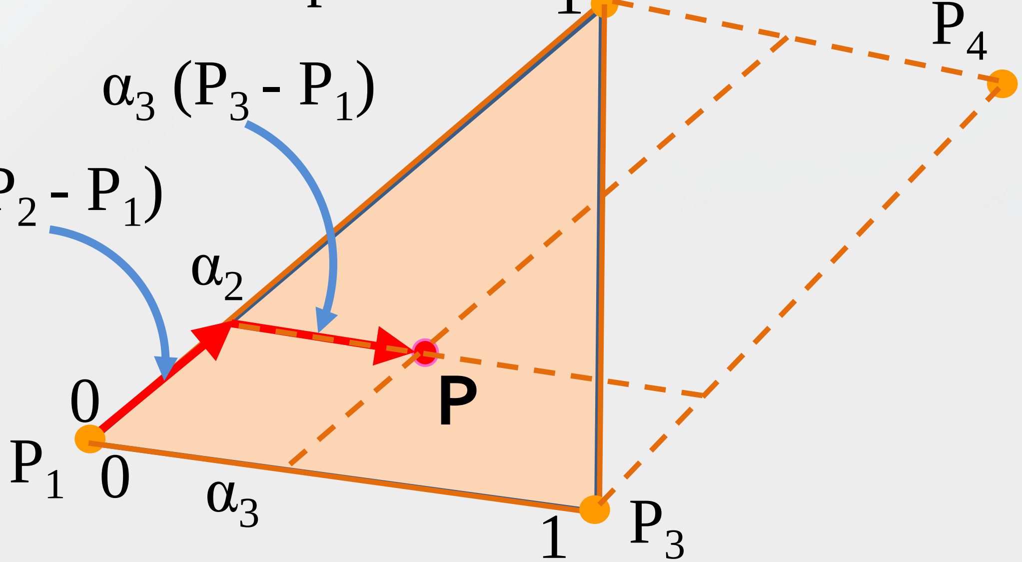
where $\alpha_1 + \alpha_2 + \alpha_3 = 1$

$0 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$

Illustration shows the point **P** generated when

$\alpha_2 = 1/4$, $\alpha_3 = 1/2$,

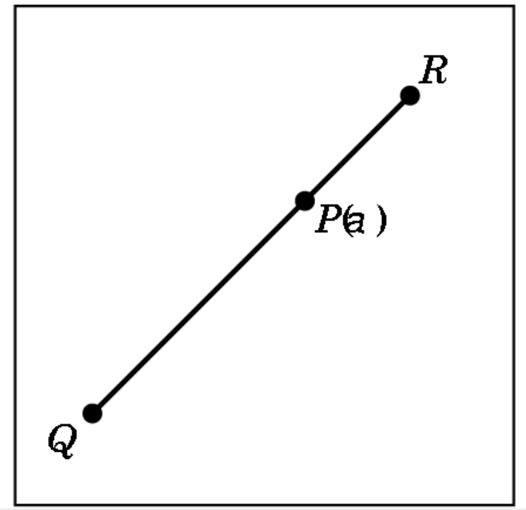
$\alpha_1 = 1 - \alpha_2 - \alpha_3 = 1/4$.



Then, each vertex of our triangle could be described in terms of its respective distance from the two walls containing the origin (P_1) and from the floor.

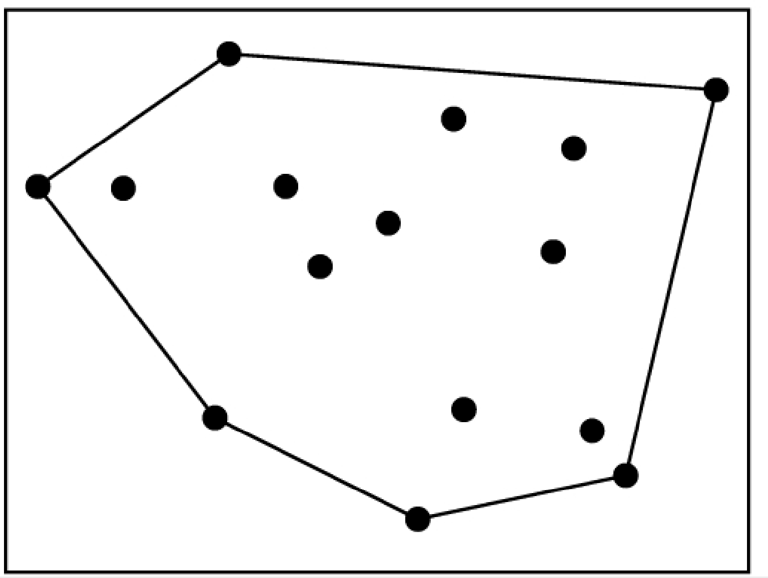
$\mathbf{P} = P_1 + \alpha_2 (P_2 - P_1) + \alpha_3 (P_3 - P_1)$ $\quad$ $\mathbf{P(u,v)} = (1 - u - v) P_1 + u P_2 + v P_3$

Convexity



A **convex object** is one for which any point lying on the line segment connecting any two points in the object is also in the object

$$P = \alpha_1 R + \alpha_2 Q \qquad \& \quad \alpha_1 + \alpha_2 = 1$$



More general form

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n$$

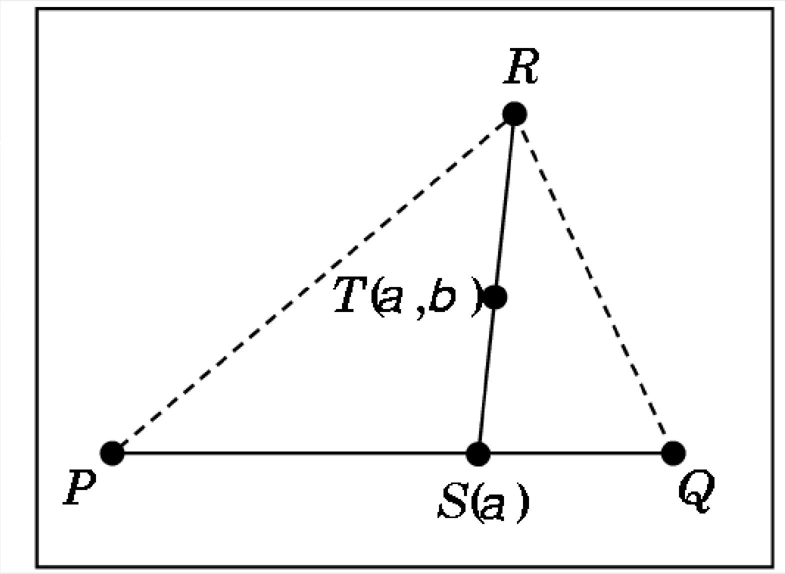
where

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$$

&

$$\alpha_i \geq 0 \,,\ i = 1, 2, \dots, n$$

Parametric Plane



Let *P*, *Q*, *R* are points defining a plane in an affine space

$S(\alpha) = \alpha P + (1- \alpha)Q \; , \; 0 \le \alpha \le 1$

$T(\beta) = \beta S + (1 - \beta) R \; , \; 0 \le \beta \le 1$

using a substitution

$T(\alpha,\beta) = \beta [\alpha P + (1- \alpha)Q] + (1 - \beta) R \; ,$

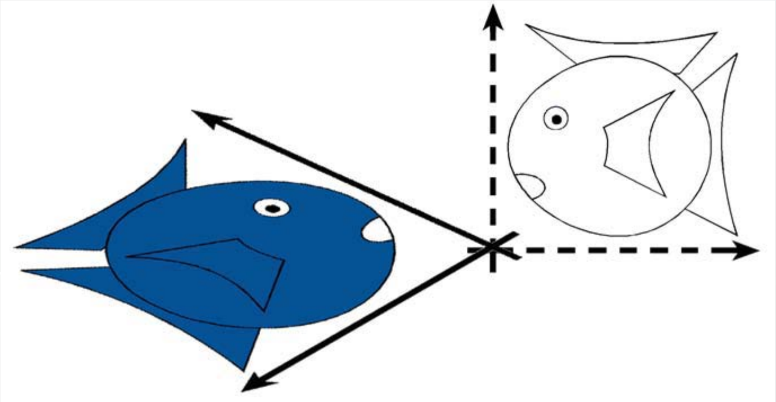
$0 \le \alpha \le 1 \; \& \; 0 \le \beta \le 1$

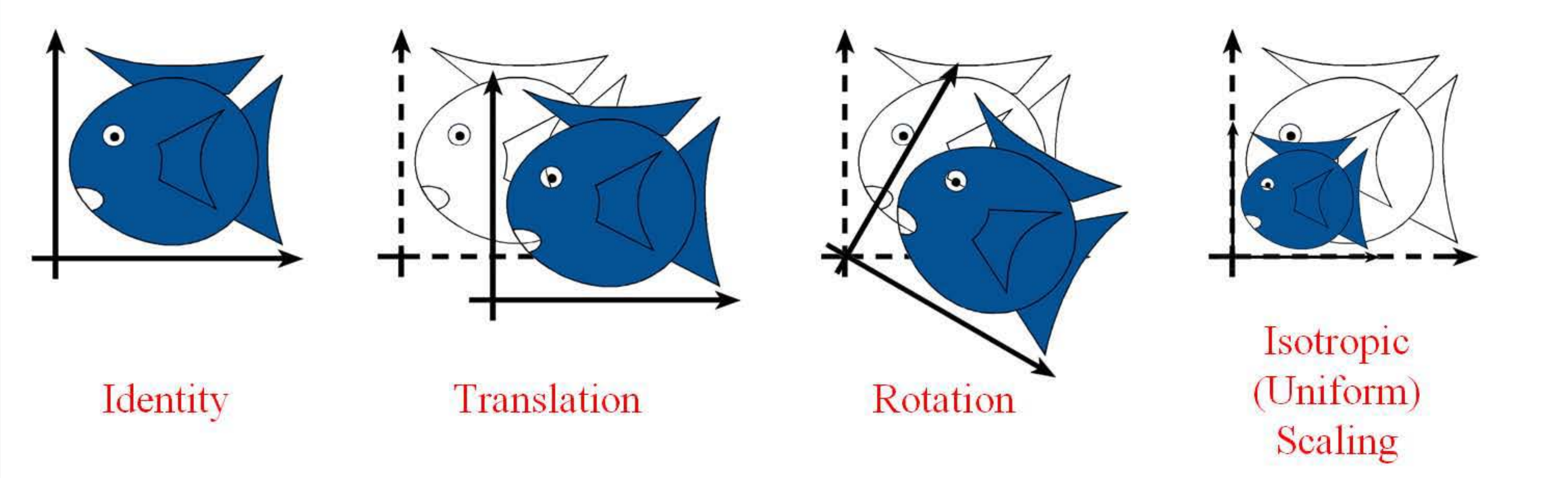
$T(\alpha,\beta) = P + \beta (1 - \alpha)(Q - P) + (1 - \beta) (R - P)$

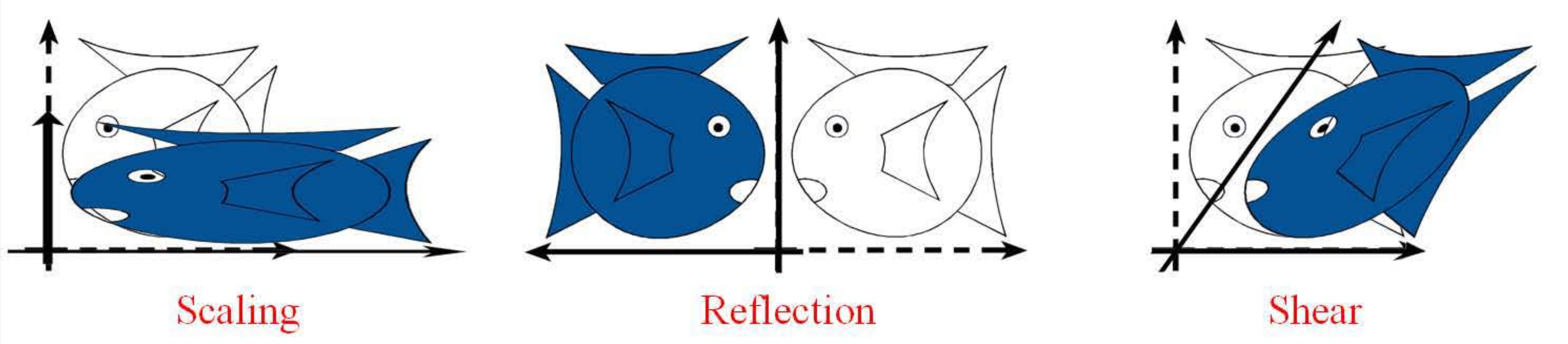
Plane given by a point P_0 and vectors *u*, *v*

$T(\alpha,\beta) = P_0 + \alpha u + \beta v \quad \& \quad 0 \le \alpha , \; \beta \le 1$

Linear Transformations



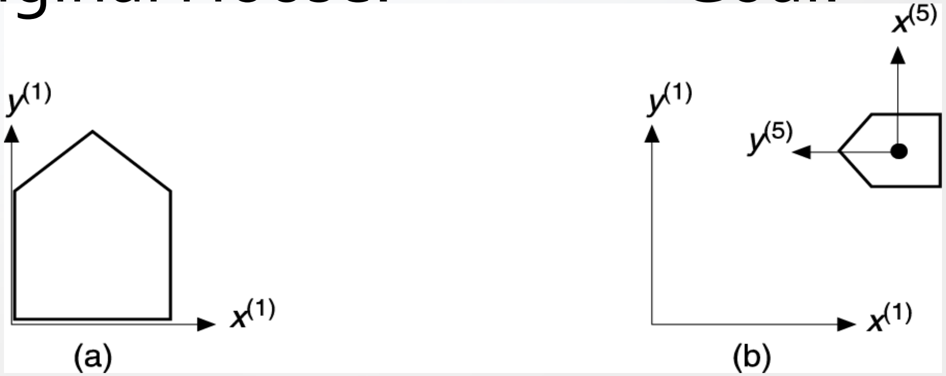




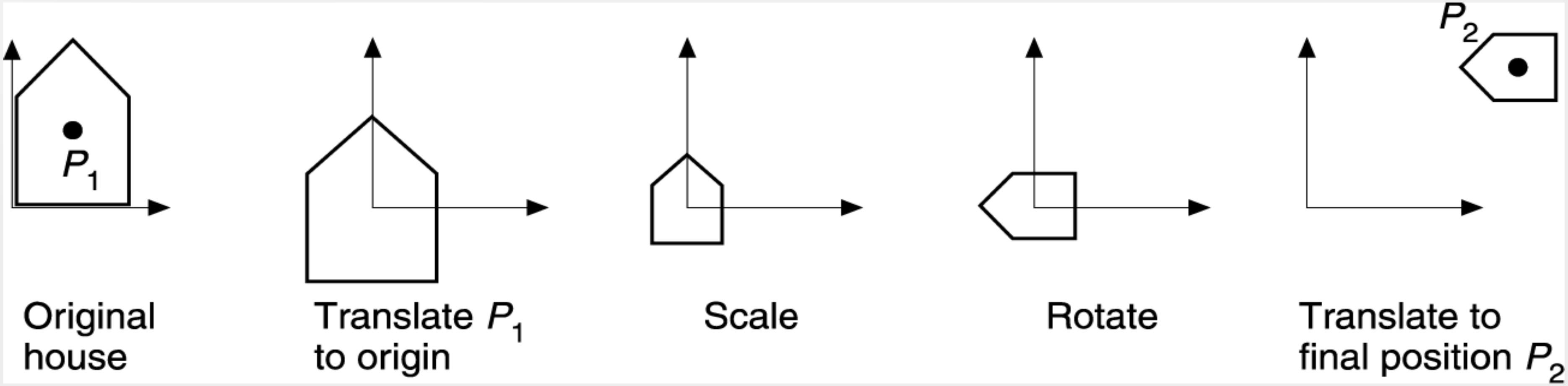
Combining Transformations

Example: Transformation of the House

Original House: Goal:

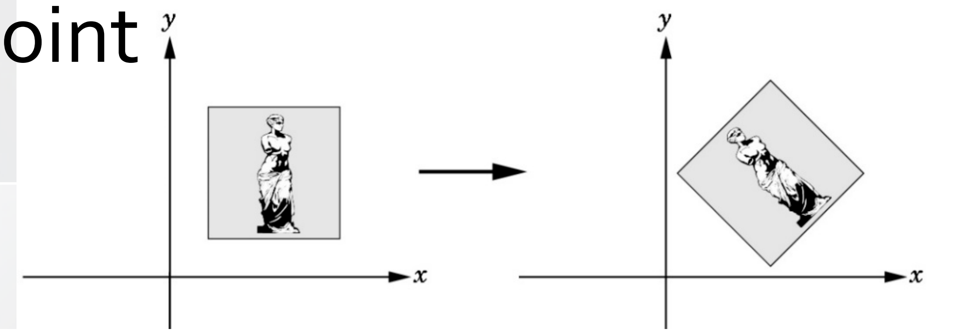


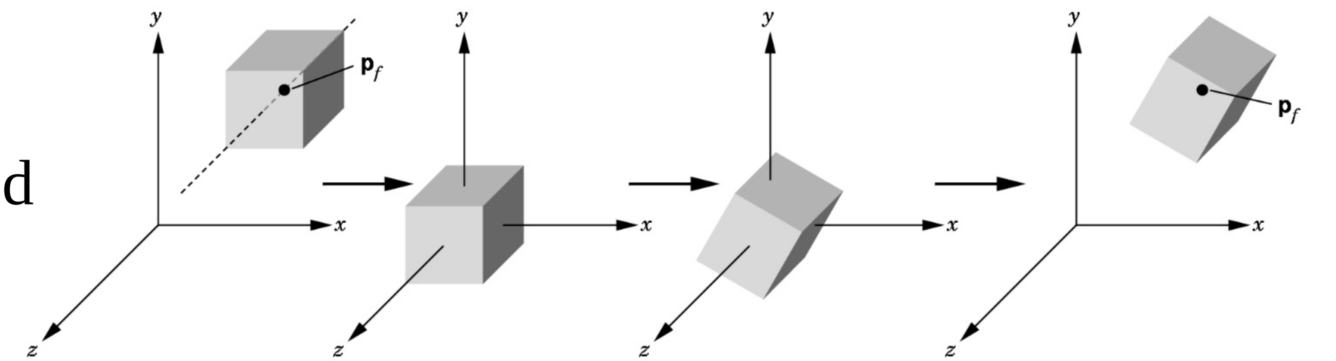
Transformation Composition:

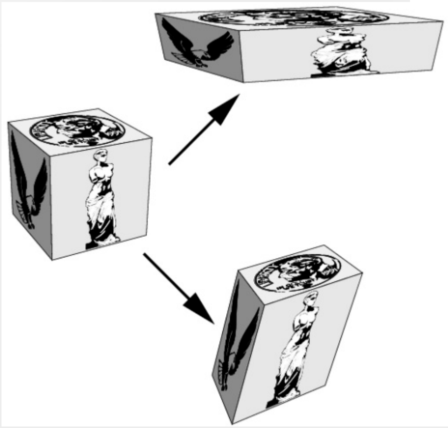


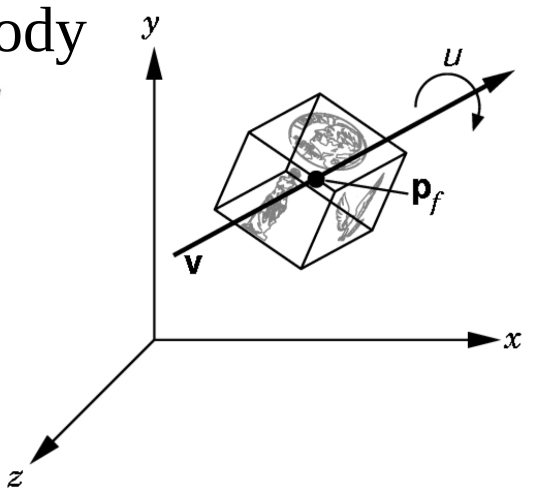
Rotation about a fixed point Transformations

For rotation – implicit point
– origin
– 2D – simple
– 3D – complicated
Transformation
– rigid-body
– non-rigid-body

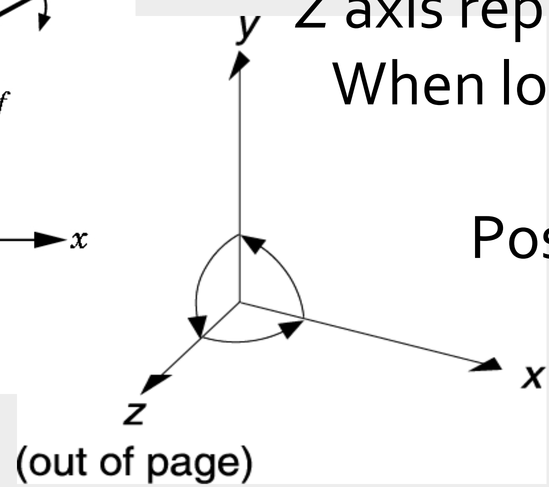






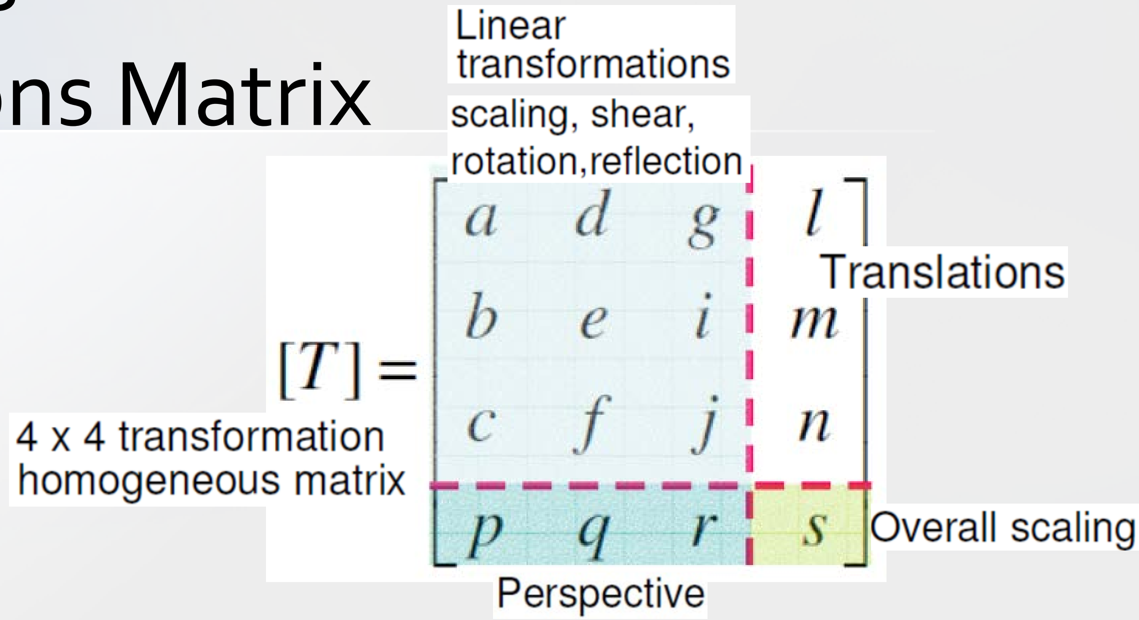


Right Handed System
Z axis represents depth
When looking “down” at the origin,
Positive rotation is CCW.



Homogeneous Transformations Matrix

Combining Transformations



Using homogeneous transformation matrix allows us use matrix multiplication to calculate all kind of transformations, so combine all in one matrix.

Scale $P' = S.P$, Translation $P'=P+d$ => $P' = T.P$,

Rotation $P'=R.P$ Combined $P'=T.R.S.T^{-1}.P$

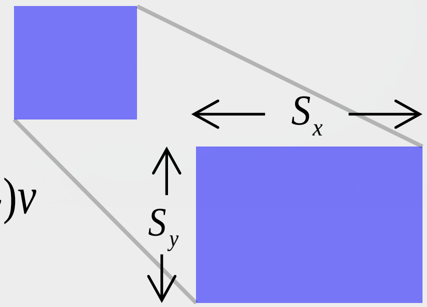
Homogenous Transformations

$$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1*x+0*y+dx*1 \\ 0*x+1*y+dy*1 \\ 0*x+0*y+1*1 \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \\ 1 \end{bmatrix}$$

Homogenous transformations for 2D space requires 3D vectors & matrices.

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \times x \\ s_y \times y \\ 1 \end{bmatrix} : v' = S(s_x, s_y)v$$



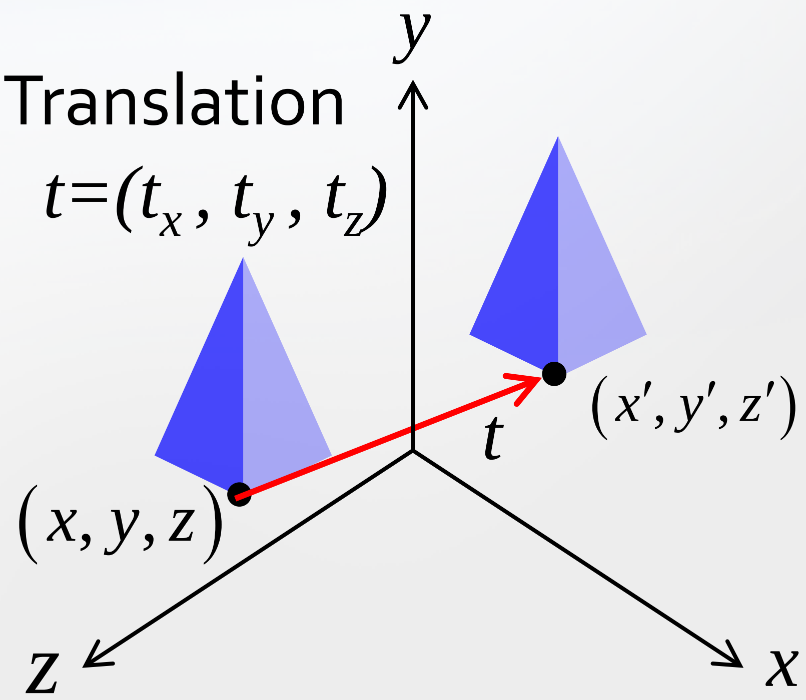
Homogenous transformations for 3D space requires 4D vectors & matrices.

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P = [x, y, z, 1]^T$

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous 3D Translation Matrix



$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

$$P' = P + t$$

$$P' = T.P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Translation of a Curve

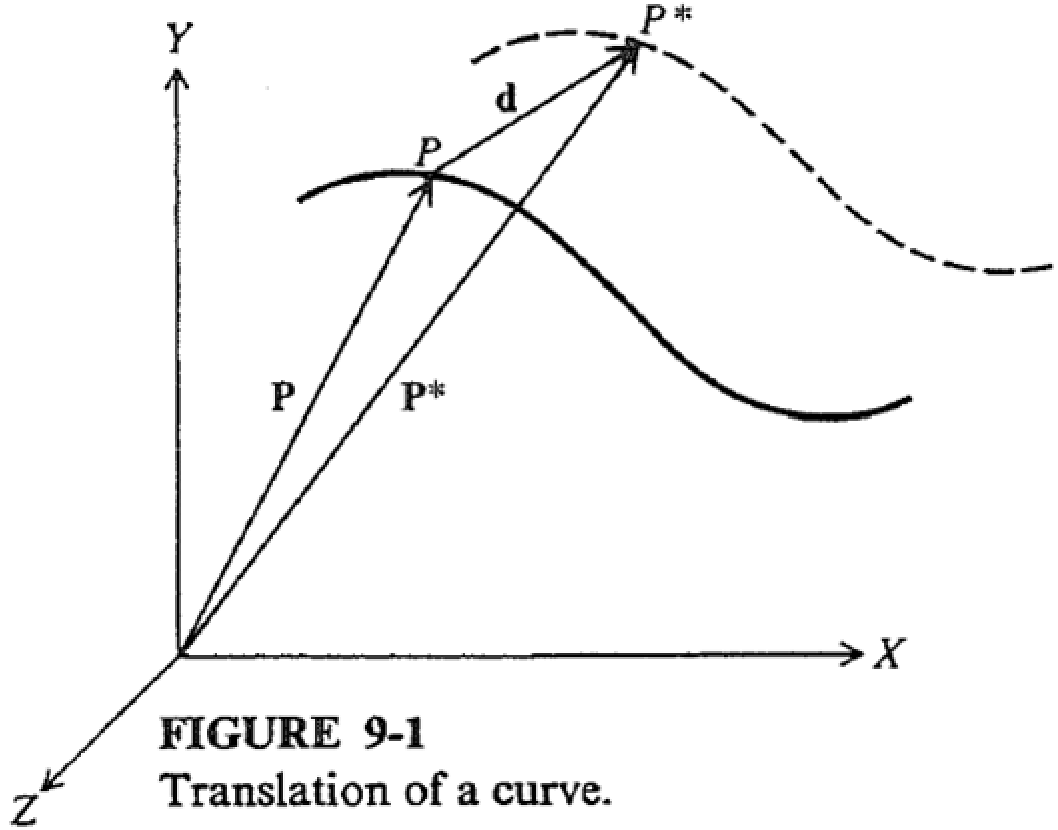


FIGURE 9-1
Translation of a curve.

$$\mathbf{P}^* = \mathbf{P} + \mathbf{d} \qquad (9.3)$$

$$x^* = x + x_d$$

$$y^* = y + y_d \qquad (9.4)$$

$$z^* = z + z_d$$

$$\mathbf{P}^* = [T]\mathbf{P}$$

where $[T]$ is the transformation matrix

3D Homogenous translation

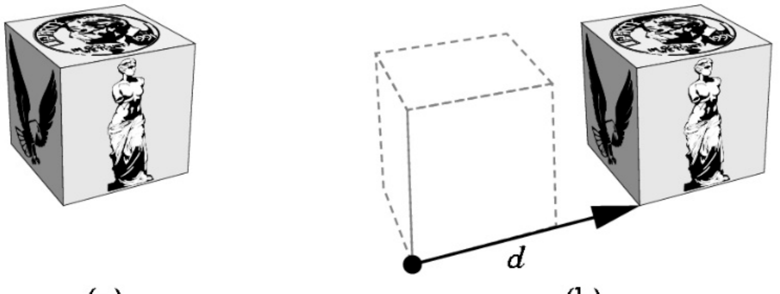
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Transformations: Scale & Translate

Scale, Parameters
for each axis direction
$P' = S.P$
Translation
$P' = T.P$
$P = [x, y, z, 1]^T$
$P' = P + d$
2D homegenous Translation

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a) (b)

$$Y^* = M^* X^* = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix}$$

Scaling

$$\mathbf{P}^* = [S]\mathbf{P} \qquad (9.9)$$

where $[S]$ is a diagonal matrix.

In three dimensions, it is given by

$$[S] = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \qquad (9.10)$$

Thus (9.9) can be expanded to give

$$x^* = s_x x \qquad y^* = s_y y \qquad z^* = s_z z$$

$$\mathbf{P}^* = s\mathbf{P} \qquad (9.12)$$

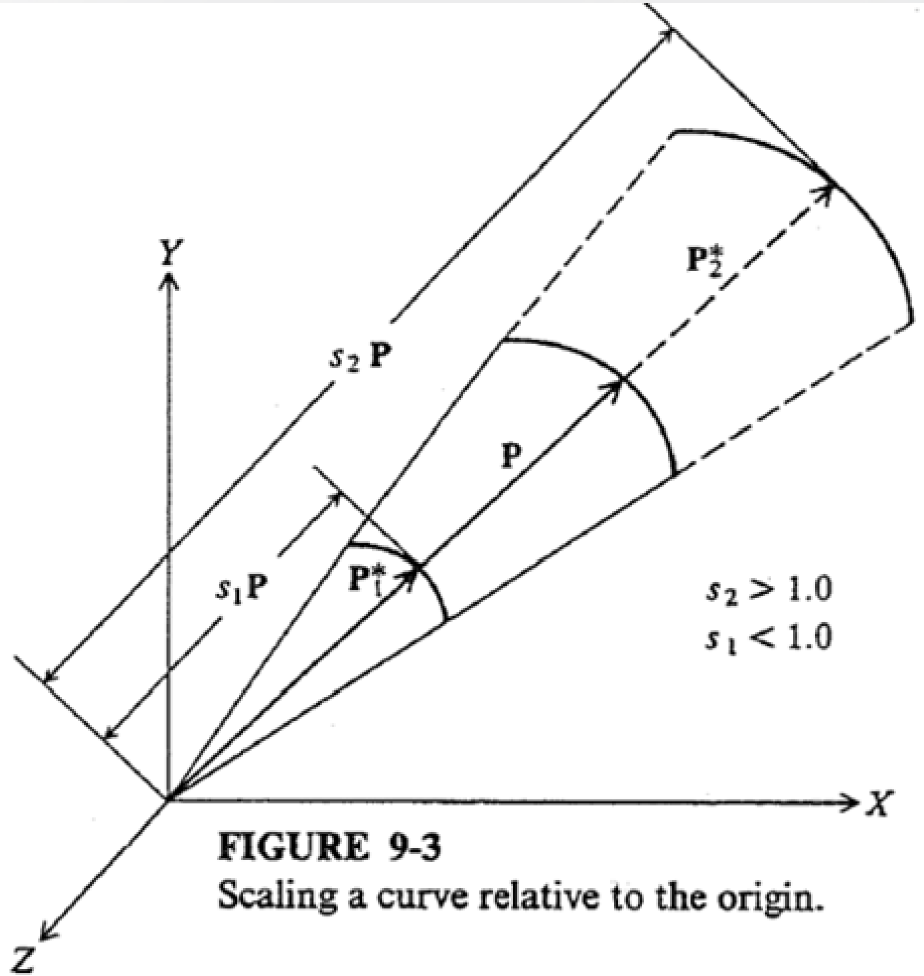


FIGURE 9-3
Scaling a curve relative to the origin.

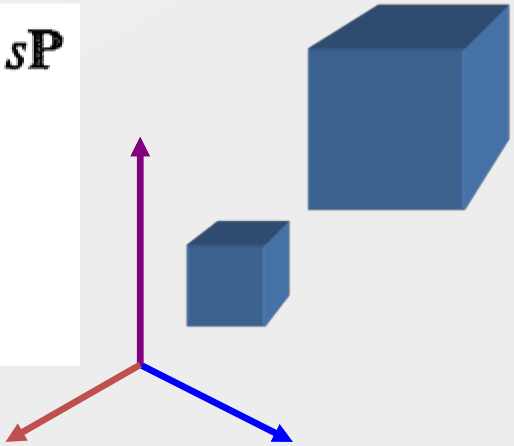
Scaling

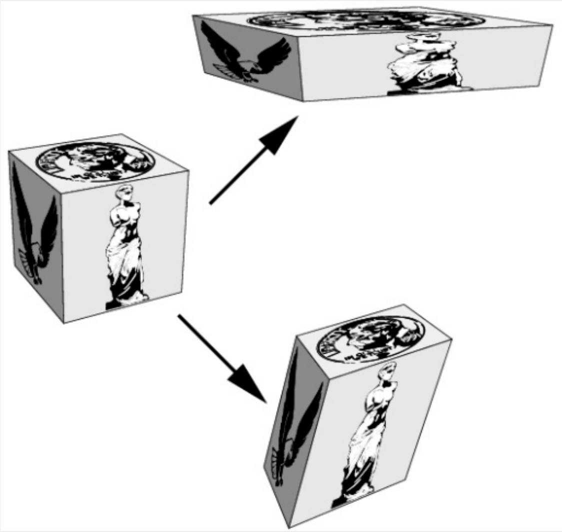
If the scale factors are equal,
$s_x = s_y = s_z = s$,
the model changes in size only
and not in shape;
this is the case of uniform scaling.

Differential scaling occurs
when $s_x \neq s_y \neq s_z$;
that is, different scaling
factors are applied
in different directions.

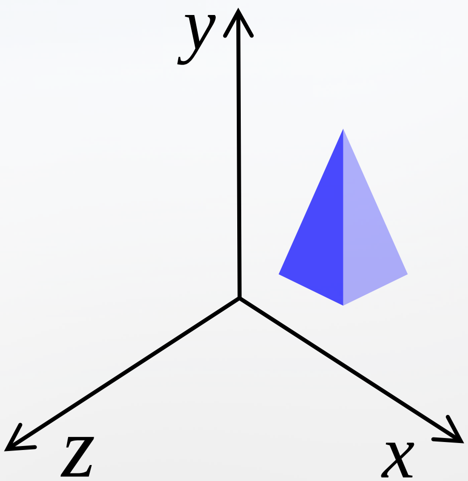
$$\mathbf{P}^* = [S]\mathbf{P} \qquad \mathbf{P}^* = s\mathbf{P}$$

$$[S] = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$





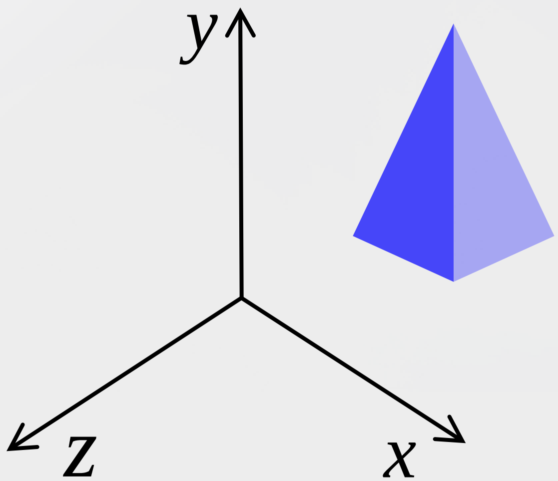
Homogenous 3D Scaling matrix



$$x' = x \cdot S_x$$

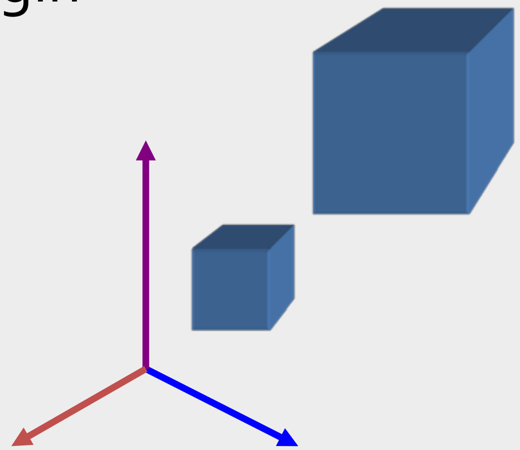
$$y' = y \cdot S_y$$

$$z' = x \cdot S_z$$

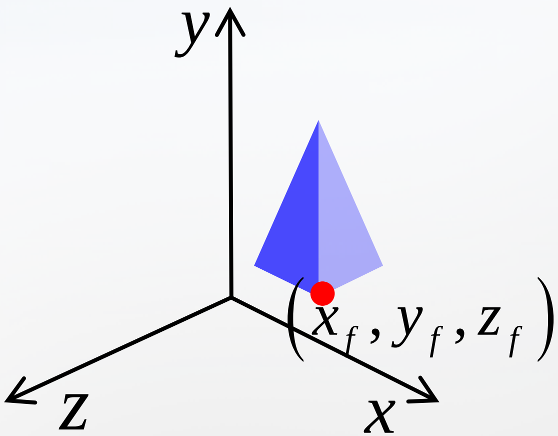


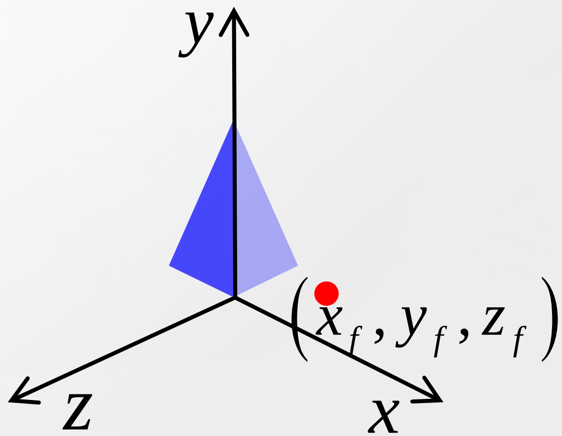
Enlarging object also moves it from origin

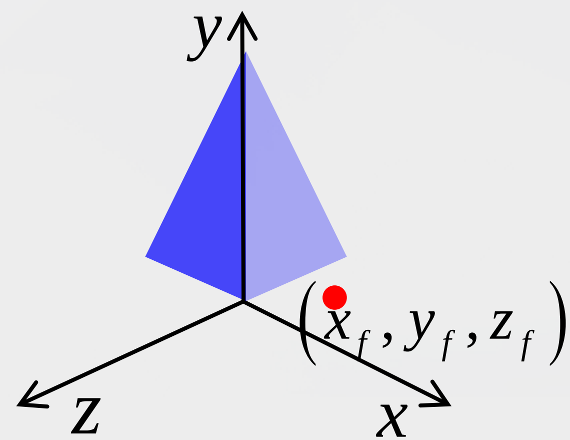
$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{S} \cdot \mathbf{P}$$

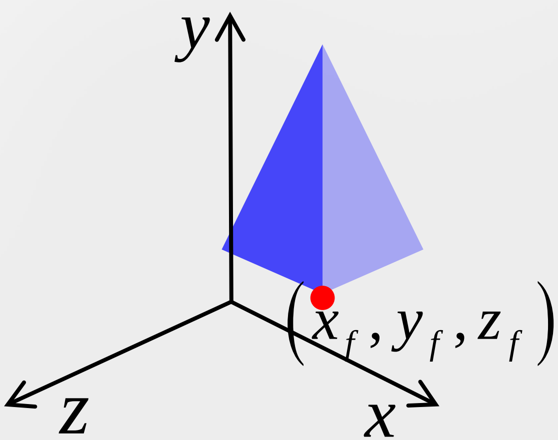


Scaling with respect to a fixed point (not necessarily of object)









$$\mathbf{T}\cdot\mathbf{S}\cdot\mathbf{T}^{-1} = \begin{bmatrix} S_x & 0 & 0 & (1-S_x)x_f \\ 0 & S_y & 0 & (1-S_y)y_f \\ 0 & 0 & S_z & (1-S_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & (1-S_x)x_f \\ 0 & S_y & 0 & (1-S_y)y_f \\ 0 & 0 & S_z & (1-S_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{T}\cdot\mathbf{S}\cdot\mathbf{T}^{-1}\cdot\mathbf{P}$$

Rotation of a point about z axis

$$x^* = r\cos(\theta + \alpha) = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta$$

$$y^* = r\sin(\theta + \alpha) = r\sin\alpha\cos\theta + r\cos\alpha\sin\theta$$

$$z^* = z$$

where $r = |\mathbf{P}| = |\mathbf{P}^*|$

Substituting

$$x = r\cos\alpha$$

$$y = r\sin\alpha$$

gives

$$x^* = x\cos\theta - y\sin\theta$$

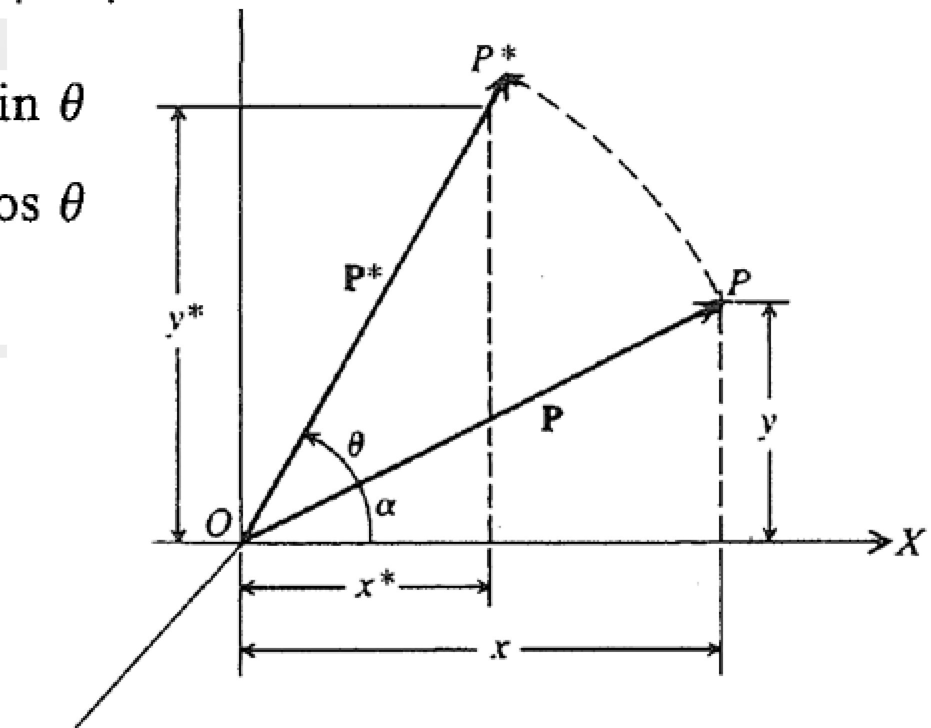
$$y^* = x\sin\theta + y\cos\theta$$

$$z^* = z$$

Rewriting Eqs. in a matrix form gives

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

or $\mathbf{P}^* = [R_Z]\mathbf{P}$



Rotation of a point about the Z axis.

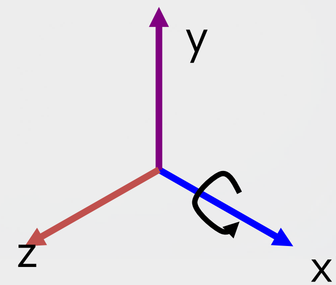
2D Rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta \times x - \sin\theta \times y \\ \sin\theta \times x + \cos\theta \times y \\ 1 \end{bmatrix} : P' = R \cdot P$$

3D Rotation about a major axis $P' = R \cdot P$

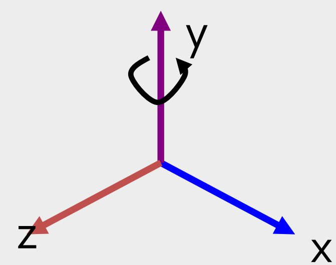
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



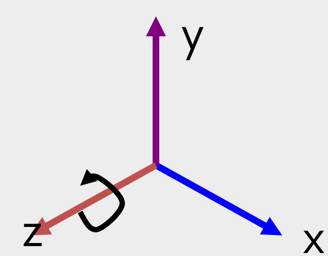
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



2D Inverse Transformations

Transformations can easily be reversed using inverse transformations

$$T^{-1} = \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

$$S^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3D inverse Transformations

Translation

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \alpha_x \\ 0 & 1 & 0 & \alpha_y \\ 0 & 0 & 1 & \alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

$$S = \begin{bmatrix} \beta_x & 0 & 0 & 0 \\ 0 & \beta_y & 0 & 0 \\ 0 & 0 & \beta_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inversion operations:

$$T^{-1} = T\,(\,-\alpha_x\,,\,-\alpha_y\,,\,-\alpha_z\,)$$

$$S^{-1} = S\,(\,1/\beta_x\,,\,1/\beta_y\,,\,1/\beta_z\,)$$

Composite translations

$$\mathbf{P}' = \mathbf{T}\left(t_{2x}, t_{2y}\right)\left\{\mathbf{T}\left(t_{1x}, t_{1y}\right)\cdot\mathbf{P}\right\} = \left\{\mathbf{T}\left(t_{2x}, t_{2y}\right)\cdot\mathbf{T}\left(t_{1x}, t_{1y}\right)\right\}\cdot\mathbf{P}$$

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x}+t_{2x} \\ 0 & 1 & t_{1y}+t_{2y} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}\left(t_{2x}, t_{2y}\right)\cdot\mathbf{T}\left(t_{1x}, t_{1y}\right) = \mathbf{T}\left(t_{1x}+t_{2x}, t_{1y}+t_{2y}\right)$$

Composite Rotations:

$$\mathbf{P}' = \mathbf{R}\left(\theta_2\right)\left\{\mathbf{R}\left(\theta_1\right)\cdot\mathbf{P}\right\} = \left\{\mathbf{R}\left(\theta_2\right)\cdot\mathbf{R}\left(\theta_1\right)\right\}\cdot\mathbf{P}$$

$$\mathbf{R}\left(\theta_2\right)\cdot\mathbf{R}\left(\theta_1\right) = \mathbf{R}\left(\theta_1+\theta_2\right)$$

$$\mathbf{P}' = \mathbf{R}\left(\theta_1+\theta_2\right)\cdot\mathbf{P}$$

Combining Transformations

The three transformation matrices are combined as follows

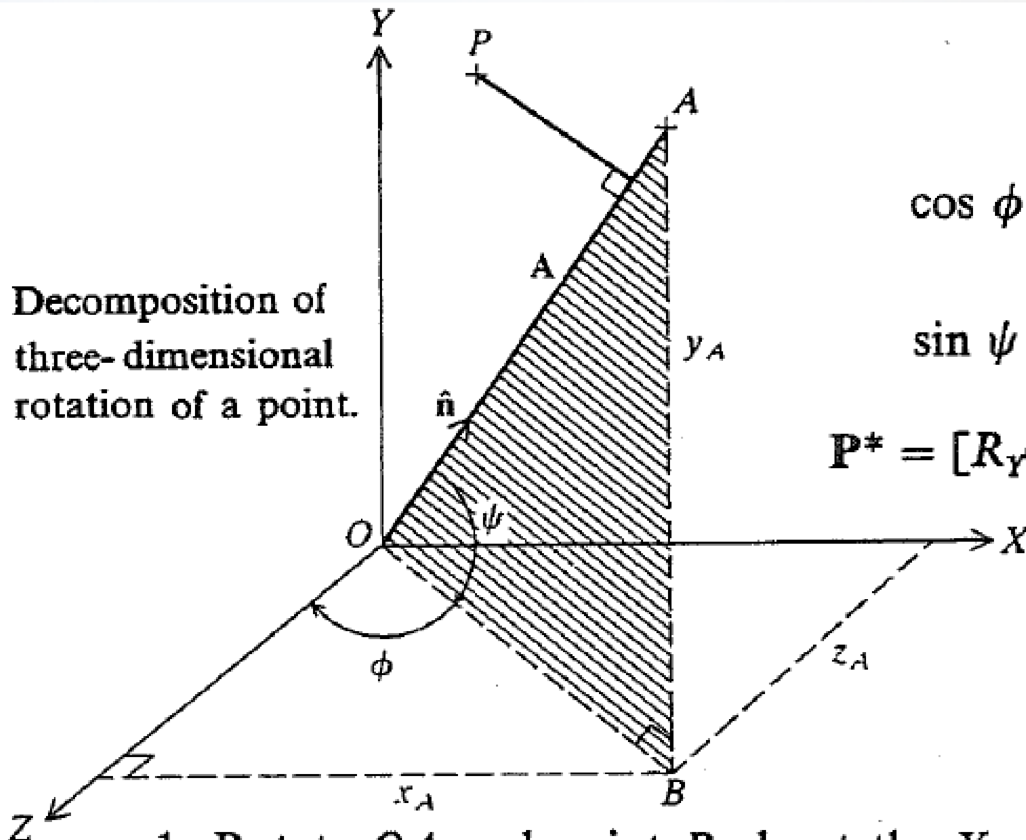
$$\begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$v' = T(-dx,-dy)R(\theta)T(dx,dy)v$$

Matrix multiplication is not commutative so order matters

$$\mathbf{P}' = \mathbf{M}_2\left(\mathbf{M}_1 \cdot \mathbf{P}\right) = \left(\mathbf{M}_2 \cdot \mathbf{M}_1\right) \cdot \mathbf{P} = \mathbf{M} \cdot \mathbf{P}$$

Rotation about an arbitrary axis $\boldsymbol{n}$ (n_x, n_y, n_z)



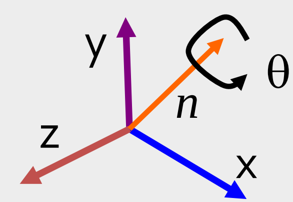
Decomposition of three-dimensional rotation of a point.

$$\tan \phi = \frac{x_A}{z_A} = \frac{x_A/|\mathbf{A}|}{z_A/|\mathbf{A}|} = \frac{n_x}{n_z}$$

$$\cos \phi = \frac{n_z}{\sqrt{n_x^2 + n_z^2}} \qquad \sin \phi = \frac{n_x}{\sqrt{n_x^2 + n_z^2}}$$

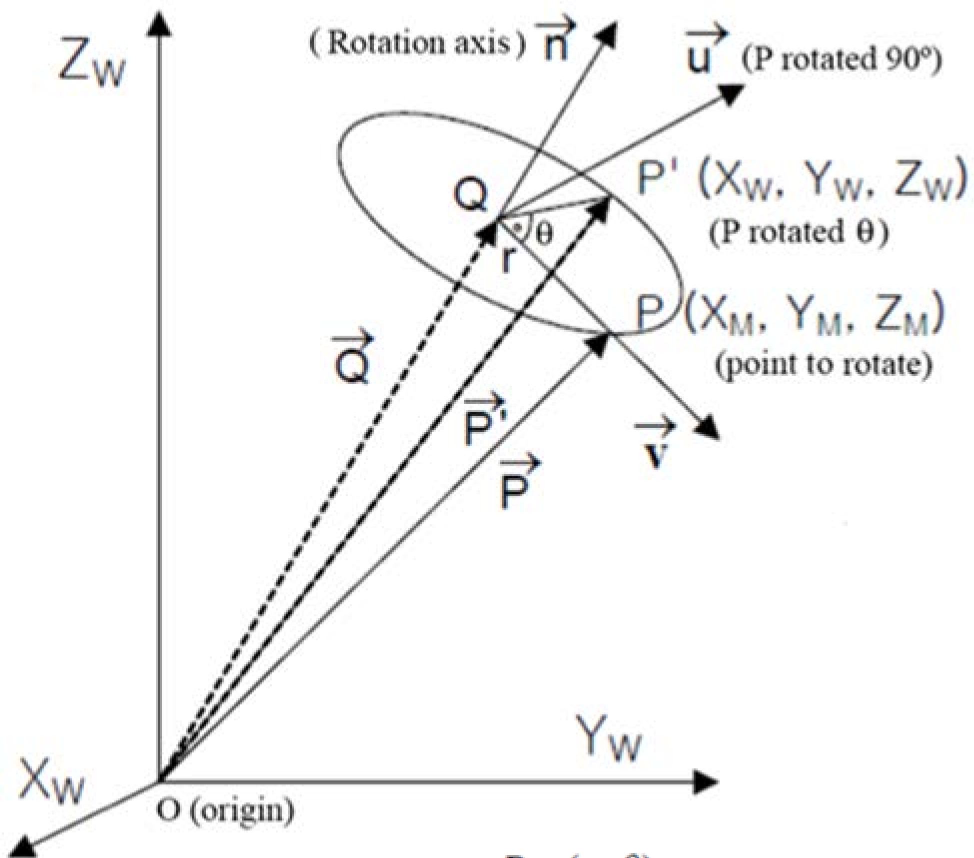
$$\sin \psi = \frac{y_A}{|\mathbf{A}|} = n_y \qquad \cos \psi = \sqrt{1 - n_y^2}$$

$$\mathbf{P}^* = [R_Y(\phi)][R_X(-\psi)][R_Z(\theta)][R_X(\psi)][R_Y(-\phi)]\mathbf{P}$$



1. Rotate OA and point P about the Y axis an angle $-\phi$
2. Following the above rotation, rotate OA and P about the X axis an angle ψ
3. Rotate point P about the Z axis an angle θ.
4. Reverse step 2, that is, rotate about the X axis an angle $-\psi$.
5. Reverse step 1, that is, rotate about the Y axis an angle ϕ.

Rotation about an axis $\boldsymbol{n}$ (n_x, n_y, n_z) by angle θ



(a) $\mathbf{P}' = \mathbf{Q} + r\cos\theta\ \mathbf{v} + r\sin\theta\ \mathbf{u}$

(b) $r\mathbf{v} = \mathbf{P} - \mathbf{Q}$

(c) $\mathbf{u} = \mathbf{n} \times \mathbf{v} = \dfrac{\mathbf{n} \times (\mathbf{P} - \mathbf{Q})}{r}$

- Substitute (b), (c) into (a)

$$\mathbf{P}' = \mathbf{Q} + (\mathbf{P} - \mathbf{Q})\cos\theta + r\sin\theta\,\frac{\mathbf{n} \times (\mathbf{P} - \mathbf{Q})}{r}$$

$$(\mathbf{n} \times \mathbf{Q} = 0)$$

$$= \mathbf{Q}(1 - \cos\theta) + \mathbf{P}\cos\theta + (\mathbf{n} \times \mathbf{P})\sin\theta \qquad \textbf{(d)}$$

- Substitute Q=(P · n)n into (d)

$$\mathbf{P}' = (\mathbf{P} \cdot \mathbf{n})\mathbf{n}(1 - \cos\theta) + \mathbf{P}\cos\theta + (\mathbf{n} \times \mathbf{P})\sin\theta$$

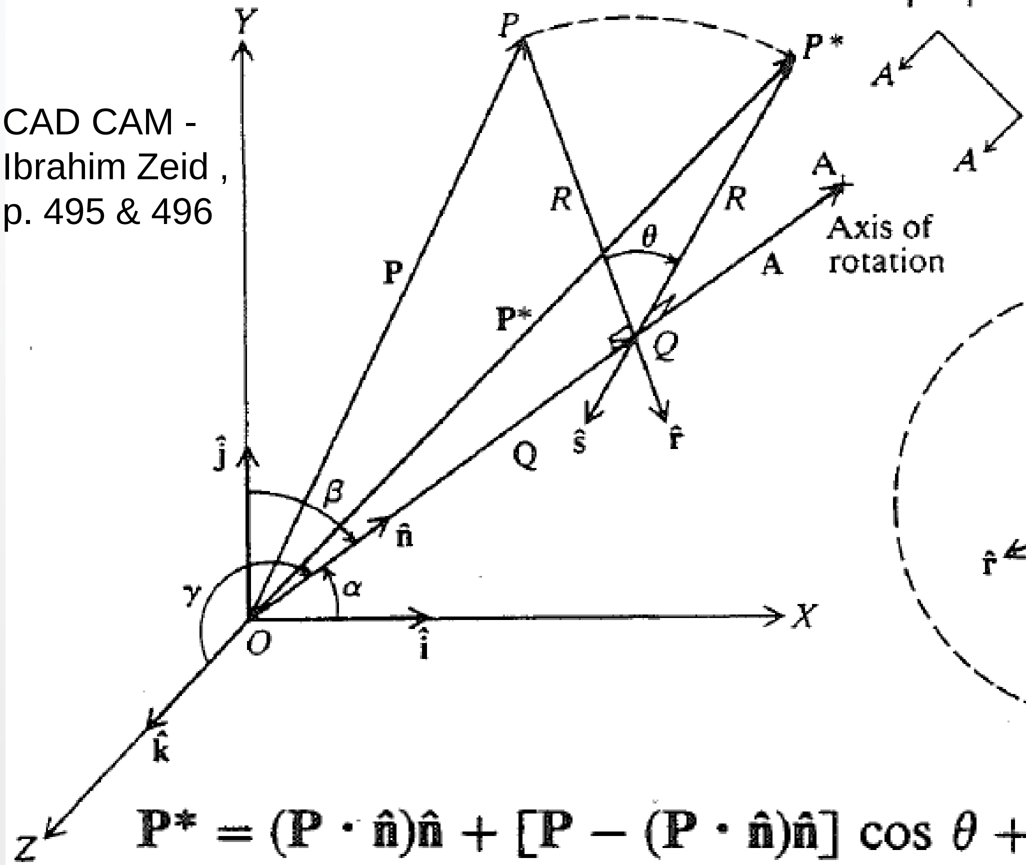
$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \mathrm{Rot}(n,\theta) \cdot \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix}$$

$$Rot(n,\theta) = \begin{bmatrix} n_x^2(1-\cos\theta)+\cos\theta & n_x n_y(1-\cos\theta)+n_z\sin\theta & n_x n_z(1-\cos\theta)-n_y\sin\theta & 0 \\ n_x n_y(1-\cos\theta)-n_z\sin\theta & n_y^2(1-\cos\theta)+\cos\theta & n_y n_z(1-\cos\theta)+n_x\sin\theta & 0 \\ n_x n_z(1-\cos\theta)+n_y\sin\theta & n_y n_z(1-\cos\theta)-n_x\sin\theta & n_z^2(1-\cos\theta)+\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation around an Arbitrary Axis

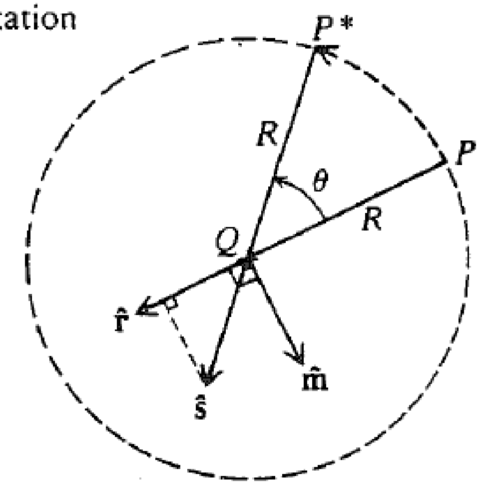
$$\mathbf{P}^* = [R]\mathbf{P} = Rot(n,\theta)\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

CAD CAM - Ibrahim Zeid , p. 495 & 496



$$|\mathbf{A}| = \sqrt{x_A^2 + y_A^2 + z_A^2}$$

$$n_x = x_A / |\mathbf{A}|$$
$$n_y = y_A / |\mathbf{A}|$$
$$n_z = z_A / |\mathbf{A}|$$



View A–A

$$\mathbf{P}^* = [R]\mathbf{P}$$

$$\mathbf{P}^* = \mathbf{P} + \mathbf{PQ} + \mathbf{QP}^*$$

$$\mathbf{P}^* = \mathbf{P} + R\hat{\mathbf{r}} - R\hat{\mathbf{s}}$$

$$\mathbf{PQ} = \mathbf{Q} - \mathbf{P}$$

$$\mathbf{Q} = (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$$

$$\hat{\mathbf{r}} = \frac{(\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{P}}{R}$$

$$\hat{\mathbf{m}} = \hat{\mathbf{n}} \times \hat{\mathbf{r}}$$

$$\hat{\mathbf{s}} = \cos\theta\hat{\mathbf{r}} + \sin\theta\hat{\mathbf{m}}$$

$$\mathbf{P}^* = (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + [\mathbf{P} - (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}]\cos\theta + (\hat{\mathbf{n}} \times \mathbf{P})\sin\theta - \hat{\mathbf{n}} \times (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\sin\theta$$

O

$$\mathbf{P}^* = (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + [\mathbf{P} - (\mathbf{P} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}]\cos\theta + (\hat{\mathbf{n}} \times \mathbf{P})\sin\theta$$

Rotation around an Arbitrary Axis

$$\mathbf{P}^* = [R]\mathbf{P} = Rot(n,\theta)\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{P}^* = (\mathbf{P}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}} + [\mathbf{P} - (\mathbf{P}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}]\cos\theta + (\hat{\mathbf{n}}\times\mathbf{P})\sin\theta$$

$$\mathbf{P}\cdot\hat{\mathbf{n}} = xn_x + yn_y + zn_z = [n_x \quad n_y \quad n_z]\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\hat{\mathbf{n}}\times\mathbf{P} = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ n_x & n_y & n_z \\ x & y & z \end{vmatrix} = (n_y z - n_z y)\hat{\mathbf{i}} + (n_z x - n_x z)\hat{\mathbf{j}} + (n_x y - n_y x)\hat{\mathbf{k}}$$

CAD CAM - Ibrahim Zeid , p. 495 & 496

$$\hat{\mathbf{n}}\times\mathbf{P} = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{P}^* = \left\{(1-\cos\theta)\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}[n_x \quad n_y \quad n_z] + \cos\theta\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \sin\theta\begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}\right\}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$\mathbf{P}^* = [R]\mathbf{P}$ The general rotation matrix $[R]$

$$[R] = \begin{bmatrix} n_x^2\,\mathrm{v}\theta + \mathrm{c}\theta & n_x n_y\,\mathrm{v}\theta - n_z\,\mathrm{s}\theta & n_x n_z\,\mathrm{v}\theta + n_y\,\mathrm{s}\theta \\ n_x n_y\,\mathrm{v}\theta + n_z\,\mathrm{s}\theta & n_y^2\,\mathrm{v}\theta + \mathrm{c}\theta & n_y n_z\,\mathrm{v}\theta - n_x\,\mathrm{s}\theta \\ n_x n_z\,\mathrm{v}\theta - n_y\,\mathrm{s}\theta & n_y n_z\,\mathrm{v}\theta + n_x\,\mathrm{s}\theta & n_z^2\,\mathrm{v}\theta + \mathrm{c}\theta \end{bmatrix}$$

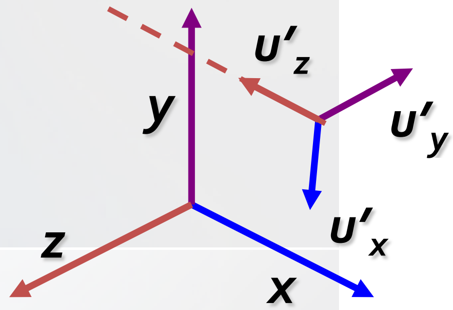
where $\mathrm{c}\theta = \cos\theta$

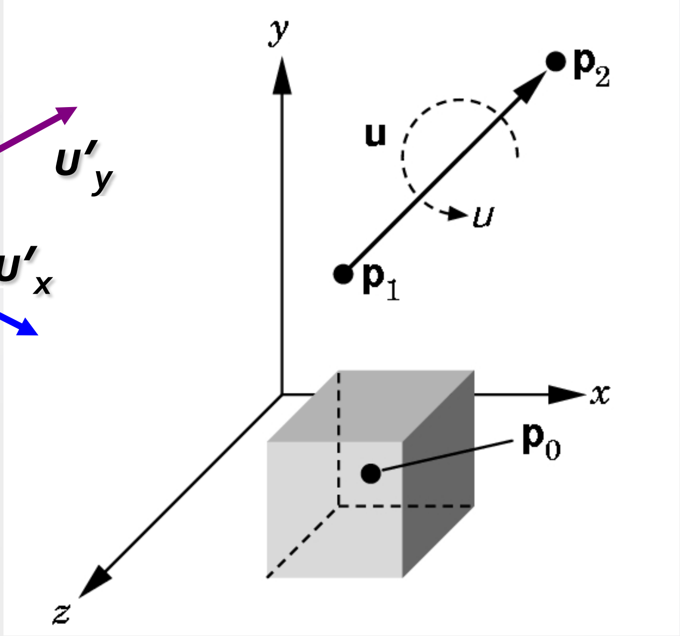
$\mathrm{s}\theta = \sin\theta$

$\mathrm{v}\theta = \text{versine}\ \theta$

$\mathrm{v}\theta = 1 - \cos\theta$

Other rotations





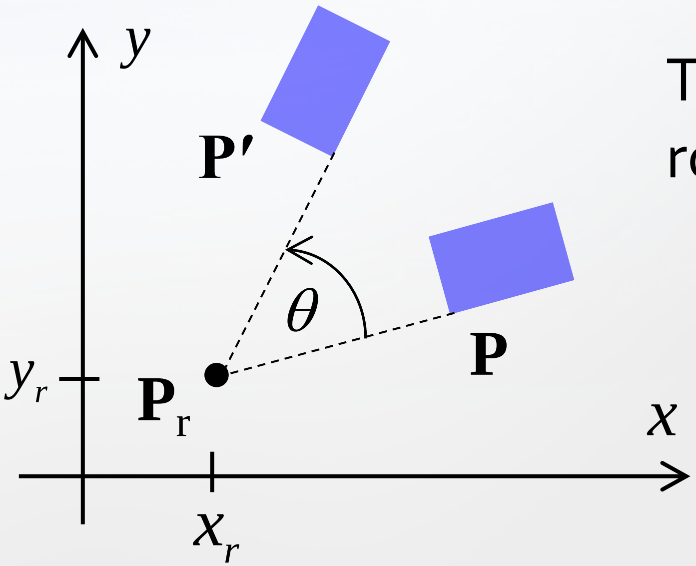
What if the axis of rotation
does not pass through the origin?

Similar process as in 2D, translate
to the origin, rotate as normal, translate back.

We just need to know a point on the axis that we can translate to the origin.

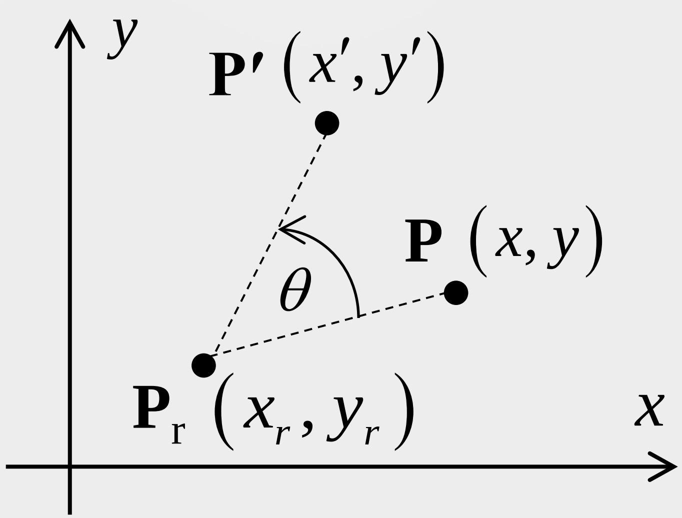
Only way to specify such a rotation is to give two points on the line or one point and a direction, so the requirement is easily satisfied.

2D Rotation about a pivot point P_r



Translate pivot point P_r to the origin, rotate as normal, translate back.

Rotation in angle θ about a pivot (rotation) point (x_r, y_r).



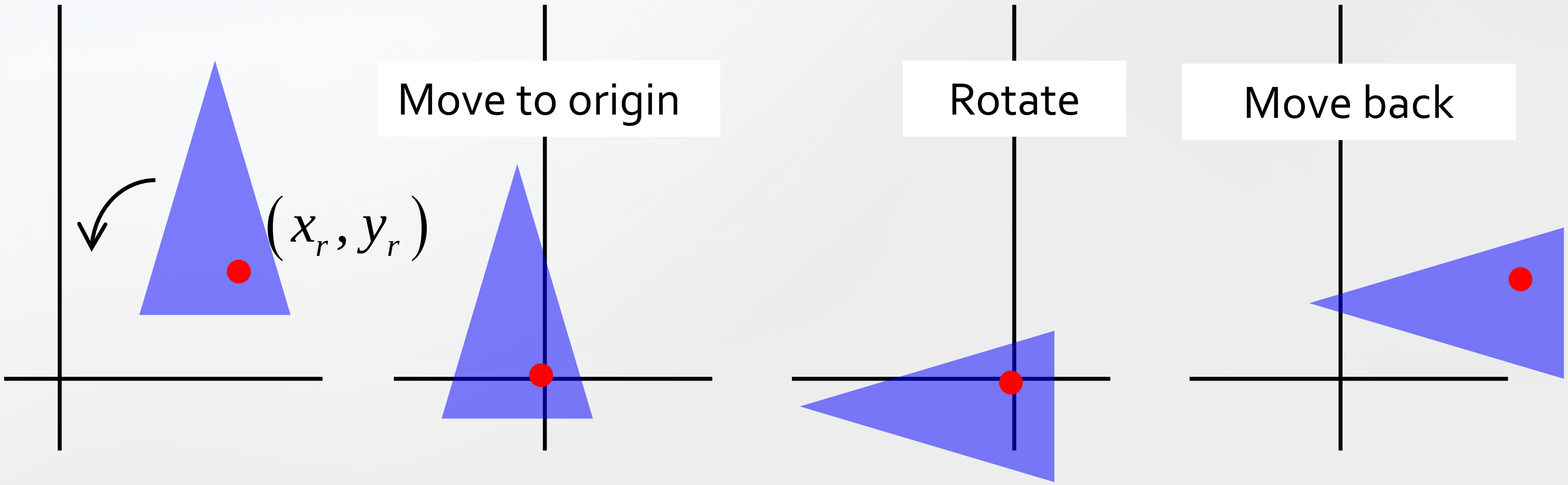
$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$

$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta$$

$$\mathbf{P}' = \mathbf{P}_r + \mathbf{R}\cdot(\mathbf{P} - \mathbf{P}_r)$$

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Rotation about a fixed point, M=T.R.T^{-1}



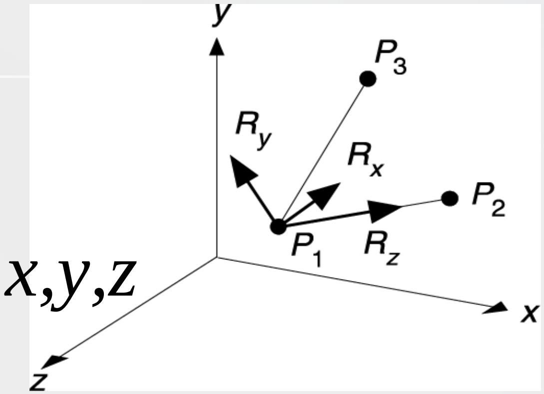
Translate the fixed point to origin,
Rotate as normal,
Translate back.

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r \sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r \sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

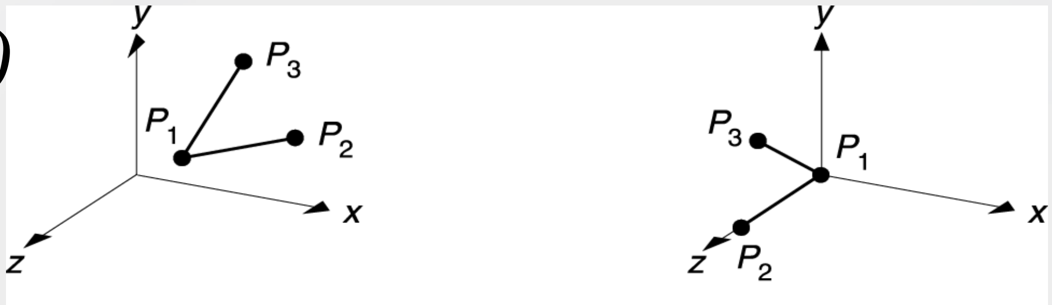
Example: Composition of 3D Transformations

Goal: Transform the local coordinate system R_x, R_y, R_z to align with the origin x,y,z

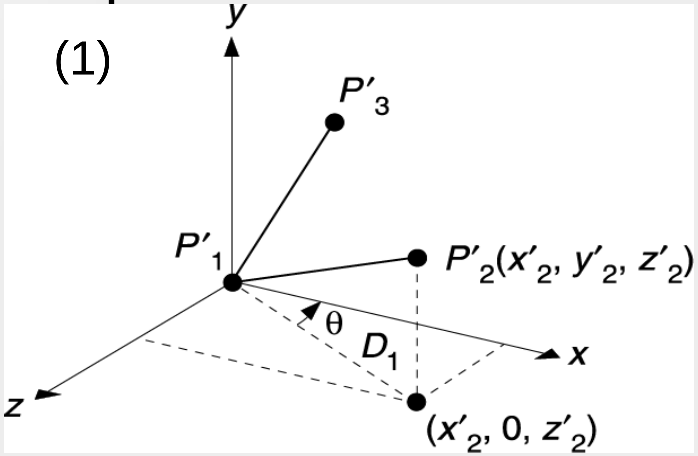


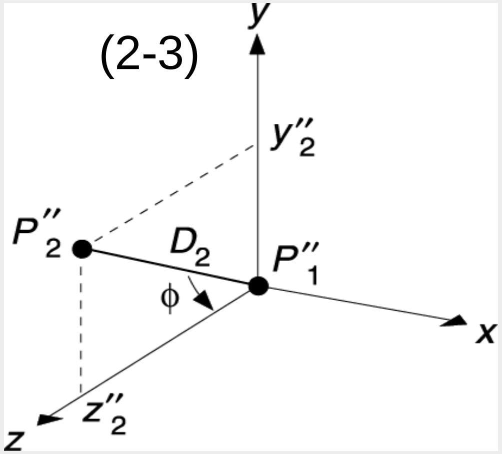
Process

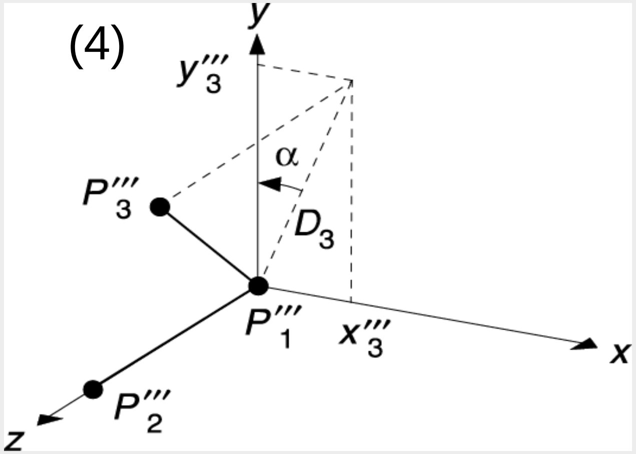
1. Translate P_1 to $(0,0,0)$
2. Rotate about y
3. Rotate about x
4. Rotate about z



(a) Initial position (b) Final position

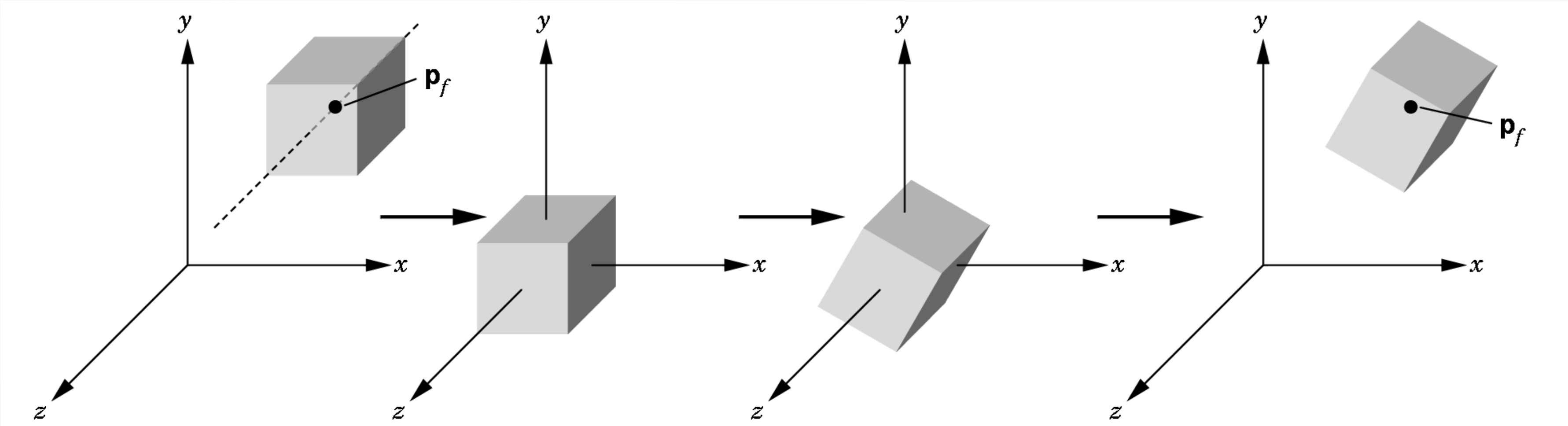




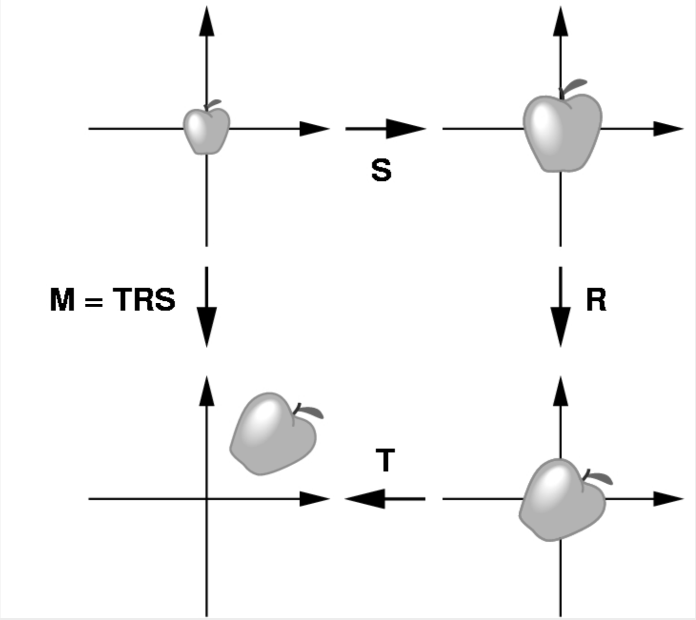


Translate the fixed point to origin,
Rotate about z axis, Translate back.

Rotation About a Fixed Point



$$M = T(p_f)\, R_z(\theta)\, T(-\, p_f)$$

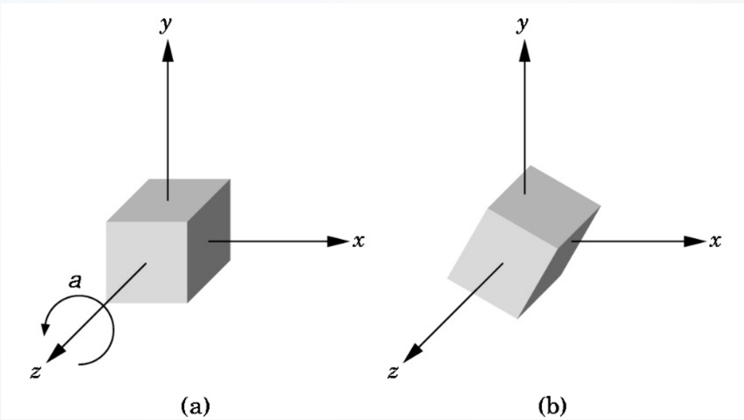


Transformation is defined by the *instance transformation M*

$$M = T\,R\,S$$

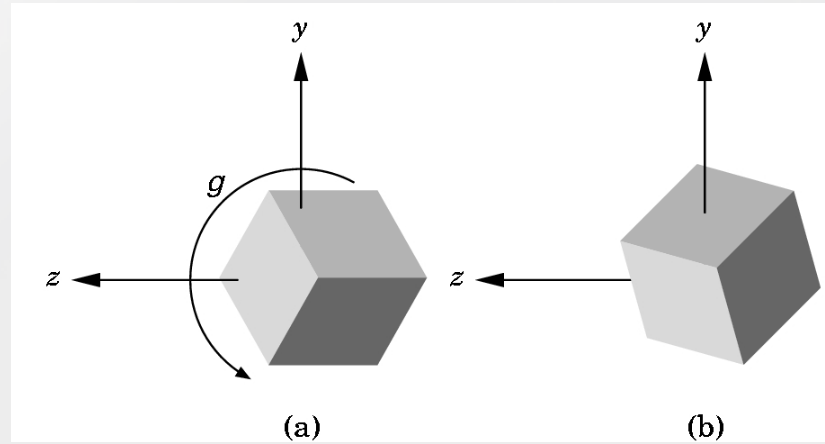
(order is substantial!)

Composite Rotations in E3

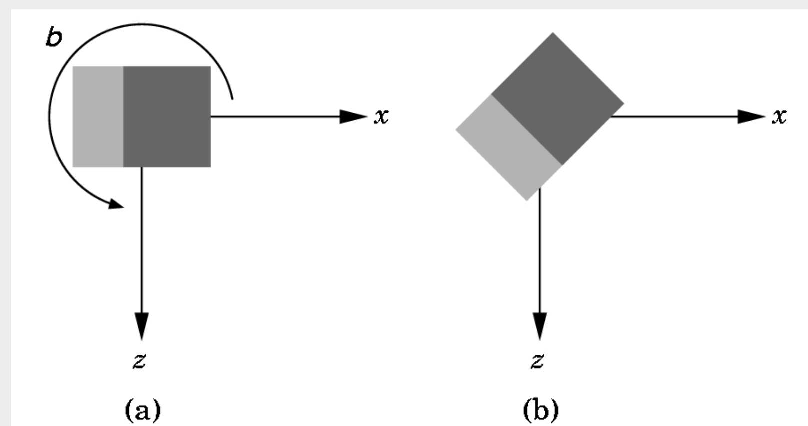


R_z

$\Rightarrow$



$\Downarrow$ R_x



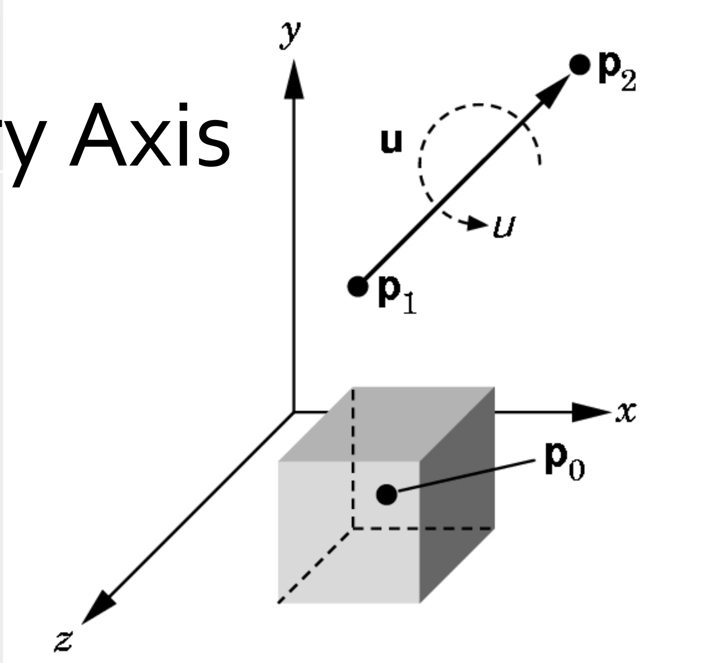
R_y

Cube can be rotated about all
x, y, z axis

In our case the transformation
matrix is defined

$M = R_y\, R_x\, R_z = R_{zx}\, R_{yz}\, R_{xy}$

Rotations About an Arbitrary Axis



Given:

- points p_1 , p_2 and rotation angle θ
- objects to be rotated

Define vectors

$u = p_1 - p_2$

and $v = u / |u|$ – normalized

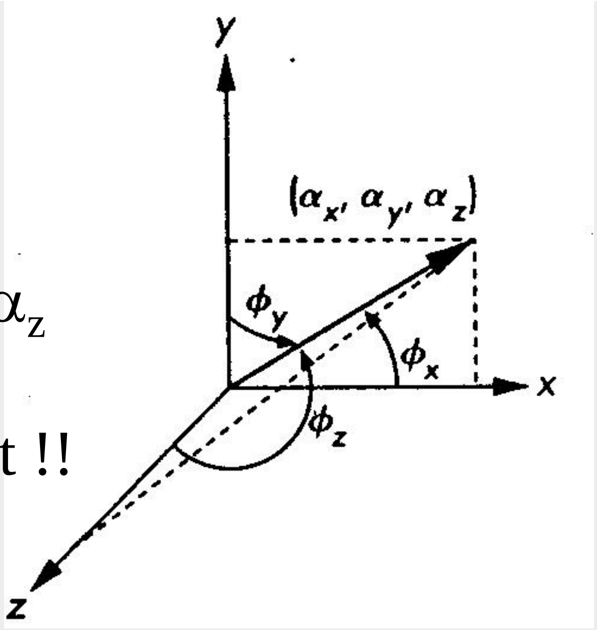
$v = [\ \alpha_x , \alpha_y , \alpha_z\ ]^T$

$\alpha_x^2 + \alpha_y^2 + \alpha_z^2 = 1$ – directional cosines

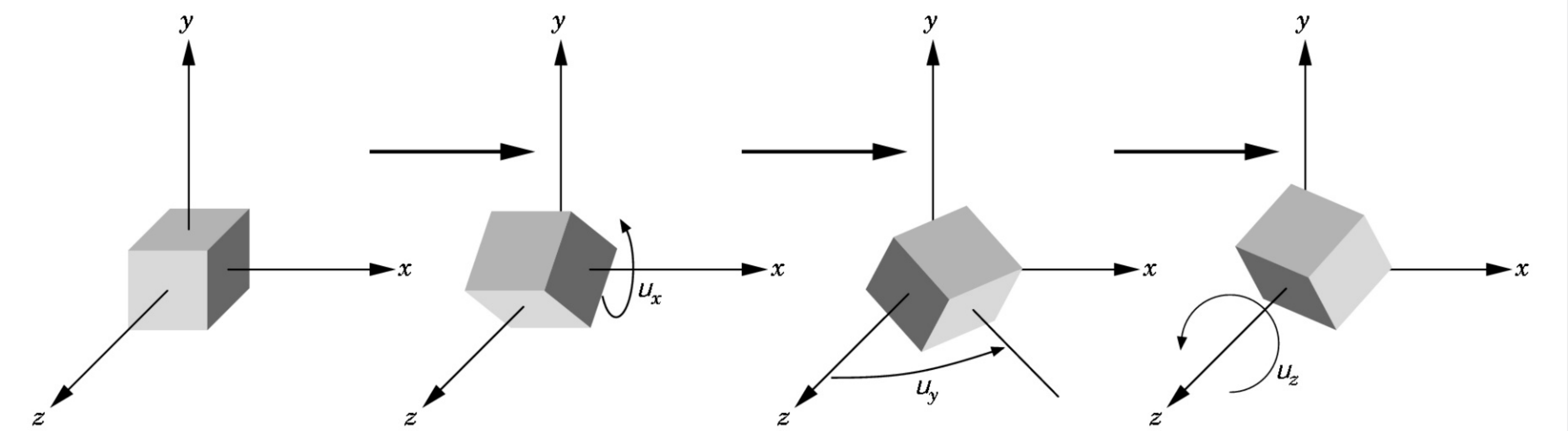
$\cos(\varphi_x) = \alpha_x$, $\cos(\varphi_y) = \alpha_y$, $\cos(\varphi_z) = \alpha_z$

$\cos^2(\varphi_x) + \cos^2(\varphi_y) + \cos^2 (\varphi_z) = 1$

$\Rightarrow$ only two directions angles are independent !!



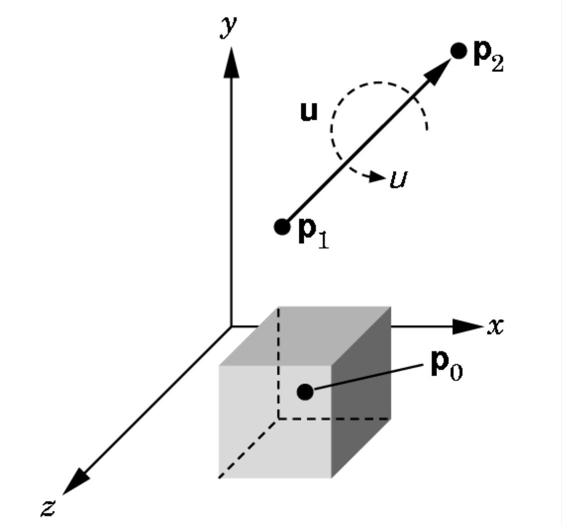
Rotations About an Arbitrary Axis



Transformation (rotation about origin)

$R = R_x(-\theta_x)\, R_y(-\theta_y)\, R_z(\theta)\, R_y(\theta_y)\, R_x(\theta_x)$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\ \theta & -sin\ \theta & 0 \\ 0 & sin\ \theta & cos\ \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



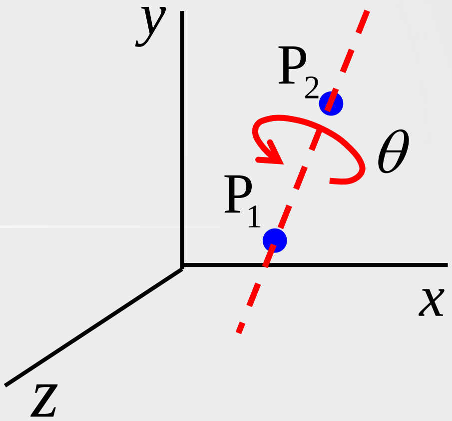
$$R_z(\theta) = \begin{bmatrix} cos\ \theta & -sin\ \theta & 0 & 0 \\ sin\ \theta & cos\ \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} cos\ \theta & 0 & sin\ \theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\ \theta & 0 & cos\ \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Complete transformation (include translations)

$M = T(p_0)\, R_x(-\theta_x)\, R_y(-\theta_y)\, R_z(\theta)\, R_y(\theta_y)\, R_x(\theta_x)\, T(-p_0)$

General 3D Rotation



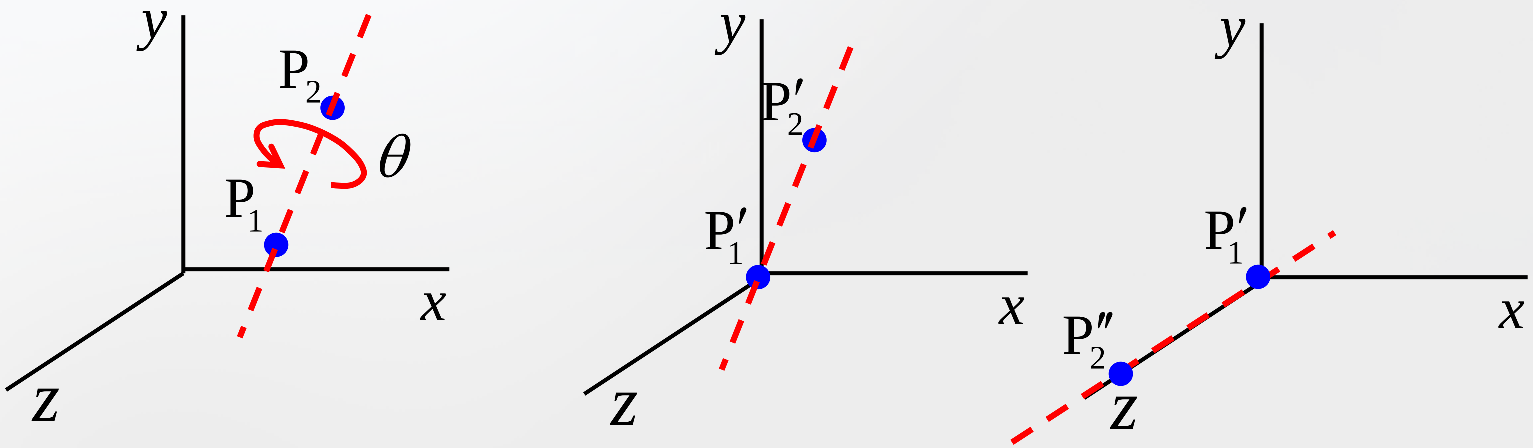
1. Translate the object such that rotation axis passes through the origin.
1. Rotate the object such that rotation axis coincides with one of Cartesian axes.
2. Perform specified rotation about the Cartesian axis.
3. Apply inverse rotation to return rotation axis to original direction.
4. Apply inverse translation to return rotation axis to original position.

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

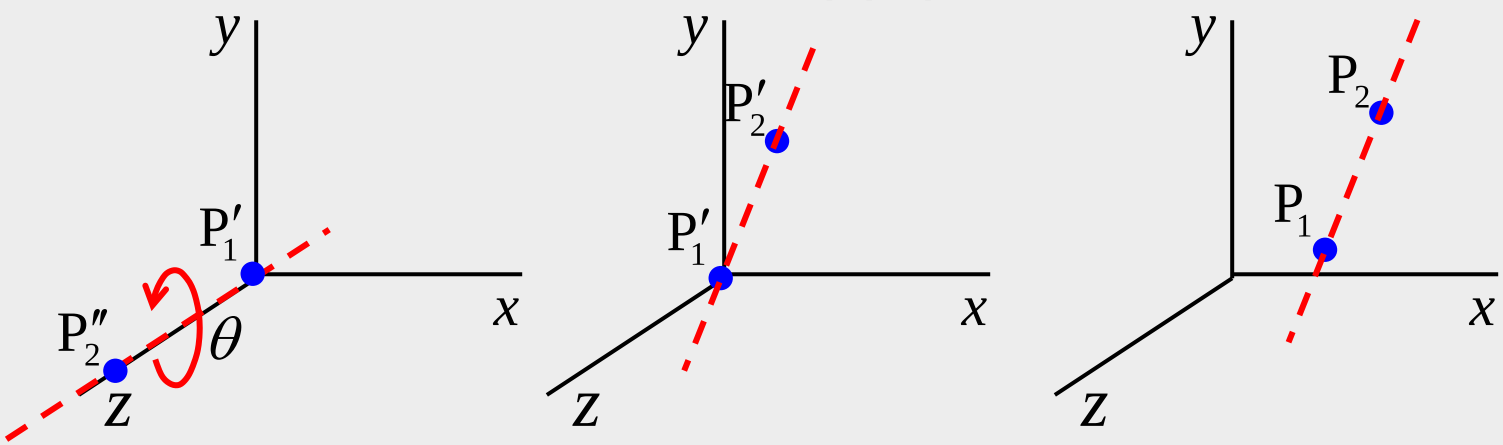
$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

General 3D Rotation

Translate to origin. Rotate on Cartesian axes.



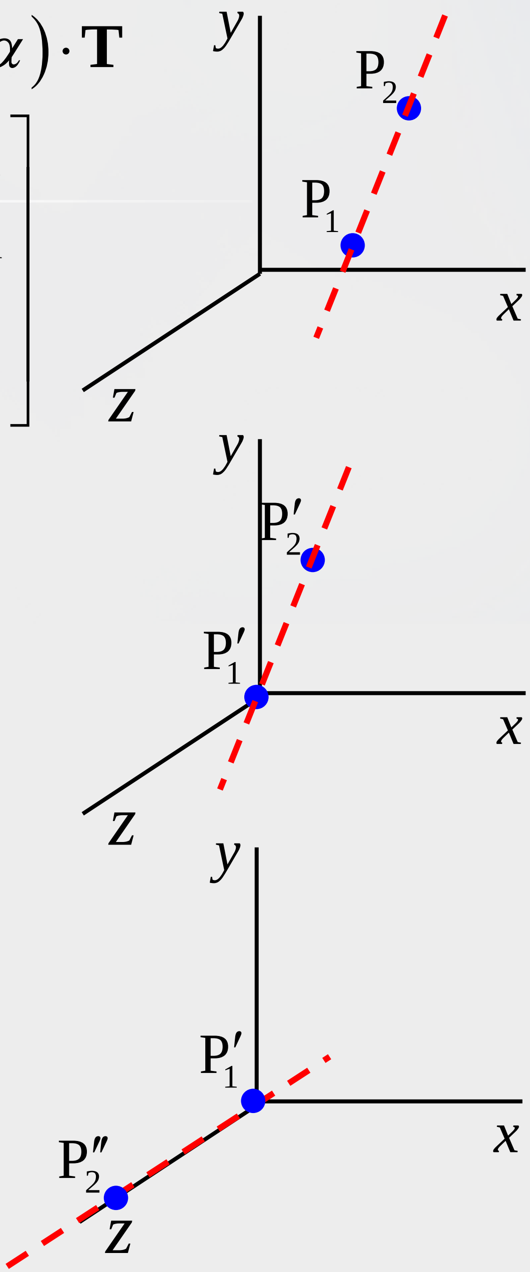
Rotation about the axis. Apply inverse translations.



$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

Translate the object to origin.

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



The vector from $\mathbf{P}_1$ to $\mathbf{P}_2$ is:

$$\mathbf{V} = \mathbf{P}_2 - \mathbf{P}_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

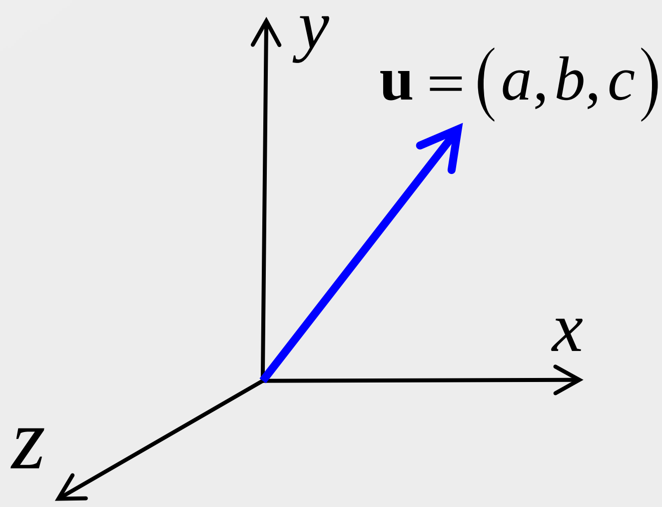
Unit rotation vector: $\mathbf{u} = \mathbf{V}/|\mathbf{V}| = (a, b, c)$

$$a = (x_2 - x_1)/|\mathbf{V}|$$

$$b = (y_2 - y_1)/|\mathbf{V}|$$

$$c = (z_2 - z_1)/|\mathbf{V}|$$

$$\sqrt{a^2 + b^2 + c^2} = 1$$



Rotating $\mathbf{u}$ to coincide with z axis

First rotate $\mathbf{u}$ around x axis to lay in $x-z$ plane.

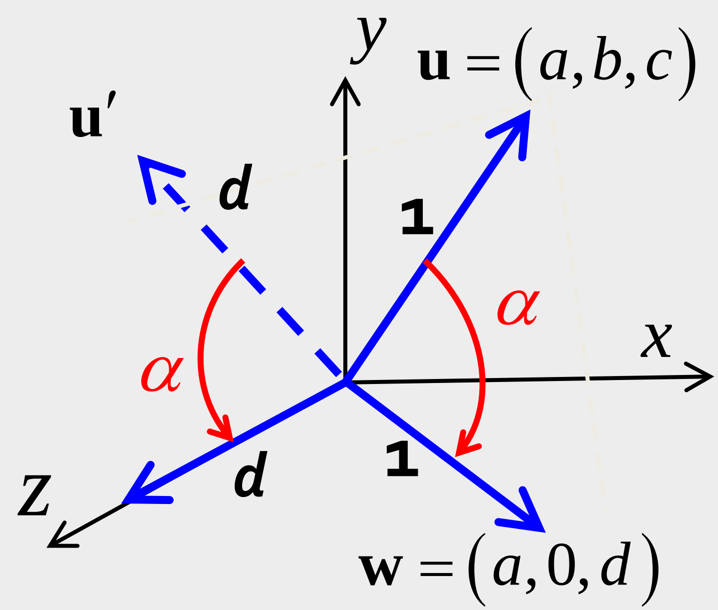
Equivqlent to rotation $\mathbf{u}$'s projection on $y-z$ plane around x axis.

$$\cos\alpha = c\Big/\sqrt{b^2+c^2} = c/d\,,\quad \sin\alpha = b/d\,.$$

We obtained a unit vector $\mathbf{w} = \left(a, 0, \sqrt{b^2+c^2} = d\right)$ in $x-z$ plane.

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\ \theta & -sin\ \theta & 0 \\ 0 & sin\ \theta & cos\ \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



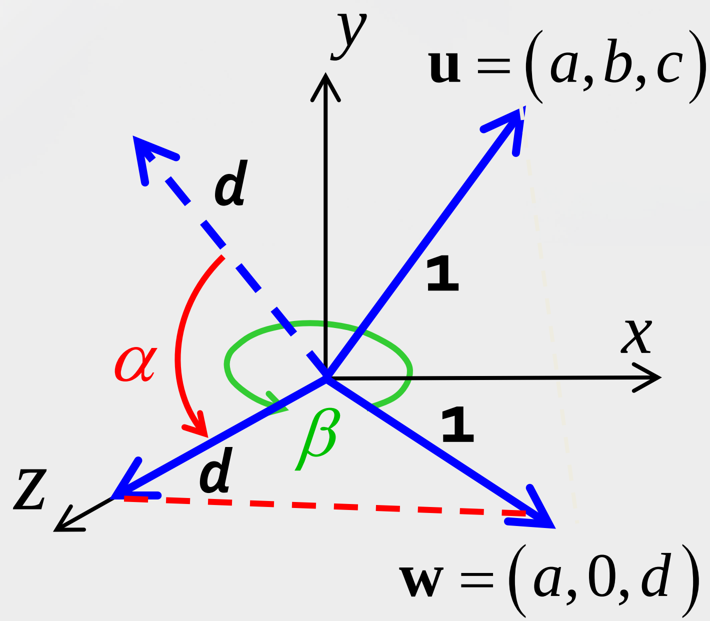
Rotate **w** counterclockwise around y axis.

w is a unit vector whose $x-$component is a, $y-$component is 0,

hence $z-$component is $\sqrt{b^2+c^2}=d$. $\cos\beta=d$, $\sin\beta=-a$

$$\mathbf{R}_y(\beta)=\begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta)=\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

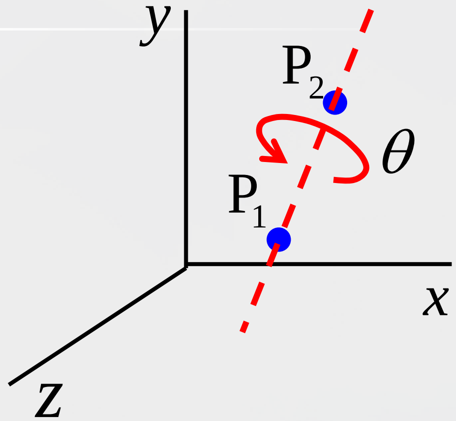


$$\mathbf{R}(\theta)=\mathbf{T}^{-1}\cdot\mathbf{R}_x^{-1}(\alpha)\cdot\mathbf{R}_y^{-1}(\beta)\cdot\mathbf{R}_z(\theta)\cdot\mathbf{R}_y(\beta)\cdot\mathbf{R}_x(\alpha)\cdot\mathbf{T}$$

$$\mathbf{R}_x(\alpha)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y(\beta)=\begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_z(\theta)=\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y(\beta)=\begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_x(\alpha)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General 3D Rotation Matrix

$$\mathbf{R}_z(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$



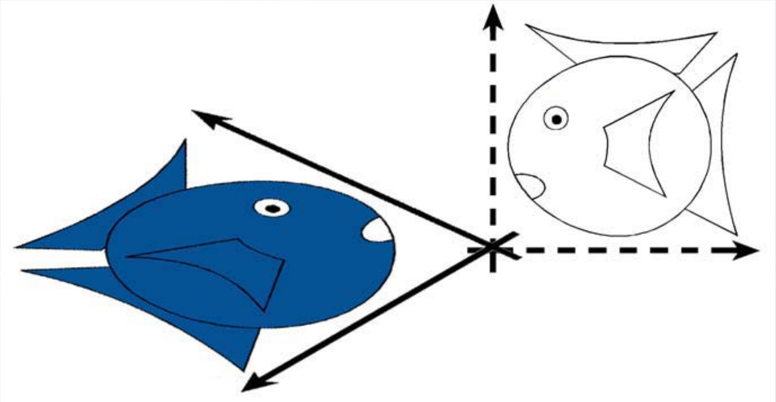
$$\mathbf{R}(\theta)=\mathbf{T}^{-1}\cdot\mathbf{R}_x^{-1}(\alpha)\cdot\mathbf{R}_y^{-1}(\beta)\cdot\mathbf{R}_z(\theta)\cdot\mathbf{R}_y(\beta)\cdot\mathbf{R}_x(\alpha)\cdot\mathbf{T}$$

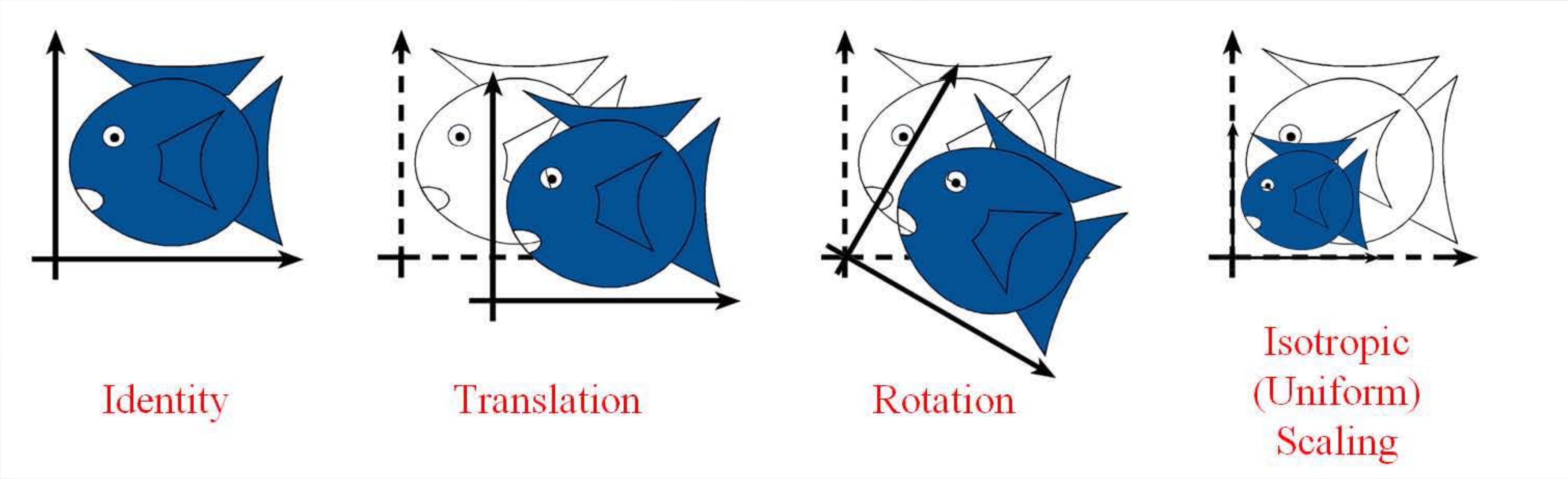
$$\mathbf{R}_x(\alpha)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & c/d & -b/d & 0\\ 0 & b/d & c/d & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \mathbf{R}_y(\beta)=\begin{bmatrix}d & 0 & a & 0\\ 0 & 1 & 0 & 0\\ -a & 0 & d & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \mathbf{R}_z(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \mathbf{R}_y(\beta)=\begin{bmatrix}d & 0 & a & 0\\ 0 & 1 & 0 & 0\\ -a & 0 & d & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \mathbf{R}_x(\alpha)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & c/d & -b/d & 0\\ 0 & b/d & c/d & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

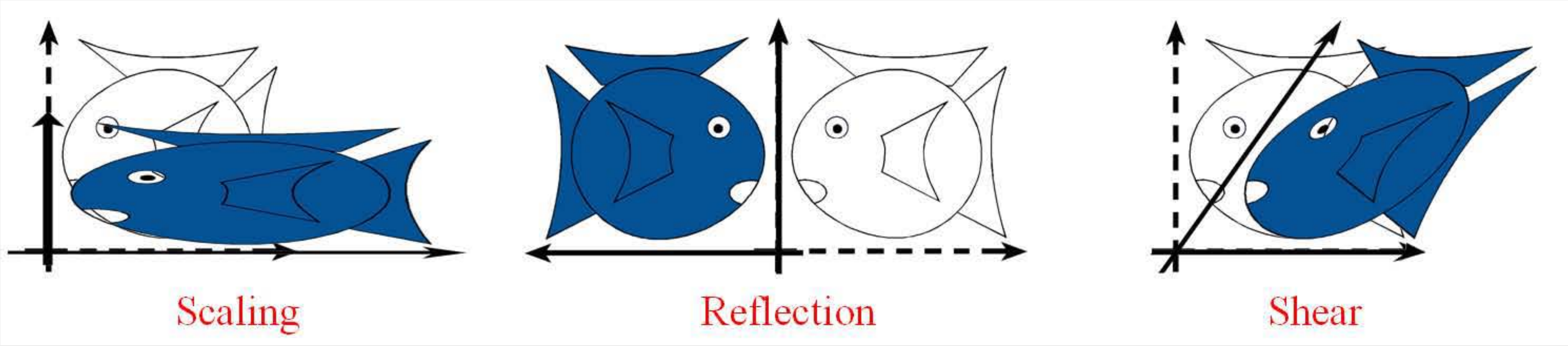
$$\mathbf{M}_R(\theta)=$$

$$\begin{bmatrix}a^2(1-\cos\theta)+\cos\theta & ab(1-\cos\theta)-c\sin\theta & ac(1-\cos\theta)+b\sin\theta\\ ba(1-\cos\theta)+c\sin\theta & b^2(1-\cos\theta)+\cos\theta & bc(1-\cos\theta)-a\sin\theta\\ ca(1-\cos\theta)-b\sin\theta & cb(1-\cos\theta)+a\sin\theta & c^2(1-\cos\theta)+\cos\theta\end{bmatrix}$$

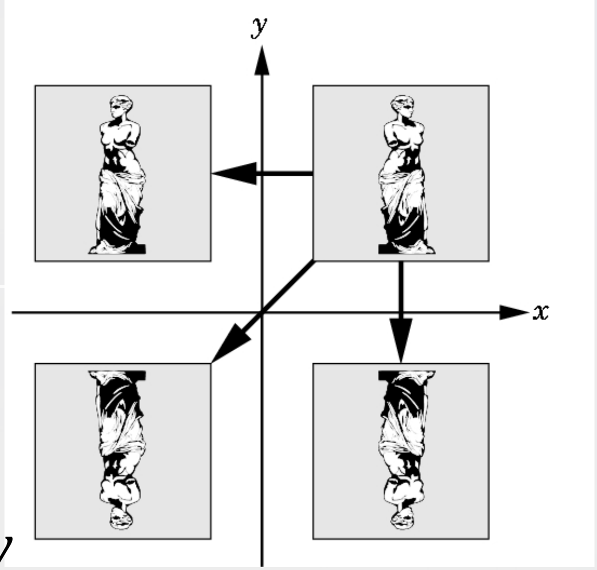
Linear Transformations



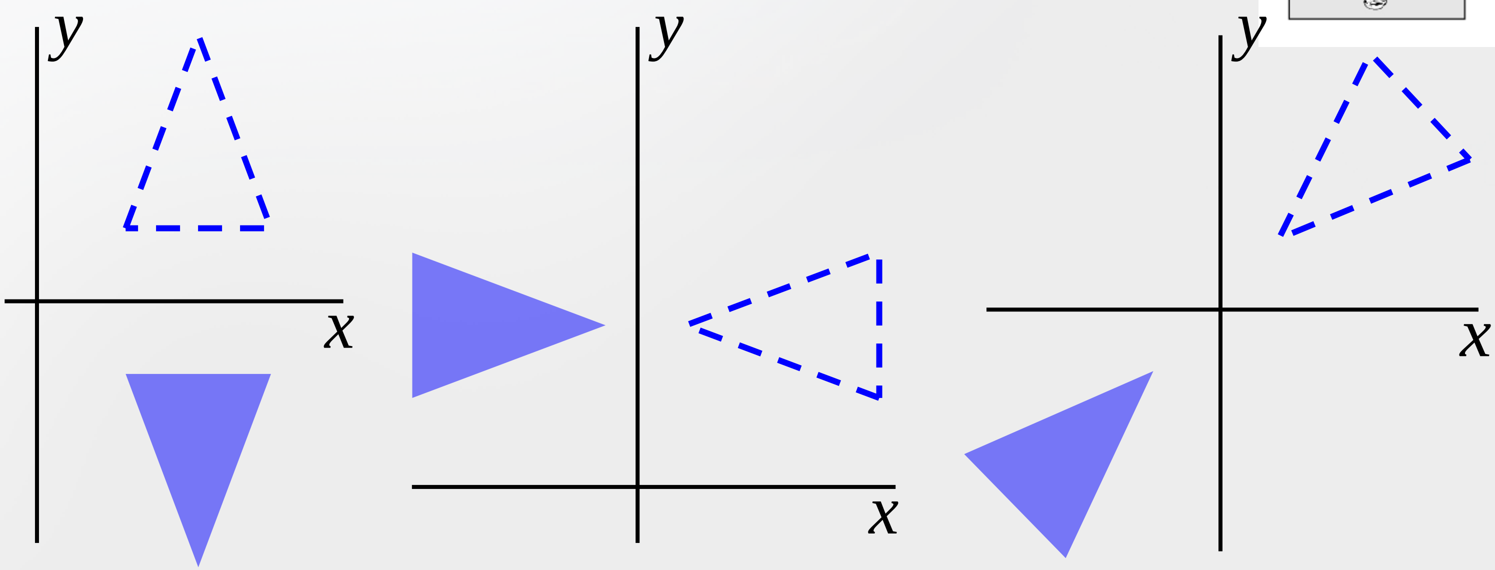




2D Reflections (Mirror)

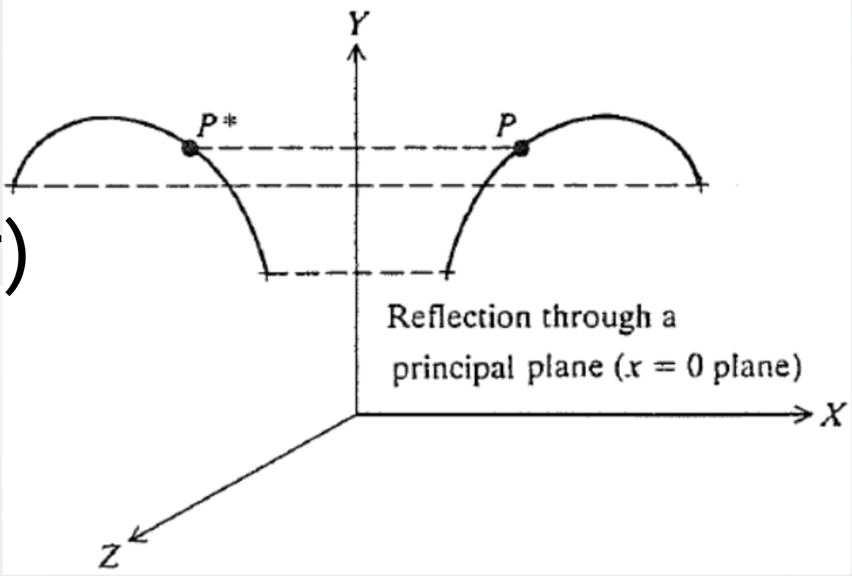


2D *reflection about x, y, (x and y)* axis :



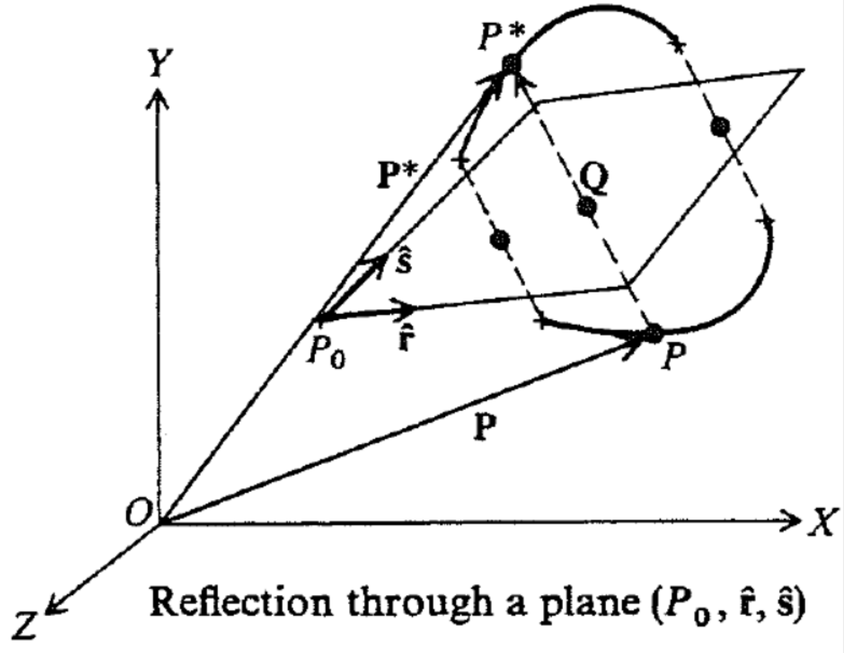
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \bullet \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3D Reflection (Mirror)



3D Reflection
about x-y plane :

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



the mirror of P

$P^* = \mathbf{M}\, P$

Complete transformation

$\mathbf{M} = T(p_0)\, R_x(-\theta_x)\, R_y(-\theta_y)\, F_z(-z)\, R_y(\theta_y)\, R_x(\theta_x)\, T(-\, p_0)$

3D Transformations: Shear

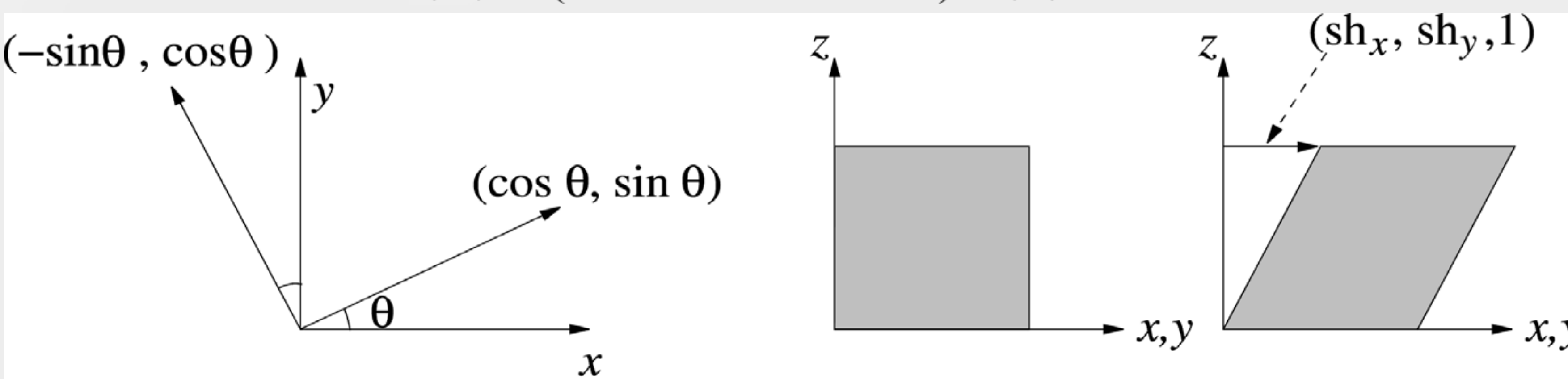
3D Shear:

(function of z)

$$H_{xy}(\theta) = \begin{pmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

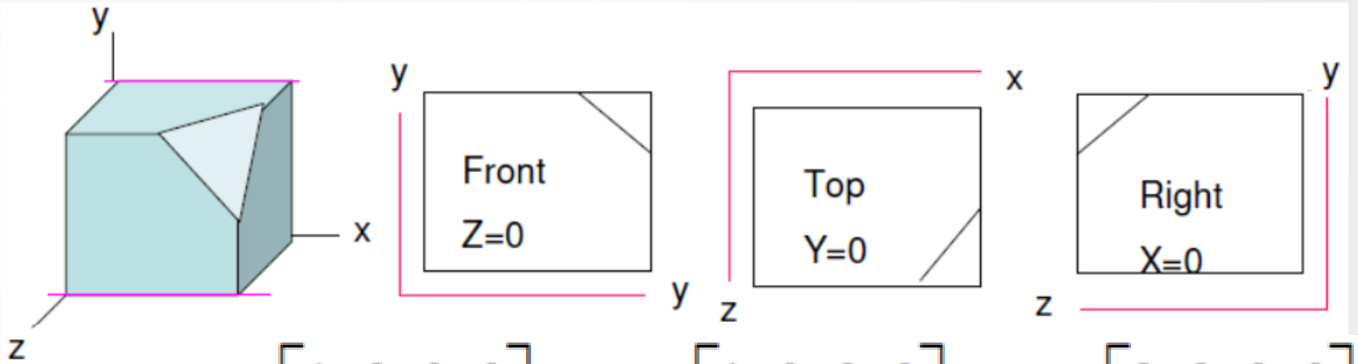
$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & Sh_x^y & Sh_x^z & 0 \\ Sh_y^x & 1 & Sh_y^z & 0 \\ Sh_z^x & Sh_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$X' = X + Sh_x^y Y + Sh_x^z Z$$
$$Y' = Sh_y^x X + Y + Sh_y^z Z$$
$$Z' = Sh_z^x X + Sh_z^y Y + Z$$



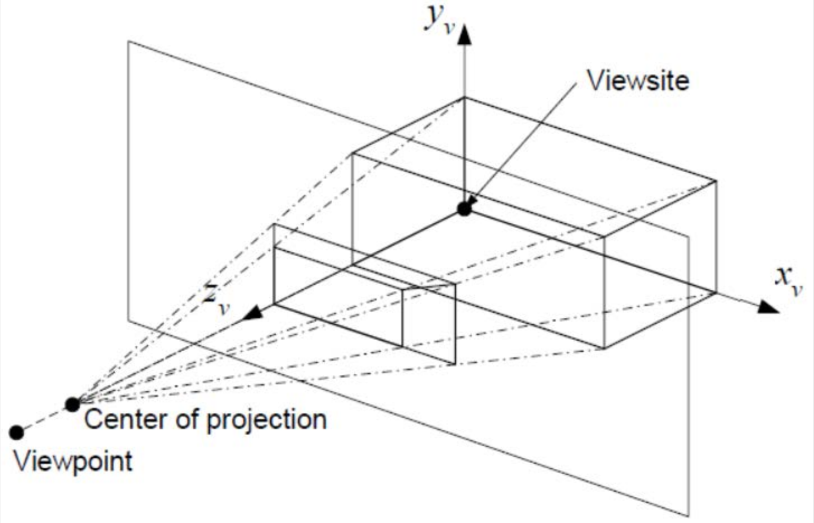
Rotation (about z) Shear (orthogonal to z)

Orthographic projection matrices

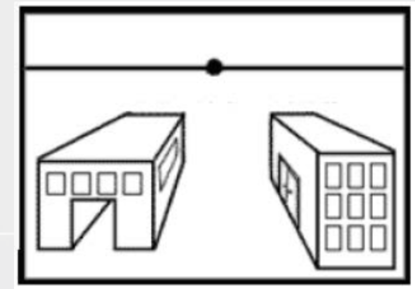


$$[T] = \left[\begin{array}{ccc|c} a & d & g & l \\ b & e & i & m \\ c & f & j & n \\ \hline p & q & r & s \end{array}\right]$$

$$[T_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [T_y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [T_x] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

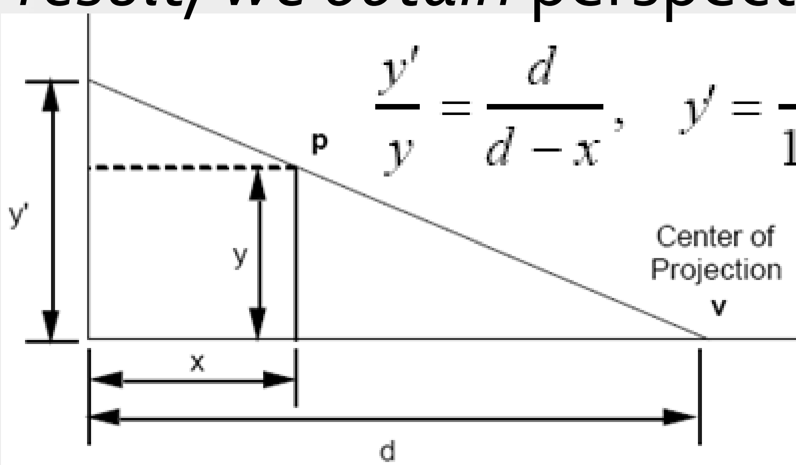


$$Y^* = M^* X^* = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix}$$



Perspective projection

Figure shows how to project a point on the *y axis* from a center of projection v lying on the *x axis at x=d. By similarity of triangles.* Thus far we have only used homogeneous matrices with a last row whose offdiagonal elements are null. What happens when they are non-null (–1/*d) term. After normalizing the result, we obtain* perspective projection of the object.

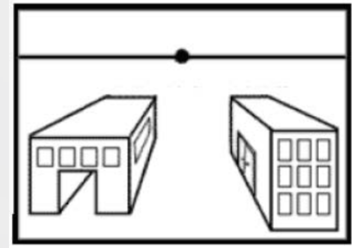


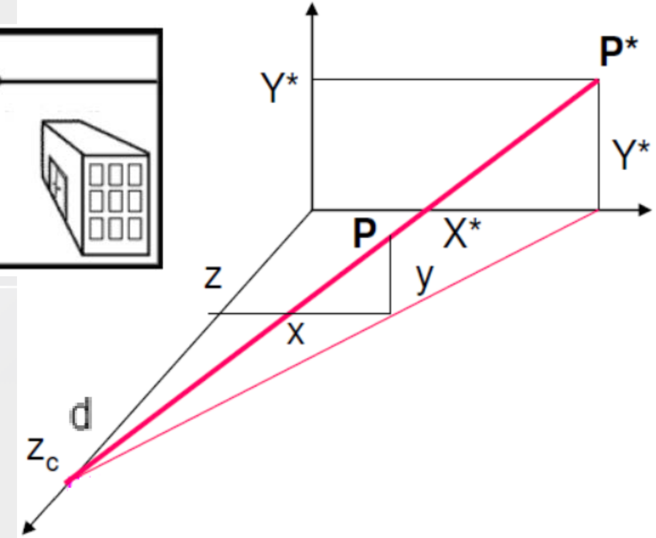
$$\frac{y'}{y} = \frac{d}{d-x}, \quad y' = \frac{y}{1 - x/d}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/d & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 - x/d \end{bmatrix}$$

$$\begin{bmatrix} \dfrac{x}{1 - x/d} \\ \dfrac{y}{1 - x/d} \\ 1 \end{bmatrix}$$

Perspective projection





Computing a planar projection involves matrix multiplication, followed by normalization and orthographic projection (z=0 plane). (r = –1/*d*)

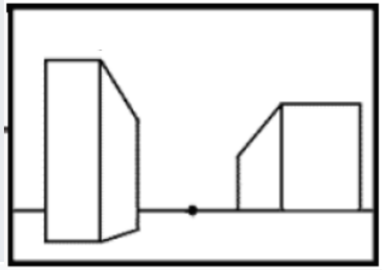
$$[x \quad y \quad z \quad 1]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&r\\0&0&0&1\end{bmatrix}=[x \quad y \quad z \quad (rz+1)]$$

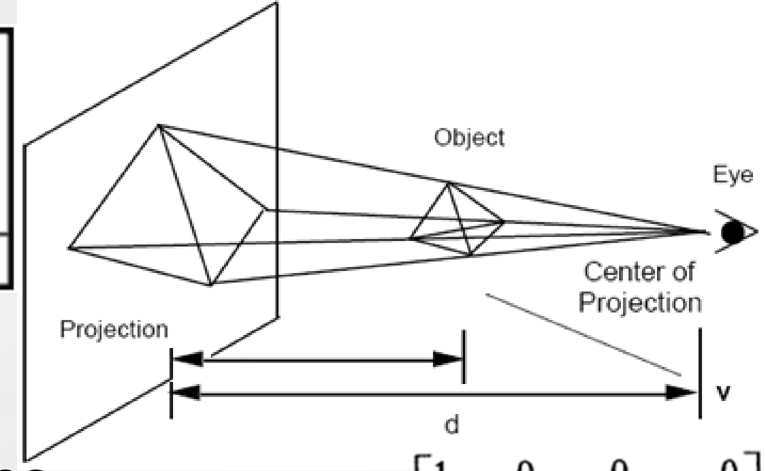
$$[x^* \quad y^* \quad z^* \quad 1]=\left[\frac{x}{rz+1} \quad \frac{y}{rz+1} \quad \frac{z}{rz+1} \quad 1\right]$$

$$[x \quad y \quad z \quad 1]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&r\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}=[x \quad y \quad z \quad 1]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&r\\0&0&0&1\end{bmatrix}=[x \quad y \quad 0 \quad (rz+1)]$$

$$[x^* \quad y^* \quad z^* \quad 1]=\left[\frac{x}{rz+1} \quad \frac{y}{rz+1} \quad 0 \quad 1\right]$$

3D Perspective





$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix}$$

In 3-D, the matrix multiplication provides us the *x and y coordinates of the projection of a point on the xy plane, from a* center of projection on the *z axis at z=d.* In 3-D the perspective transformation produces a deformed 3-D object, which must be projected orthographically onto the *xy plane to generate the desired 2-D image. Computing* a planar projection involves matrix multiplication, followed by normalization and orthographic projection.

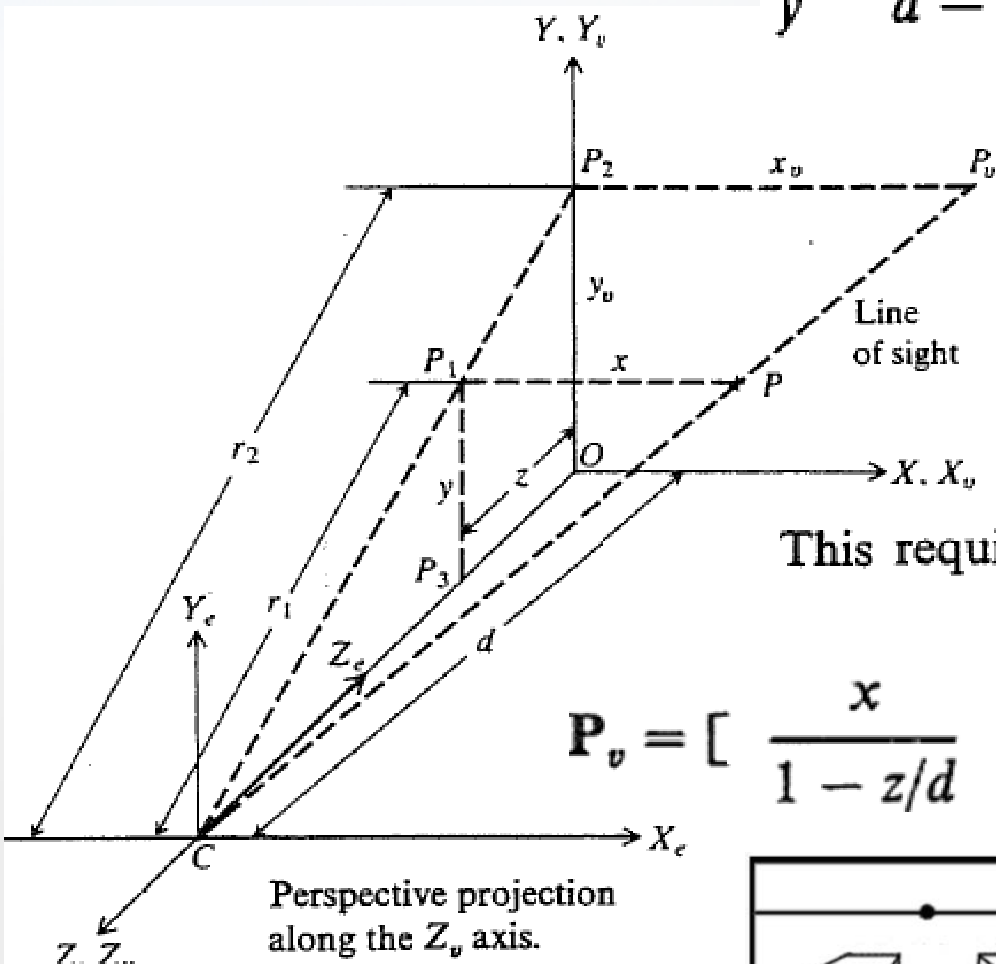
$$\mathbf{P}_v = \begin{bmatrix} \dfrac{x}{1 - z/d} \\ \dfrac{y}{1 - z/d} \\ 0 \end{bmatrix}$$

3D Perspective

$$\frac{x_v}{x} = \frac{r_2}{r_1} = \frac{d}{d-z} = \frac{1}{1-z/d}$$

$$\frac{y_v}{y} = \frac{d}{d-z} = \frac{1}{1-z/d}$$

$$\mathbf{P}_v = \begin{bmatrix} \dfrac{x}{1-z/d} \\ \dfrac{y}{1-z/d} \\ 0 \end{bmatrix}$$



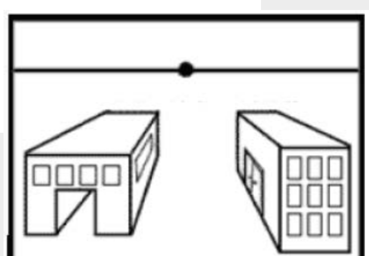
Perspective projection along the Z_v axis.

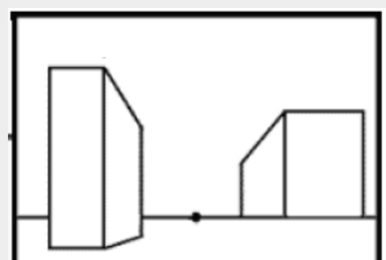
$$\mathbf{P}_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\mathbf{P}_v = [x \quad y \quad 0 \quad (1-z/d)]^T$$

This require the division of x and y by $(1-z/d)$

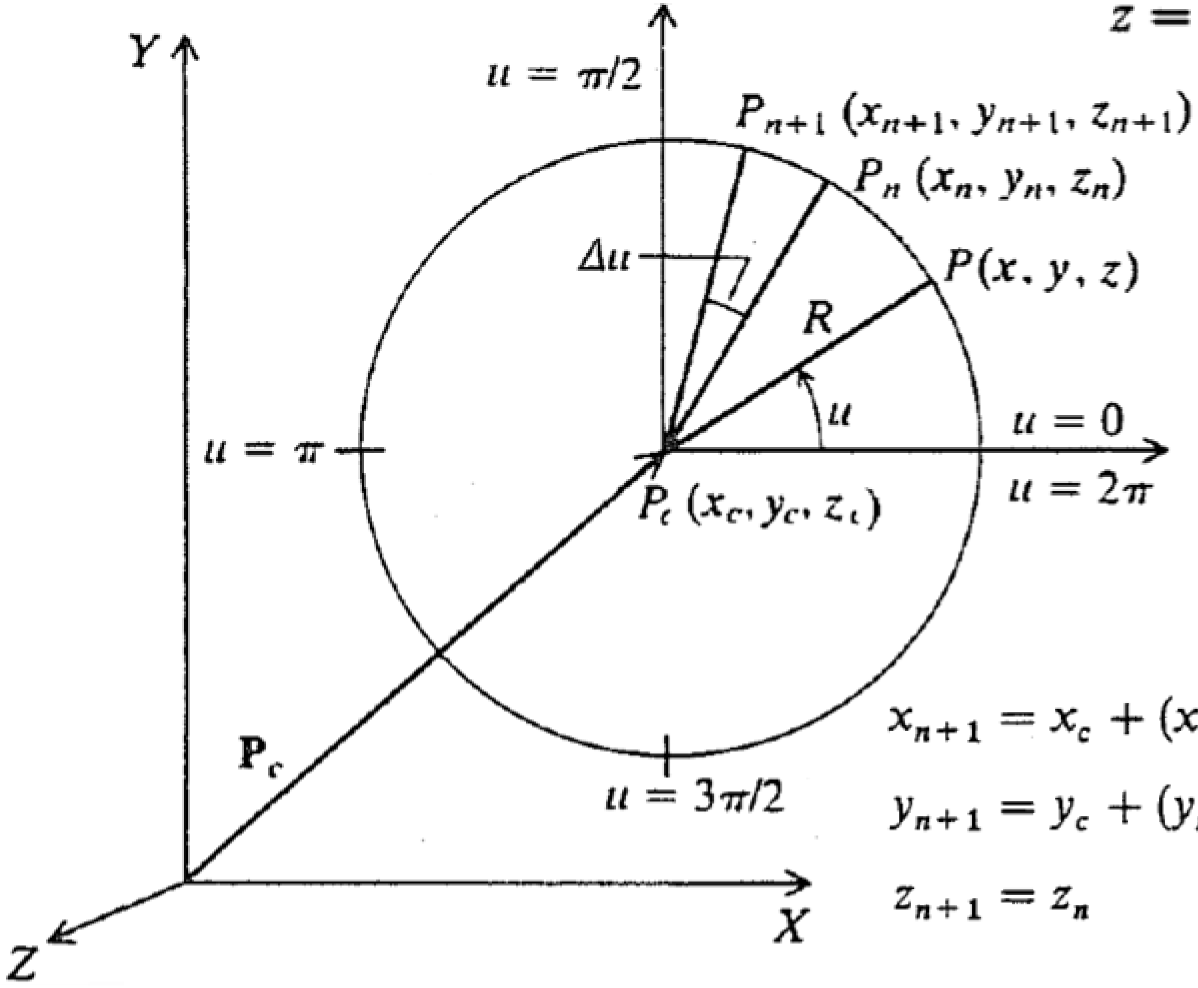
$$\mathbf{P}_v = \left[\frac{x}{1-z/d} \quad \frac{y}{1-z/d} \quad 0 \quad 1 \right]^T = \begin{bmatrix} \dfrac{x}{1-z/d} \\ \dfrac{y}{1-z/d} \\ 0 \\ 1 \end{bmatrix}$$





Parametric Circle

$$\left.\begin{aligned} x &= x_c + R\cos u \\ y &= y_c + R\sin u \\ z &= z_c \end{aligned}\right\} \quad 0 \le u \le 2\pi$$



$$\begin{aligned} x_n &= x_c + R\cos u \\ y_n &= y_c + R\sin u \\ x_{n+1} &= x_c + R\cos(u + \Delta u) \\ y_{n+1} &= y_c + R\sin(u + \Delta u) \\ z_{n+1} &= z_n \end{aligned}$$

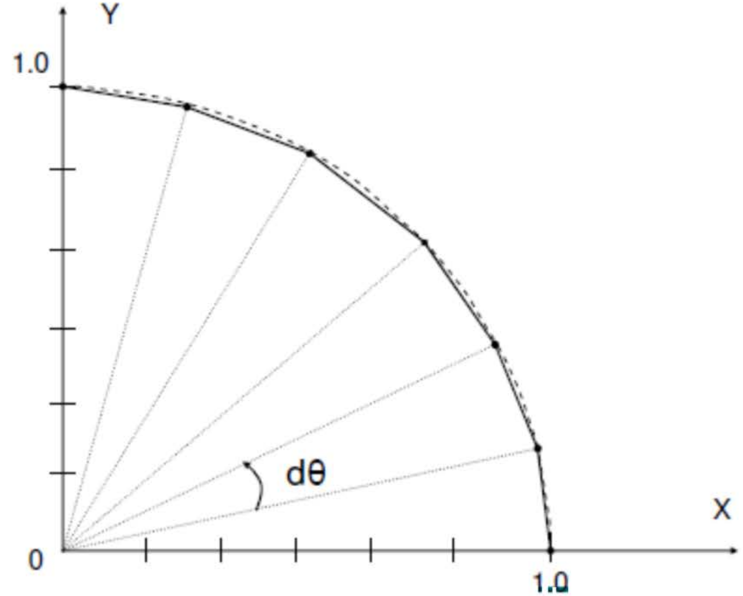
$$\begin{aligned} x_{n+1} &= x_c + (x_n - x_c)\cos\Delta u - (y_n - y_c)\sin\Delta u \\ y_{n+1} &= y_c + (y_n - y_c)\cos\Delta u + (x_n - x_c)\sin\Delta u \\ z_{n+1} &= z_n \end{aligned}$$

Parametric Circle

$$\begin{cases} x_n = r\cos\theta \\ y_n = r\sin\theta \end{cases}$$

$$x_{n+1} = r\cos(\theta + d\theta) = r\cos\theta\cos d\theta - r\sin\theta\sin d\theta$$

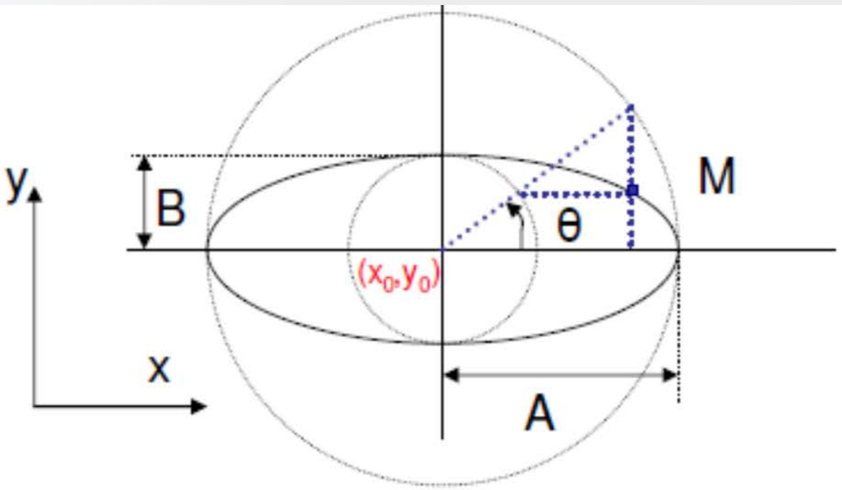
$$\begin{cases} x_{n+1} = x_n\cos d\theta - y_n\sin d\theta \\ y_{n+1} = y_n\cos d\theta + x_n\sin d\theta \end{cases}$$



Other Parametric Curves

Ellipse

$$\begin{cases} x = x_o + A\cos\theta \\ y = y_o + B\sin\theta \quad\quad 0 \le \theta \le 2\pi \\ z = z_o \end{cases}$$



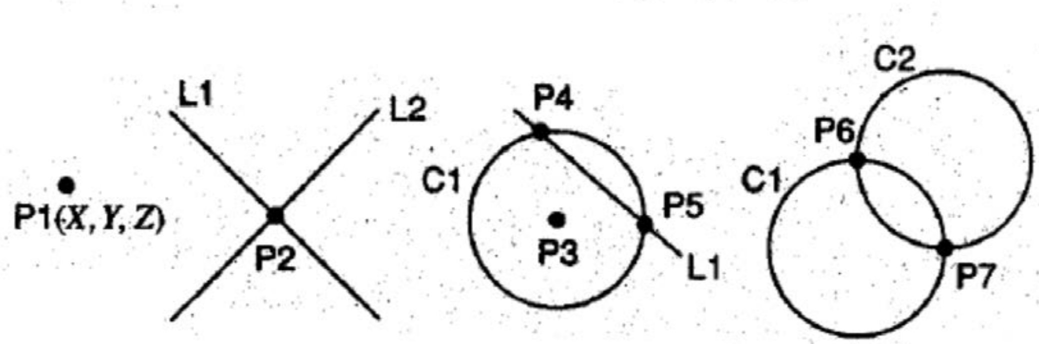
Parabola

$$\begin{cases} x = x_o + Au^2 \\ y = y_o + 2Au \quad\quad 0 \le u \le \infty \\ z = z_o \end{cases}$$

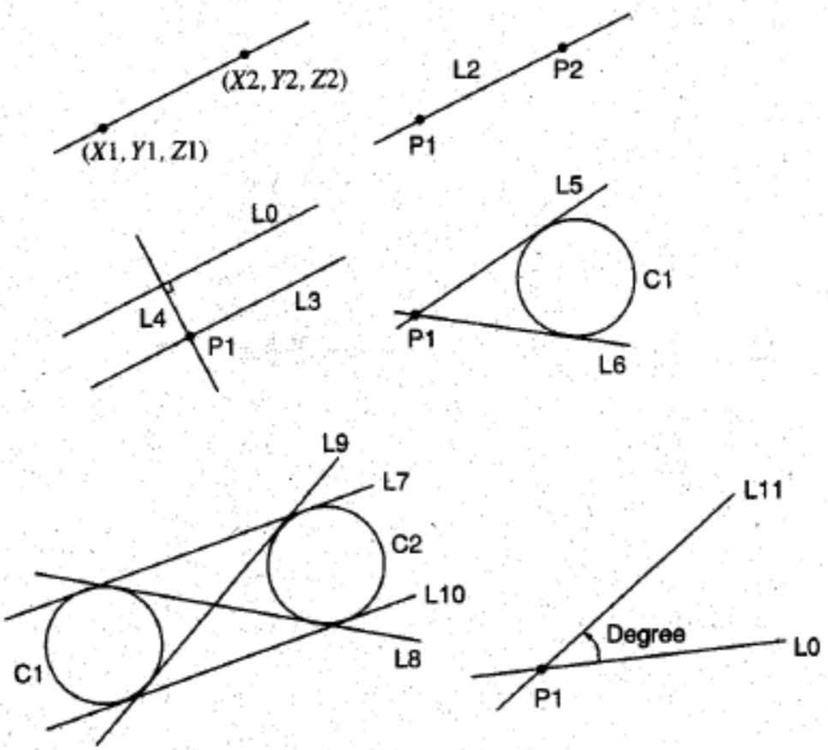
2D CAD

APT Statements

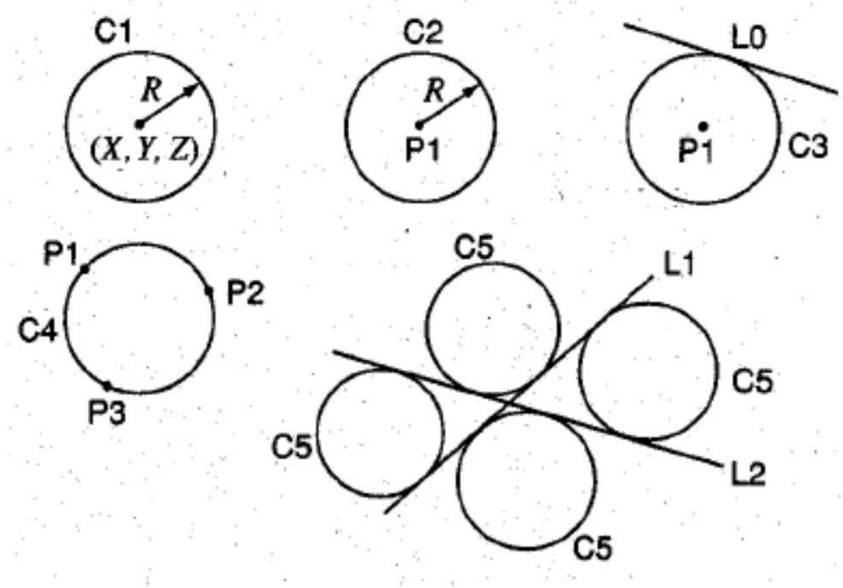
Points



Lines



Circles



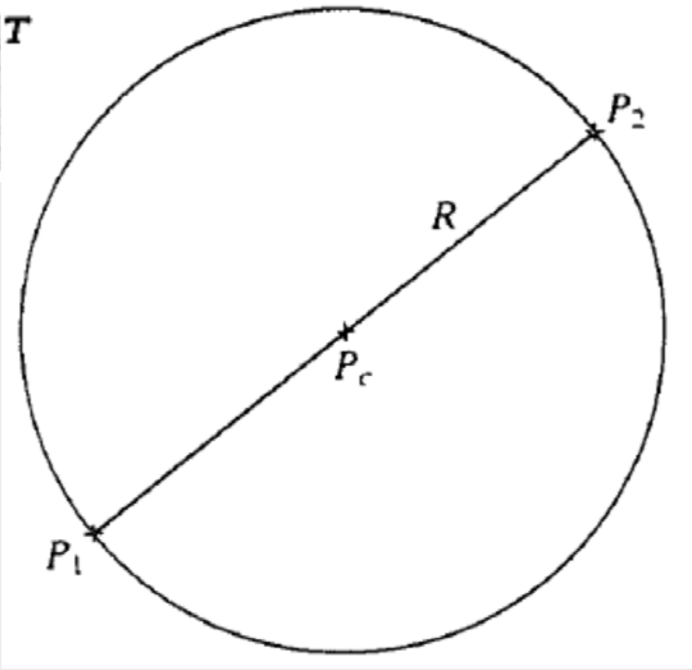
Circle defined by diameter P1 P2

Circle radius ***R*** and center ***P***$_c$ are

$$R = \frac{1}{2}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\mathbf{P}_c = \frac{1}{2}(\mathbf{P}_1 + \mathbf{P}_2)$$

$$[x_c \quad y_c \quad z_c]^T = \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \quad \frac{z_1 + z_2}{2}\right]^T$$



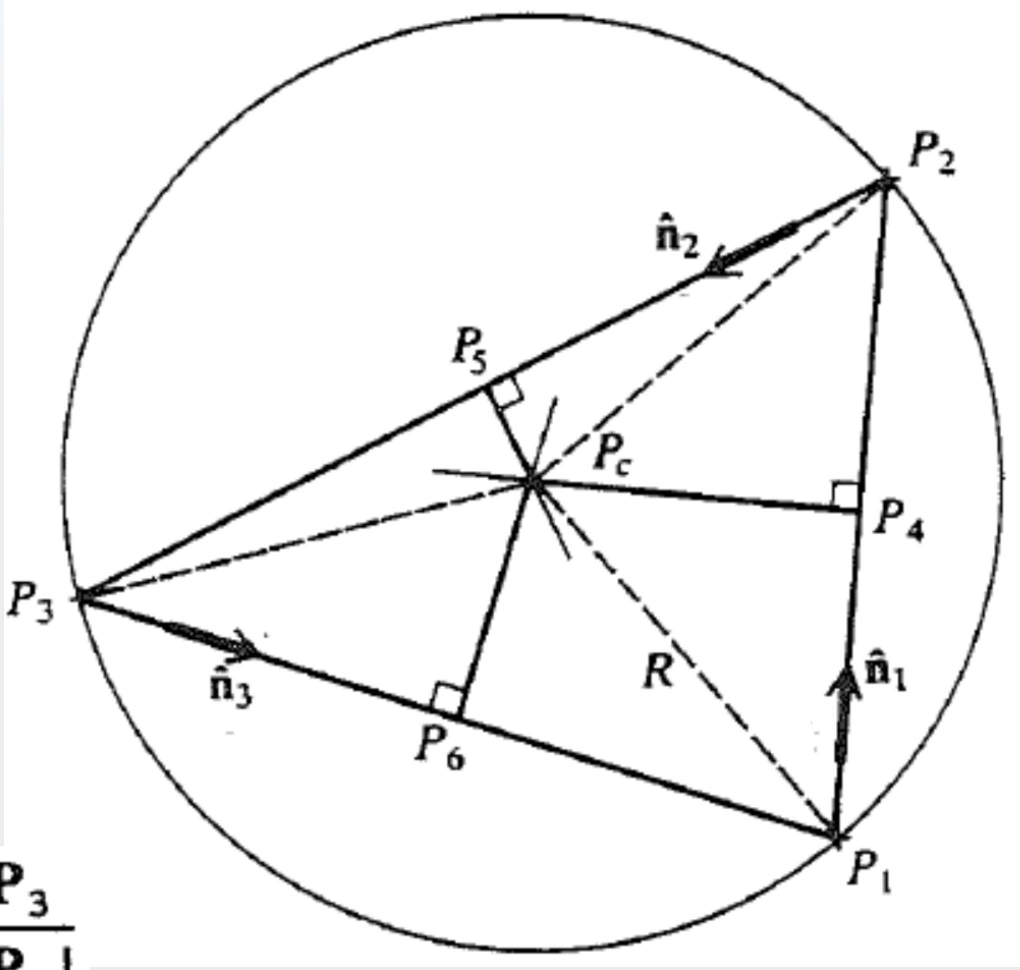
Circle passing through three points

Circle center **Pc** is the intersection of the perpendicular lines to the chords **P1P2** , **P2P3** , **P2P1** from their midpoints **P6** , **P4** , **P5** .

$$\hat{\mathbf{n}}_1 = \frac{\mathbf{P}_2 - \mathbf{P}_1}{|\mathbf{P}_2 - \mathbf{P}_1|}$$

$$\hat{\mathbf{n}}_2 = \frac{\mathbf{P}_3 - \mathbf{P}_2}{|\mathbf{P}_3 - \mathbf{P}_2|}$$

$$\hat{\mathbf{n}}_3 = \frac{\mathbf{P}_1 - \mathbf{P}_3}{|\mathbf{P}_1 - \mathbf{P}_3|}$$

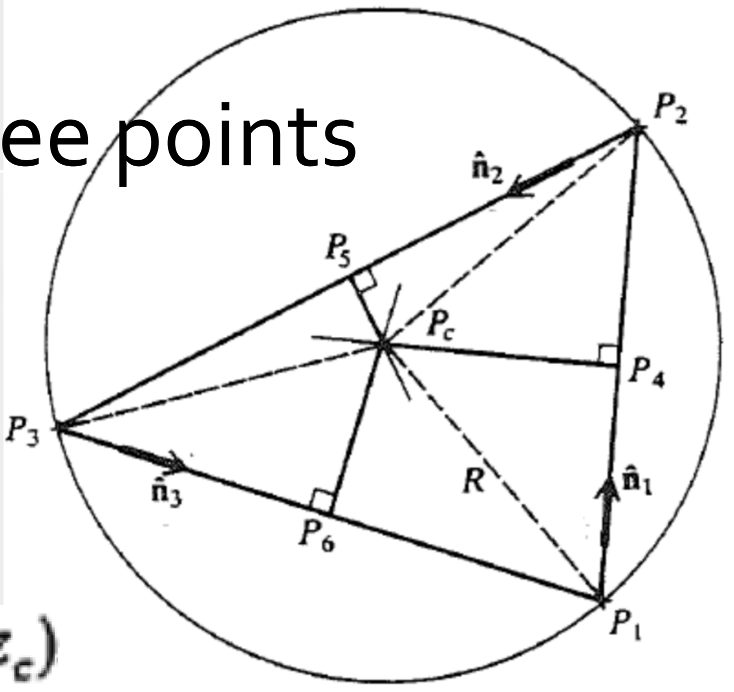


Circle passing through three points

$$(\mathbf{P}_c - \mathbf{P}_1) \cdot \hat{\mathbf{n}}_1 = \frac{|\mathbf{P}_2 - \mathbf{P}_1|}{2}$$

$$(\mathbf{P}_c - \mathbf{P}_2) \cdot \hat{\mathbf{n}}_2 = \frac{|\mathbf{P}_3 - \mathbf{P}_2|}{2}$$

$$(\mathbf{P}_c - \mathbf{P}_3) \cdot \hat{\mathbf{n}}_3 = \frac{|\mathbf{P}_1 - \mathbf{P}_3|}{2}$$



$$P_c\ (x_c\,,\ y_c\,,\ z_c)$$

$$\begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ n_{3x} & n_{3y} & n_{3z} \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

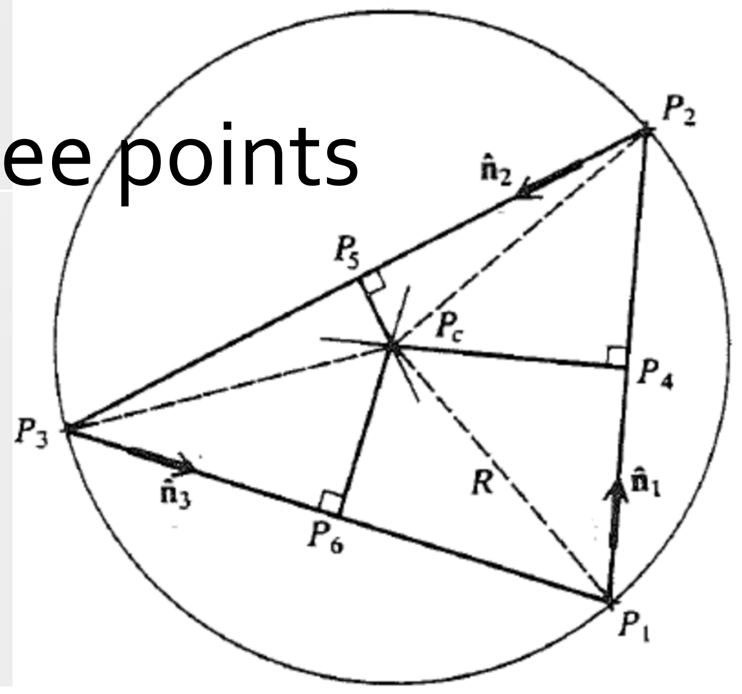
$$[A]\mathbf{P}_c = \mathbf{b}$$

$$b_1 = \frac{|\mathbf{P}_2 - \mathbf{P}_1|}{2} + (x_1 n_{1x} + y_1 n_{1y} + z_1 n_{1z})$$

$$b_2 = \frac{|\mathbf{P}_3 - \mathbf{P}_2|}{2} + (x_2 n_{2x} + y_2 n_{2y} + z_2 n_{2z})$$

$$b_3 = \frac{|\mathbf{P}_1 - \mathbf{P}_3|}{2} + (x_3 n_{3x} + y_3 n_{3y} + z_3 n_{3z})$$

Circle passing through three points



$$[A]\mathbf{P}_c = \mathbf{b}$$

$$P_c\ (x_c\,,\ y_c\,,\ z_c)$$

$$\mathbf{P}_c = [A]^{-1}\mathbf{b} = \frac{\text{Adj}\ ([A])}{|A|}\,\mathbf{b}$$

The cofactor C_{ij} is given by

$$C_{ij} = (-1)^{i+j} M_{ij}$$

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$$\mathbf{P}_c = \frac{[C]^T}{|A|}\,\mathbf{b}$$

$$C_{11} = n_{2y}n_{3z} - n_{2z}n_{3y} \qquad C_{12} = n_{2z}n_{3x} - n_{2x}n_{3z} \qquad C_{13} = n_{2x}n_{3y} - n_{2y}n_{3x}$$

$$C_{21} = n_{1z}n_{3y} - n_{1y}n_{3z} \qquad C_{22} = n_{1x}n_{3z} - n_{1z}n_{3x} \qquad C_{23} = n_{1y}n_{3x} - n_{1x}n_{3y}$$

$$C_{31} = n_{1y}n_{2z} - n_{1z}n_{2y} \qquad C_{32} = n_{1z}n_{2x} - n_{1x}n_{2z} \qquad C_{33} = n_{1x}n_{2y} - n_{1y}n_{2x}$$

$$|A| = n_{1x}(n_{2y}n_{3z} - n_{2z}n_{3y}) - n_{1y}(n_{2x}n_{3z} - n_{2z}n_{3x}) + n_{1z}(n_{2x}n_{3y} - n_{2y}n_{3x})$$

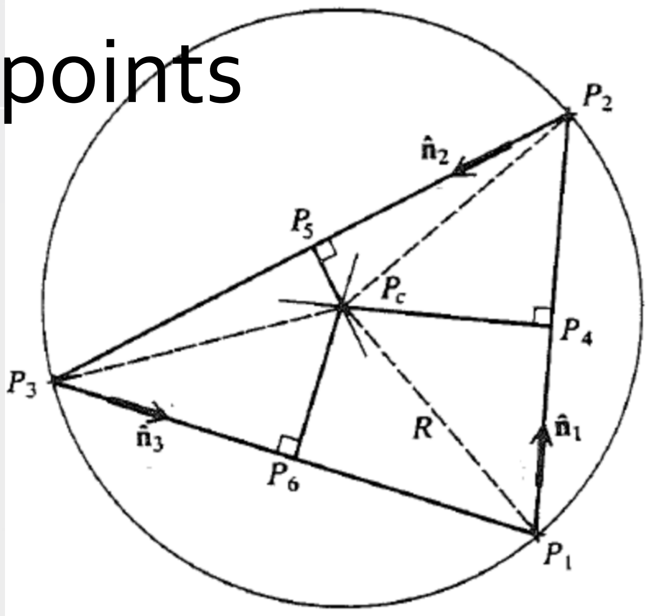
Circle passing through three points

$$\mathbf{P}_c = \frac{[C]^T}{|A|}\mathbf{b}$$

$$x_c = \frac{1}{|A|}(C_{11}b_1 + C_{21}b_2 + C_{31}b_3)$$

$$y_c = \frac{1}{|A|}(C_{12}b_1 + C_{22}b_2 + C_{32}b_3)$$

$$z_c = \frac{1}{|A|}(C_{13}b_1 + C_{23}b_2 + C_{33}b_3)$$



$$R = |\mathbf{P}_c - \mathbf{P}_1| = |\mathbf{P}_c - \mathbf{P}_2| = |\mathbf{P}_c - \mathbf{P}_3|$$

$$R = \sqrt{(x_c - x_1)^2 + (y_c - y_1)^2 + (z_c - z_1)^2}$$

For 2D case:

$$\begin{bmatrix} n_{1x} & n_{1y} \\ n_{2x} & n_{2y} \end{bmatrix}\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$b_1 = \frac{|\mathbf{P}_2 - \mathbf{P}_1|}{2} + (x_1 n_{1x} + y_1 n_{1y})$$

$$b_2 = \frac{|\mathbf{P}_3 - \mathbf{P}_2|}{2} + (x_2 n_{2x} + y_2 n_{2y})$$

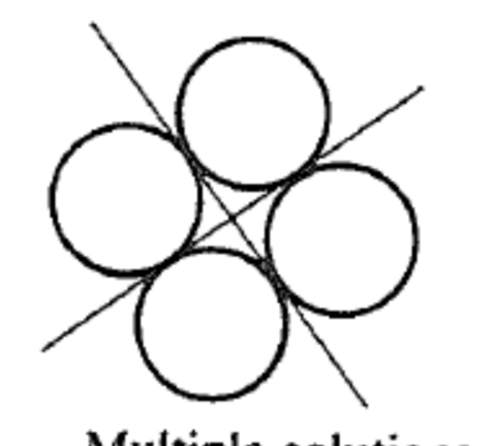
$$P_c(x_c,\ y_c)$$

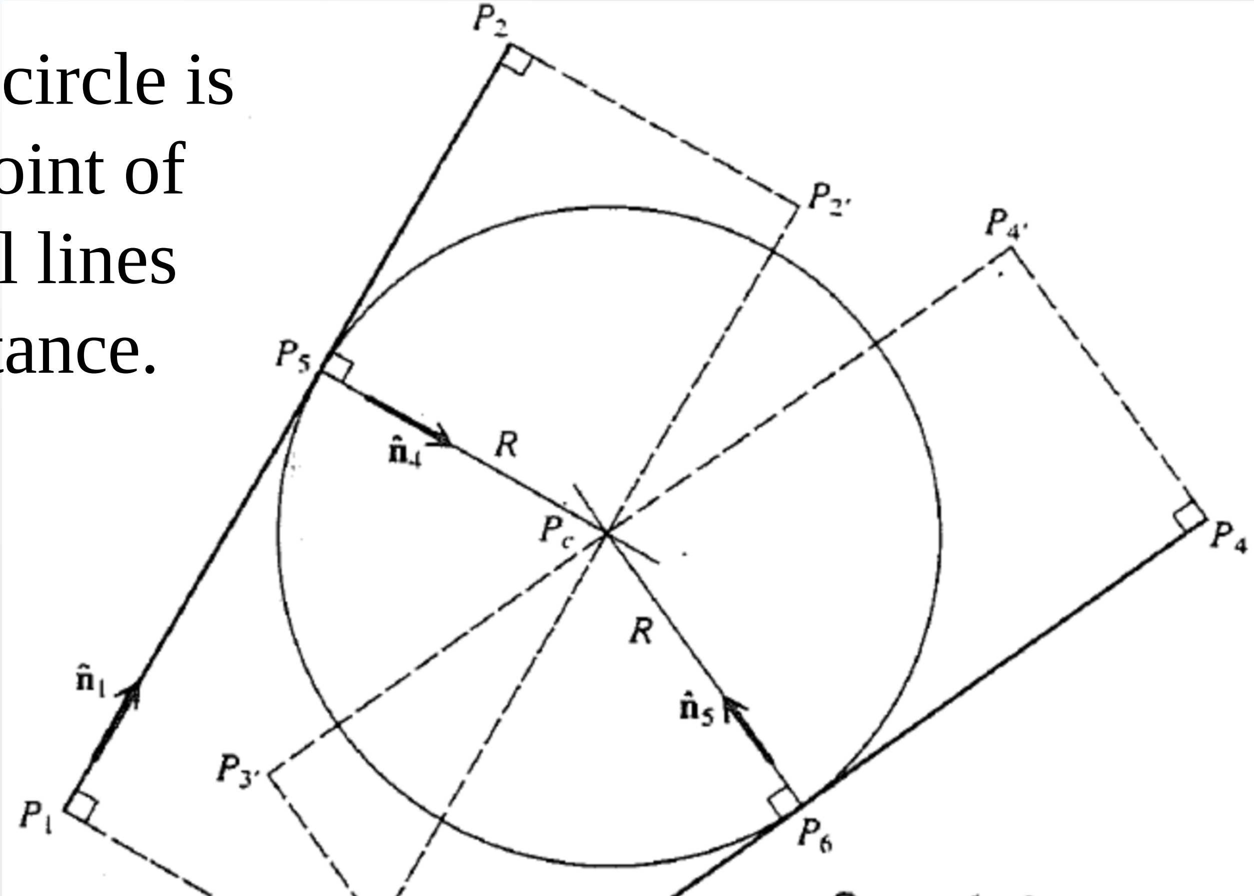
$$x_c = \frac{n_{2y}b_1 - n_{1y}b_2}{n_{1x}n_{2y} - n_{1y}n_{2x}}$$

$$y_c = \frac{n_{1x}b_2 - n_{2x}b_1}{n_{1x}n_{2y} - n_{1y}n_{2x}}$$

Circle tangent to two lines with a given R

The center of the circle is the intersection point of two offset parallel lines with radius R distance.

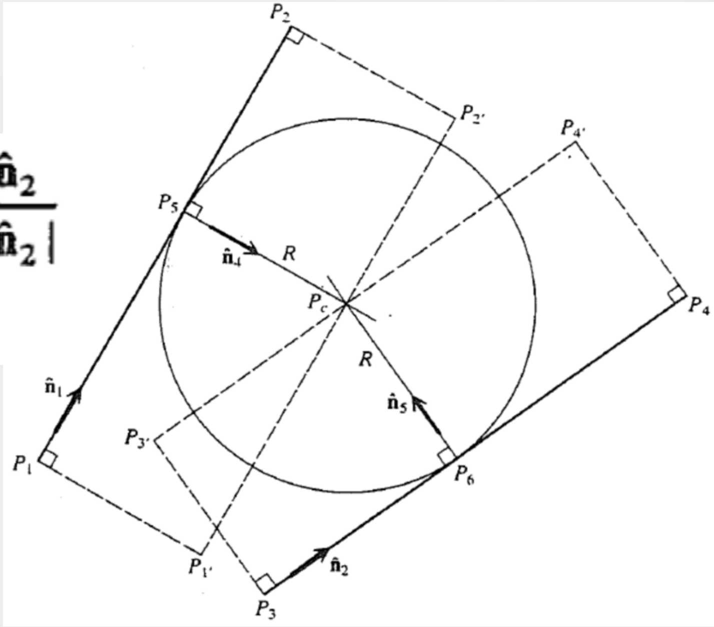

Multiple solutions



One solution

$$\hat{\mathbf{n}}_1 = \frac{\mathbf{P}_2 - \mathbf{P}_1}{|\mathbf{P}_2 - \mathbf{P}_1|} \qquad \hat{\mathbf{n}}_2 = \frac{\mathbf{P}_4 - \mathbf{P}_3}{|\mathbf{P}_4 - \mathbf{P}_3|} \qquad \hat{\mathbf{n}}_3 = \frac{\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2}{|\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2|}$$

$$\hat{\mathbf{n}}_4 = \hat{\mathbf{n}}_3 \times \hat{\mathbf{n}}_1 \qquad \hat{\mathbf{n}}_5 = \hat{\mathbf{n}}_2 \times \hat{\mathbf{n}}_3$$



$$\mathbf{P}_{1'} = \mathbf{P}_1 + R\hat{\mathbf{n}}_4 \qquad \mathbf{P}_{2'} = \mathbf{P}_2 + R\hat{\mathbf{n}}_4$$

$$\mathbf{P}_{3'} = \mathbf{P}_3 + R\hat{\mathbf{n}}_5 \qquad \mathbf{P}_{4'} = \mathbf{P}_4 + R\hat{\mathbf{n}}_5$$

The parametric vector equations of parallel lines

$$\mathbf{P} = \mathbf{P}_1 + u(\mathbf{P}_2 - \mathbf{P}_1) + R\hat{\mathbf{n}}_4$$

$$\mathbf{P} = \mathbf{P}_3 + v(\mathbf{P}_4 - \mathbf{P}_3) + R\hat{\mathbf{n}}_5$$

Intersection point of two lines

$$\mathbf{P}_1 + u(\mathbf{P}_2 - \mathbf{P}_1) + R\hat{\mathbf{n}}_4 = \mathbf{P}_3 + v(\mathbf{P}_4 - \mathbf{P}_3) + R\hat{\mathbf{n}}_5$$

$$u = \frac{(\mathbf{P}_3 - \mathbf{P}_1) \cdot \hat{\mathbf{n}}_5 + (1 - \hat{\mathbf{n}}_4 \cdot \hat{\mathbf{n}}_5)R}{(\mathbf{P}_2 - \mathbf{P}_1) \cdot \hat{\mathbf{n}}_5}$$

$$\mathbf{P}_c = \mathbf{P}_1 + \left[\frac{(\mathbf{P}_3 - \mathbf{P}_1) \cdot \hat{\mathbf{n}}_5 + (1 - \hat{\mathbf{n}}_4 \cdot \hat{\mathbf{n}}_5)R}{(\mathbf{P}_2 - \mathbf{P}_1) \cdot \hat{\mathbf{n}}_5}\right](\mathbf{P}_2 - \mathbf{P}_1) + R\hat{\mathbf{n}}_4$$

Fillet circle to perpendicular corner

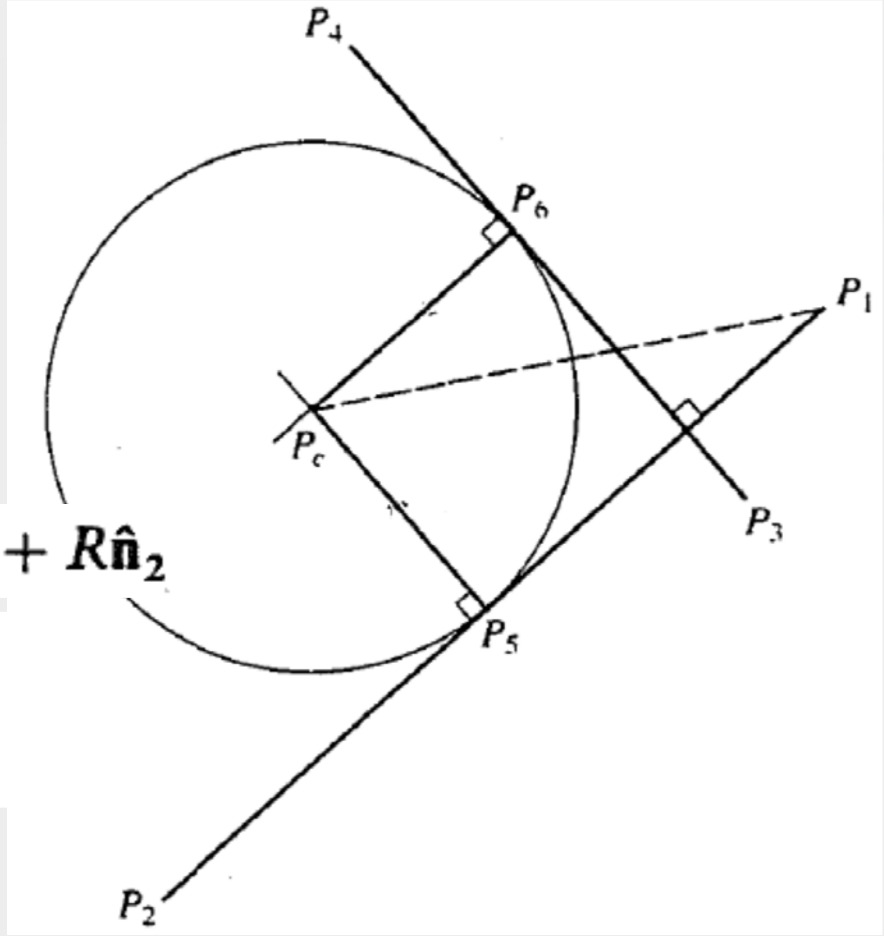
$$\mathbf{P}_c = \mathbf{P}_1 + \left[\frac{(\mathbf{P}_3 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_5 + (1 - \hat{\mathbf{n}}_4\cdot\hat{\mathbf{n}}_5)R}{(\mathbf{P}_2 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_5}\right](\mathbf{P}_2 - \mathbf{P}_1) + R\hat{\mathbf{n}}_4$$

$$u = \frac{(\mathbf{P}_3 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_5 + (1 - \hat{\mathbf{n}}_4\cdot\hat{\mathbf{n}}_5)R}{(\mathbf{P}_2 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_5}$$

$$\hat{\mathbf{n}}_4 = \hat{\mathbf{n}}_2,\ \ \hat{\mathbf{n}}_5 = \hat{\mathbf{n}}_1,\ \text{and}\ \ \hat{\mathbf{n}}_4\cdot\hat{\mathbf{n}}_5 = 0$$

$$u = \frac{(\mathbf{P}_3 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_1 + R}{|\mathbf{P}_2 - \mathbf{P}_1|}$$

$$\mathbf{P}_c = \mathbf{P}_1 + [(\mathbf{P}_3 - \mathbf{P}_1)\cdot\hat{\mathbf{n}}_1 + R]\hat{\mathbf{n}}_1 + R\hat{\mathbf{n}}_2$$



Trim points are

$$\mathbf{P}_5 = \mathbf{P}_c - R\hat{\mathbf{n}}_4$$

$$\mathbf{P}_6 = \mathbf{P}_c - R\hat{\mathbf{n}}_5$$

Ellipses

$$x_{n+1} = x_c + (x_n - x_c)\cos\Delta u - \frac{A}{B}(y_n - y_c)\sin\Delta u$$

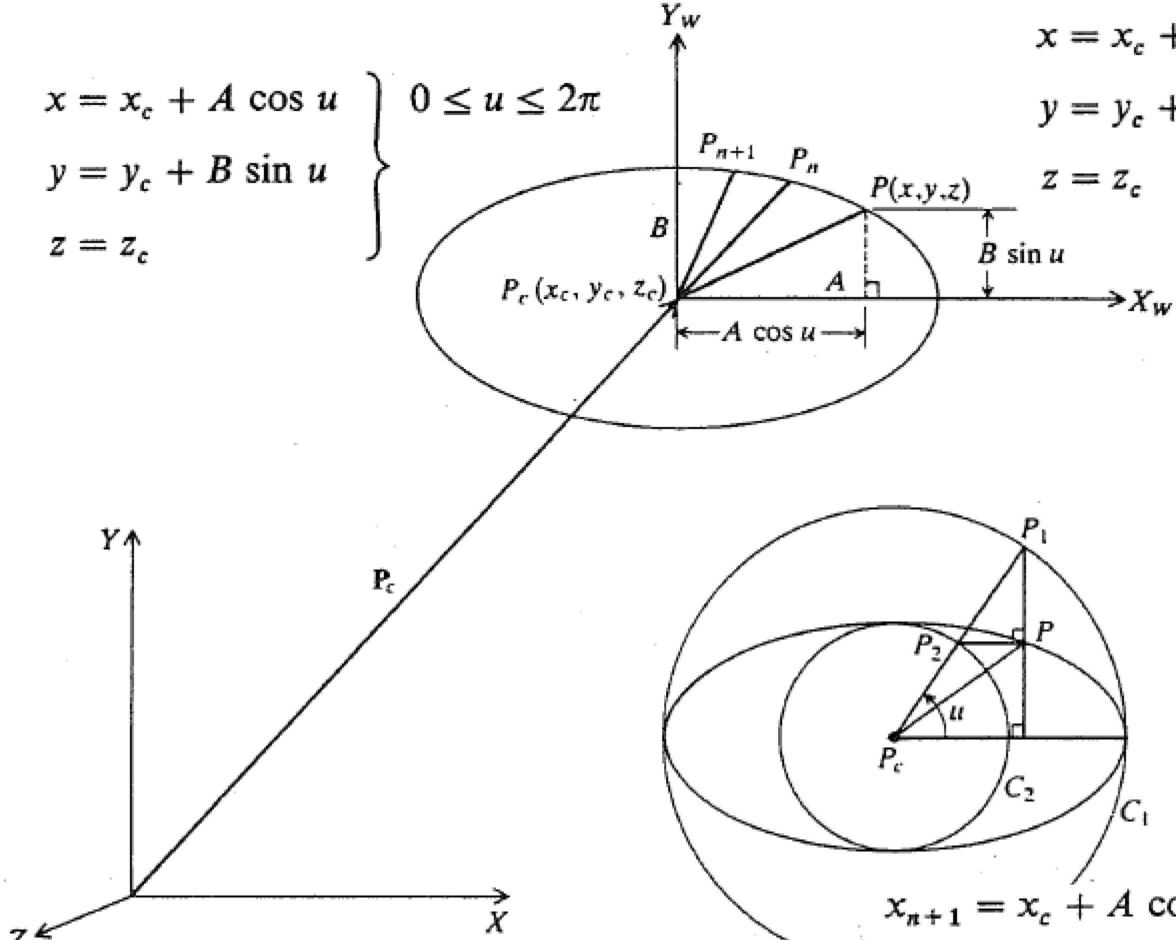
$$y_{n+1} = y_c + (y_n - y_c)\cos\Delta u + \frac{A}{B}(x_n - x_c)\sin\Delta u$$

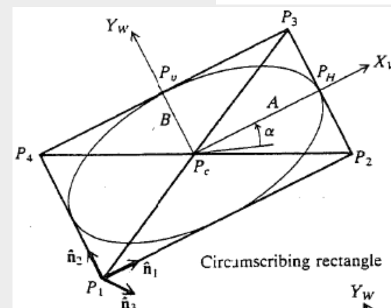
$$z_{n+1} = z_n$$

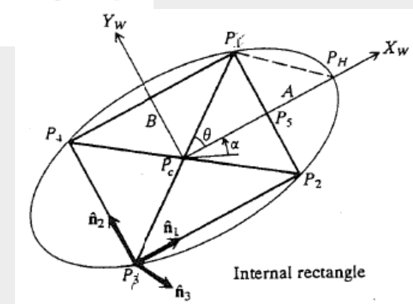
$$\left.\begin{aligned} x &= x_c + A\cos u \\ y &= y_c + B\sin u \\ z &= z_c \end{aligned}\right\} \quad 0 \le u \le 2\pi$$

$$\left.\begin{aligned} x &= x_c + A\cos u\cos\alpha - B\sin u\sin\alpha \\ y &= y_c + A\cos u\sin\alpha + B\sin u\cos\alpha \\ z &= z_c \end{aligned}\right\}$$

$$0 \le u \le 2\pi$$







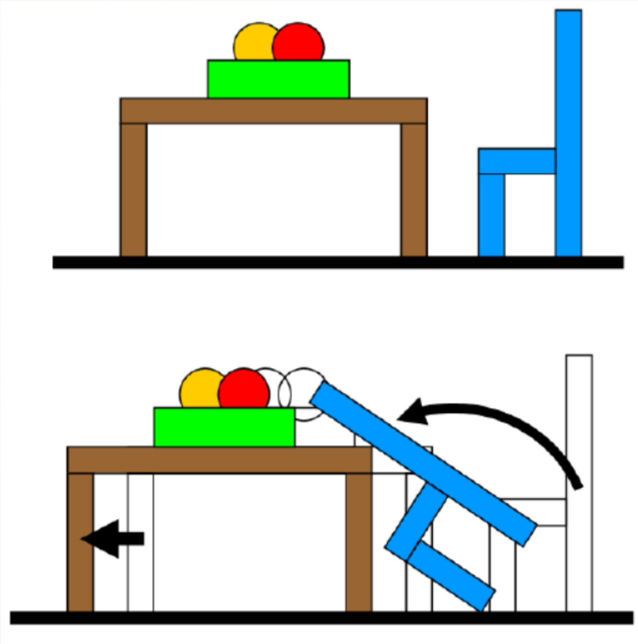
$$x_{n+1} = x_c + A\cos(u_n + \Delta u)\cos\alpha - B\sin(u_n + \Delta u)\sin\alpha$$

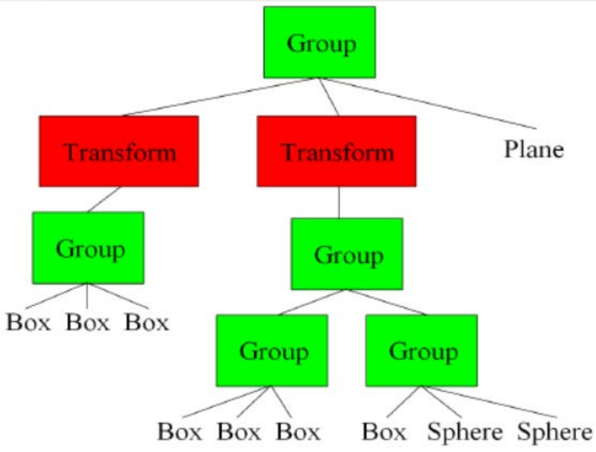
$$y_{n+1} = y_c + A\cos(u_n + \Delta u)\sin\alpha + B\sin(u_n + \Delta u)\cos\alpha$$

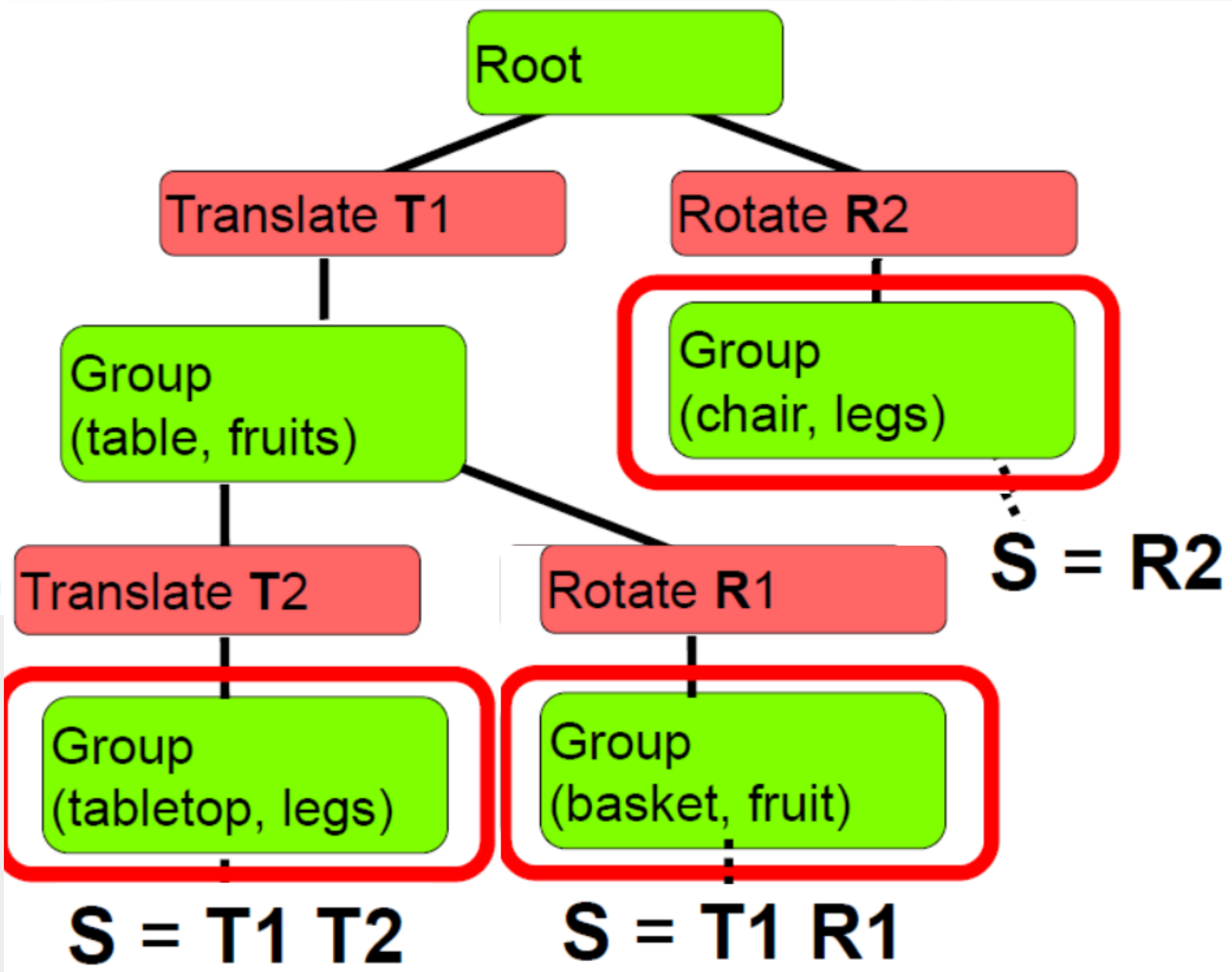
$$z_{n+1} = z_n$$

Ellipse defined by a center, major and minor axes.

Transformations in model. p' = C * T * p

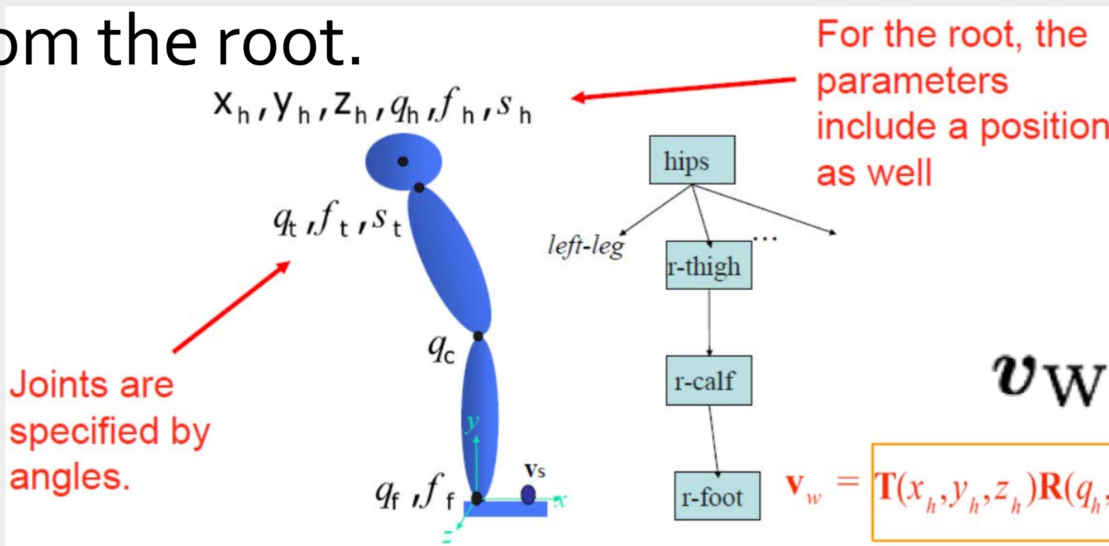






Forward Kinematics, Skeleton Hierarchy

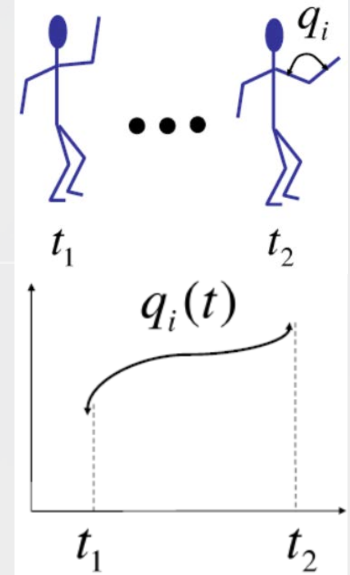
Each bone position/orientation described relative to the parent in the hierarchy. Given the skeleton parameters **p** (position of the root and the joint angles) and the position of the point in local coordinates **v**s, what is the position of the point in the world coordinates **v**w? Just apply transform accumulated from the root.



$$\boldsymbol{v}_{\mathrm{WS}} = \boldsymbol{S}(\boldsymbol{p})\,\boldsymbol{v}_{\mathrm{s}}$$

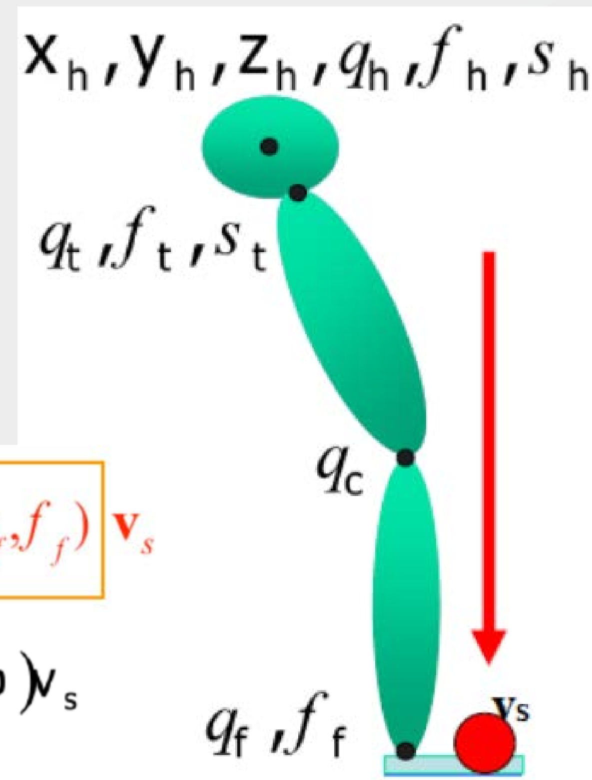
$$\mathbf{v}_w = \boxed{\mathbf{T}(x_h,y_h,z_h)\mathbf{R}(q_h,f_h,s_h)\ \mathbf{TR}(q_t,f_t,s_t)\ \mathbf{TR}(q_c)\ \mathbf{TR}(q_f,f_f)}\ \mathbf{v}_s$$

Hierarchical modeling, animation



- Hierarchical structure modeling
- Forward and inverse kinematics
- Eyes move with head
- Hands move with arms
- Feet move with legs
- Models can be animated

by specifying the joint angles
as functions of time.



$$\mathbf{v}_w = \boxed{\mathbf{T}(x_h, y_h, z_h)\mathbf{R}(q_h, f_h, s_h)\ \mathbf{TR}(q_t, f_t, s_t)\ \mathbf{TR}(q_c)\ \mathbf{TR}(q_f, f_f)}\ \mathbf{v}_s$$

$$v_w = S\left(\underbrace{x_h, y_h, z_h, \theta_h, \phi_h, \sigma_h, \theta_t, \phi_t, \sigma_t, \theta_c, \theta_f, \phi_f}_{\text{parameter vector } \mathbf{p}}\right) v_s = S(p) v_s$$

$$\boldsymbol{v}_{\mathrm{WS}} = \boldsymbol{S}(\boldsymbol{p})\, \boldsymbol{v}_{\mathrm{s}}$$

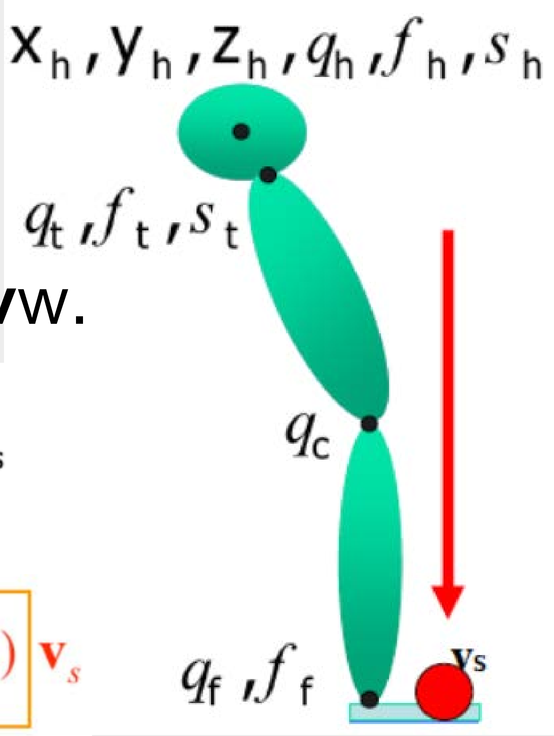
$$\left[\frac{\partial (\boldsymbol{v}_{\mathrm{WS}})_i}{\partial p_j} \right]$$

Forward Kinematics

Transformation matrix **S** for a point **v**s is a matrix composition of all joint transformations between the foot point and the root of the hierarchy.

S is a function of all the joint angles between foot point and root.

Inverse Kinematics requires solving for **p**, given **v**s and the desired position **v**w.



$$v_w = S\left(\underbrace{x_h, y_h, z_h, \theta_h, \phi_h, \sigma_h, \theta_t, \phi_t, \sigma_t, \theta_c, \theta_f, \phi_f}_{p}\right) v_s = S(p)\, v_s$$

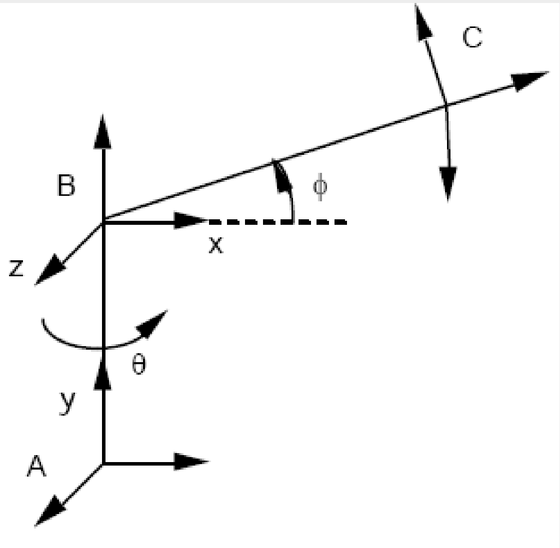
parameter vector **p**

$$\mathbf{v}_w = \boxed{\mathbf{T}(x_h, y_h, z_h)\mathbf{R}(q_h, f_h, s_h)\ \mathbf{TR}(q_t, f_t, s_t)\ \mathbf{TR}(q_c)\ \mathbf{TR}(q_f, f_f)}\ \mathbf{v}_s$$

Applications in Robotics and Simulation

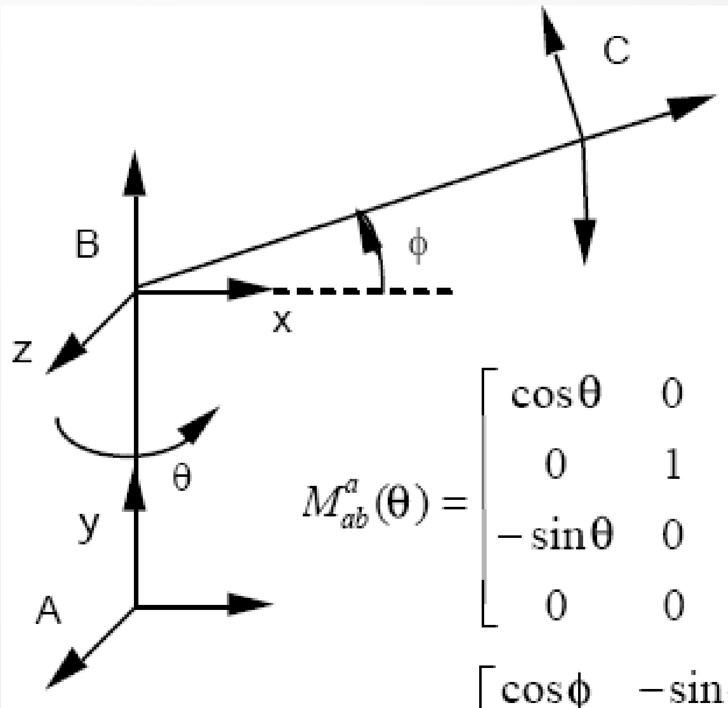
A robotic manipulator is a **kinematic chain**, i.e., a collection of solid bodies—called *links—connected at joints. The most common joints are the **revolute joint**, which* corresponds to rotational motion between two links, and the ***prismatic joint***, *which* corresponds to a translation. Most of the industrial robot "arms" in use today have only revolute joints.

Figure shows an idealized robot
with two links and two revolute joints.



Applications in Robotics and Simulation

Stick-figure model for a 2-link robot



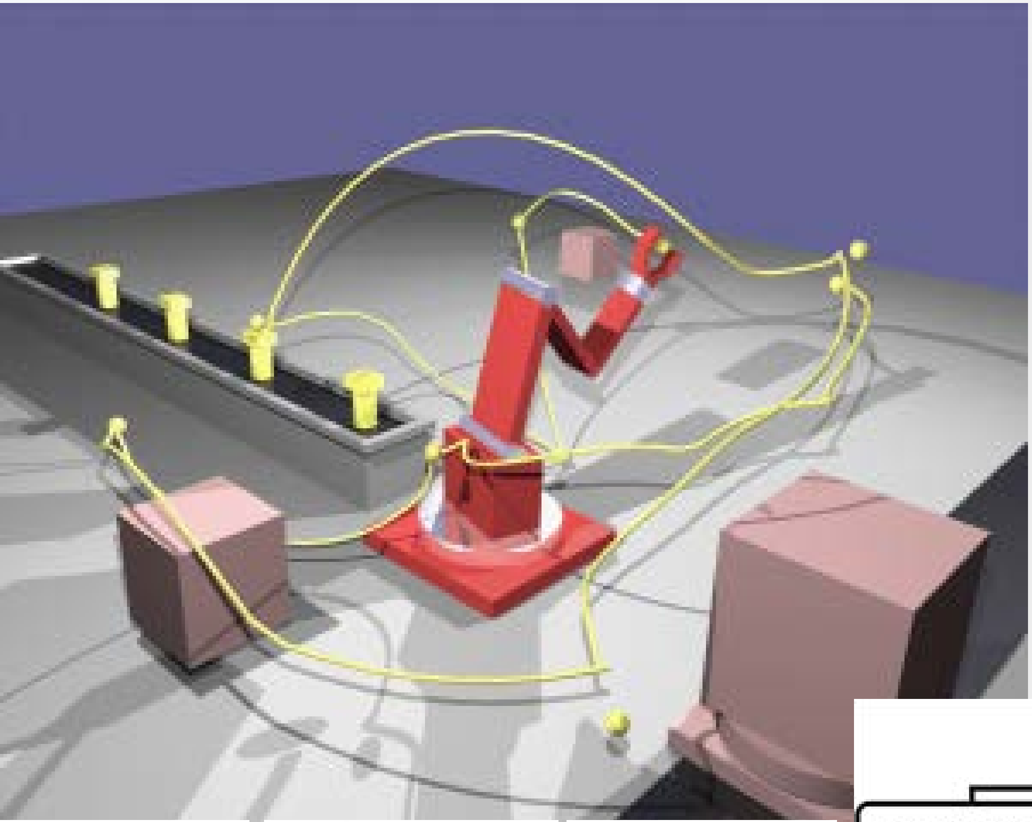
$$X^a = M^a_{ab} X^b = M^a_{ab} M^b_{bc} X^c$$

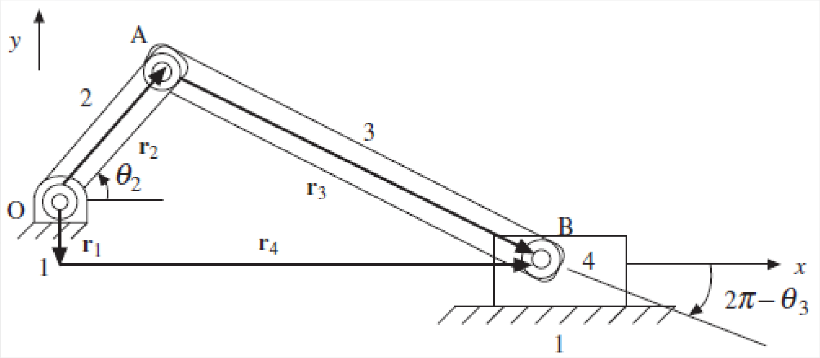
$$X^a = M^a_{ab}(\theta) M^b_{bc}(\phi) X^c$$

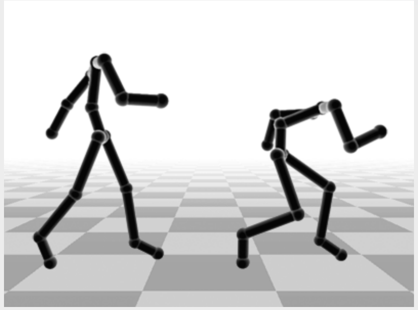
$$M^a_{ab}(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & L_1 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

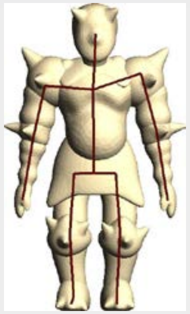
$$M^b_{bc}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & L_2\cos\phi \\ \sin\phi & \cos\phi & 0 & L_2\sin\phi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

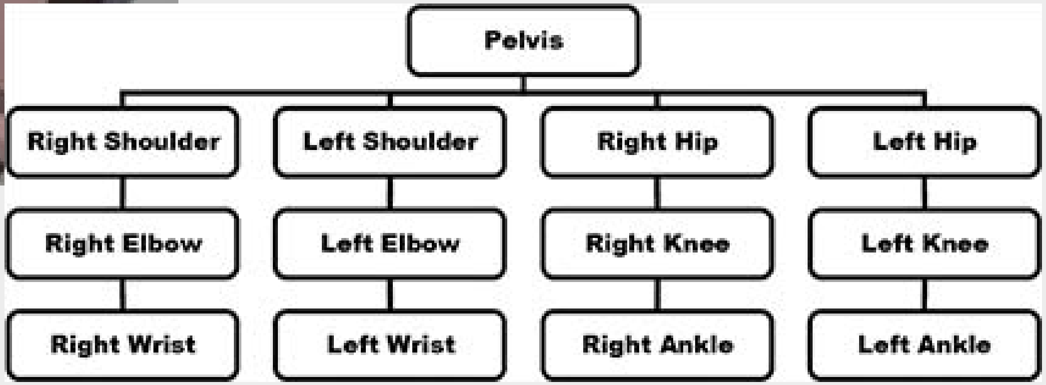
Example:
CAD Assemblies & Animation Models

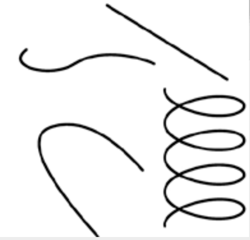


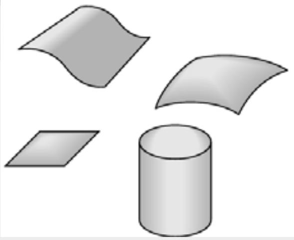


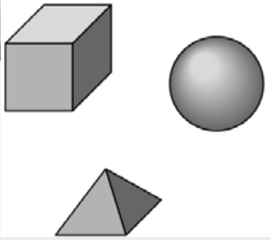












References

- *CAD/CAM Theory and Practice* , Ibrahim Zeid, McGraw Hill , 1991

- *Mathematical Elements for Computer Graphics*, Rogers, D.F., Adams, J.A., McGraw Hill, 1990.

- Computer Aided Geometric Design, Thomas W. Sederberg, 2003.