PPT

ADVANCED COMPUTER AIDED DESIGN

DEPARTMENT OF MECHANICAL ENGINEERING

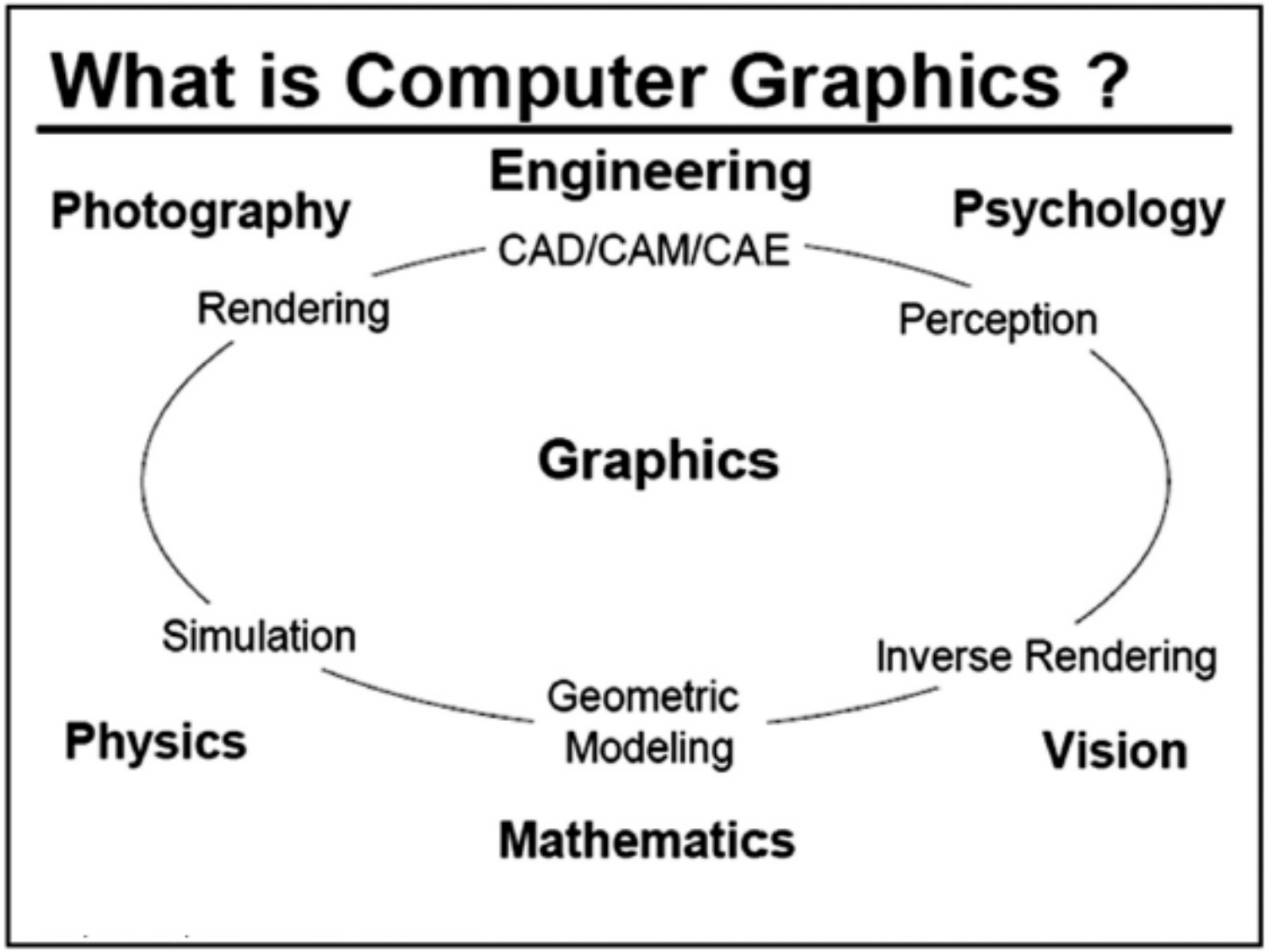
M.TECH : CAD / CAM - I SEM

by

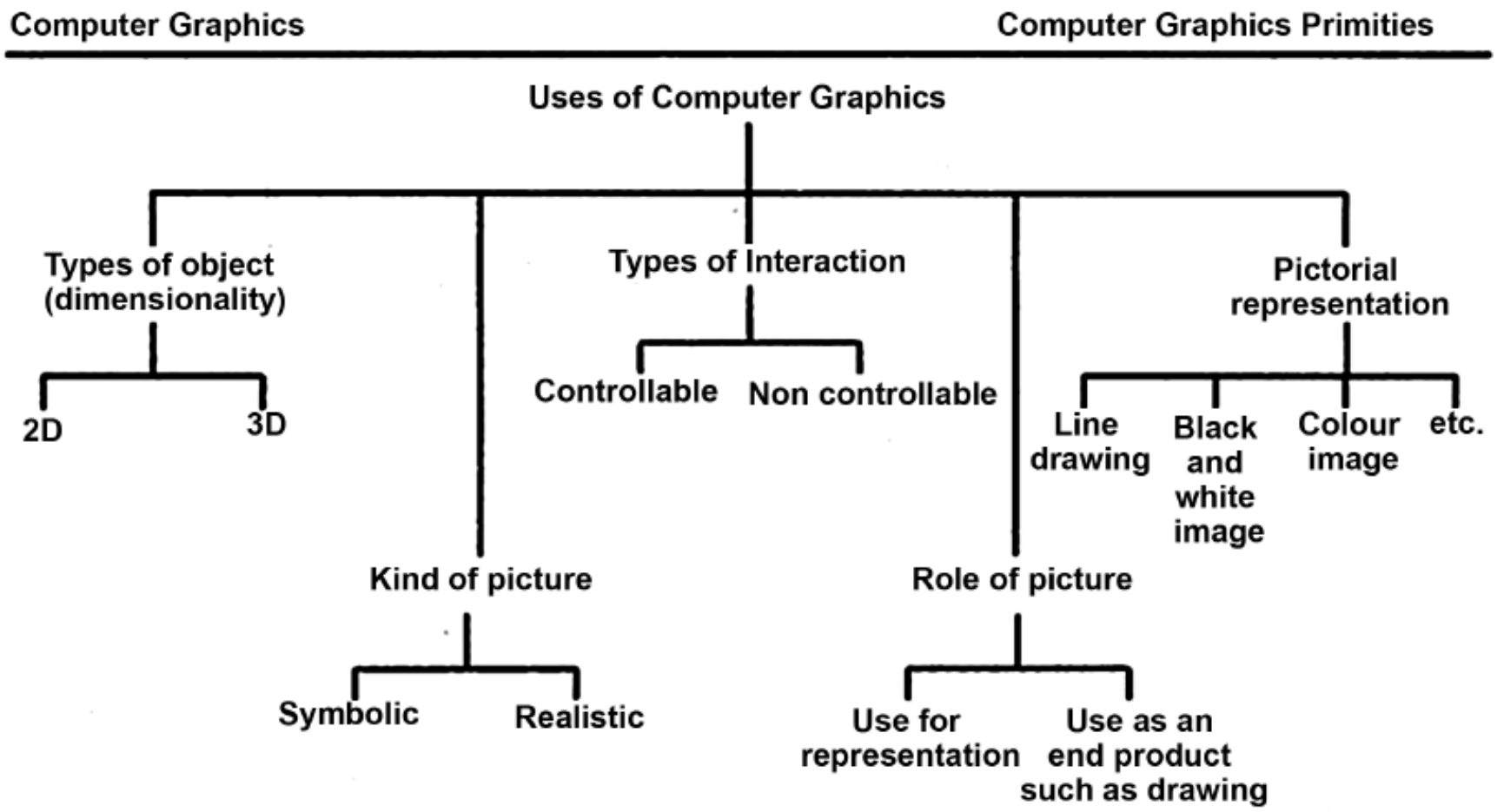
Dr. K RAGHU RAM MOHAN REDDY

Syllabus

Introduction, graphic primitives, point plotting, lines, Bresenham”s circle algorithm, ellipse, transformation in graphics, coordinate systems, view port, 2D and 3D transformation, hidden surface removal, reflection, shading and generation of character.


What is Computer Graphics ?
Engineering
Photography
Psychology
CAD/CAM/CAE
Rendering
Perception
Graphics
Simulation
Inverse Rendering
Geometric
Modeling
Physics
Vision
Mathematics

Classification of computer graphics



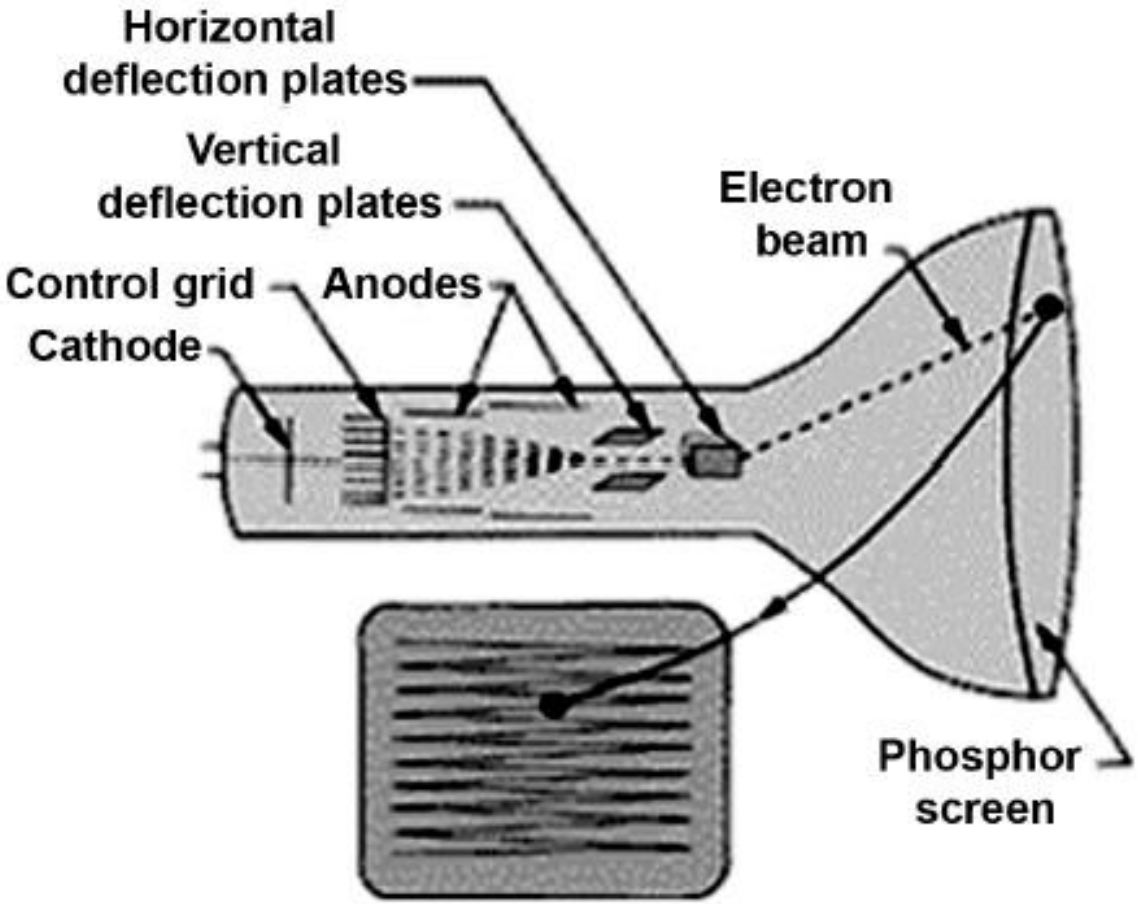
Graphic Primitives

Display devices

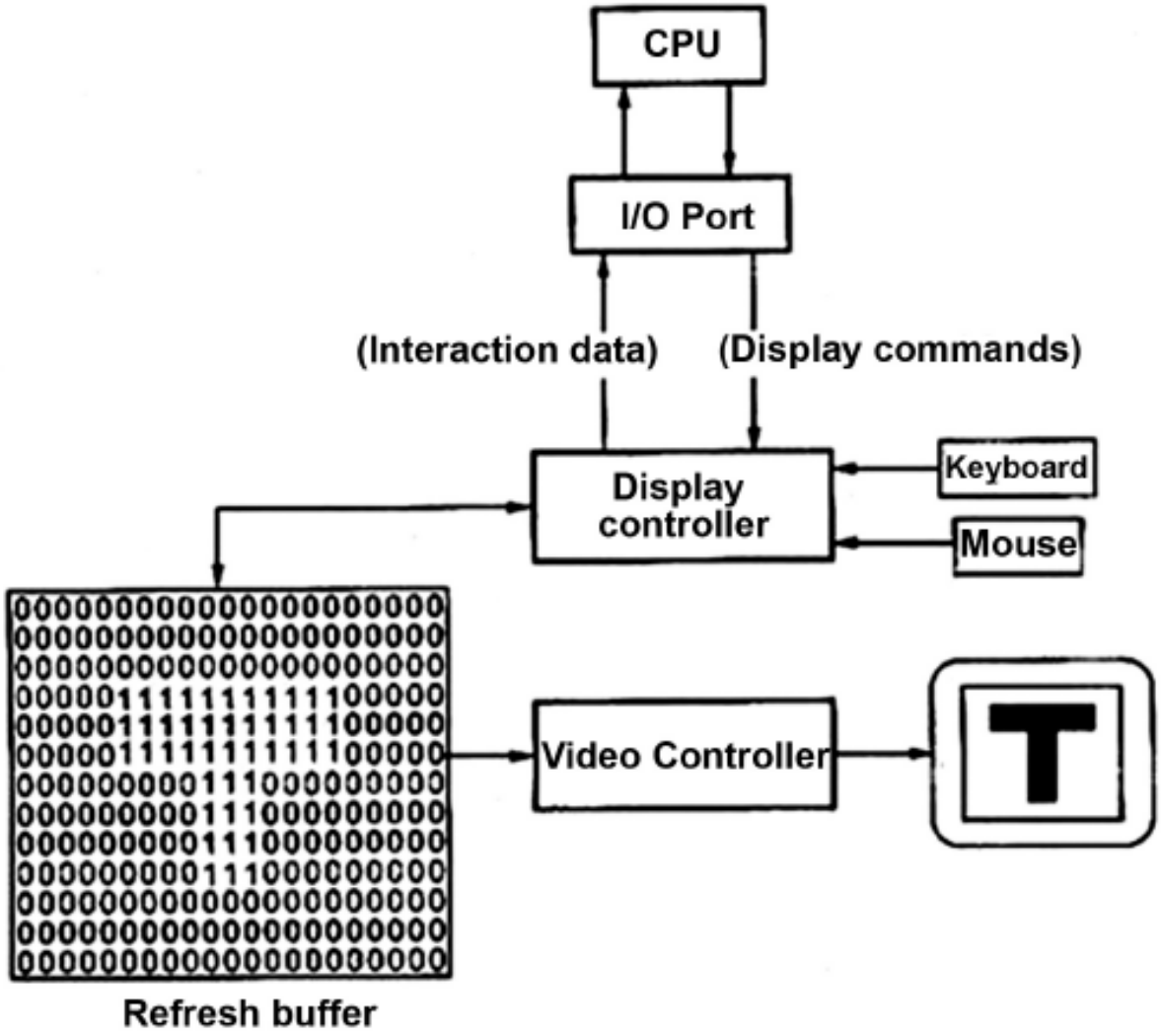
Output devices include:

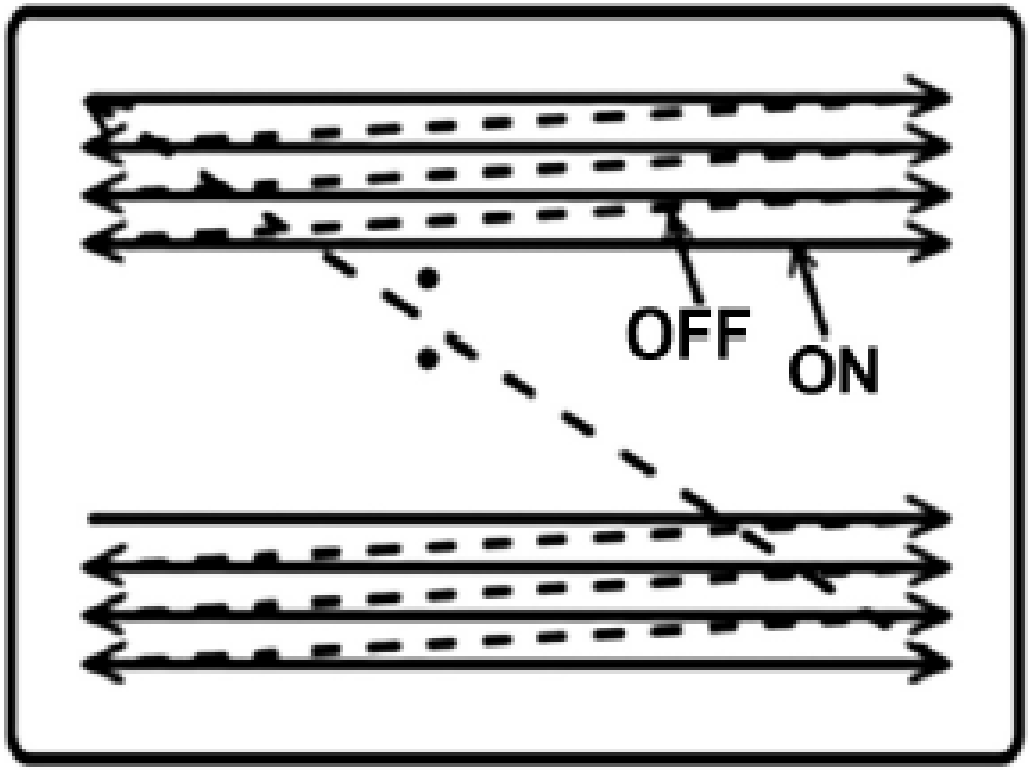
- **CRT**
- **Random scan display**
- **Rastar scan display**
- **Monitors**
- **Speakers**
- **Headphones**
- **Printer**

Cathode ray tube



Rastar Scan Display





Raster scan CRT

Bresenham's line algorithm

Bresenham's Line Algorithm

- An accurate and efficient raster line-generating algorithm, developed by Bresenham which scan converts line using only incremental integer calculations that can be modified to display circles and other curves.
- Figure shows section of display screen where straight line segments are to be drawn.

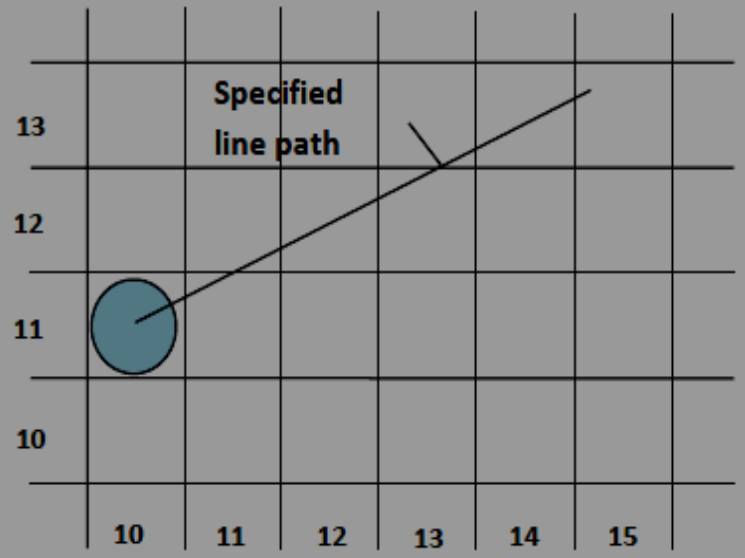


Fig. 2.4: - Section of a display screen where a straight line segment is to be plotted, starting from the pixel at column 10 on scan line 11.

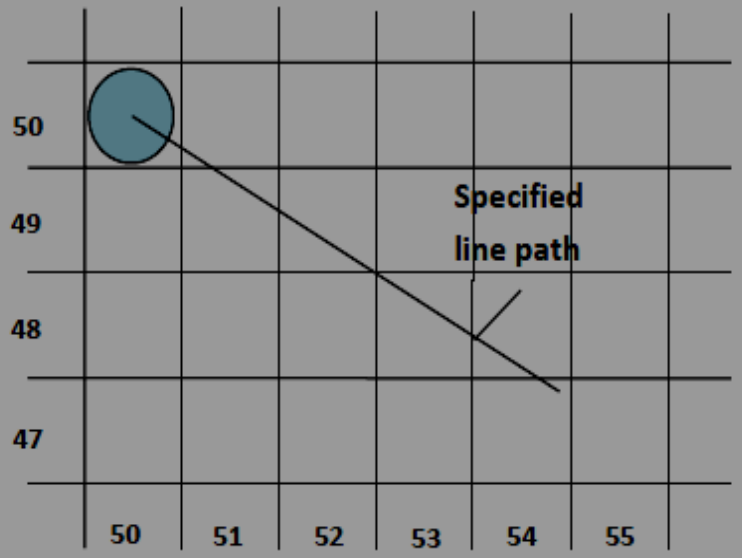
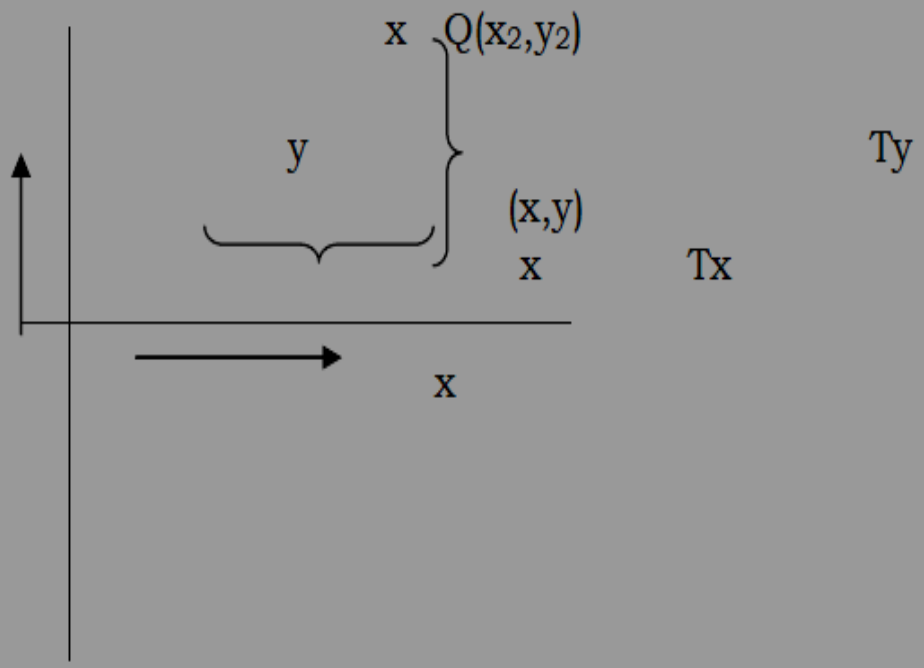


Fig. 2.5: - Section of a display screen where a negative slope line segment is to be plotted, starting from the pixel at column 50 on scan line 50.

Two Dimensional Transformations

1. Introduction
2. What is transformation
3. Basic information
4. Translation
5. Rotation
6. Scaling

Translation



Consider a point $P(x_1, y_1)$ to be translated to another point $Q(x_2, y_2)$. If we know the point value (x_2, y_2) we can directly shift to Q by displaying the pixel (x_2, y_2). On the other hand, suppose we only know that we want to shift by a distance of Tx along x axis and Ty along Y axis. Then obviously the coordinates can be derived by $x_2 = x_1 + T_x$ and $Y_2 = y_1 + T_y$.

Rotation

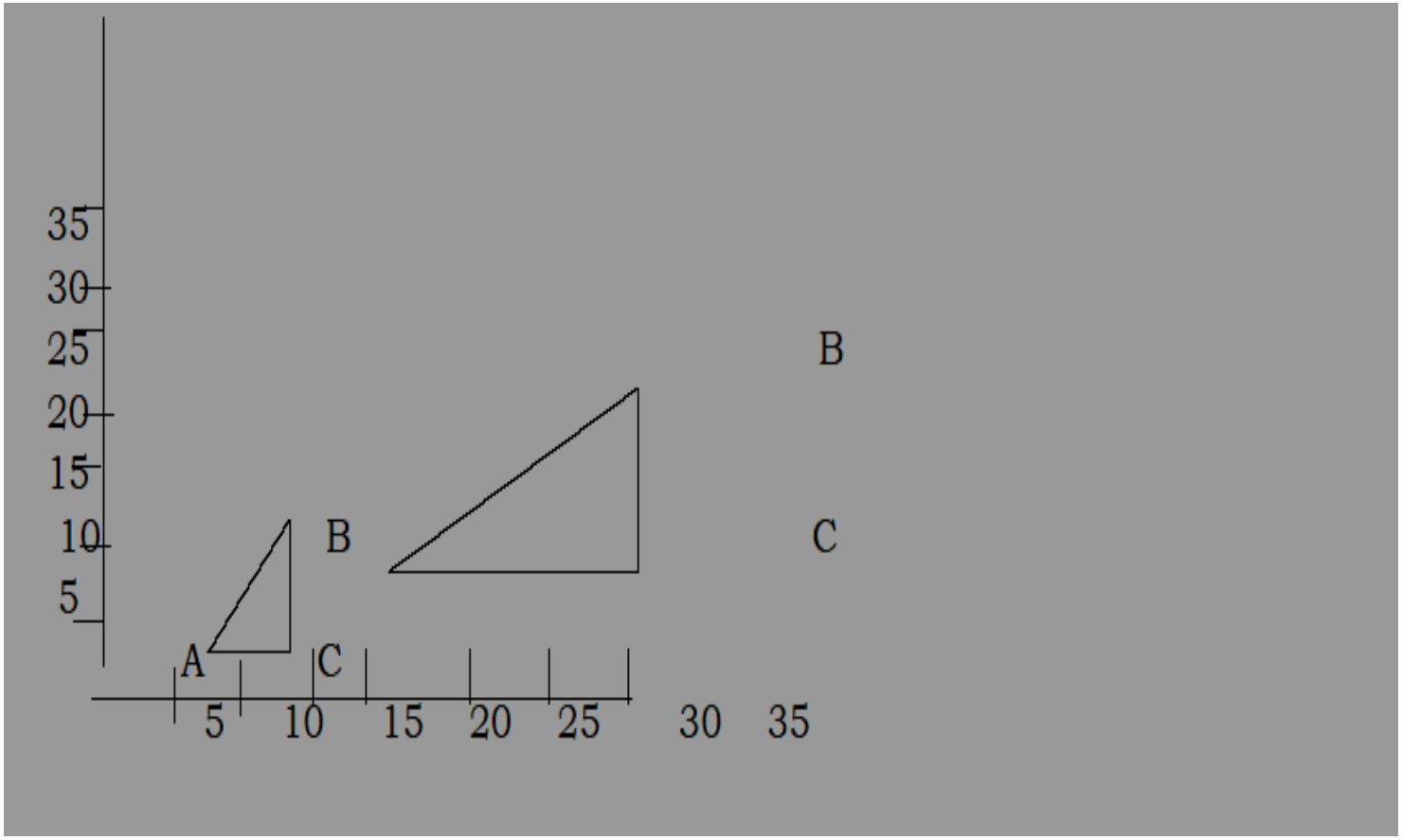
Suppose we want to rotate a point $(x_1\ y_1)$ clockwise through an angle/ about the origin of the coordinate system. Then mathematically we can show that

$$x_2 = x_1\cos\theta + y_1\sin\theta \quad \text{and}$$
$$y_2 = x_1\sin\theta - y_1\cos\theta$$

These equations become applicable only if the rotation is about the origin.

In the matrix for $[x_2\ y_2\ 1] = [x_1\ y_1\ 1] * \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Scaling



Three Dimensional Transformations

1. Introduction
2. 3D transformation
3. Translation
4. Rotation
5. Scaling

Translations

Without repeating the earlier methods, we simply write

$$[x_1 \; y_1 \; z_1 \; 1] = [x \; y \; z \; 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{pmatrix}$$

Where the point [x y z 1] gets transformed to [x_1 y_1 z_1 1] after translating by Tx, Ty and Tz along the x,y,z directions respectively.

Scaling

A given point [x y z 1] gets transformed to [x_1 y_1 z_1 1] after getting scaled by factors Sx, Sy and Sz in the three dimensions to

$$[x_1 \; y_1 \; z_1 \; 1] = [x \; y \; z \; 1] \begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation

$$[x_1 \; y_1 \; z_1 \; 1] = [x \; y \; z \; 1] \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 1 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation Matrix

Hidden Surface Removal

The Depth Buffer Algorithm
Warnock's Algorithm

The Depth – Buffer algorithm

The concept of this algorithm is extremely simple and straightforward. Given a given resolution of the screen, every pixel on the screen can represent a point on one (and only one) object (or it may be set to the back ground if it does not form a part of any object). I.e. irrespective of the number of objects in line with the pixel, it should represent the object nearest to the viewer. The algorithm aims at deciding for every pixel the object whose features it should represent. The depth-buffer algorithm used two arrays, depth and intensity value. The size of the arrays equals the number of pixels. As can be expected, the corresponding elements of the array store the depth and intensity represented by each of the pixels.

a. For every pixel, set it's depth and intensity pixels to the back ground value ie. At the end of the algorithm, if the pixel does not become a part of any of the objects it represents the background value.

b. For each polygon on the scene, find out the pixels that lie within this polygon (which is nothing but the set of pixels that are chosen if this polygon is to be displayed completely).

For each of the pixels

i) Calculate the depth Z of the polygon at that point (note that a polygon, which is inclined to the plane of the screen will have different depths at different points)

ii) If this Z is less than the previously stored value of depth in this pixel, it means the new polygon is closer than the earlier polygon which the pixel was representing and hence the new value of Z should be stored in it. (i.e from now on it represents the new polygon). The corresponding intensity is stored in intensity vector.

If the new Z is greater than the previously stored vale, the new polygon is at a farther distance than the earlier one and no changes need be made. The polygon continues to represents the previous polygon.

One may note that at the end of the processing of all the polygons, every pixel, will have the intensity value of the object which it should display in its intensity location and this can be displayed.

This simple algorithm, as can be expected, works on the image space. The scene should have properly projected and clipped before the algorithm is used.

The basic limitation of the algorithm is it's computational intensiveness. On a 1024 X 1024 screen it will have to evaluate the status of each of these pixels in a limiting case. In it's present form, it does not use any of the coherence or other geometric properties to reduce the computational efforts.

To reduce the storage, some times the screen is divided into smaller regions like say 50 X 50 or 100 X 100 pixels, computations made for each of this religions, displayed on the screen and then the next region is undertaken. However this can be both advantageous and disadvantageous. It is obvious that such a division of screen would need each of the polygons to be processed for each of the regions – thereby increasing the computational efforts. This is disadvantage. But when smaller regions are being considered, it is possible to make use of various coherence tests, thereby reducing the number of pixels to be handled explicitly.

Warnock's Algorithm

This is one of the class of "area" algorithms. It tries to solve the hidden surface problem recursively. The algorithm proceeds on the following lines.

i. Try to solve the problem by taking the entire screen as one window. If no polygons overlap either in x or y or even if they do, overlap so that they do not obscure, then return the screen.

ii. If the problem is not easily solvable in step (i) the algorithm divides the screen into 4 equal parts and tries to apply step (i) each of them. If it is not solvable, again divides into smaller windows and so on.

iii. The recursive process continues till each window is trivially solvable or one endsup with single pixels.

We have still not described how the actual "solution" is done. To do this, in any window, the algorithm classifies the polygons into three groups

i) Disjoint Polygons: Polygons that do not overlap in the window and hence can be trivially passed.

ii) A bigger and a smaller polygon overlapping so that the smaller one will be completely blocked by the bigger one (if the Z of the larger polygon is smaller than Z of the smaller one).

iii) Intersected polygons: Polygons that partly obscure each other.

Polygons that fall into category (i) and (ii) are removed at each level. If the remaining polygons can be easily solved, the recursive process stops at that level, else the process continues (with the polygons of category (i) and (ii) removed).

Since at each recursive level a few polygons are removed, as the windows become smaller and smaller with the advance of recursion, the list of polygons falling into them also reduces and hopefully the problem of hidden surfaces gets solved trivially.

One main draw back of algorithm is that the windows get divided into smaller and smaller rectangles. In many cases it would be efficient if one can divide the window roughly in the shape of the polygons themselves. Such an algorithm, developed by Wieler and Atherton, was found more efficient, though more complex in terms of larger complexities of recursive divisions and clippings.

Unit-II: CAD TOOLS

Syllabus

Definition of CAD Tools, Types of system, CAD/CAM system evaluation criteria, brief treatment of input and output devices. Graphics standard, functional areas of CAD, Modeling and viewing, software documentation, efficient use of CAD software; Geometric modeling: Types of mathematical representation of curves, wire frame models wire frame entities parametric representation of synthetic curves hermite cubic splines Bezier curves Bezier splines rational curves.

CAD/CAM Systems Evaluation Criteria

The various types of CAD/CAM systems are Mainframe-Based Systems, Minicomputer-Based Systems, Microcomputer-Based Systems and Workstation-Based Systems.

The implementation of these types by various vendors, software developers and hardware manufacturers result in a wide variety of systems, thus making the selection process of one rather difficult. CAD/CAM selection committees find themselves developing long lists of guidelines to screen available choices.

These lists typically begin with cost criteria and end with sample models or benchmarks chosen to test system performance and capabilities. In between comes other factors such as compatibility requirements with in-house existing computers, prospective departments that plan to use the systems and credibility of CAD/CAM systems' suppliers.

System Considerations

(i) Hardware

- Each workstation is connected to a central computer, called the server, which has enough large disk and memory to store users' files and applications programs as well as executing these programs.

(ii) Software

- Three major contributing factors are the type of operating system the software runs under, the type of user interface (syntax) and the quality of documentation.

(iii) Maintenance

- Repair of hardware components and software updates comprise the majority of typical maintenance contracts. The annual cost of these contracts is substantial (about 5 to 10 percent of the initial system cost) and should be considered in deciding on the cost of a system in addition to the initial capital investment.

(iv) Vendor Support and Service

- Vendor support typically includes training, field services and technical support. Most vendors provide training courses, sometimes on-site if necessary.

Geometric Modeling Capabilities

(i) Representation Techniques

- The geometric modeling module of a CAD/CAM system is its heart. The applications module of the system is directly related to and limited by the various representations it supports. Wireframes, surfaces and solids are the three types of modeling available.

(ii) Coordinate Systems and Inputs

- In order to provide the designer with the proper flexibility to generate geometric models, various types of coordinate systems and coordinate inputs ought to be provided. Coordinate inputs can take the form of cartesian (x, y, z), cylindrical (r, θ, z) and spherical (θ, ϕ, z).

(iii) Modeling Entities

- The fact that a system supports a representation scheme is not enough. It is important to know the specific entities provided by the scheme. The ease to generate, verify and edit these entities should be considered during evaluation.

(iv) Geometric Editing and Manipulation

- It is essential to ensure that these geometric functions exist for the three types of representations. Editing functions include intersection, trimming and projection and manipulations include translation, rotation, copy, mirror, offset, scaling and changing attributes.

(v) Graphics Standards Support

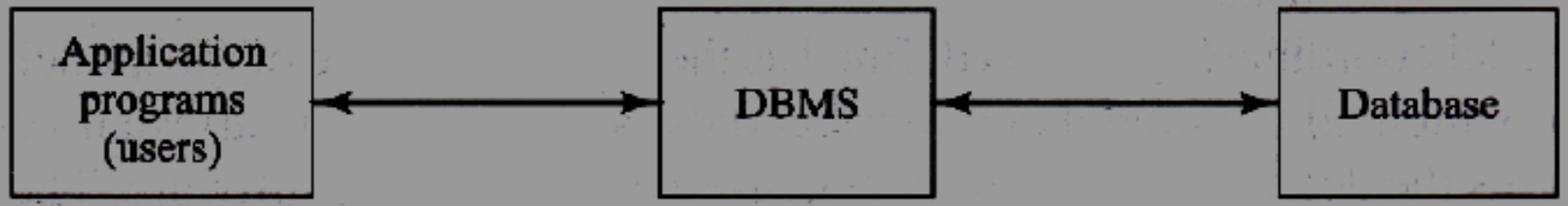
- If geometric models' databases are to be transferred from one system to another, both systems must support exchange standards.

CAD Softwares

- Softwares can be defined as an interpreter or translator which allows the user to perform specific type of application or job related to CAD.
- The user may utilize the software for drafting or designing of machine parts or components subjected to stresses or analysis of any type of system.
- The CAD application software can be prepared in variety of languages such as BASIC (Beginner's All-purpose Symbolic Instruction Code), FORTRAN, Java, PASCAL & C-Language.
- C- Language has been preferred for CAD software development because of no. of advantages as compared to other languages.
- The Java language has the capacity to operate with high level graphics and animation.

Database Management System (DBMS)

- A DBMS is defined as the software that allows access to use and/or modify data stored in a database. The DBMS forms a layer of software between the physical database itself (i.e., stored data) and the users of this database as shown in Fig. 1.31(a).
- DBMS shields users from having to deal with hardware-level details by interpreting their input commands and requests from the database. For example, a command such as retrieve a line could involve few lower-level steps to execute.



(*a*) Simplified DBMS

- In general, a DBMS is responsible for all database-related activities such as creating files, checking for illegal users of the database and synchronizing user access to the database.
- DBMSs designed for commercial business systems are too slow for CAD/ CAM. The handling of graphics data is an area where the conventional DBMSs tend to break down under the shear volume of data and the demand for quick display.
- By contrast, data handled in the commercial realm is mostly alphanumeric and the objects described are usually not very complex. A DBMS is directly related to the database model it is supposed to manage. For example, relational DBMSs require relatively large amounts of CPU time for searching and sorting data stored in the relations or tables.
- Therefore, the concept of database machines exists where a DBMS is implemented into hardware that can lie between the CPU of a computer and its database disks,

Graphics Exchange standards

- With the proliferation of computers and software in the market, it became necessary to standardize certain elements at each stage, so that investment made by companies in certain hardware or software was not totally lost and could be used without much modification on the newer and different systems.
- Standardization in engineering hardware is well known. Further, it is possible to obtain hardware and software from a number of vendors and then be integrated into a single system.
- This means that there should be compatibility between various software elements as also between the hardware and software. This is achieved by maintaining proper interface standards at various levels.

Both CAD/CAM vendors as well as users identified some needs to have some graphics standards. The needs are as follows:

i. **Software portability**: This avoids hardware dependence of the software. If the program is written originally for random scan display, when the display device is changed to raster scan display the program should work with minimum effort.
ii. **Image data portability**: Information and storage of images should be independent of different graphics devices.
iii. **Text data portability**: The text associated with graphics should be independent of different input/output devices.
iv. **Model database portability**: Transporting of design and manufacturing data from one application software to another should simple and economical.

The search for standards began in 1974 to fulfill the above needs both at the USA and International levels. As a result of worldwide efforts, various standards at different levels of the graphics systems were developed. The standards are as follows:

Both CAD/CAM vendors as well as users identified some needs to have some graphics standards. The needs are as follows:

i. **Software portability**: This avoids hardware dependence of the software. If the program is written originally for random scan display, when the display device is changed to raster scan display the program should work with minimum effort.
ii. **Image data portability**: Information and storage of images should be independent of different graphics devices.
iii. **Text data portability**: The text associated with graphics should be independent of different input/output devices.
iv. **Model database portability**: Transporting of design and manufacturing data from one application software to another should simple and economical.

The search for standards began in 1974 to fulfill the above needs both at the USA and International levels. As a result of worldwide efforts, various standards at different levels of the graphics systems were developed. The standards are as follows:

1. **GKS (Graphics kernel system):** It is an ANSI (American National Standards Institute and ISO (International Standards Organization) standard. It interfaces the application program with graphics support package.
2. **IGES (Initial graphics exchange specification):** It is an ANSI standard. It enables an exchange of model database among CAD/CAM software.
3. **PHIGS (Programmer's hierarchical interactive graphics system):** It supports workstations and their related CAD/CAM applications. It supports 3-dimensional modeling of geometry segmentation and dynamic display.
4. **CGM (Computer graphics metafile):** It defines functions needed to describe an image. Such description can be stored or transported from one graphics device to another.
5. **CGII (Computer graphics interface)**: It is designed to interface plotters to GKS or PHIGS. It is the lowest device independent interface in a graphics system.

6. **Drawing Exchange Format (DXF):** The DXF format has been developed and supported by Autodesk for use with the AutoCAD drawing files. It is not an industry standard developed by any standards organisation, but in view of the widespread use of AutoCAD made it a default standard for use of a variety of CAD/CAM vendors. A Drawing Interchange File is simply an ASCII text file with a file extension of .DXF and specially formatted text.

7. **Standard for the Exchange of Product Model Data (STEP)**, officially the ISO standard 10303, Product Data Representation and Exchange, is a series of International Standards with the goal of defining data across the full engineering and manufacturing life cycle. The ability to share data across applications, across vendor platforms and between contractors, suppliers and customers, is the main goal of this standard.

8. **Parasolid:** It is a portable "kernel" that can be used in multiple systems - both high-end and mid-range. By adopting Parasolid, start-up software companies have eliminated a major barrier to application development - a high initial investment. This enabled them to effectively market softwares with strong solid modeling functionality at lower-cost.

9. **PDES (Product Data Exchange Specification)** is an exchange for product data in support of industrial automation. "Product data" encompasses data relevant to the entire life cycle of a product such as design, manufacturing, quality assurance, testing and support. In order to support industrial automation, PDES files are fully interpretable by computer. For example, tolerance information would be carried in a form directly interpretable by a computer rather than a computerized text form which requires human intervention to interpret.

Curve Representation

Curve is defined as the locus of point moving with one degree freedom.

A curve can be represented by following two methods either by storing its analytical equation or by storing an array of co-ordinates of various points

Curves can be described mathematically by following methods:

(i) Non-parametric form
   a) Explicit form
   b) Implicit form

(ii) Parametric form

Non-parametric form:

- In this, the object is described by its co-ordinates with respect to current reference frame in use.

Explicit form: (Clearly expressed)

In this, the co-ordinates of y and z of a point on curve are expressed as two separate functions of x as independent variable.

$$P = [x \quad y \quad z]$$
$$= [x \quad f(x) \quad g(x)]$$

Parametric form:

In this, a parameter is introduced and the co-ordinates of x, y and z are expressed as functions of this parameters. This parameter acts as a local co-ordinate for points on curve.

$$P(u) = [x \quad y \quad z]$$

$$= [x(u) \quad y(u) \quad z(u)]$$

Example 2.1: For the position vectors $P_1[1 \quad 2]$ and $P_2[4 \quad 3]$, determine the parametric representation of line segment between them. Also determine the slope and tangent vector of line segment.

A parametric representation of line is

$$P = P_1 + u(P_2 - P_1)$$

$$= [1 \quad 2] + u([4 \quad 3] - [1 \quad 2])$$

$$= [1 \quad 2] + u[3 \quad 1]$$

Parametric representation of x and y components are

$$x(u) = x_1 + u(x_2 - x_1)$$

$$= 1 + 3u$$

$$y(u) = y_1 + u(y_2 - y_1)$$

$$= 2 + u$$

The tangent vector is obtained by differentiating P(u)

$$P'(u) = [x'(u) \quad y'(u)]$$

$$= [3 \quad 1]$$

The slope of line segment is

$$\frac{dy}{dx} = \frac{dy/du}{dx/du}$$

$$= \frac{y'}{x'}$$

$$= \frac{1}{3}$$

Example 2.2 : The end points of line are P1(1,3,7) and P2(–4,5, –3). Determine

i. Tangent vector of the line

ii. Length of line

iii. Unit vector in the direction of line

Parametric representation of x, y and z components are

$$x(u) = x_1 + u\,(x_2 - x_1)$$
$$= 1 - 5u$$
$$y(u) = y_1 + u\,(y_2 - y_1)$$
$$= 3 + 2u$$
$$z(u) = z_1 + u\,(z_2 - z_1)$$
$$= 7 - 10u$$

Tangent vector, $P'(u) = [x'(u)\ \ y'(u)\ \ z'(u)]$

$$= [-5\ \ 2\ \ -10]$$
$$= -5i + 2j - 10k$$

Length of line, $L = |P_2 - P_1|$

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

$$= \sqrt{(-4-1)^2 + (5-3)^2 + (-3-7)^2}$$

$$= 11.358$$

Unit vector in the direction of line, $\hat{n} = \frac{P_2 - P_1}{|P_2 - P_1|} = \frac{P_2 - P_1}{L}$

$$= \frac{1}{11.358}[-5 \quad 2 \quad -10]$$

$$= [-0.44 \quad 0.176 \quad -0.88]$$

$$= -0.44i + 0.176j - 0.88k$$

Example 2.3: A line is represented by the end point $P_1(2, 4, 6)$ and $P_2(-3, 6, 9)$. If the value of parameter u at P_1 and P_2 is 0 and 1 respectively, determine the tangent vector for the line. Also determine the coordinate of a point represented by; u equal to 0, 0.25, -0.25, 1 and 1.5. Also find the length and unit vector of line between two points P_1 and P_2.

Parametric representation of x, y and z components are

$$x(u) = x_1 + u\,(x_2 - x_1)$$
$$= 2 - 5u$$
$$y(u) = y_1 + u\,(y_2 - y_1)$$
$$= 4 + 2u$$
$$z(u) = z_1 + u\,(z_2 - z_1)$$
$$= 6 - 3u$$

$$\text{Tangent vector, } P'(u) = [x'(u)\ \ y'(u)\ \ z'(u)]$$
$$= [-5\ \ 2\ \ -3]$$
$$= -5i + 2j - 3k$$

u	0	0.25	– 0.25	1	1.5
x (u)	2	0.75	3.25	–3	–5.5
y (u)	4	4.5	3.5	6	7
z (u)	6	5.25	6.75	3	1.5

Length of line, $L = |P_2 - P_1|$

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

$$= \sqrt{(-3-2)^2 + (6-4)^2 + (9-6)^2}$$

$$= 6.16$$

Unit vector in the direction of line, $\hat{n} = \frac{P_2 - P_1}{|P_2 - P_1|} = \frac{P_2 - P_1}{L}$

$$= \frac{1}{6.16}[-5 \quad 2 \quad -3]$$

$$= [-0.81 \quad 0.324 \quad -0.487]$$

$$= -0.81\mathrm{i} + 0.324\mathrm{j} - 0.487\mathrm{k}$$

Example 2.4: The two endpoints of diameter of a circle are $P_1(13,15,7)$ and $P_2(35,40,7)$. Determine the centre and radius of circle.

The centre of a circle, $P_c = \frac{1}{2}(P_1 + P_2)$

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

$$[x_c \;\; y_c \;\; z_c] = \left[\frac{13+35}{2} \;\; \frac{15+40}{2} \;\; \frac{7+7}{2}\right]$$

$$[x_c \;\; y_c \;\; z_c] = [24 \;\; 27.5 \;\; 7]$$

The radius of circle

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

$$R = \frac{1}{2}\left(\sqrt{(35-13)^2 + (40-15)^2 + (7-7)^2}\right)$$

$$R = 16.65$$

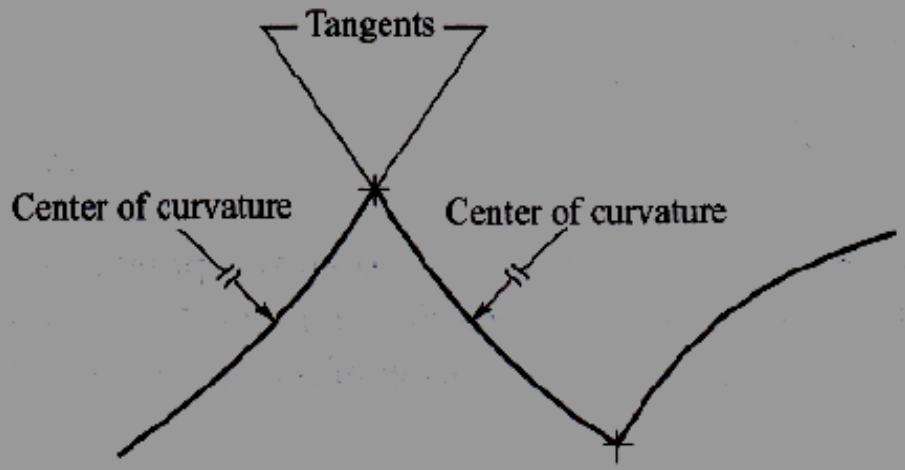
Parametric representation of synthetic curve

Analytic curves, are usually not sufficient to meet geometric design requirements of mechanical parts. Products such as car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles and bottles are a few examples that require free-form, or synthetic, curves and surfaces.

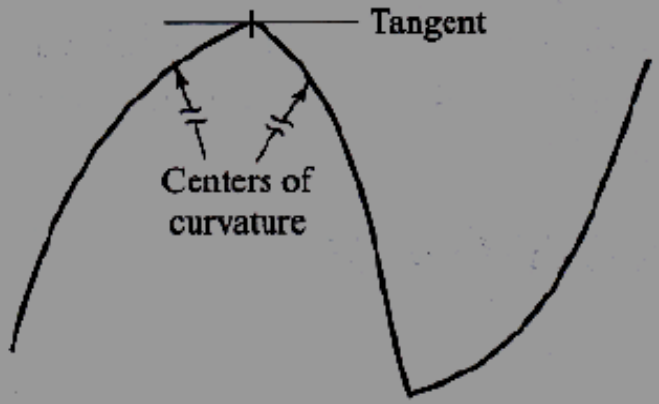
The need for synthetic curves in design arises on two occasions: when a curve is represented by a collection of measured data points and when an existing curve must change to meet new design requirements.

In the latter occasion, the designer would need a curve representation that is directly related to the data points and is flexible enough to bend, twist, or change the curve shape by changing one or more data points.

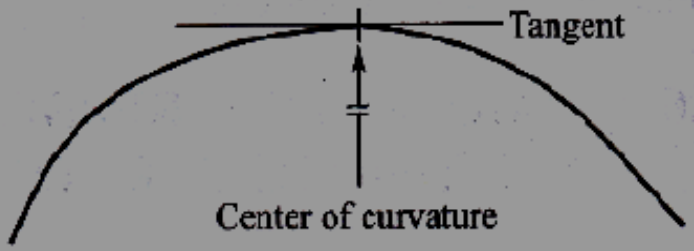
Data points are usually called control points and the curve itself is called an interpolant if it passes through all the data points.



(*a*) Zero-order continuity (C^0 curve)



(*b*) First-order continuity (C^1 curve)



(*c*) Second-order continuity (C^2 curve)

Fig. 2.11 Various Orders of Continuity of Curves

Unit- III: SURFACE MODELING

Syllabus

Mathematical representation surfaces, surface model, surface entities surface representation.

Parametric representation of surfaces, plane surface, rule surface, surface of revolution, tabulated cylinder.

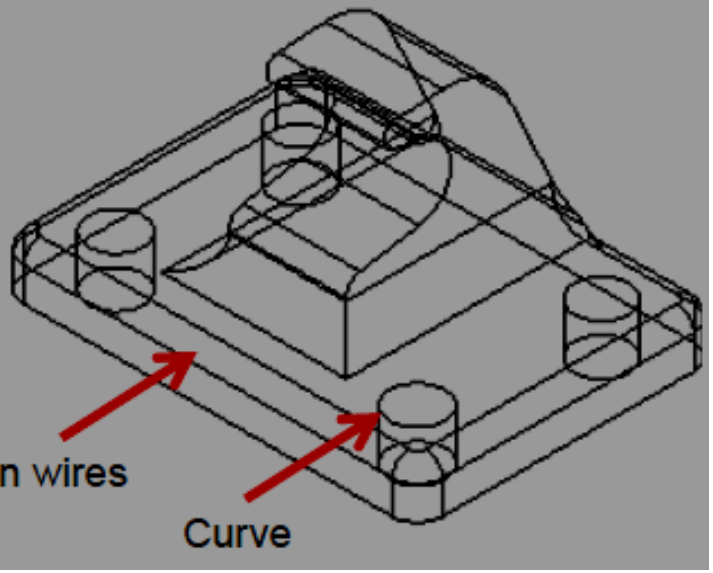
Basic modeling techniques are

1. **Wire frame modeling**
2. **Surface modeling**
3. **Solid modeling**

Wireframe Modeling

A wireframe representation is a 3-D line drawing of an object showing only the edges without any side surface in between.

A frame constructed from thin wires representing the edges and projected lines and curves.



Wireframe modeling is the process of visual presentation of a three-dimensional or physical object used in 3-D computer graphics. It is an abstract edge or skeletal representation of a real-world 3-D object using lines and curves.

Merits:

Simple to construct for 2D and simple and symmetric 3D objects
Designer needs little training
System needs little memory
Take less manipulation time
Retrieving and editing can be done easy
Consumes less time
Best suitable for manipulations as orthographic isometric and perspective views.

Surface modelling

Surface models define the surface features, as well as the edges, of objects. Different types of spline curves are used to create surface patches with different modelling characteristics.

Merits:

It is less ambiguous.

Complex surfaces can be easily identified.

It removes hidden line and adds realism.

Demerits:

Difficult to construct.

Difficult to calculate mass property.

More time is required for creation.

Requires high storage space as compared to wire frame modelling.

Also requires more time for manipulation.

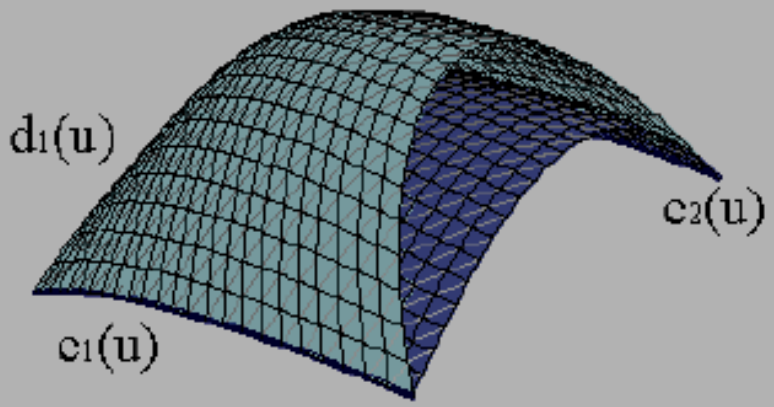
Surface modeling was essentially the situation in the early 1940s.

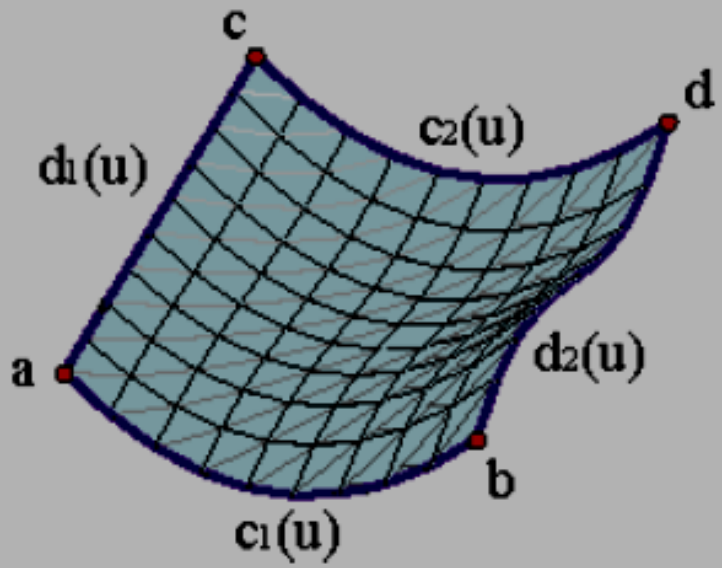
The pressures of wartime production, particularly in the aircraft industry, led to changes in the way the geometry was represented.

N. Lidbro [1956] describes a system used by Saab-Scania in Sweden in the 1950s.

Surface Modeling

The use of parametric techniques became popular in the 1960s, largely due to the pioneering work of Coons [1964].





Surface Modeling

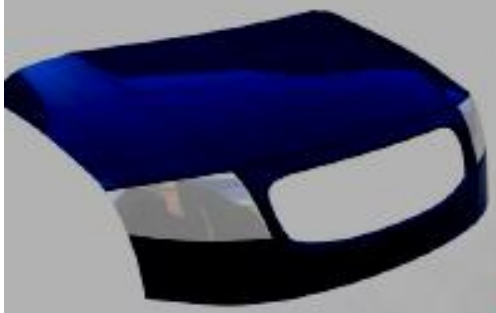
Widespread in

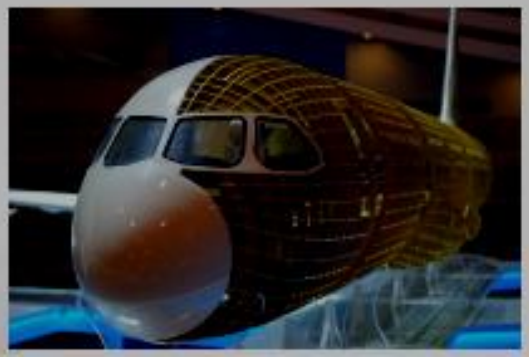
- shipbuilding
- automotive manufacture
- shoe industry
- companies manufacturing
  - forgings,
  - castings.

Surface Modeling

It is particularly useful for modeling objects, which can be modeled as

- shells,
- car body panels,
- aircraft fuselages
- fan blades.
- wind turbine

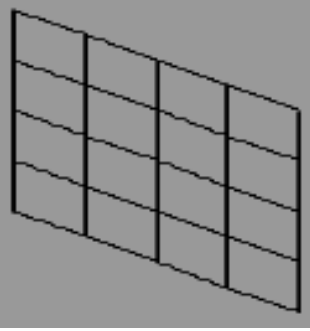


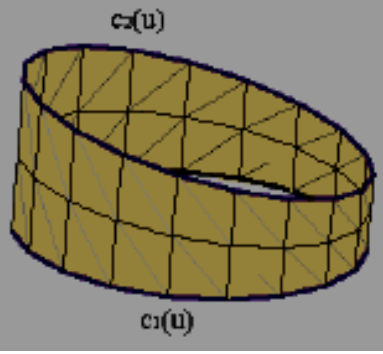


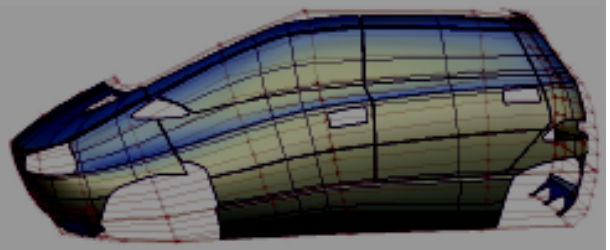
Surface: definition

- An area bounded by an identifiable perimeter.
- In Computer Graphics, is an area within which every position is defined by mathematical methods.

Planar surface | **Cylindrical/conic** | **Sculptured**







Surface modeling

As a surface model defines adequate data on a component's surface geometry hidden lines and surfaces are readily and automatically removed as required.

This gives rise to non-ambiguous visualization of the object when viewed from any direction.

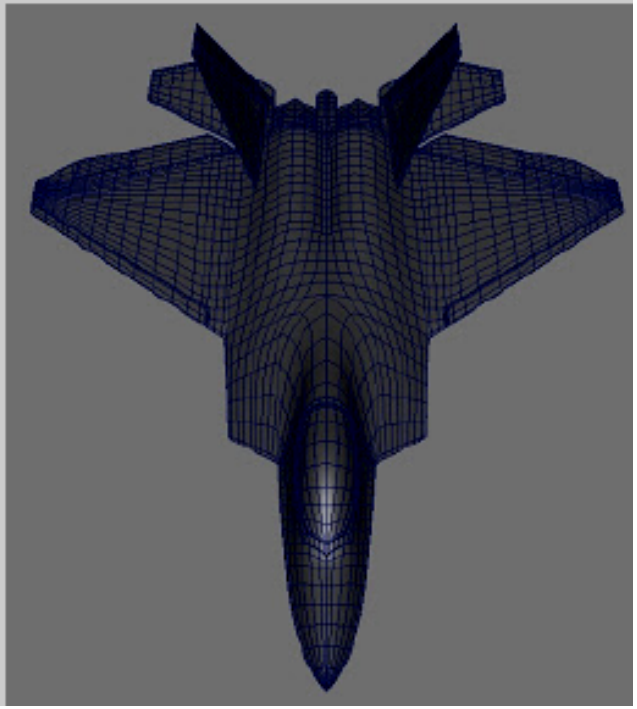
The software calculates the amount of light reflected back to the user from different areas on the surface and each area is color filled with varying shades accordingly.

Surface Modeling

Complex objects such as car or airplane body can not be achieved utilizing wireframe modeling.

Surface modeling are used in

- ✓calculating mass porperties
- ✓checking for interference
- ✓between mating parts
- ✓generating cross-section views
- ✓generating finite elements meshes
- ✓generating NC tool paths for
- ✓continuous path machining

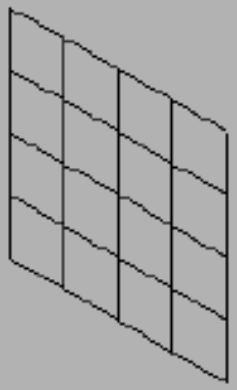


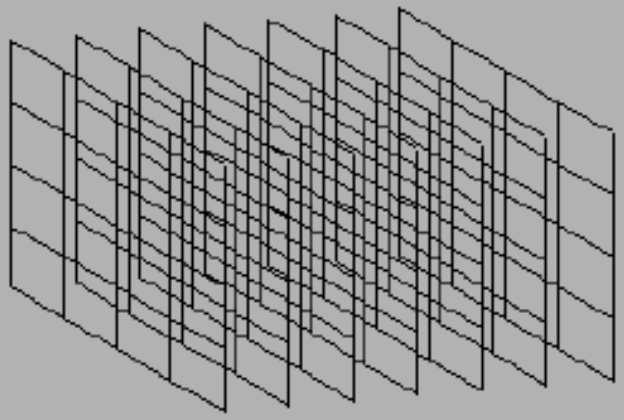
Advantages of Solid & Surface Modelling

Solid Modelling	Surface Modelling
Easy to learn/use	More flexible in modelling complex geometry
Parametric/associative capabilities	Interactive modelling capabilities
Quicker creation and updating of assemblies	Quicker creation and updating of complex components and tooling
Excellent for creating functional models	Excellent for creating aesthetic or ergonomic free-form models

Plane surface

This is the simplest surface. It requires three non-coincident points to define an infinite plane. The plane surface can be used to generate cross-sectional views by intersecting a surface model with it, generate cross sections for mass property calculations, or other similar applications where a plane is needed.





Plane surface

A plane surface that passes through three points, P_0, P_1 and P_2 is given by

$$\mathbf{P}(u,v) = \mathbf{P}_0 + u(\mathbf{P}_1 - \mathbf{P}_0) + v(\mathbf{P}_2 - \mathbf{P}_0), \quad 0 \le u \le 1, 0 \le v \le 1$$

The surface normal vector then is

$$\mathbf{n}(u,v) = \frac{(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_2 - \mathbf{P}_0)}{\left|(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_2 - \mathbf{P}_0)\right|}, \quad 0 \le u \le 1, 0 \le v \le 1$$

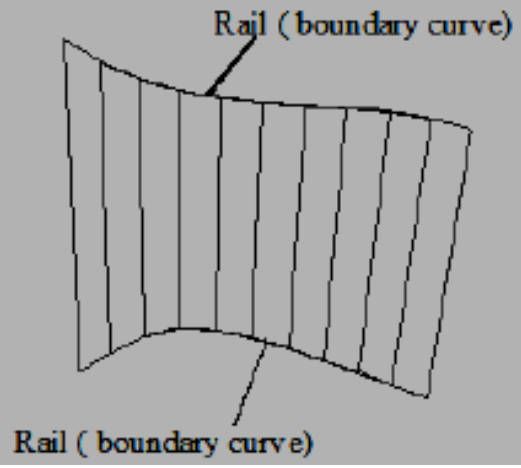
Once the normal unit vector is known, the surface can be also expressed in nonparametric form as

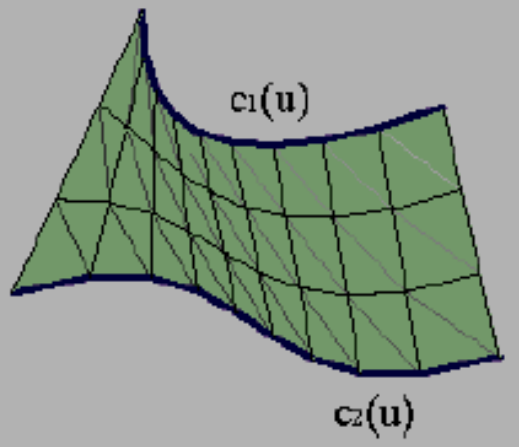
$$(\mathbf{P} - \mathbf{P}_0) \cdot \mathbf{n} = 0$$

Ruled Surface (lofted surface)

A ruled surface is generated by joining two space curves (rails) with a straight line (ruling or generator). If two curves are denoted by F(u) and G(u) respectively, for a value of u, then the parametric equation is given by

$$\mathbf{P}(u,v) = (1-v)\mathbf{G}(u) + v\mathbf{F}(u), \quad 0 \le u \le 1, \; 0 \le v \le 1$$

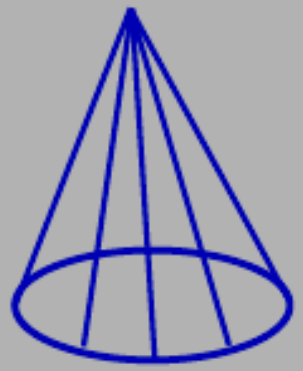




Ruled surface

- Linear interpolation between two bounding geometric elements (curves).
- Elemental division the same for each curve.
- Bounding curves must both be either geometrically open (line, arc) or closed (circle, ellipse).
- Curvature in one direction only.

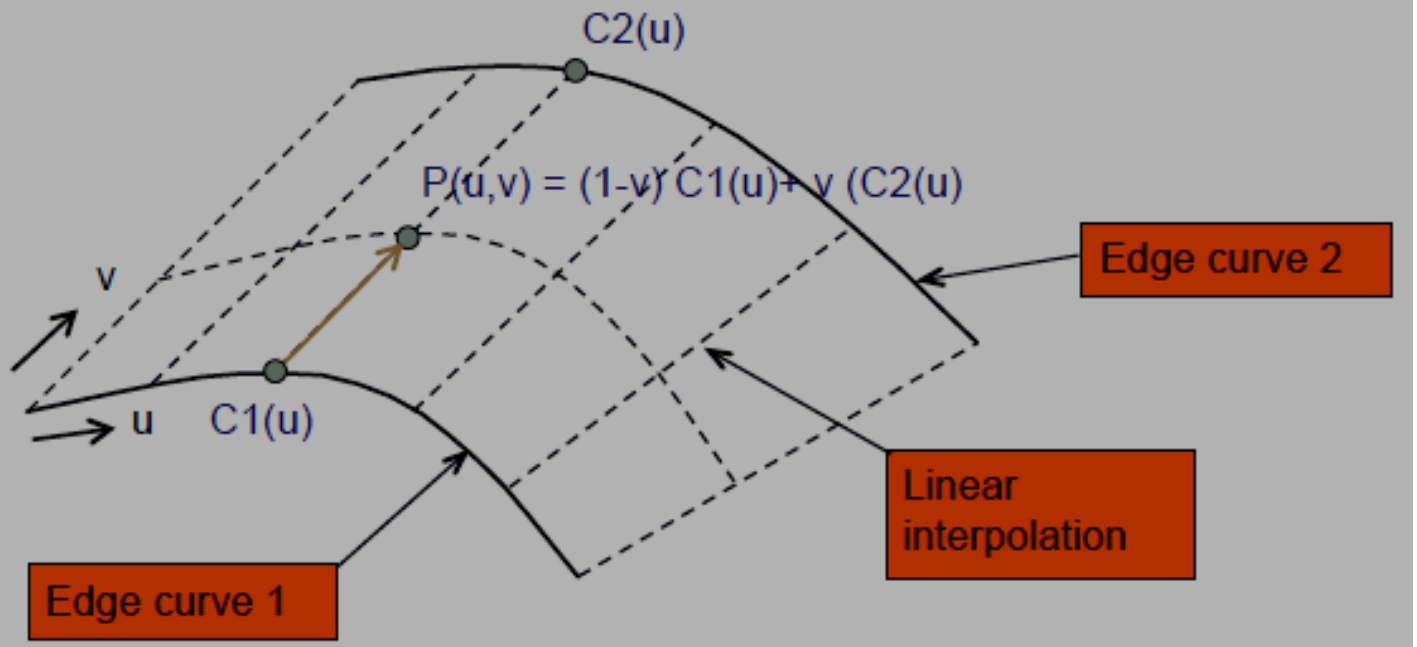
Both geometries open
(line & arc)



Both geometries closed
(circle & point *)

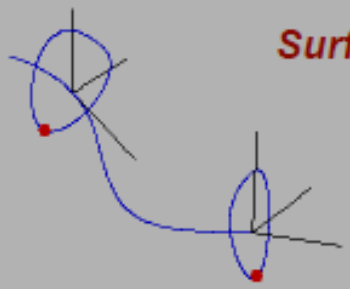
Ruled Surface

- Linear interpolation between two edge curves
- Created by lofting through cross sections

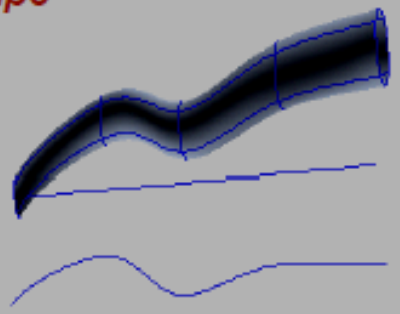


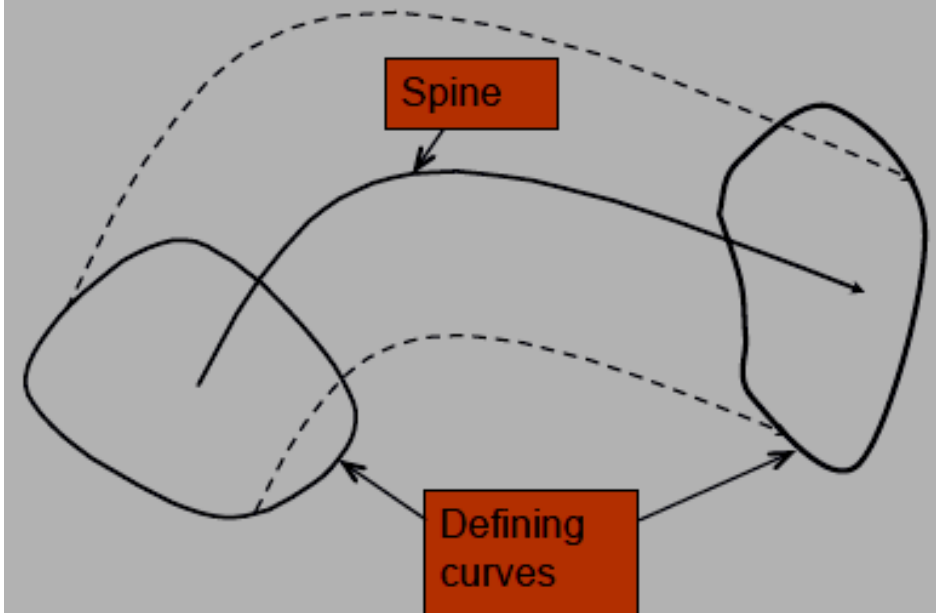
Lofted Surface

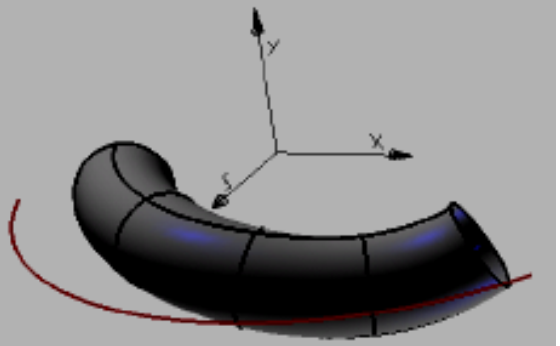
Defining curve swept along an arbitrary spine curve



Surface Pipe





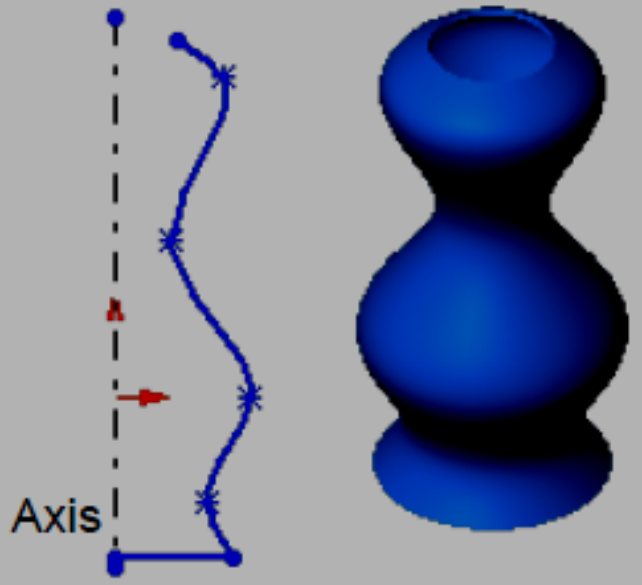


Revolved surface

A revolved surface is generated a space curve about an axis of rotation. The parametric representation of the curve in the working coordinate system that has x- and y-axes on the perpendicular plane to the axis of rotation that is z-axis can be expressed by

$$\mathbf{P}(u,v) = r_x(u)\cos v\mathbf{n}_x + r_z(u)\sin v\mathbf{n}_y + z(u)\mathbf{n}_z,\ 0 \le u \le 1,\ 0 \le v \le 2\pi$$

This is an axisymmetric surface that can model axisymmetric objects.
It is generated by rotating a planar wireframe entity in space about the axis of symmetry a certain angle.



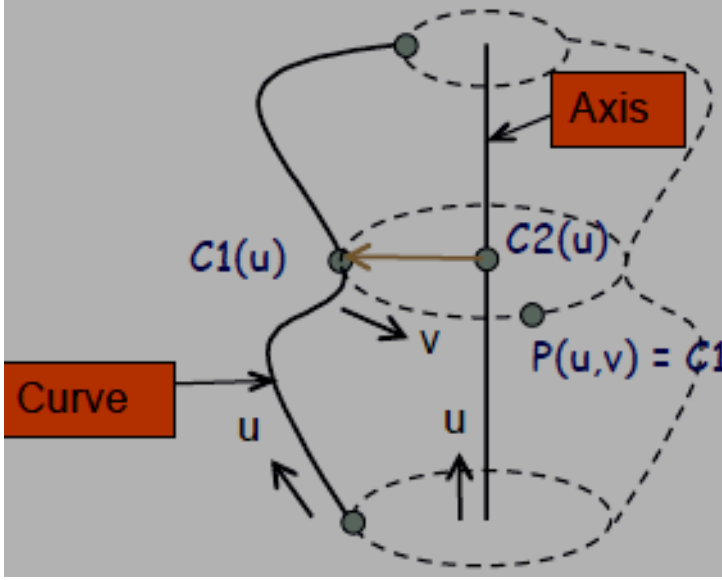
Revolution of Surface

Surface of revolution requires:

– a shape curve (must be continuous)

– a specified angle

– an axis defined in 3D modelspace.



Positive rotation direction usually based upon direction of axis vector..

Tabulated cylinder

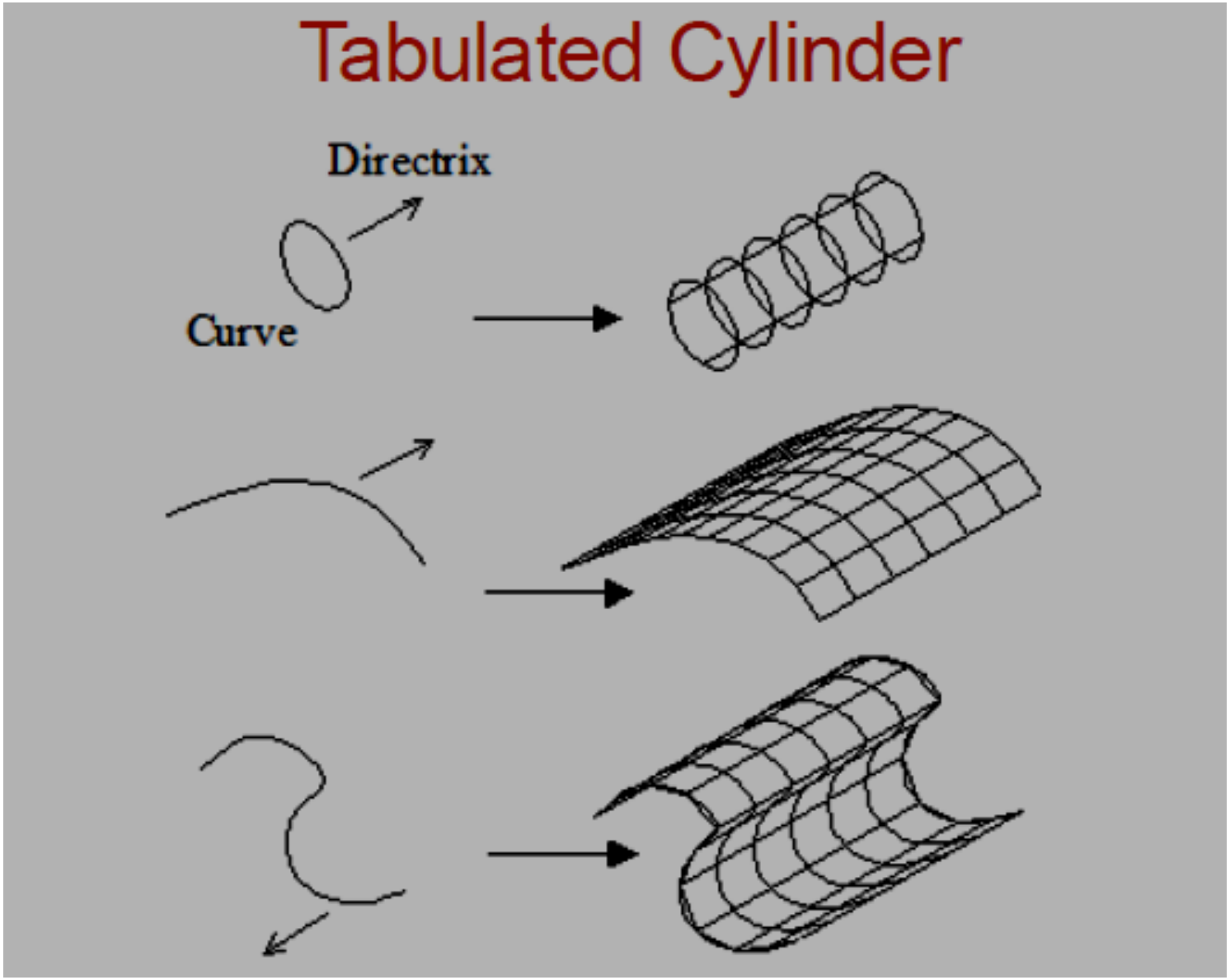
A tabulated cylinder is generated by moving a straight line along a space curve. The parametric representation of the curve can be expressed by

$$\mathbf{P}(u,v) = \mathbf{F}(u) + v\mathbf{n}_v, \quad 0 \le u \le u_{max}, \quad 0 \le v \le v_{max}$$

where n_v is the unit vector in the direction of generatix.

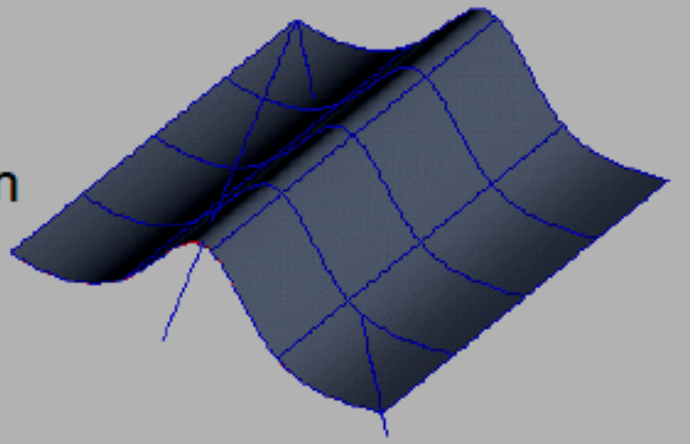
- Defined by projecting a shape curve (or profile) along a direction vector.
- Curvature in one direction only (along shape curve), linear in other direction.

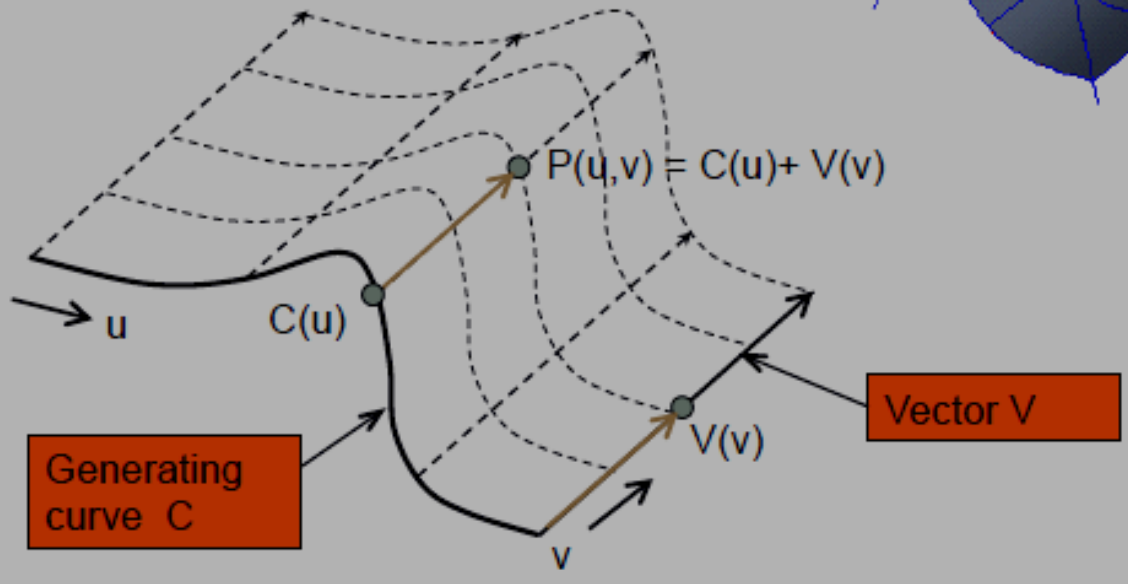
Tabulated Cylinder



Tabulated Cylinder

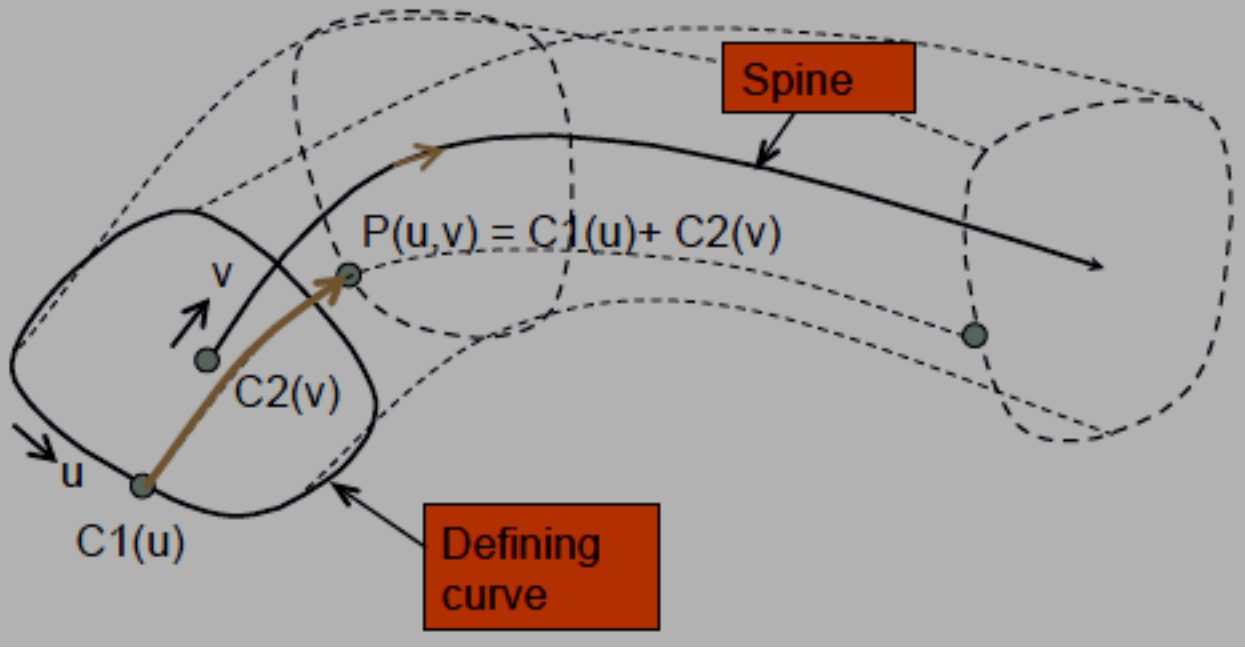
- Project curve along a vector
- In SolidWorks, created by extrusion





Swept Surface

- Defining curve swept along an arbitrary spine curve



Unit- IV :PARAMETRIC REPRESENTATION OF SYNTHETIC SURFACES

Syllabus

Parametric representation of synthetic surfaces: Hermite Bicubic surface, Bezier surface, Bezier Spline surface, COONs surface, Blending surface Sculptured surface, Surface manipulation; Displaying, Segmentation, Trimming, Intersection, Transformations (both 2D and 3D).

$$\mathbf{P}(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3}C_{ij}u^i v^j,\quad 0\le u,v\le 1$$

In a matrix form it can be expressed.

$$\mathbf{P}(u,v)=\mathbf{U}^T[\mathbf{C}]\mathbf{V}\equiv\mathbf{U}^T[\mathbf{M}_H][\mathbf{B}][\mathbf{M}_H]^T\mathbf{V}$$

where

$$\mathbf{U}^T=\{u^3,u^2,u,1\},\ \mathbf{V}=\{v^3,v^2,v,1\}^T$$

[B] has 4 x 4 = 16

Hermite Bicubic Surface

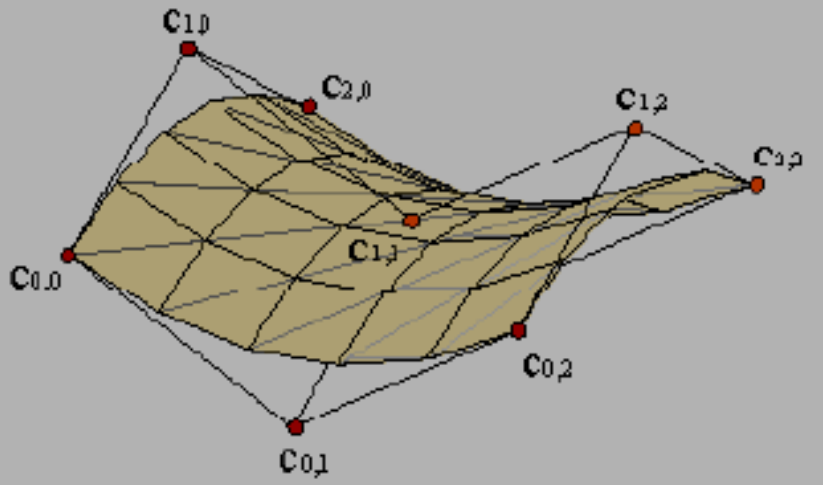
Applying the boundary conditions (continuity and tangency) at data points determines all coefficients. Here

$$[\mathbf{B}] = \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \left.\frac{\partial \mathbf{P}}{\partial v}\right|_{00} & \left.\frac{\partial \mathbf{P}}{\partial v}\right|_{01} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \left.\frac{\partial \mathbf{P}}{\partial v}\right|_{10} & \left.\frac{\partial \mathbf{P}}{\partial v}\right|_{11} \\ \left.\frac{\partial \mathbf{P}}{\partial u}\right|_{00} & \left.\frac{\partial \mathbf{P}}{\partial u}\right|_{01} & \left.\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\right|_{00} & \left.\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\right|_{01} \\ \left.\frac{\partial \mathbf{P}}{\partial u}\right|_{10} & \left.\frac{\partial \mathbf{P}}{\partial u}\right|_{11} & \left.\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\right|_{10} & \left.\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\right|_{11} \end{bmatrix}$$

This matrix can be determined by imposing the smoothness conditions at data points joining two adjacent panels.

Bezier Surface

This is a surface that approximates given input data.



It is different from the previous surfaces in that it is a synthetic surface, it does not pass through all given data points.

Bezier surface is an extension of Bezier curve and interpolates to a finite number of data point. It can be expressed as

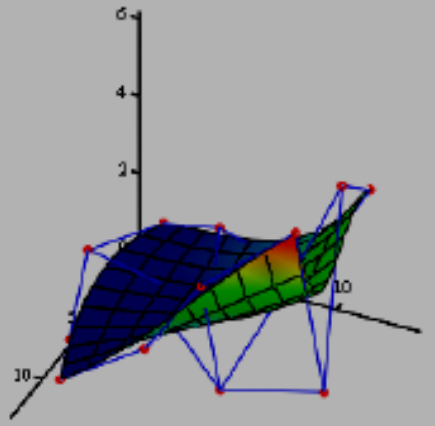
$$\mathbf{P}(u,v)=\sum_{j=0}^{n}\sum_{i=0}^{m}\mathbf{P}_{i,j}B_{i,m}(u)B_{j,n}(v),\ \ 0\le u,v\le 1$$

Bezier Surfaces

Bezier curves can be extended to surfaces

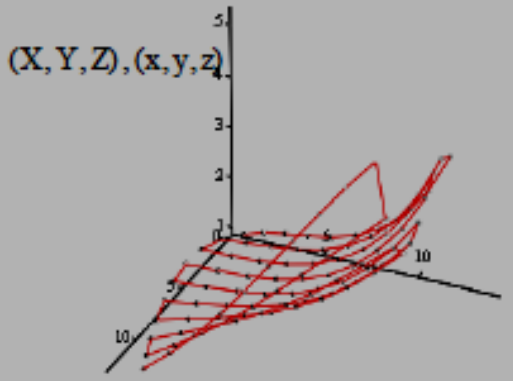
Same problems as for Bezier curves:

- no local modification possible
- smooth transition between adjacent patches difficult to achieve



$$C_{n,i} := \frac{n!}{i! \cdot n - i!} \qquad B(u,i) := C_{n,i} \cdot u^{i} \cdot (1-u)^{n-i}$$

$$P(u,v,r) := \sum_{i=0}^{m} \sum_{j=0}^{n} \left(p_{i,j}\right)_{r} \cdot B(u,i) \cdot B(v,j)$$



Isoparametric curves used for tool path generation.

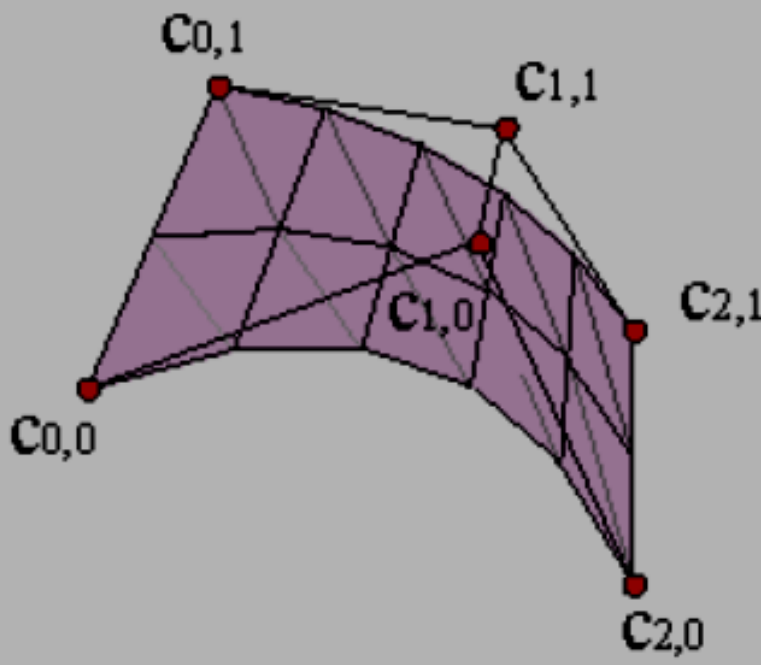
(x,y,z)

B-Spline Surface

This is a surface that can approximate or interpolate given input data.

It is a synthetic surface.

It is a general surface like the Bezier surface but with the advantage of permitting local control of the surface.

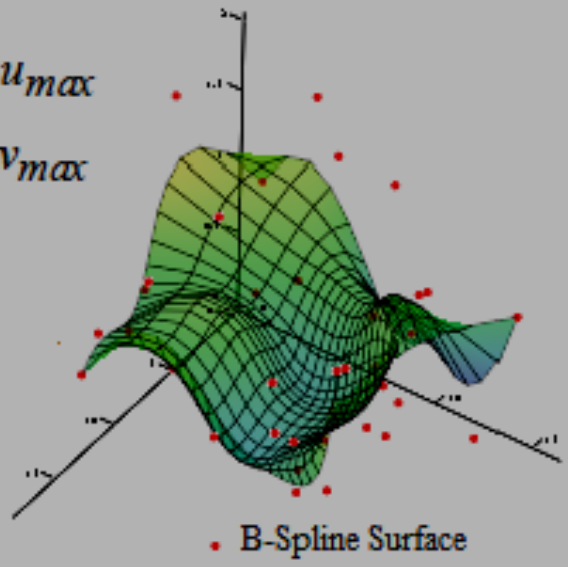


B-Spline Surfaces

- As with curves, B-spline surfaces are a generalization of Bezier surfaces
- The surface approximates a control polygon
- Open and closed surfaces can be represented

$$P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{ij} N_{i,k}(u) \cdot N_{j,l}(v) \qquad \begin{matrix} 0 \le u \le u_{max} \\ 0 \le v \le v_{max} \end{matrix}$$

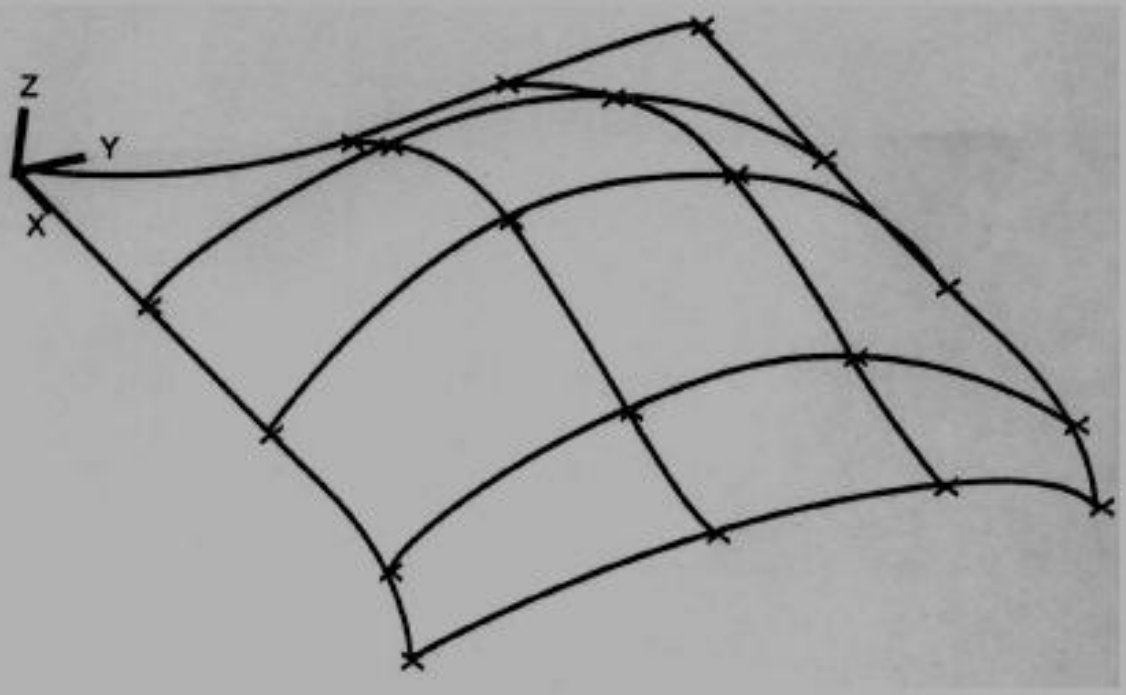
$$N_{i,k}(u) = (u - u_i) \cdot \frac{N_{i,k-1}}{u_{i+k-1} - u_i} + (u_{i+k} - u) \cdot \frac{N_{i+1,k-1}}{u_{i+k} - u_{i+1}}$$



B-Spline Surface

B-spline surfaces

- Rational parametric surfaces
- Sphere, cylinders, cones

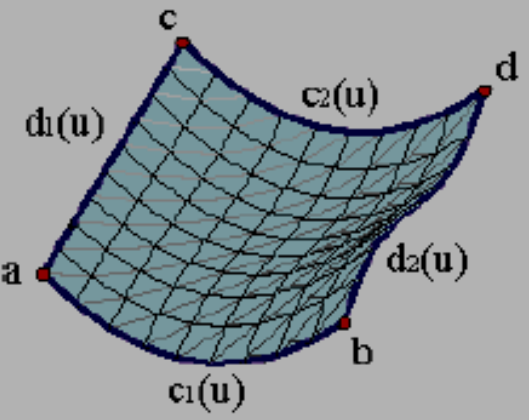


Interpolated B-spline surface patch

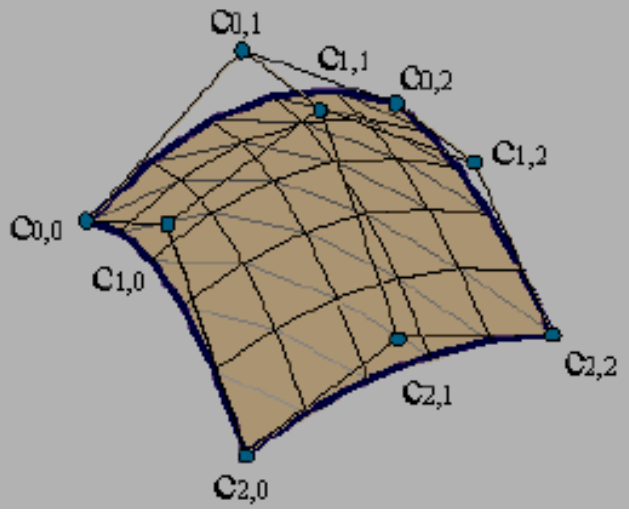
Coons Surface

The above surfaces are used with either open boundaries or given data points.

The **Coons patch** is used to create a surface using curves that form closed boundaries .

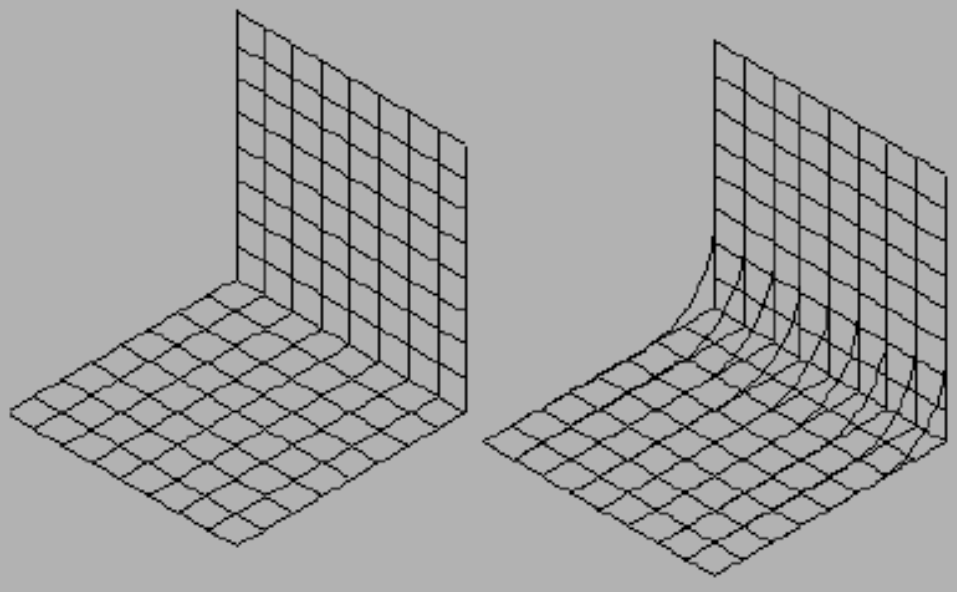


Steven Anson Coons was an early pioneer in the field of computer graphical methods. He was a professor at the MIT. He had a vision of interactive computer graphics as a design tool to aid the engineer.



Fillet Surface

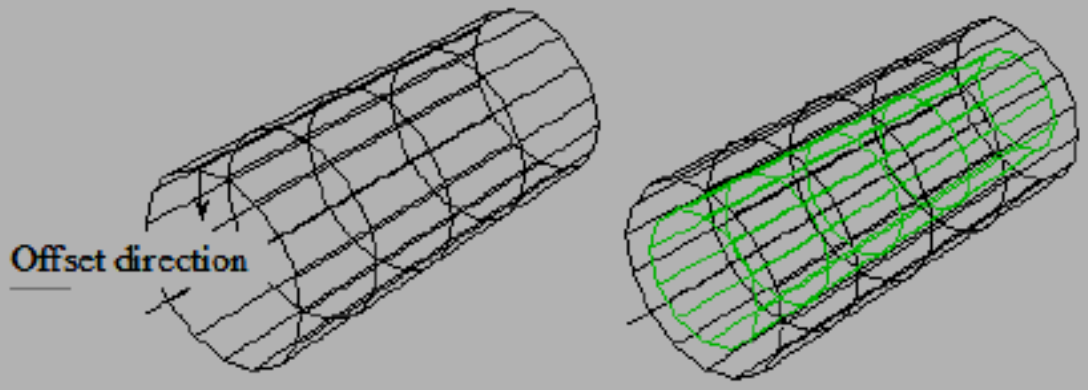
This is a B-spline surface that blends two surfaces together.



Offset Surface

Existing surfaces can be offset to create new ones identical in shape but may have different dimensions.

To create a hollow cylinder, the outer or inner cylinder can be created using a cylinder command and the other one can be created by an offset command.



Parametric Bi-cubic Surfaces

We have $Q(t) = T \cdot M \cdot G$ from parametric curves.

Replace t with s (need a parameter in the other direction).

$$Q(s) = S \cdot M \cdot G$$

The parametric bi-cubic surface is given by

$$Q(s,t) = S \cdot M \cdot G \cdot M^T \cdot T^T, \quad 0 \le s,t \le 1$$

where

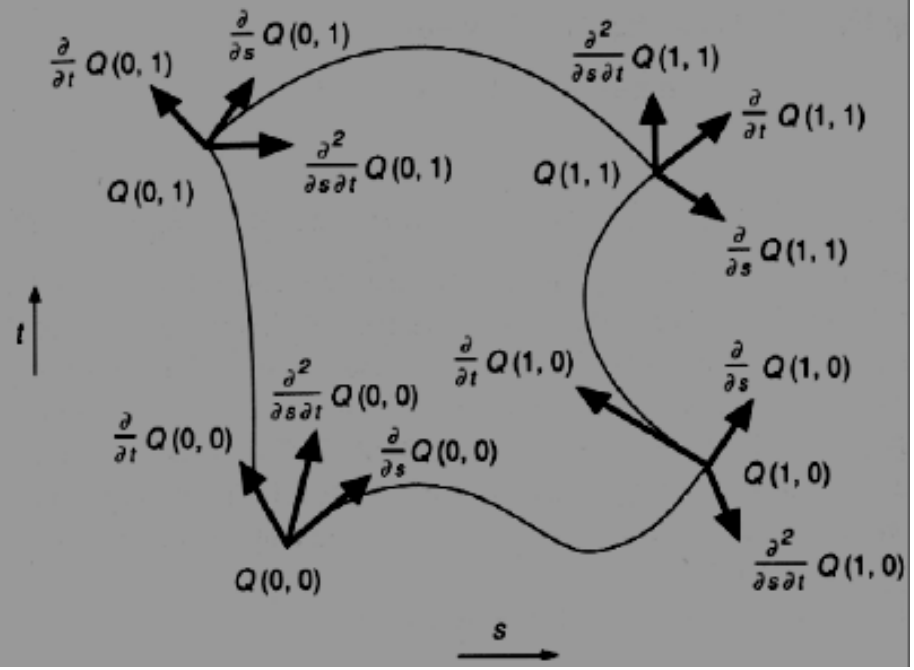
$$G = \begin{bmatrix} g_{11} & g_{21} & g_{31} & g_{41} \\ g_{12} & g_{22} & g_{32} & g_{42} \\ g_{13} & g_{23} & g_{33} & g_{43} \\ g_{14} & g_{24} & g_{34} & g_{44} \end{bmatrix}$$

Hermite Surfaces

Hermite permits C^1 and G^1 continuity from one curve segment to the next.

Conditions for C^1 continuity are that the control points along the edge, the tangent vectors and the twist vectors to be equal.

For G^1 the vectors must have the same direction (not the same magnitude).



Bezier Surfaces

A Bézier surface can be displayed using the following equation:

$$x(s,t) = \begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{bmatrix} \cdot G_{B_x} \cdot \begin{bmatrix} (1-s)^3 \\ 3s(1-s)^2 \\ 3s^2(1-s) \\ s^3 \end{bmatrix}$$

B-Spline Surfaces

B-spline surfaces are represented as:

$$x(s,t) = S \cdot M_{Bs} \cdot G_{Bs_x} \cdot M_{Bs}^T \cdot T^T$$

$$y(s,t) = S \cdot M_{Bs} \cdot G_{Bs_y} \cdot M_{Bs}^T \cdot T^T$$

$$z(s,t) = S \cdot M_{Bs} \cdot G_{Bs_z} \cdot M_{Bs}^T \cdot T^T$$

B-spline surfaces come naturally with C^2 continuity.

All techniques for subdivision and display carry over to the bi-cubic case.

Quadric Surfaces

A frequently used class of objects, which are described with second-degree equations.

The implicit surface equation

$$f(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0$$

defines a family of quadric surfaces. An alternative to rational surfaces if only quadric surfaces are being represented.

Useful in specialized applications, such as molecular modeling.

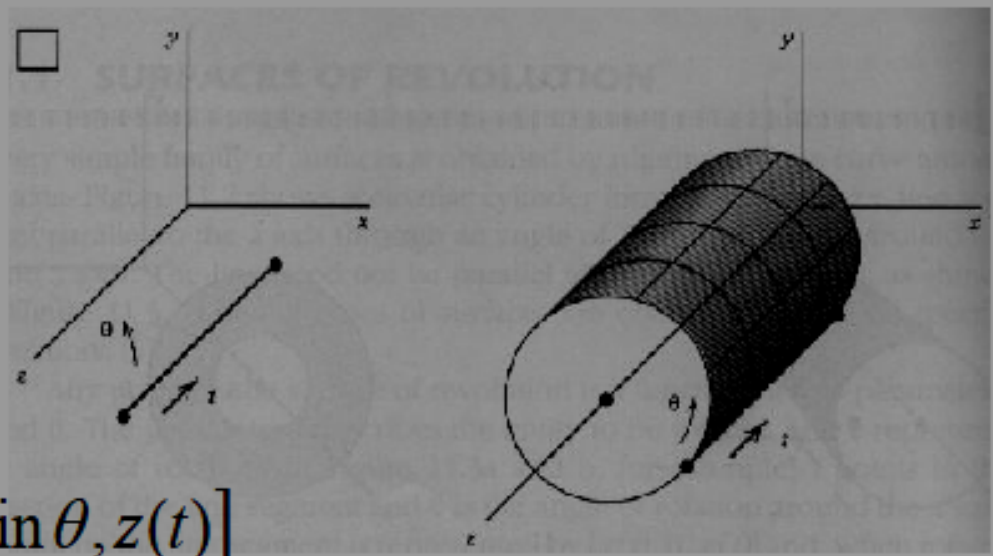
Other objects are spheres, ellipsoids, tori, parapoloids and hyperbolas.

Surface of Revolution

A very simple way of defining objects is obtained by rotating a curve or a line around an axis.

A circular cylinder is formed by rotating a line segment parallel to the z-axis through an angle of 360° around the same axis.

Any point on the surface of revolution is a function of t and θ.



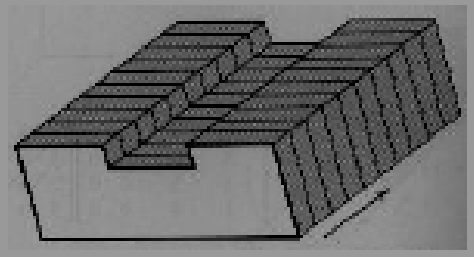
$$P(t,\theta) = [x(t)\cos\theta, x(t)\sin\theta, z(t)]$$

Sweep Representation

A technique to produce 3D objects.

Specify a 2D shape and sweep the shape through a region.

The sweep transformation can contain translation, scaling or rotation.



Translational Sweep

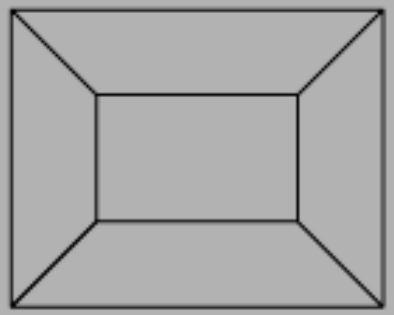
Unit – V : GEOMETRICMODELLING-3D

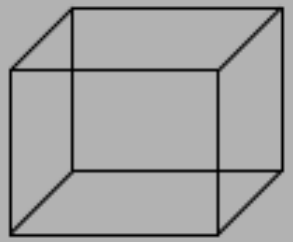
Syllabus

Geometricmodelling-3D: Solid modeling, solid representation, boundary representation (B-rep), Constructive solid geometry (CSG). CAD/CAM exchange: Evaluation of data, exchange format, IGES data representations and structure, STEP Architecture, implementation, ACIS and DXF; Design applications: Mechanical tolerances, mass property calculations, finite element modeling and analysis and mechanical assembly; Collaborative engineering: Collaborative design, principles, approaches, tools, design systems.

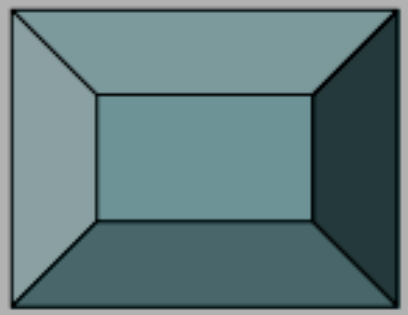
Evolution of Geometric Modeling

A wireframe representaion of an object is done using edges (lines curves) and vertices. Surface representation then is the logical evolution using faces (surfaces), edges and vertices. In this sequence of developments, the solid modeling uses topological information in addition to the geometrical information to represent the object unambiguously and completely.





Wireframe Model



Solid Model

Solid Modelling

Solid Modelling is a modelling that provides a complete representation of an object than a wire frame modelling and surface modelling. In this model, the appearance of an object is displayed in solid design.

Merits:

Complete modelling.
Unambiguous.
Best suitable for calculating mass properties.
Very much suitable for automated applications.
Fast creation.
Gives huge information.

Demerits:

Requires large memory.
Slow manipulation.
Some manipulations can be complex and require tedious procedure.

Advantages of Solid Models

Unlike wireframes and surface representations which contain only geometrical data, the solid model uses topological information in addition to the geometrical information to represent the object unambiguously and completely. Solid model results in accurate design, helps to further the goal of CAD/ CAM like CIM, Flexible manufacturing leading to better automation of the manufacturing process.

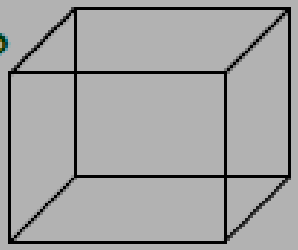
Geometry: The graphical information of dimension, length, angle, area and transformations

Topology: The invisible information about the connectivity, neighborhood, associatively etc

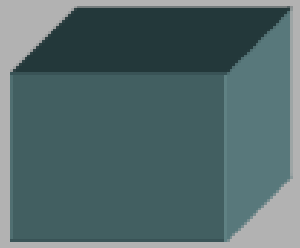
Is a solid model just a shaded image?

Three dimensional addressability?

Suitable for automation?



Wireframe Model



Solid Model

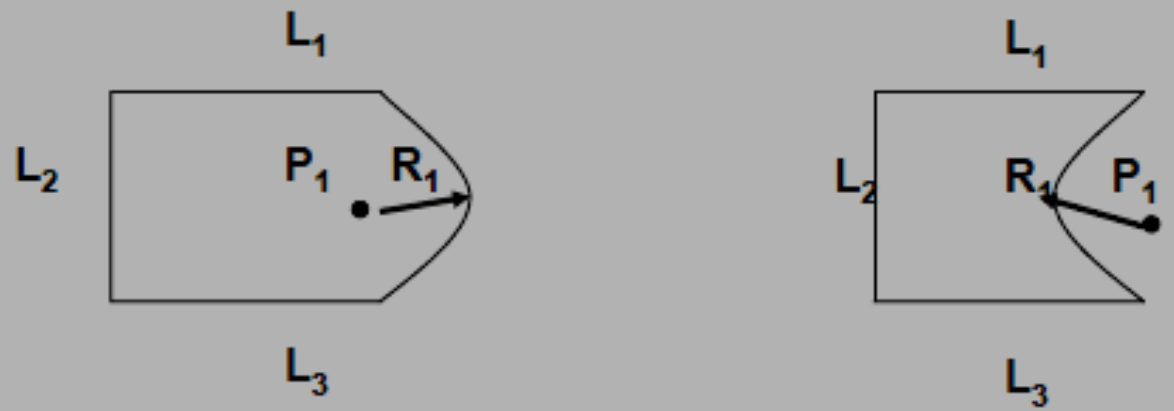
Geometry Vs Topology

Geometry:

Metrics and dimensions of the solid object. Location of the object in a chosen coordinate system

Topology:

Combinatorial information like connectivity, associativity and neighborhood information. Invisible relationship information.



Same geometry and different Topology

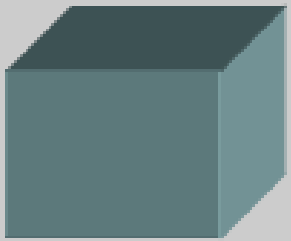
Manifold Vs Non-manifold

Two Manifold Representations:

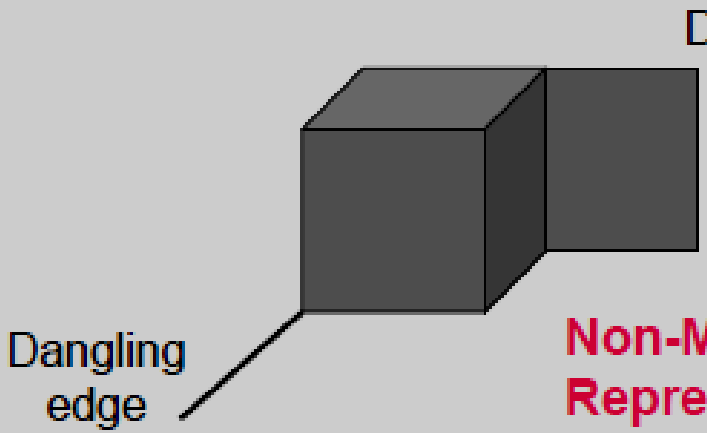
Two manifold objects are well bound, closed and homomorphic to a topological ball.

Non-manifold Representations:

When restrictions of closure and completeness are removed from the solid definition, wireframe entities can coexist with volume based solids.



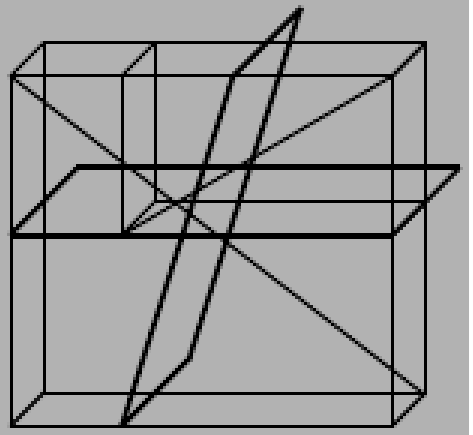
Two-Manifold Object



Non-Manifold Representation

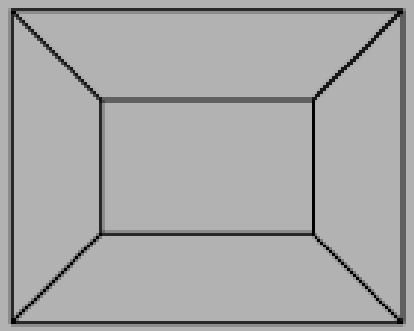
Disadvantages of Wireframe Models

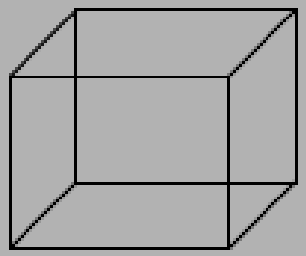
- Ambiguity
- Subjective human interpretation
- Complex objects with many edges become confusing
- Lengthy and verbose to define
- Not possible to calculate Volume and Mass properties, NC tool path, cross sectioning etc



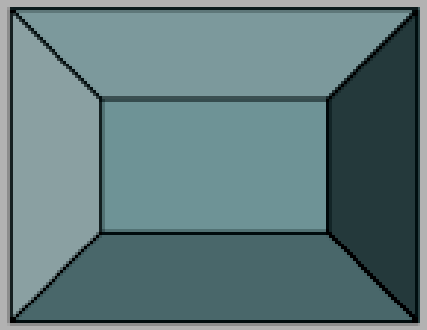
Definition of a Solid Model

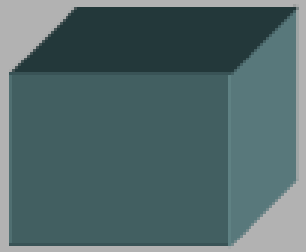
A solid model of an object is a more complete representation than its surface (wireframe) model. It provides more topological information in addition to the geometrical information which helps to represent the solid unambiguously.





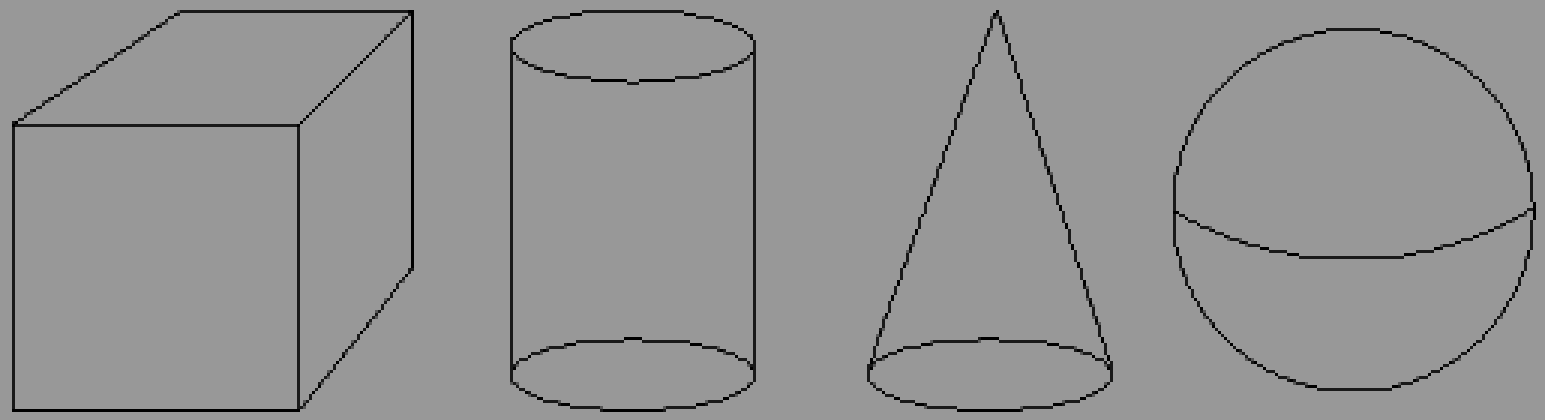
Wireframe Model



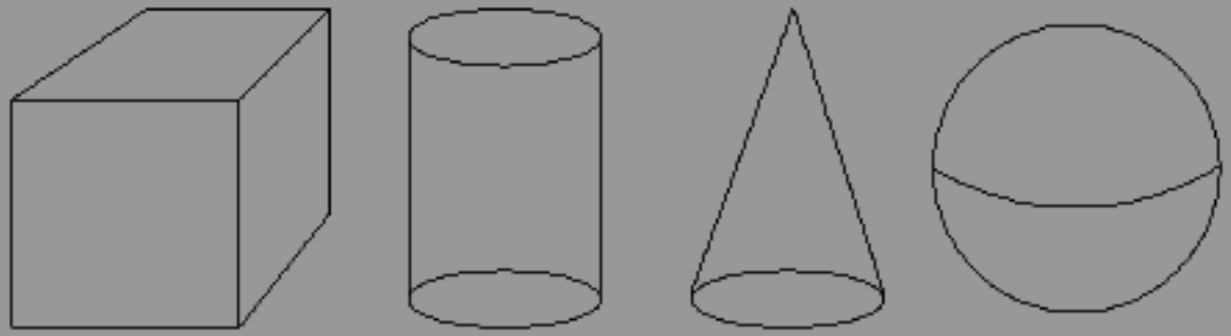


Solid Model

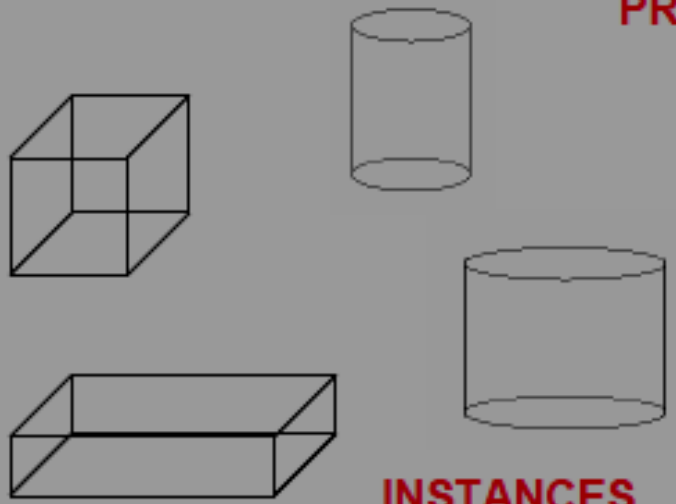
Geometric Modeling - Primitives



Primitives and Instances



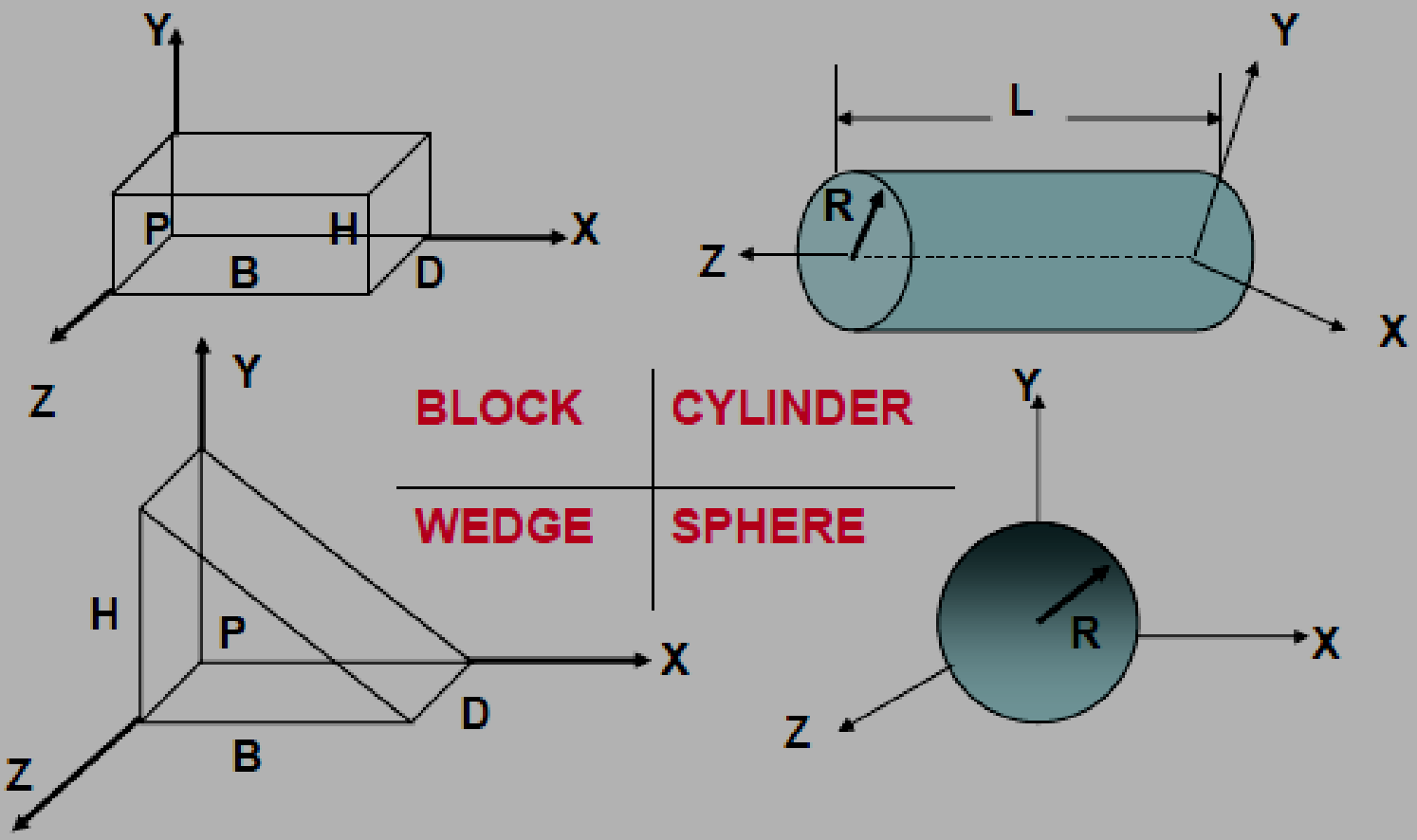
PRIMITIVES



INSTANCES

Instancing: An instance is a scaled / transformed replica of its original. The scaling may be uniform or differential. It is a method of keeping primitives in their basic minimum condition like unit cube and unit sphere etc.

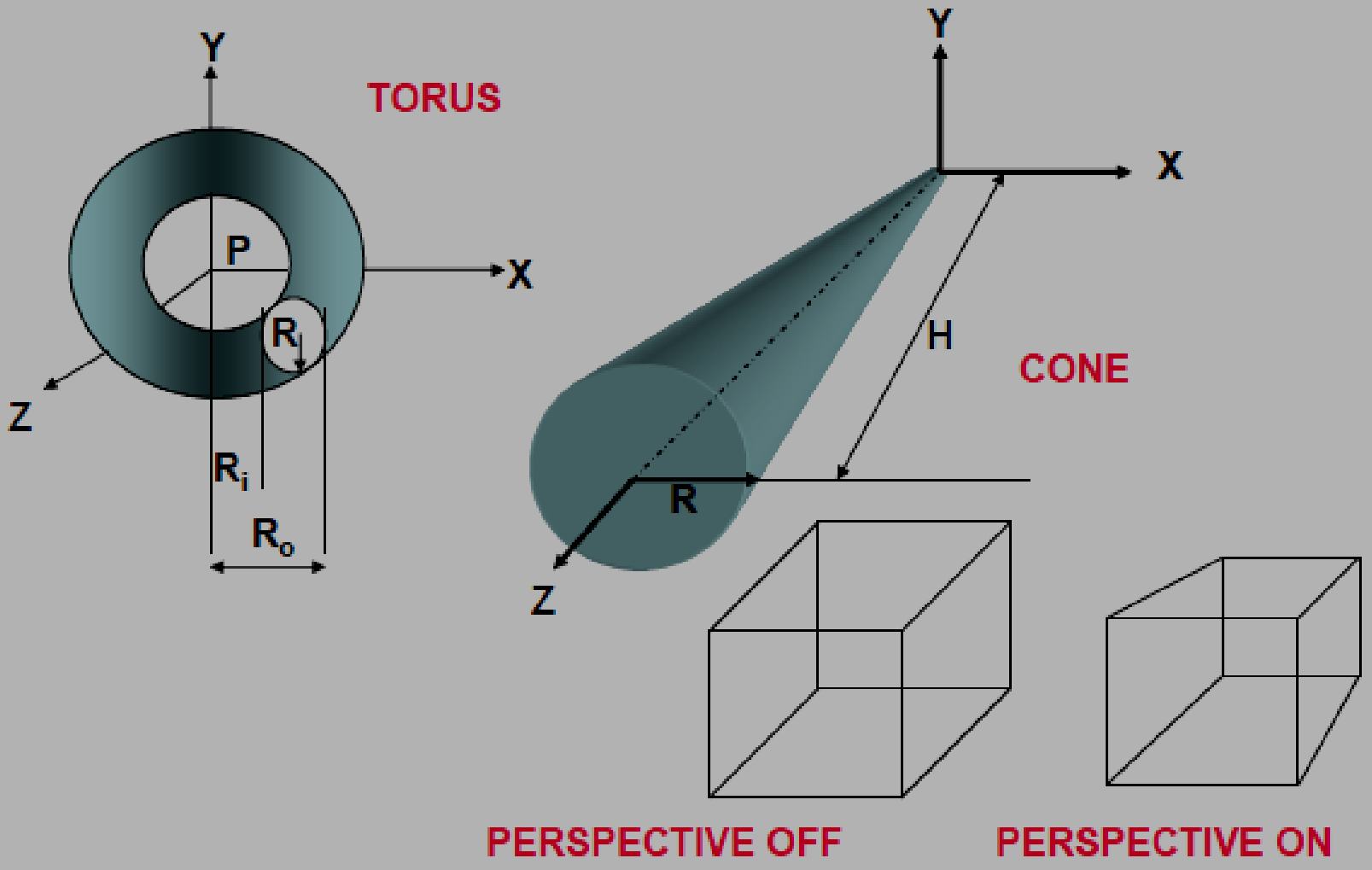
Solid Primitives: Ready starting objects



Origin of MCS: P Orientation Topology and shape

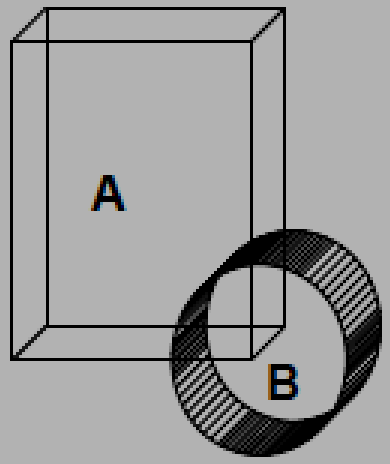
Dimension (size) L,B,H.D,R Perspective on/off

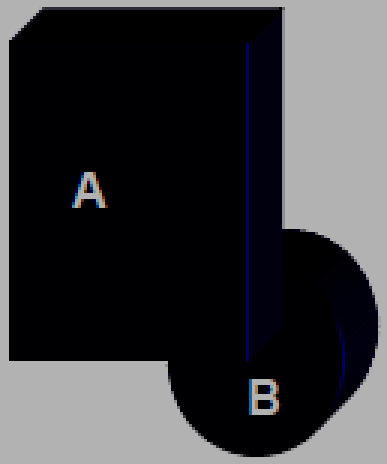
Solid Primitives: Ready starting objects



Operation on Primitives

- A desired solid can be obtained by combining two or more solids
- When we use Boolean (set) operations the validity of the third (rusulting) solid is ensured





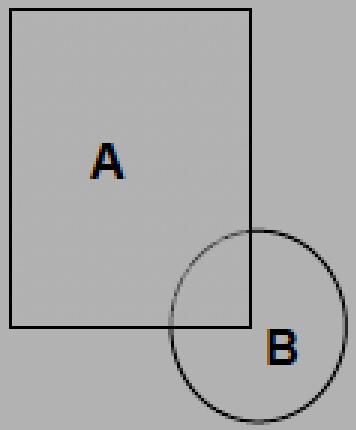
3 Dimensional

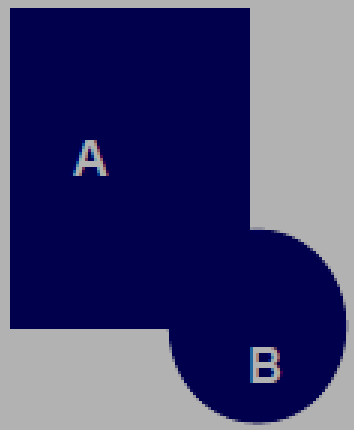
UNION: BLOCK ∪ CYLINDER

$A \cup B$

Operation on Primitives

- A desired solid can be obtained by combining two or more solids
- When we use Boolean (set) operations the validity of the third (rusulting) solid is ensured

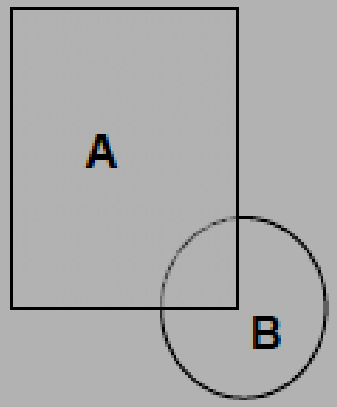


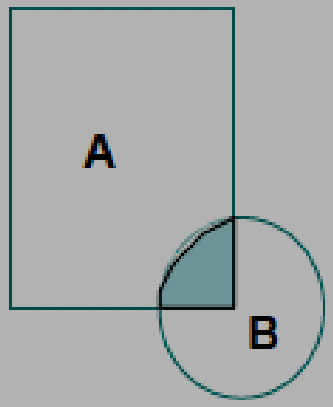


UNION: BLOCK ∪ CYLINDER

$A \cup B$

Operation on Primitives

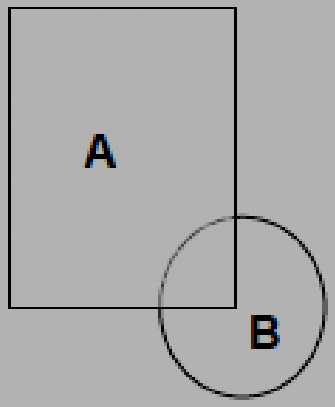


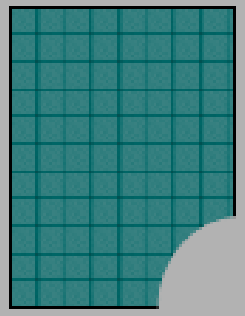


INTERSECTION:

BLOCK ∩ CYLINDER

$A \cap B$





A - B

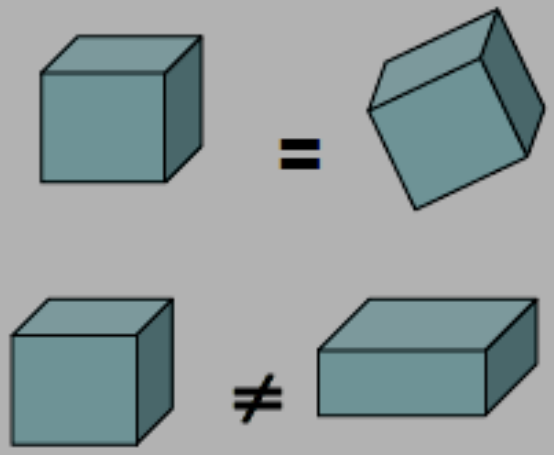
DIFFERENCE:

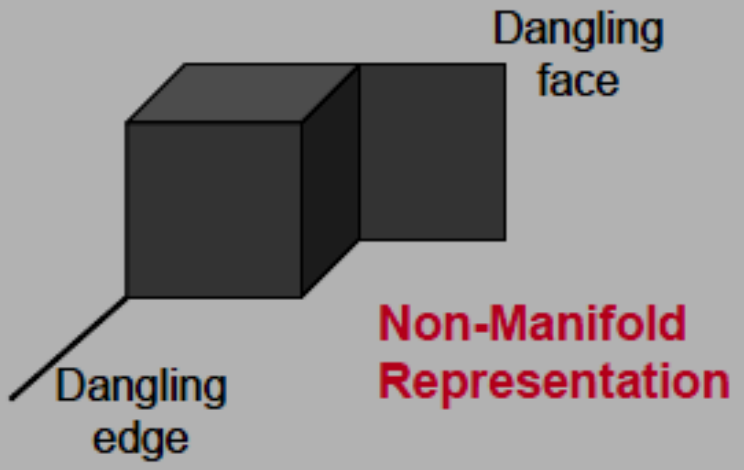
BLOCK - CYLINDER

B - A

Properties of Solid Models

- **Rigidity:** Shape of the solid is invariant w.r.t location/orientation
- **Homogeneous 3-dimensionality**: The solid boundaries must be in contact with the interior. No isolated and dangling edges are permitted.

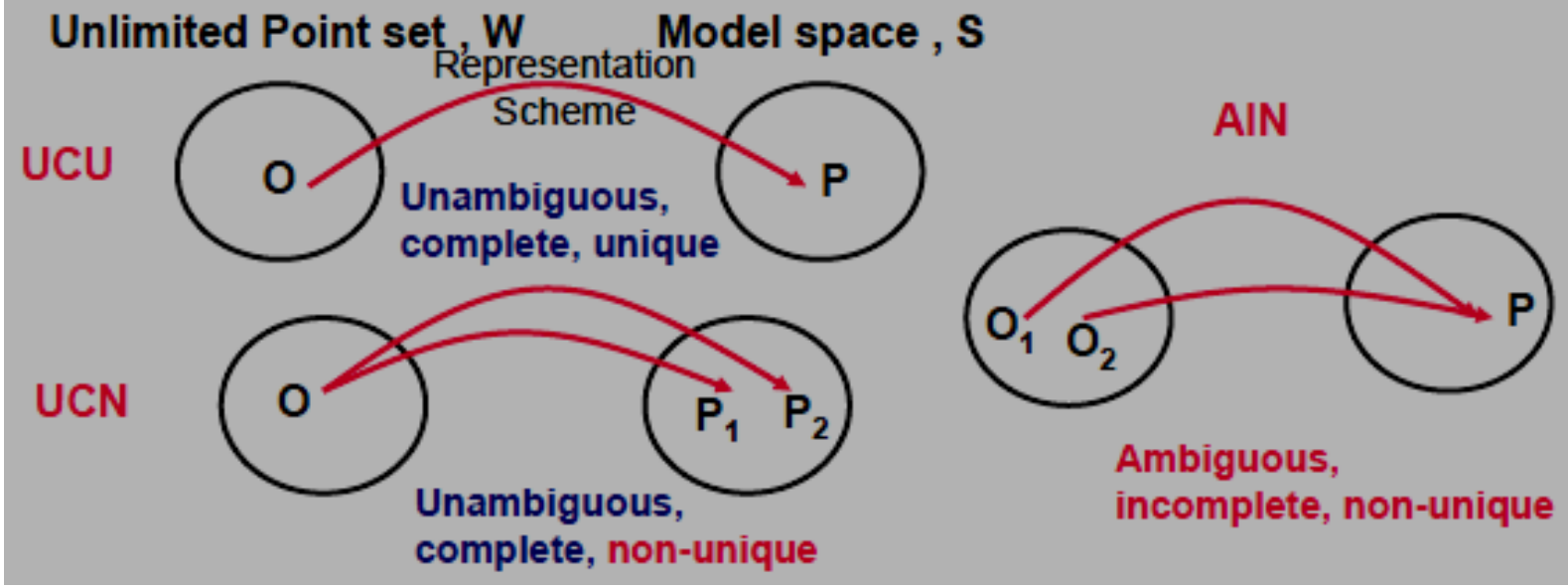




Properties of Solid Modeling Schemes

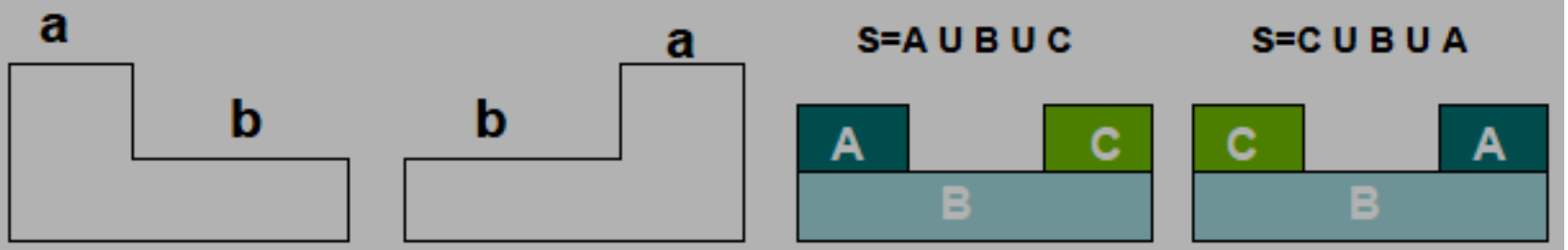
The schemes operate on r-sets or point sets to produce the desired solid model, which is automatically a valid solid

- For unambiguous, complete and unique representation the scheme maps points form point set (Euclidian) into a valid model to represent the physical object.



Common Features of Modeling Schemes

- **Domain**: The type of objects that can be represented or the geometric coverage.
- **Validity**: The resulting solid model when subjected to algorithms should succeed as valid solid.
- **Completeness** and **Unambiguousness**.
- **Uniqueness**: Positional and Permutational uniqueness.



Positionally non-unique Permutationally non-unique

Data Exchange formats and standards

IGES specification defines the format of the file, language format, and the product definition data in these formats. The product definition includes geometric, topological, and non-geometric data. The geometry part defines the geometric entities to be used to define the geometry. The topology part defines the entities to describe the relationships between the geometric entities. The geometric shape of a product is described using these two parts (i.e. geometry and topology). The non-geometric part can be divided into annotation, definition, and organization. The annotation category consists of dimensions, drafting notations, text, etc. The definition category allows users to define specific properties of individual or collections of entities. The organization category defines groupings of geometric, annotation, or property elements.

An IGES file consists of six sections: Flag, Start, Global, Directory Entry, Parameter Data, and Terminate. Each entity instance consists of a directory entry and parameter data entry. The directory entry provides an index and includes attributes to describe the data. The parameter data defines the specific entity. Parameter data are defined by fixed length records, according to the corresponding entity. Each entity instance has bi-directional pointers between the directory entry and the parameter data section.

DXF (Data eXchange Format) was originally developed by Autodesk, Inc., the vendor of AutoCAD. It has become a "de-facto" standard among most CAD vendors and is in wide use to exchange 2D/3D wireframe data. All implementations of AutoCAD accept this format and are able to convert it to and from their internal representation. A DXF file is a complete representation of the AutoCAD drawing database thus some features or concepts can't be used by other CAD systems. The DXF version R13 supports wireframe, surface, and solid representations.

A DXF file consists of four sections: Header, Table, Block, and Entity section. The header section contains general information about the drawing. Each parameter has a variable name and an associated value. The table section contains definitions of line types, layers, text styles, views, etc. The block section contains entities for block definitions. These entities define the blocks used in the drawing. The format of the entities in the block section is identical to entities in the entity section. The entity section contains the drawing entities, including any block references. Items in the entity section exist also in the block section and the appearance of entities in the two sections is identical.

STEP (STandard for the Exchange of Product model data) is a new International Standard (ISO 10303) for representing and exchanging product model information. It includes an object-flavored data specification language, EXPRESS, to describe the representation of the data. STEP defines also implementation methods, for instance, a physical transfer file, and offers different resources, e.g. geometric and topological representation.

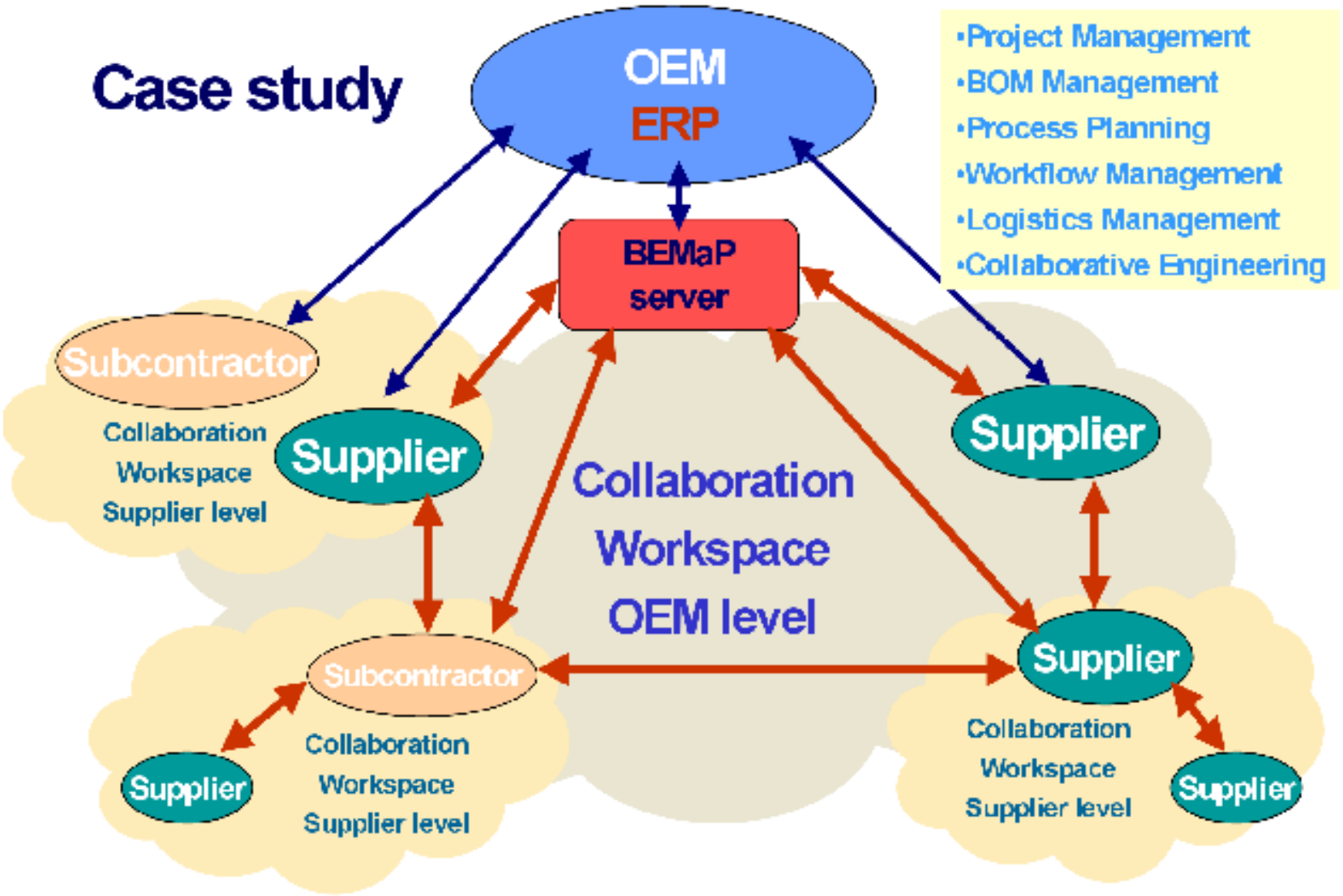
The objective of STEP is to offer system-independent mechanism to describe the product information in computer aided systems throughout its lifetime. It separates the representation of product information from the implementation methods. Implementation methods are used for data exchange. The representation offers a definition of product information to many applications. STEP provides also a basis for archiving product information and a methodology for the conformance testing of implementations.

STEP does not only define the geometric shape of a product: it also includes topology, features, tolerance specifications, material properties, etc. necessary to completely define a product for the purposes of design, analysis, manufacture, test, inspection and product support. The use of STEP is still very modest but it is growing all the time. The majority of CAD system vendors has implemented or is implementing STEP pre- and postprocessors for their CAD systems. STEP is an evolving standard that will cover the whole product life cycle in terms of data sharing, storage and exchange. It is the most important and largest effort ever established in engineering domain and will replace current CAD exchange standards.

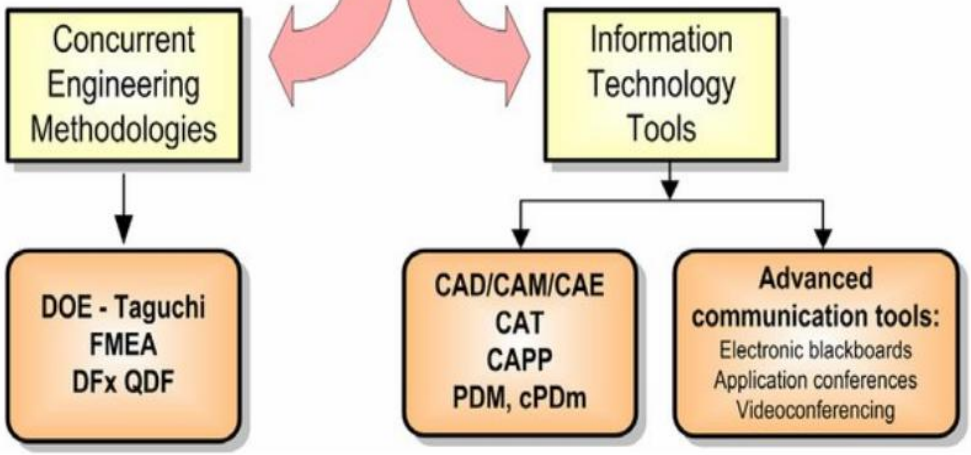
Collaborative Engineering

Collaborative Engineering

Collaborative Engineering



Collaborative Engineering



Collaborative Engineering