COMPUTER AIDED DESIGN
(BME-42)

Unit-IV: 3D Graphics
(7 Lectures)

- Introduction, Wireframe modeling, Surface modeling,
- Polygon surfaces-polygon meshes, polygon equations,
- Quadric and Superquadric surfaces, Blobby objects,
- Solid modeling-
  - Boolean set operations,
  - regularized set operations,
  - **Primitive instancing,**
  - **Sweep representation-translational, rotational and hybrid sweeps,**
  - Boundary representation-topology, geometry, boundary models
  - Constructive solid geometry-unbounded and bounded primitives

Lecture 30

Topics Covered

Limitations of Surface Modeling

Solid Modeling

Boolean Set Operations

Ordinary Boolean Operations

Regularized Boolean Operations

Prepared By

Prof. S. K. SRIVASTAVA

MED, MMMUT, Gorakhpur (UP)
sksme@mmmut.ac.in

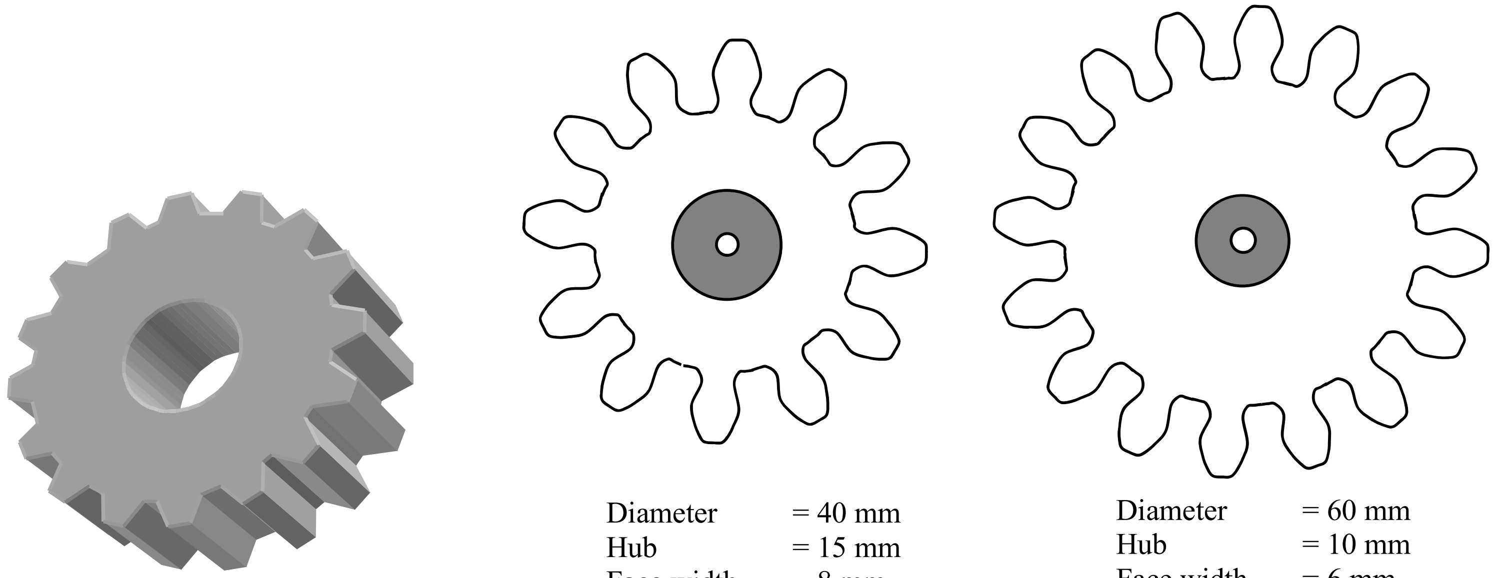
PRIMITIVE INSTANCING

- In this scheme of solid modeling, each element shape may be considered as a solid model with associated mass properties.
- The modeling system defines a set of three-dimensional solid shapes that are relevant to a particular area/discipline of engineering such as mechanical engineering, civil engineering, etc.
- For examples, along with the common 3D primitives such as block, cylinder, sphere, cone, wedge, etc., the primitives related to the modeling of mechanical engineering components such as gears, shafts, pulleys, etc. are also included.
- These primitives are parameterized in terms of the properties, e.g., one primitive object may be a regular pyramid with a user defined number of faces meeting at the apex.
- Primitive instancing is preferred for the complex objects such as gears, pulleys, bolts, etc. which are very difficult to define with Boolean combinations of simple objects.

PRIMITIVE INSTANCING

- The complex objects are characterized by few high-level parameters. For example, a spur gear is characterized by its pitch circle diameter, hub diameter, face width, number of teeth and hole diameter, as shown in Fig. (a).
- Similarly, the bolts are characterized by the number of sides of the bolt head *n*, bolt height *H*, length of threaded portion *L*, pitch of thread *p* and bolt diameter *d*, as shown in Fig. (b).
- In this method, no provisions are made for combining the primitive objects to form a new high-level object, using the regularized Boolean set operations.
- The complex high-level object can be created only by writing the code that defines it. Similarly, the computer program must be written to draw the objects as well as for determining the mass properties of individual object.

PRIMITIVE INSTANCING



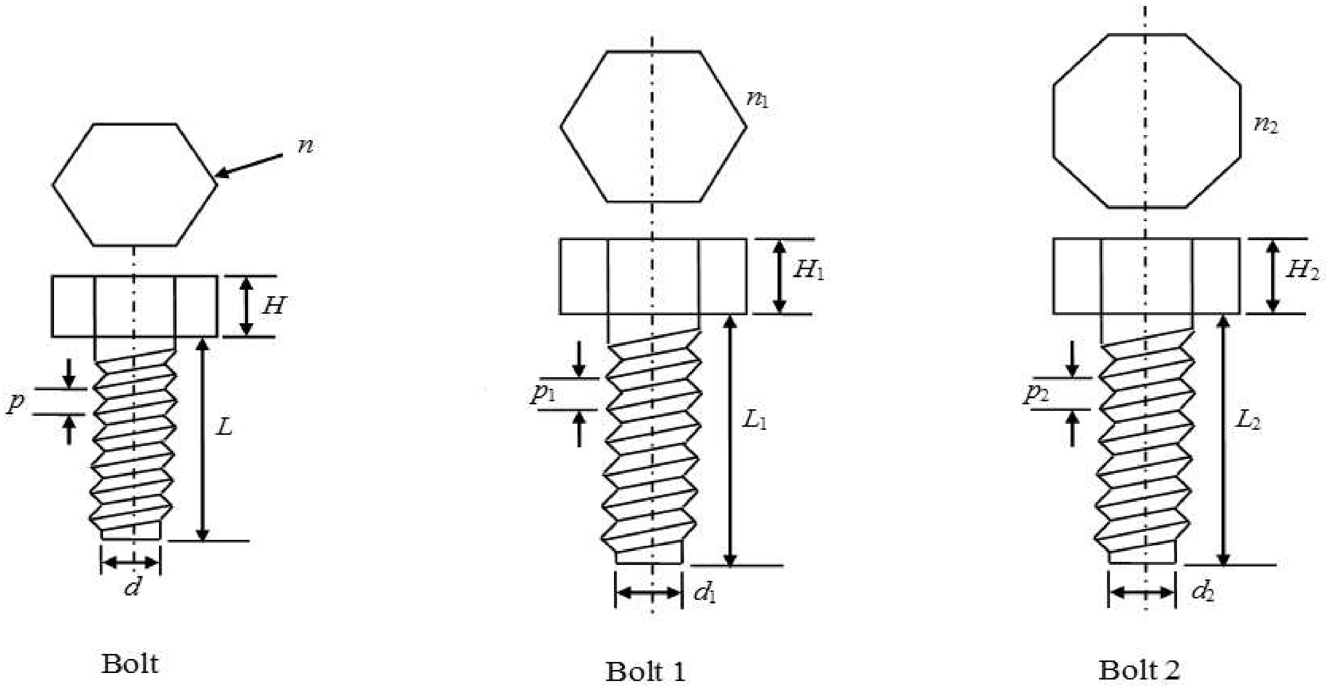
	Spur gear 1	Spur gear 2
Diameter	= 40 mm	= 60 mm
Hub	= 15 mm	= 10 mm
Face width	= 8 mm	= 6 mm
Teeth	= 12	= 16
Hole diameter	= 3 mm	= 3 mm

Solid model of spur gear   Spur gear 1   Spur gear 2

(a)

Spur gear defined by primitive instancing method of Solid Modeling

PRIMITIVE INSTANCING



(b)

Bolt defined by primitive instancing method of Solid Modeling

SWEEP REPRESENTATION

- Sweeping an object along the known trajectory in space defines a solid, called as *sweep*.
- This method is used for creating the thee-dimensional objects possessing translational, rotational or other symmetries.
- A set of two-dimensional primitives such as circles, rectangles, ellipses are used for the sweep representations.
- Other two-dimensional options may be closed spline curve or slices of a solid object.
- Sweeping is used, in general, as a means of entering the object descriptions into B-rep or CSG based solid modelers.
- Sweeping is based on moving a point, surface, curve, polygon, etc. along a specified path.

SWEEP REPRESENTATION

- Depending upon the types of motion to be performed by the object boundary, in general, there are three types of sweep representations of solids:

  I. Translational sweep
  II. Rotational sweep
  III. Hybrid sweep

I. Translational Sweep

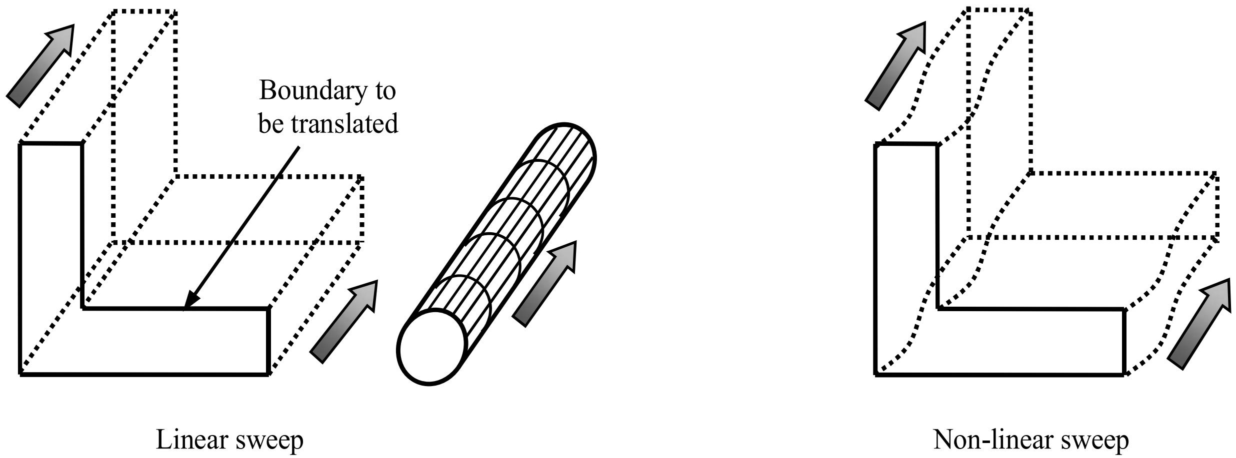
- The simplest type of area/surface movement through a given distance along the specified path, normal to the plane of area, results into a solid known as *translational* sweep or *extrusion*.

- This is a natural way to create a volume by extruding the metal or plastic through a die having the desired cross-section.

I. Translational Sweep…

- The sweeping may be *linear* or *non-linear* depending upon the path of movement of area/surface as shown in Fig. (a).

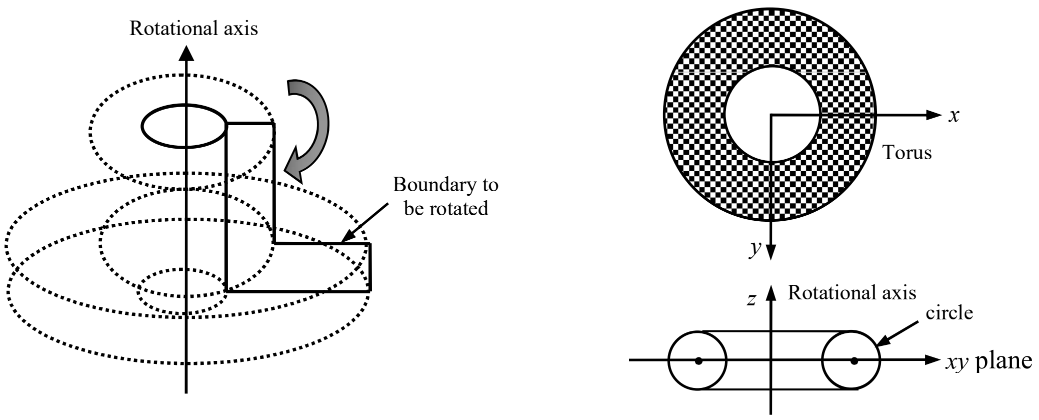
- In linear sweep, the path is linear described by a linear parametric equation, while in non-linear sweep, the path is a curve described by the higher order quadratic, cubic, etc. equations.



(a) Translational sweep

II. Rotational Sweep

- Rotational sweeps are defined by rotating an area/surface about the rotational axis, as shown in Fig. (b).
- If we use rotation axis perpendicular to the plane of spline cross-section, two-dimensional shapes are generated.
- However, if we rotate the cross-section about the axis containing the cross-section, three- dimensional shapes such as torus can be generated.



(b) Rotational sweep

SWEEP REPRESENTATION

II. Rotational Sweep…

- A torus of circular cross-section can be generated by sweeping a circle about the rotational axis through 360^o.
- We can move the area along a circular path through any angular distance from 0^o to 360^o.
- For non-circular path, we can specify the curve function describing the path and distance traveled along the path.
- It is also possible to vary the shape or size of the cross-section along the sweep path, e.g., crane hook.

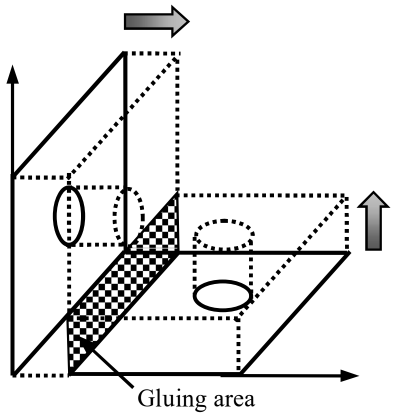
III. Hybrid Sweep

- The objects with hybrid sweeps possess motions along the two coordinate axes; followed by gluing at the contact surface.

SWEEP REPRESENTATION

III. Hybrid Sweep…

- Fig. (c) shows the hybrid sweeps, wherein, one rectangular surface moves along the x-axis and other rectangular surface, having a circular hole, along the y-axis.
- Thus, the two boundaries are swept in different directions and the two volumes are glued together to form the object.
- Sweeps are the natural and intuitive way to produce the objects.
- Many solid modelers allow user to create objects as sweeps but store the objects in some other representation scheme of solid modeling.



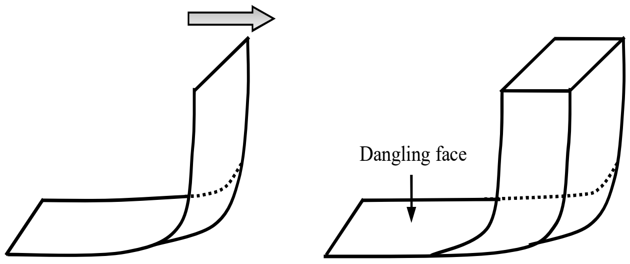
(c) Hybrid sweep

SWEEP REPRESENTATION

III. Hybrid Sweep…

- If sweeping directions is not proper, it may result into invalid solids or non-regular sets as shown in Fig.

Invalid Sweep



Sweeping operation of solid modeling is useful in engineering applications that involve modeling of swept volume in space. The two widely used applications are:

- Simulation of material removal process during the machining operations such as milling, turning, etc. The volume swept by a machine tool moving cutter along the cutting path is intersected with the stock of raw material.
- Interference detection of moving object such as robot arm. A moving object will collide with the fixed object it the swept volume due to the motion of first intersects with the fixed object.

BOUNDARY REPRESENTATION (B-REP)

- Boundary representation is one of the most popular solid modeling schemes for creating a solid model.
- This is based on the concept of representing a solid model using the *faces*, *edges* and *vertices*.
- The model is represented in the form of boundaries/faces formed by the edges and vertices; therefore, also termed as *perimeter* modeling.
- The faces are the subsets of closed and orientable surfaces.
- A closed surface is represented by the continuous surface without break; and orientable surfaces are distinguished by the direction of surface normal to point either inside or outside the solid model.
- A boundary model represented by the faces, edges and vertices should be topologically valid.
- The boundary models are not unique because an object may be represented by different combinations of faces, edges and vertices.

BOUNDARY REPRESENTATION (B-REP)

- However, verification of uniqueness is not verified because it is computationally expensive and not preferred.
- Despite surface representation for the boundary models, it is possible to compute volumetric properties such as mass properties of the model by assuming uniform density of the object by virtue of Gauss divergence theorem, which relates volume integral to the surface integral.

The database of a boundary representation model contains both its *topology* and *geometry*.

Topology and homeomorphism

- Topology is related to the number of vertices, edges and faces of the model.
- The other frequently used terms to define the primitives of a B-rep solid model and other topological items, which are used to create the B-rep solid model, are *loops*, *not through holes* and *through holes*.

BOUNDARY REPRESENTATION (B-REP)

Vertex

A vertex is a unique point in the space. The edges of an object meet at the vertex.

Edge

An edge is a finite, non-self-intersecting, directed space curved (or straight line) connected between the two vertices that are not necessarily distinct.

Face

- Face is a finite, connected, non-self-intersecting, region of a closed oriented surface bounded by one or more loops.
- The face of a solid is subset of the solid boundary, and union of all the faces defines the boundary of a solid.
- Faces are two-dimensional homogeneous regions possessing areas (not volume) without any dangling edges.
- A point on a face has a surface normal that has a sign associated with it to indicate whether it points into or away from the solid interior.

Loop

A loop is defined as a non-self-intersecting, piecewise closed space curve, may be a boundary of a face termed as outer loop. An inner loop is represented by the region bounded by a boundary hole on the face of a polyhedral object.

Not through hole

This is defined as a depression (or boundary hole) in a face of the object.

Through hole

A through hole is also termed as handle. A handle is defined as a hole that passes through the entire object. The topological name of the through hole is *genus*.

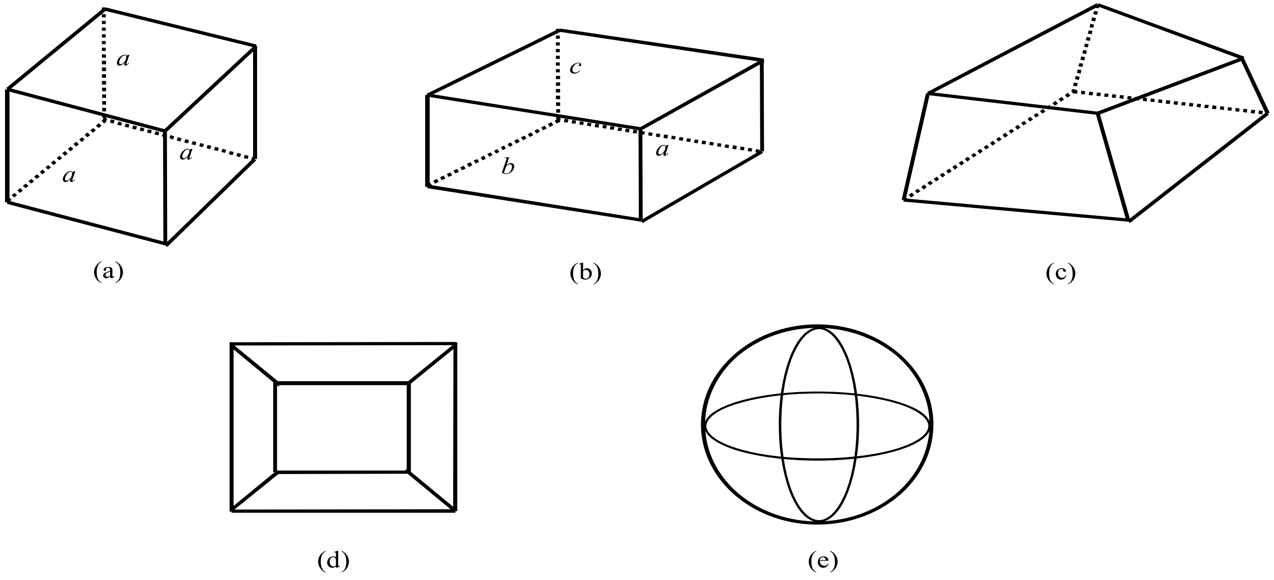
BOUNDARY REPRESENTATION (B-REP)

- The aim of topologically valid model is to set rules or procedures to recognize the geometrical features of the object.
- Two objects will belong to the same categories if they have same basic, overall structure even though differ much in detail.

- For example, in a cube (Fig. a), the internal angles are 90^o and the edge length a is same.
- This remains a cube irrespective of the value of edge length a.
- Similarly, the sphere form remains the same irrespective of the radius specified.
- A cube is a special case of a block with different lengths a, b, and c along the three mutually perpendicular edges (Fig. b).
- If we choose the internal angles other than 90^o and consider different edge lengths, the shape of the object changes drastically (Fig. c).

BOUNDARY REPRESENTATION (B-REP)

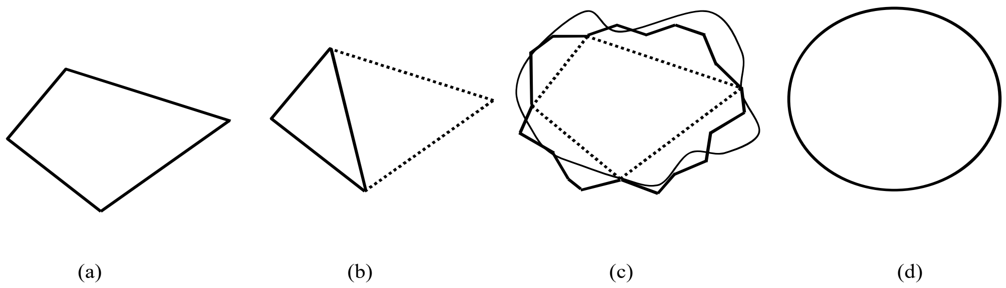
- Still there is a common thing among these three objects; they are all hexahedrons (of six sides).
- Therefore, if we do not consider the intricacies of lengths and angles, all solids (from a-c) are topologically identical.
- Similarly, 2D objects such as square, a parallelogram and a rectangle are topologically identical.



Hexahedral topology of objects (a-d), all homeomorphic to a sphere(e)

BOUNDARY REPRESENTATION (B-REP)

- ❖ Since, we can ignore the edge length between the two topological valid objects; thus, one side of zero length of a quadrilateral result into a triangle (Fig. 8b), which would be topologically identical to the quadrilateral.
- ❖ A further reshaping or re-morphing the straight edges of a quadrilateral is to have bends.
- ❖ A polygon (shown by thick lines in Fig. c) approaches to simple closed curve (shown by thin lines in Fig. c), if edge lengths of the quadrilateral approaches to infinity.
- ❖ Thus, all three planar shapes are topologically identical.



Closed planar shapes of identical topology (a-c), all homeomorphic to a circle (d)

Regularized Boolean Operations…

The regularized Boolean set operator are defined as

$$A\ (\text{operator})^*\ B = \text{Closure}\ (\text{interior}\ (A\ (\text{operator})\ B))$$

where operators are either Union (∪) or Intersection (∩) or Difference (-).

For intersection operator (say), we write

$$A \cap^* B = \text{Closure}\ (\text{interior}\ (A \cap B))$$

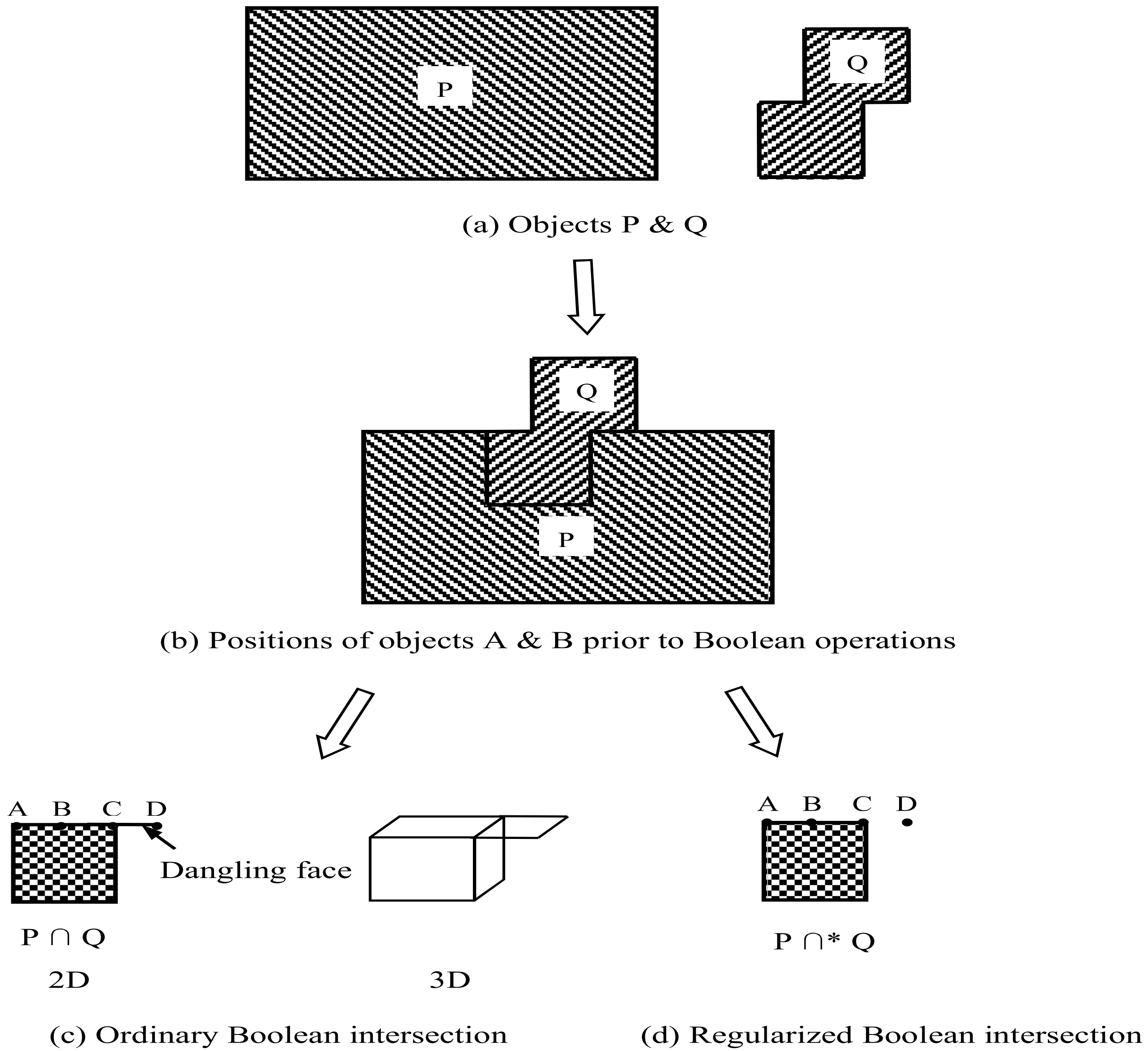
- The regularized Boolean set operator produces only regular set when applied onto an ordinary set.
- When regularized Boolean intersection is applied on the cubes, then Figs. a & e yield same results as their ordinary Boolean intersection, but is empty in Figs. b, c & d.

Comparison of Ordinary and Regularized Boolean Operations

- The ordinary Boolean intersection of two objects P and Q, shown in Fig., contains the intersection of the interior and boundary points of each object with the interior and boundary of the other.
- They result into a dangling face CD, shown as a line CD, in the cross-section.
- It includes a shared boundary (AB) in the resulting boundary if both objects lie on the same side of it, and excludes the shared boundary (CD) if the objects lie on the opposite sides.
- However, boundary interior intersection (BC) is always included to maintain the closure.
- The regularized Boolean set operators are used as a user interface to develop the complex objects from simple ones, in most of the three-dimensional object representation schemes.
- They also explicitly used in the Constructive Solid Geometry (CSG) scheme of solid modeling.

BOOLEAN SET OPERATIONS…

Boolean Intersection of Objects P and Q



COMPUTER AIDED DESIGN
(BME-42)

Unit-IV: 3D Graphics

(7 Lectures)

- Introduction, Wireframe modeling, Surface modeling,
- Polygon surfaces-polygon meshes, polygon equations,
- Quadric and Superquadric surfaces, Blobby objects,
- Solid modeling-
  - Boolean set operations,
  - regularized set operations,
  - Primitive instancing,
  - Sweep representation-translational, rotational and hybrid sweeps,
  - Boundary representation-**topology, geometry, boundary models**
  - Constructive solid geometry-unbounded and bounded primitives

Lecture 31

Topics Covered

Solid Modeling

Boundary representation

Topological Validity of Object

Geometry

Types of Boundary Models

Closed Polyhedral Objects

Open Polyhedral Objects

Data Structure of Boundary Models

Euler Operators of Boundary Models

Example

Prepared By

Prof. S. K. SRIVASTAVA

MED, MMMUT, Gorakhpur (UP)
sksme@mmmut.ac.in

Topological Validity of Object

- A boundary representation model must be *topologically valid*.
- *Euler- Poincaré law* (generally called as Euler's law) is used to validate the topology of the object model.
- Euler proved that polyhedra that are homeomorphic to a sphere, i.e., their faces are non-self-intersecting and belong to a closed orientable surfaces (viz., surfaces can be distinguished by using the direction of surface normal to point to the inside or outside of the solid model), are topologically valid if the faces, edges and vertices are manipulated such that

$$F - E + V - L = 2(B - G)$$

where F = Number of faces

E = Number of edges

V = Number of vertices

L = Number of inner loops

B = Number of bodies

G = Number of through holes/genus

Topological Validity of Object

- Euler's law applies to the *closed polyhedral* objects only.
- The above eqn. ensures the validity of closed boundary solid models.
- Open polyhedral objects do not satisfy above eqn. For open polyhedral objects, Euler's law is modified as

$$F - E + V - L = B - G$$

where B refers to an *open body* such as wire, area or volume.

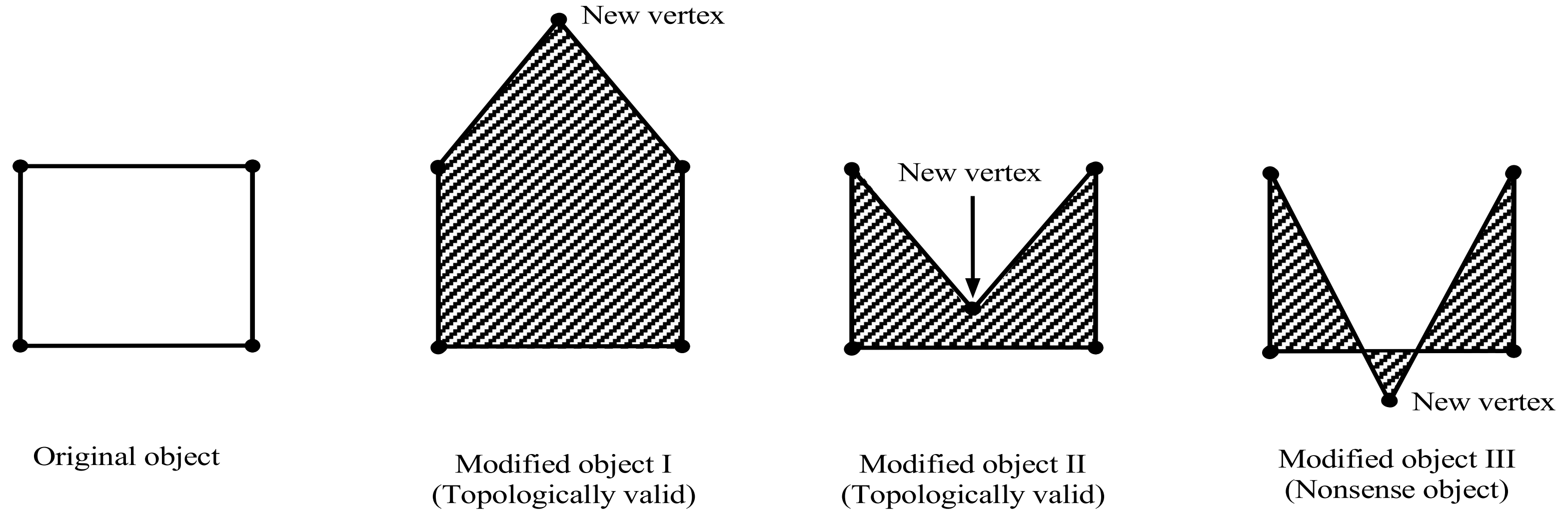
- Similar to the regularized Boolean operators, Euler law ensures the integrity of the boundary models.
- The boundary models may be built up incrementally by adding the faces, vertices and edges so that it satisfies the Euler's law during the solid development.

Geometry

- The *geometry* of boundary models includes coordinates of vertices, rigid motion and geometric transformation (translation, rotation, etc.).
- It also includes *metric* information such as distances, areas, volumes, angles, mass, etc. The geometry is created by performing Euclidean calculations.
- The topology and geometry of a boundary model is interrelated. Both must be compatible otherwise a *nonsense* object is created.
- Fig. shows the modifications in the shape of a square formed by the edges and vertices.
- The introduction of a new vertex and one edge modifies the shapes of the square object.
- For example, modified objects I & II are topologically valid whereas object III is a *nonsense* object, and it is topologically invalid boundary model.

BOUNDARY REPRESENTATION (B-REP)…

Geometry



Modified shapes of boundary model by introducing a new vertex

Types of Boundary Models

The modeling of boundary objects mainly depends on the type of primitive surfaces (plane, curved, etc.) used. In general, a solid modeling system consists of the following components:

I. *A particular solid modeling scheme must define the type of object that can be modeled.*

II. *The type of primitives required to model the object must be identified.*

III. *The type of operators that enable users to build the complex shaped objects.*

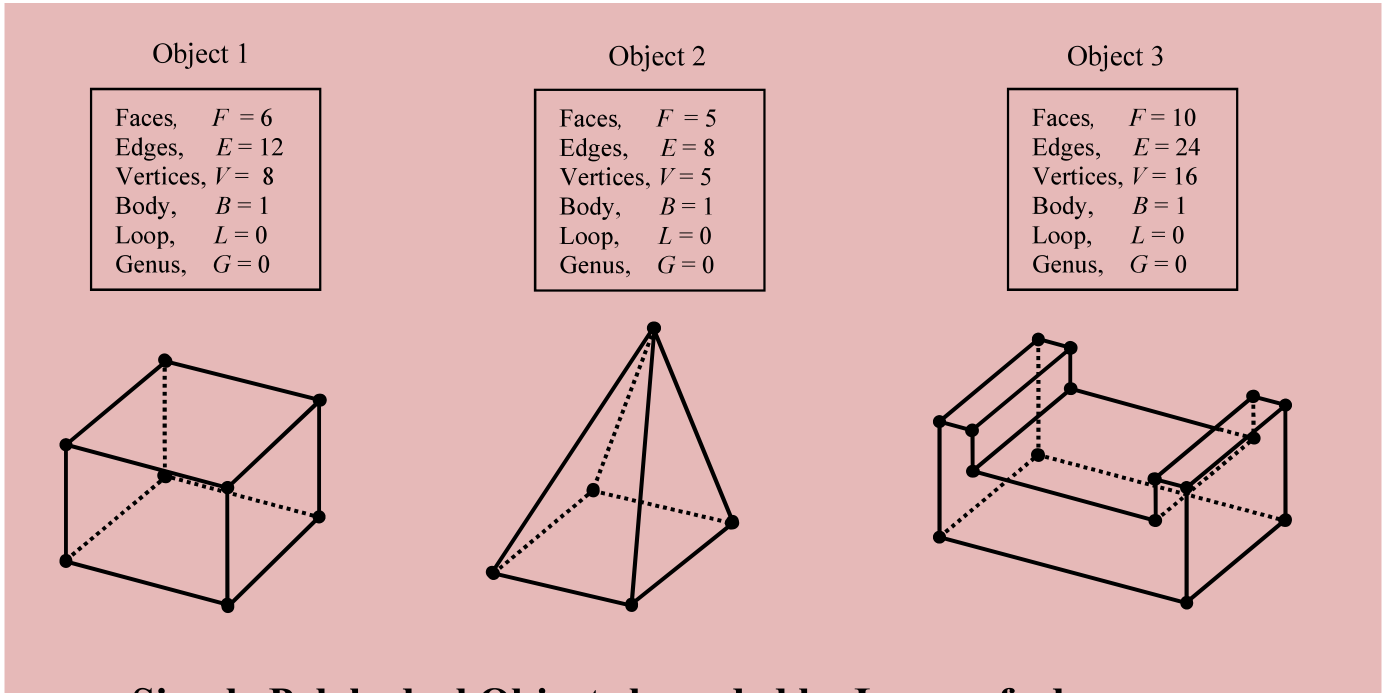
IV. *The facility to store the relevant data and information, by designing a suitable data structure of the solid model.*

Polyhedral objects or *curved* objects are used in many engineering applications. A polyhedral object may be closed or open. Since polyhedral objects are simple; therefore, described first.

CLOSED POLYHEDRAL OBJECTS

- A *closed polyhedral object* consists of plane-faced polyhedrons connected at straight edges, which in turn, connected at the vertices, e.g., a cube (object 1).

- The counts of various variables (faces, edges, vertices, bodies, inner loops and genus) for each closed polyhedron object is shown in Figs.



Simple Polyhedral Objects bounded by Loops of edges

CLOSED POLYHEDRAL OBJECTS…

Closed polyhedral objects are classified into the four categories:

I. Simple Polyhedra

- For simple polyhedra, each face of the object is bounded by a single set of connected edges, i.e., bounded by single loop of edges (called outer loop), connected at the vertices.

- These polyhedral objects neither have *through holes* nor have *not through holes*. The objects 1, 2 & 3, shown in Fig., represent *simple polyhedral* objects.

- For simple closed polyhedrons L = G = 0, B = 1, eqn. reduces to

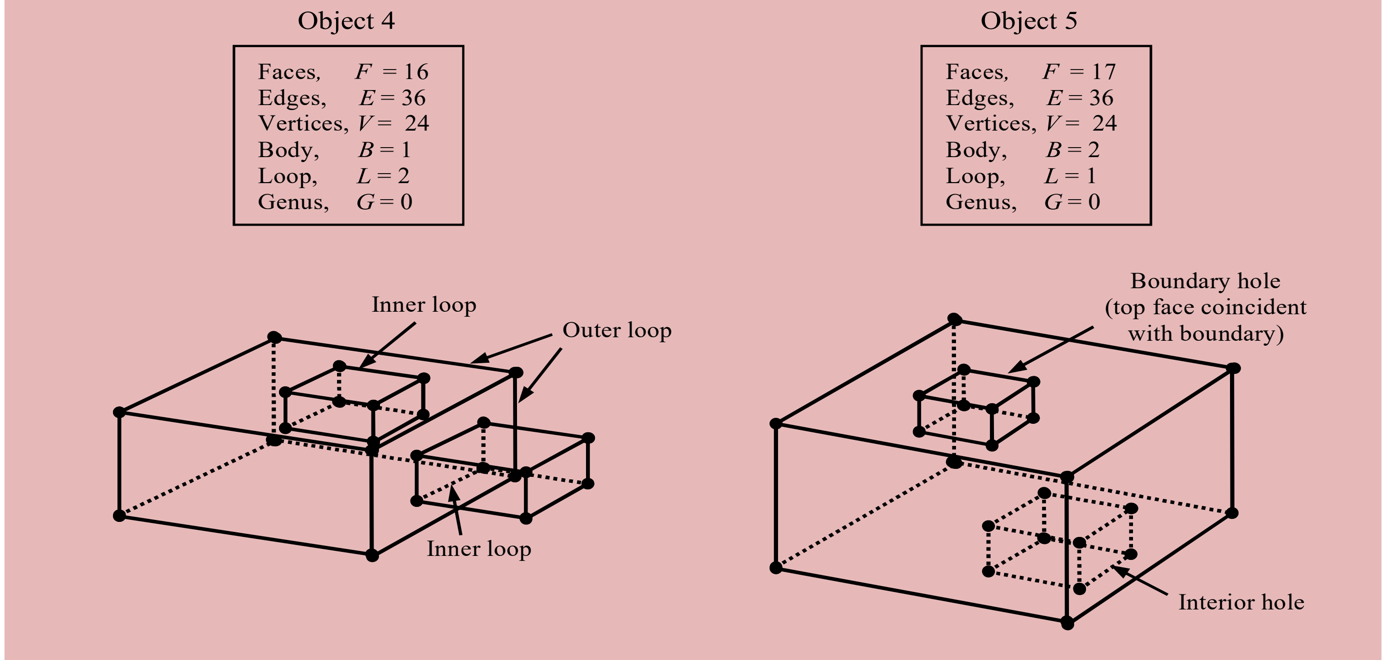
$$F - E + V = 2$$

The boxes shown in Fig. shows the number of faces, vertices and edges for the polyhedral objects 1, 2 & 3, each satisfy above eqn.

CLOSED POLYHEDRAL OBJECTS…

II. Polyhedra with Faces of Inner Loops

- In this category, one or more faces of the object contain more than one loop of edges.
- The face of an object contains inner loops along with the outer loops.
- The inner loops are also called as *rings*. Fig. shows object 4 that contains *two inner* loops.



(a) Polyhedral object with faces of inner loop (b) Polyhedral object with not through hole

CLOSED POLYHEDRAL OBJECTS…

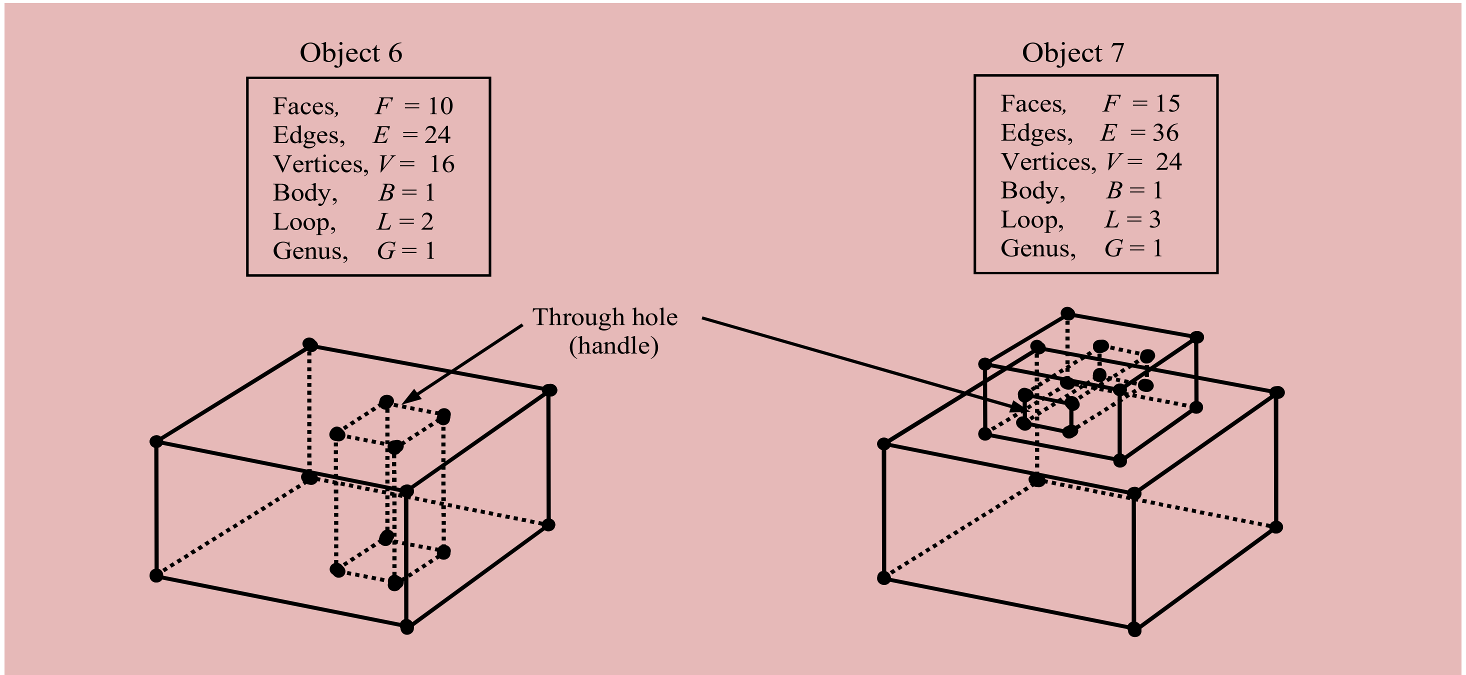
III. Polyhedra with not through holes

- These polyhedra do not contain holes through the entire object.
- Either one face of the hole is coincident with the boundary of the faces called boundary holes or it may be an interior hole in the form of void or crack inside the object, with no faces coincident with the boundary.
- The object 5 in Fig. contains one boundary hole and one interior hole.
- The exterior cube (main cube) and interior cube (hole) forms two bodies of the polyhedral objects.

CLOSED POLYHEDRAL OBJECTS…

IV. Polyhedra with through holes

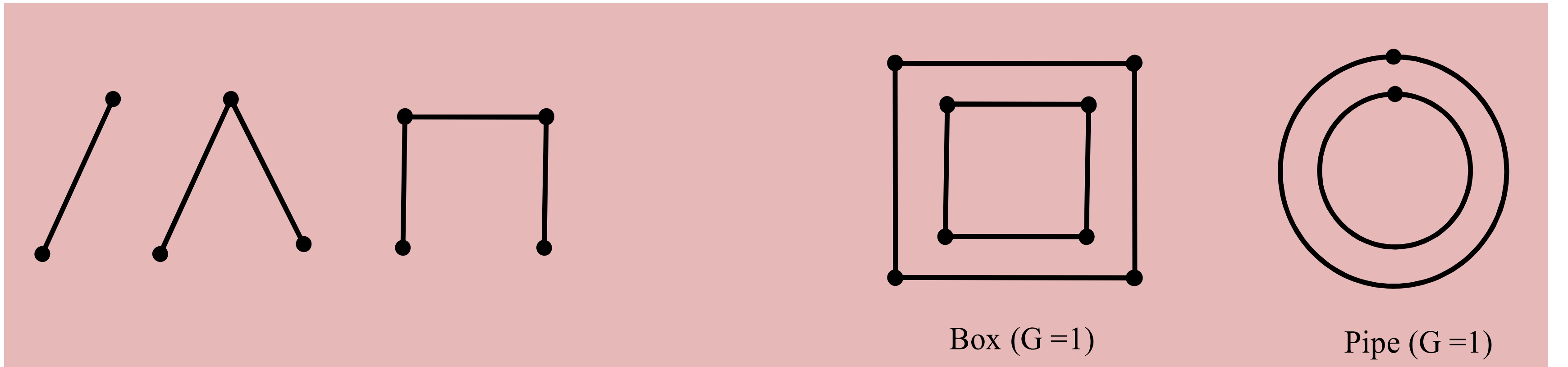
- This is the fourth category of objects contains *through holes*.
- These through holes are termed as *handles*.
- Fig. shows the examples of polyhedral boundary models (objects 6 & 7) containing through holes.



(c) Polyhedral object with through holes

OPEN POLYHEDRAL OBJECTS…

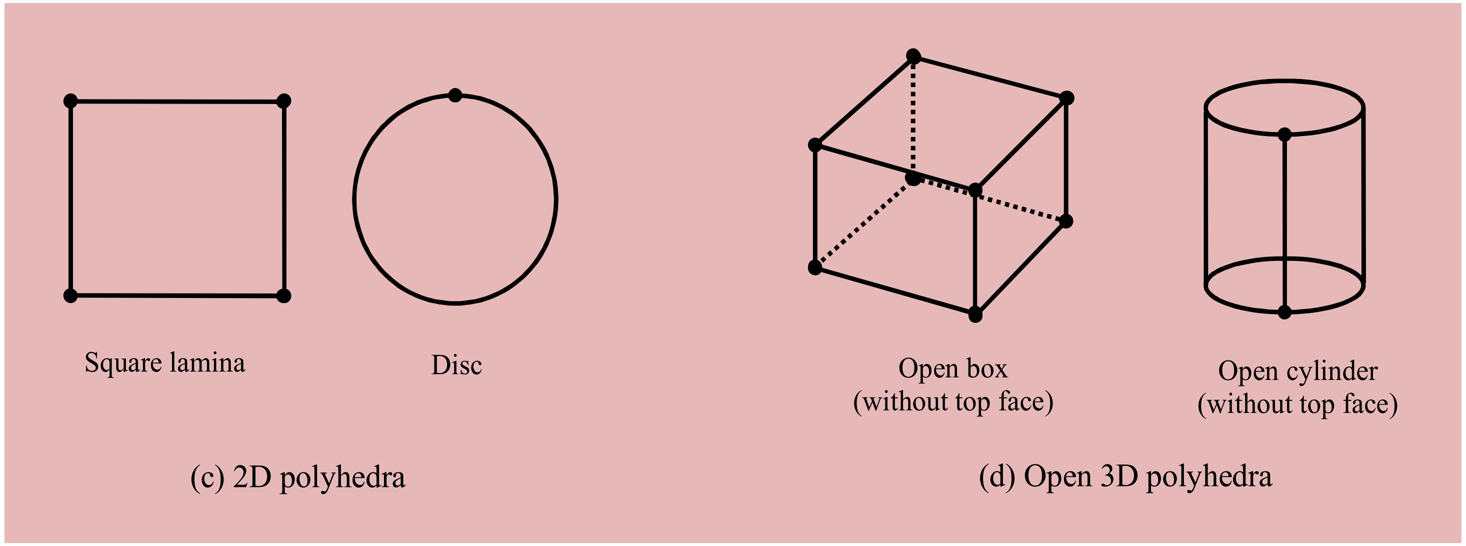
- Open polyhedral objects are obtained during the construction of closed polyhedron boundary models and all *two-dimensional* polygonal objects.
- Modified Euler's law ($F - E + V - L = B - G$) is used to ensure the *validity of open polyhedral* objects.
- Moreover, above eqn. can also be used for creating the wireframe boundary models. Fig. shows some open polygonal objects.



(a) Wire polyhedra

(b) Open shell polyhedra

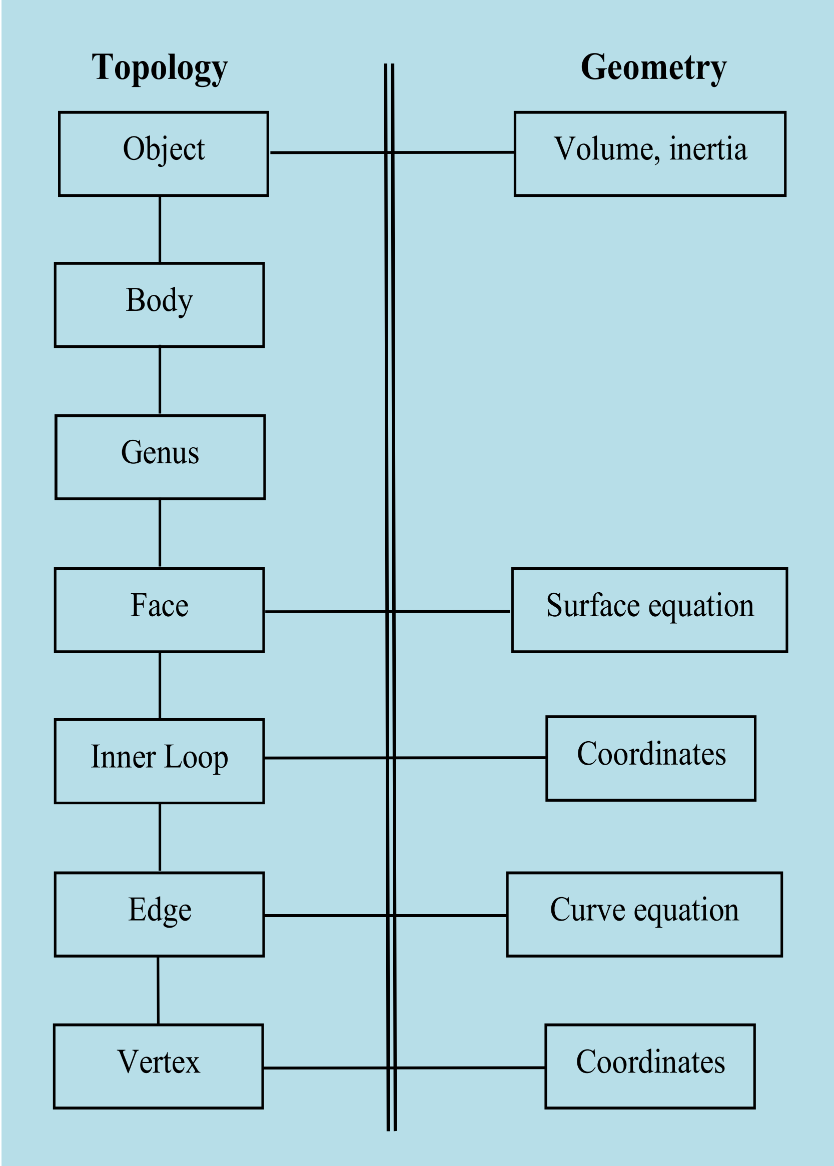
II. Open Polyhedral Objects…



DATA STRUCTURE OF BOUNDARY MODELS

- The data structure of a boundary model should have both *topological* and *geometrical* information of three-dimensional objects.
- Fig. shows the general *data structure* of a solid model based on boundary representation scheme for closed polyhedral objects.
- Two tables are created for the data storage, one for the topological information, which provides a list of bodies, faces, loops, edges and vertices; and other for the geometrical information that defines the faces, edges and vertices of the boundary model.
- For faces and edges, the surface and curve equations, respectively, are stored in the table.
- For edges consist of synthetic curves, the control points, slopes and knot vector may be stored depending on the Hermite, Bézier and B-spline segments modeled.
- The topology of inner loops and vertices are stored in the form of point coordinates. Each line in the data structure represents a pointer in the database.

DATA STRUCTURE OF BOUNDARY MODELS



EULER OPERATORS FOR BOUNDARY MODELS

- Euler-Poincaré operators are used for *creating/editing* the edges, vertices and faces of the polyhedron boundary models.
- The operators are designed such that *Euler's law is always valid for the intermediate shapes.*
- There are two categories of Euler's operators: *Make* groups and *Kill* groups for *adding* and *deleting* the primitives (points, edges, etc.), respectively.

Make group

- *Make* operator is expressed as *Mxyz* where parameters *x*, *y* and *z* stands for the vertex, edge, face, loop, etc.
- For example, *MEV* stands for adding (making) the edges and vertices; similarly, *MFE* operator is used for adding the faces and edges in the boundary model.

Make group…

- The *MSG* operator makes a shell with a hole. Euler operators form a complete set of modeling primitives in such a way that any polyhedron at intermediate stages satisfies Euler-Poincaré law.

- Thus, Euler's operators are highly significant during the construction of B-rep solid models. Table describes some of the useful *Make* group Euler's operators to add the boundary model elements (edges, vertices, faces, etc.) in the existing polyhedron models.

- The combined *Make-Kill* operators are used to add and delete the elements simultaneously during the development of boundary models.

- It should be noted that the addition/deletion of elements do not change the Euler-Poincaré relationship, as shown in the rightmost column of Table

EULER OPERATORS FOR BOUNDARY MODELS

Some Make group Euler's operators

Operator	Operation	V	E	F	L	B	G	Change in Euler-Poincaré law
MEV	Make Edge, Vertex	+1	+1	-	-	-	-	0
MFE	Make Face, Edge	-	+1	+1	-	-	-	0
MBFV	Make Body, Face, Vertex	+1	-	+1	-	+1	-	0
MME	Make Multiple Edges	-	+n	-	-	-	-	0
MFEVB	Make Face, Edge, Vertex, Body	+1	+1	+1	-	-	-	0
MEKL	Make Edge, Kill Loop	-	+1	-	-1	-	-	0
MEKBFL	Make Edge, Kill Body, Face, Loop	-	+1	-1	-1	-1	-	0
MFKLG	Make Face, Kill Loop, Genus	-	-	+1	-1	-	-1	0
MFEVKG	Make Face, Edge, Vertex, Kill Genus	+1	+1	+1	-	-	-1	0

Kill group

- *Kill* operator is expressed as *Kxyz* and used for killing or deleting the elements.
- By exchanging *M* with *K* in the Make Table, yields Kill group, as shown in Kill Table.
- We can partially delete some of the entities of existing polyhedron and then use the Make group to reconstruct the modified boundary model.
- For example, *KEV* operator (complement of MEV) is used for deleting (killing) the *edges* and *vertices* in the boundary model.
- *Kill* Table describes some of the useful *Kill* group Euler's operators to delete the elements (edges, vertices, faces, etc.) in the existing polyhedron models.
- The combined *Kill-Make operators* are used to delete and add the elements simultaneously during the creation of boundary models.
- However, deletion/addition of elements do not change the Euler-Poincaré relationship, as shown in the rightmost column of Kill Table.

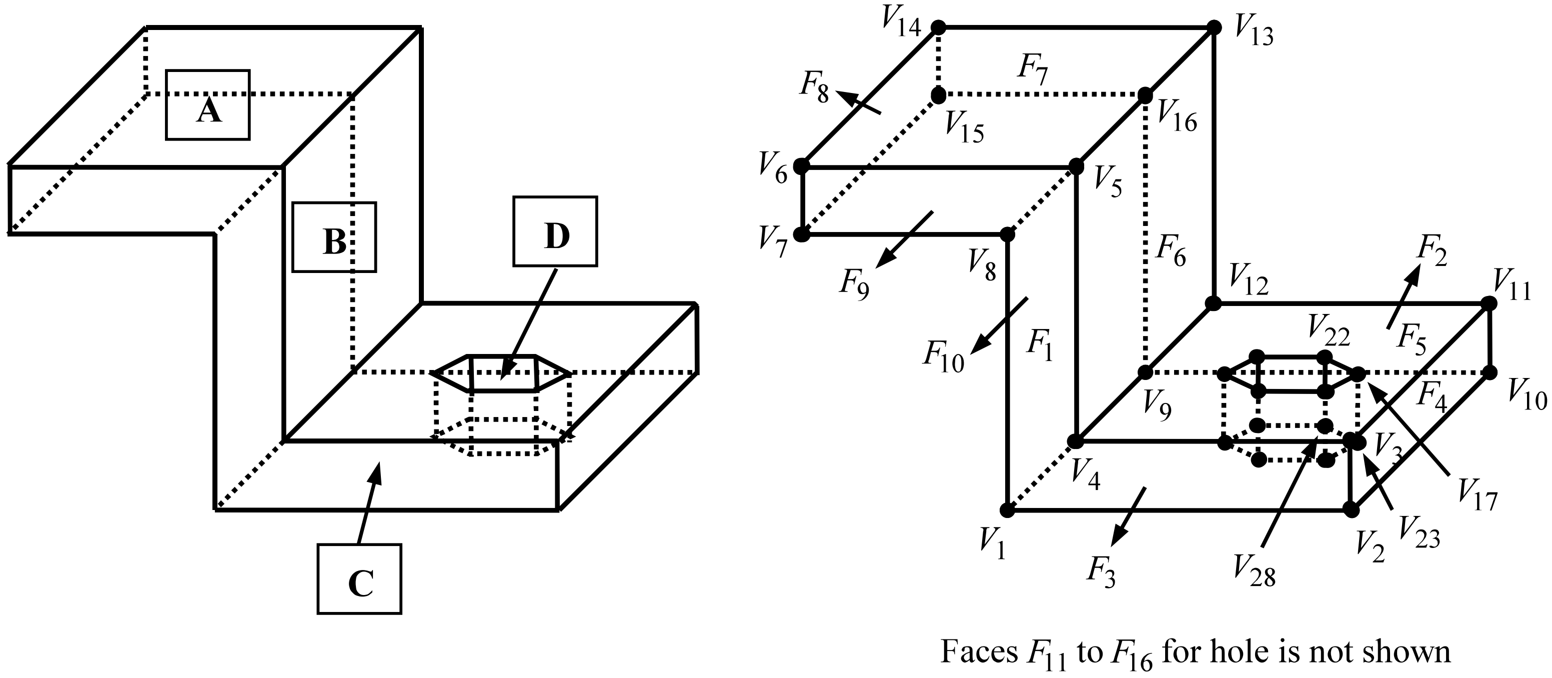
EULER OPERATORS FOR BOUNDARY MODELS

Some Kill group Euler's operators

Operator	Operation	V	E	F	L	B	G	Change in Euler-Poincaré law
KEV	Kill Edge, Vertex	-1	-1	-	-	-	-	**0**
KFE	Kill Face, Edge	-	-1	-1	-	-	-	**0**
KBFV	Kill Body, Face, Vertex	-1	-	-1	-	-1	-	**0**
KME	Kill Multiple Edges	-	-n	-	-	-	-	**0**
KFEVB	Kill Face, Edge, Vertex, Body	-1	-1	-1	-	-	-	**0**
KEML	Kill Edge, Make Loop	-	-1	-	+1	-	-	**0**
KEMBFL	Kill Edge, Make Body, Face, Loop	-	-1	+1	+1	+ 1	-	**0**
KFMLG	Kill Face, Make Loop, Genus	-	-	-1	+1	-	+1	**0**
KFEVMG	**Kill Face, Edge, Vertex, Make Genus**	**-1**	**-1**	**-1**	**-**	**-**	**+1**	**0**

BOUNDARY REPRESENTATION OF SOLID MODELS

The boundary model of the solid S has 16 faces, 28 vertices, 42 edges, 2 loops, 1 body and 1 genus. They all together satisfy Euler's law, i.e., $F - E + V - L = 2(B - G)$.

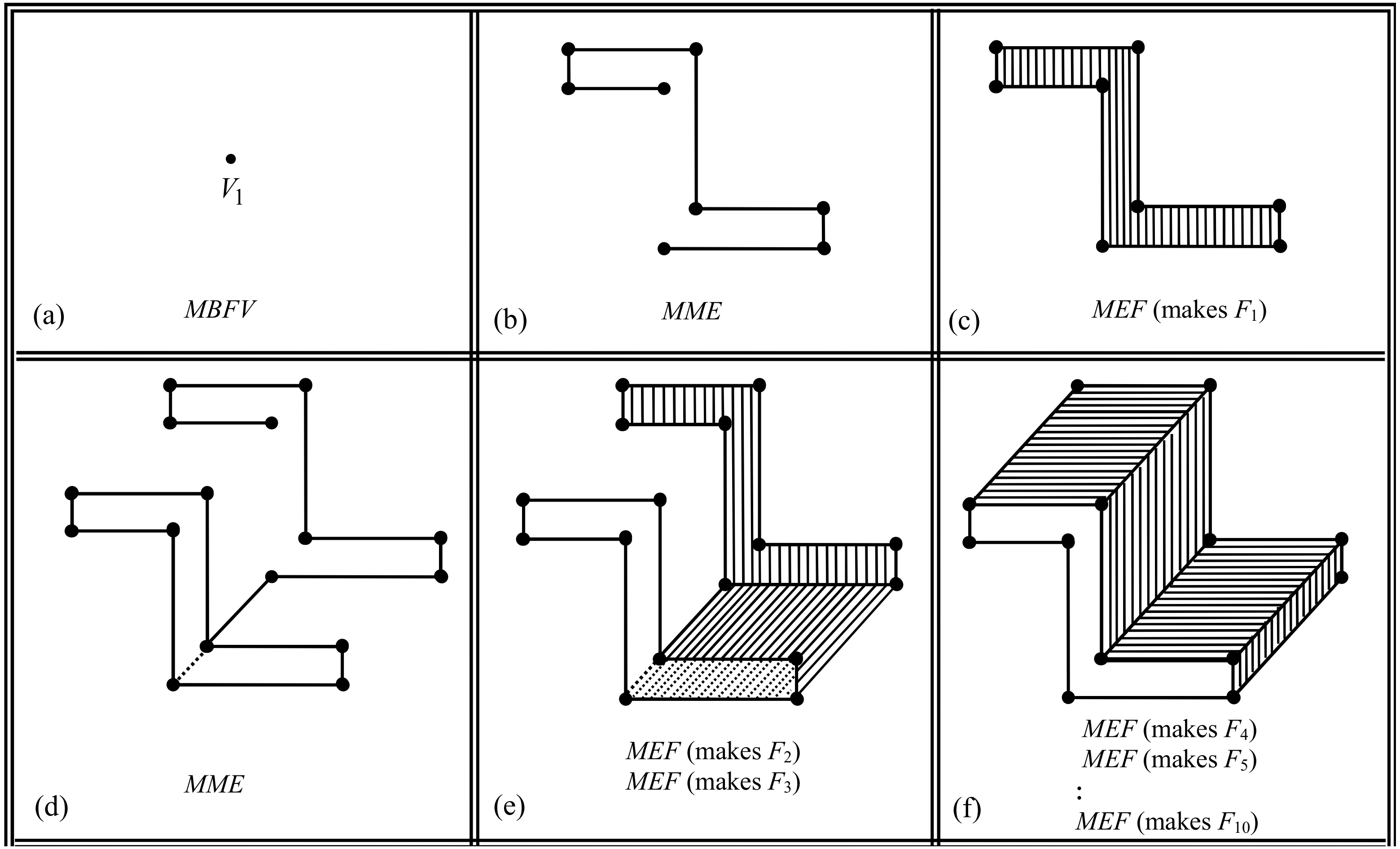


(a) (b)

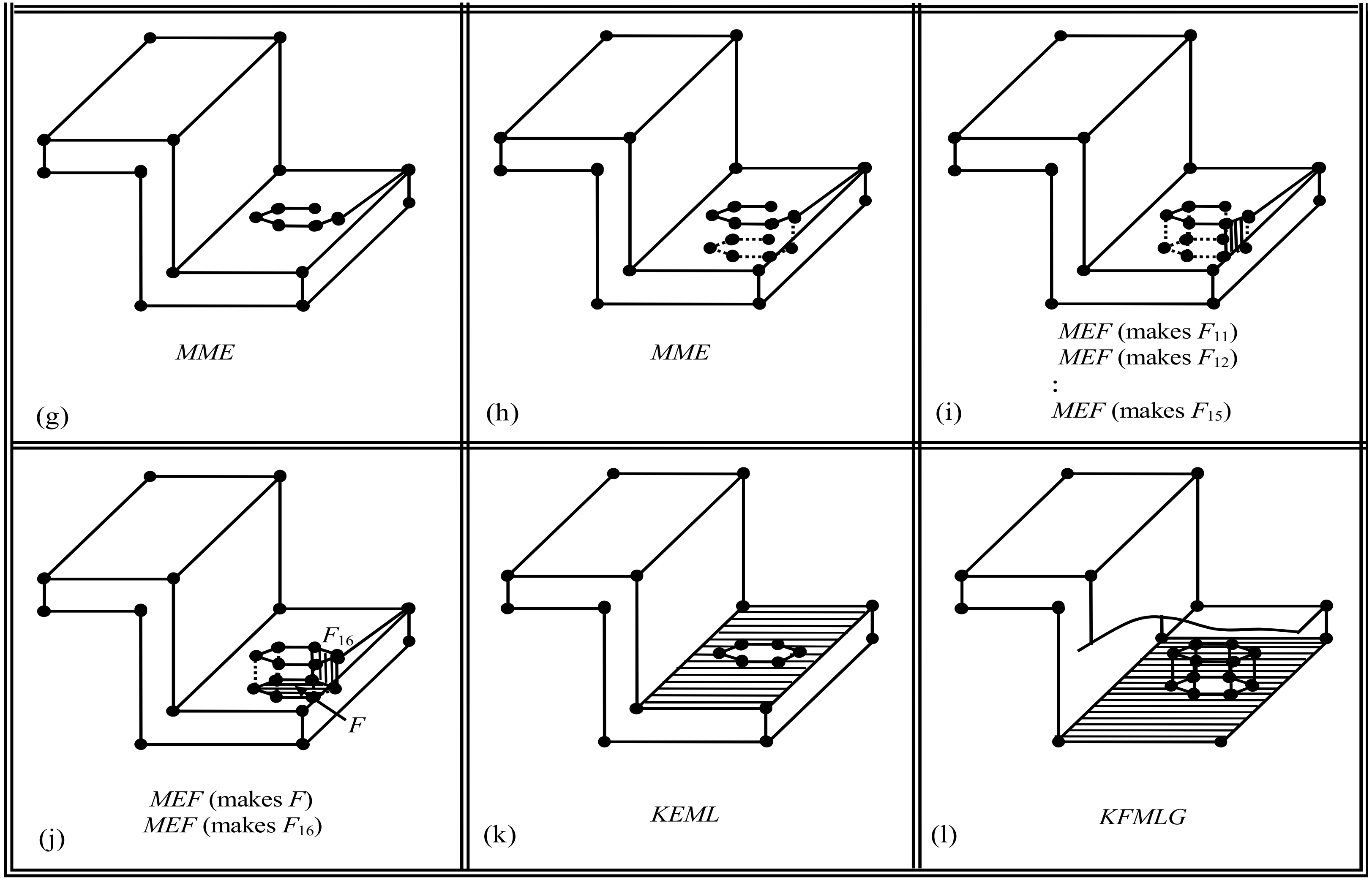
(a) Simple polyhedral object (b) boundary model of solid *S*

BOUNDARY REPRESENTATION OF SOLID MODELS

Fig. shows the sequence of steps involved for creating the boundary model of solid S through the steps from (a) to (l)



BOUNDARY REPRESENTATION OF SOLID MODELS



COMPUTER AIDED DESIGN
(BME-42)

Unit-IV: 3D Graphics

(7 Lectures)

- Introduction, Wireframe modeling, Surface modeling,
- Polygon surfaces-polygon meshes, polygon equations,
- Quadric and Superquadric surfaces, Blobby objects,
- Solid modeling-
  - Boolean set operations,
  - regularized set operations,
  - Primitive instancing,
  - Sweep representation-translational, rotational and hybrid sweeps,
  - Boundary representation-topology, geometry, boundary models
  - **Constructive solid geometry-unbounded and bounded primitives**

Lecture 32

Topics Covered

Constructive Solid Geometry

Topology and Homeomorphism

Prepared By

Prof. S. K. SRIVASTAVA

MED, MMMUT, Gorakhpur (UP)
sksme@mmmut.ac.in

CSG Primitives

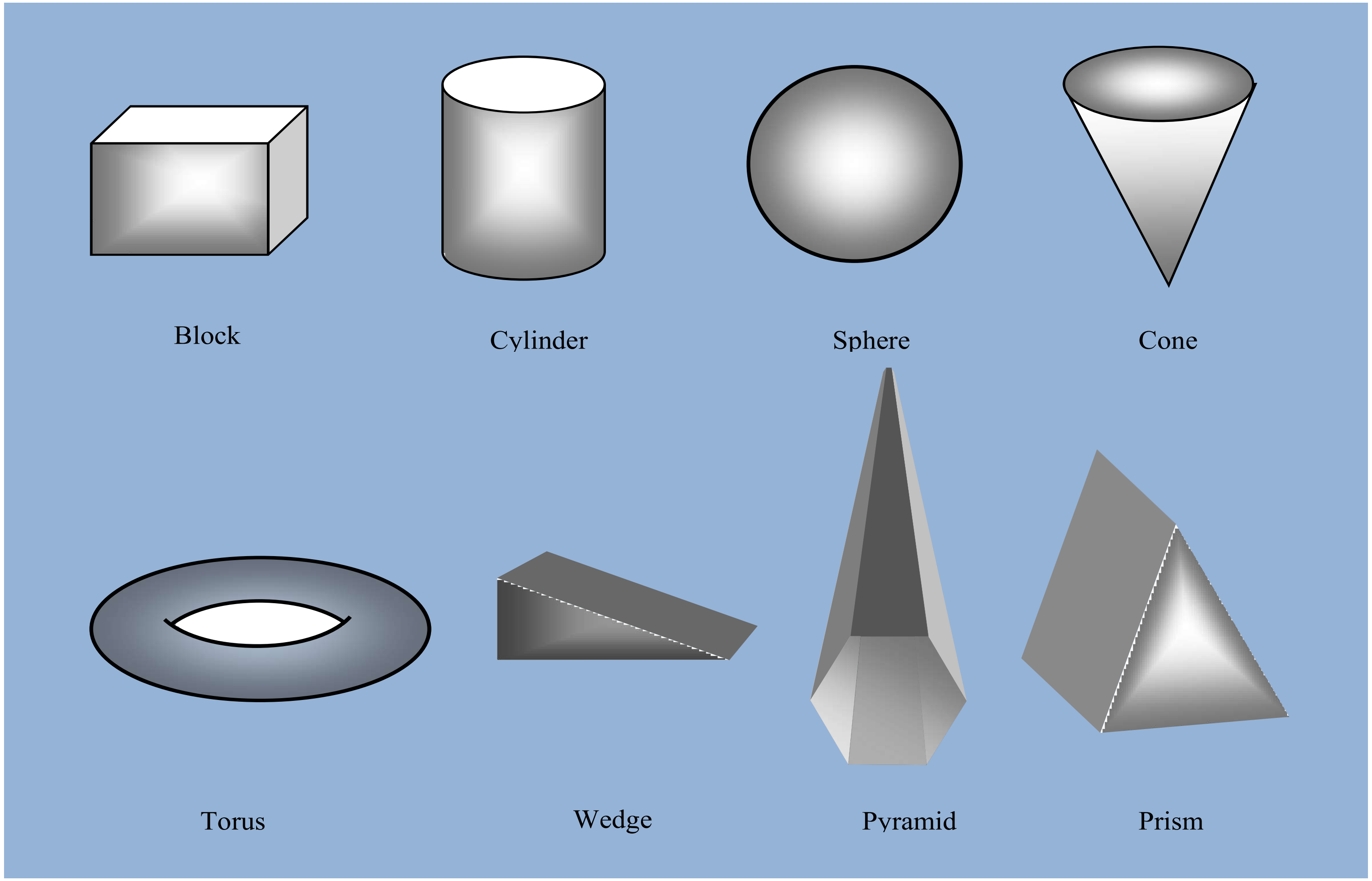
- Similar to B-rep, CSG method is another most popular scheme to create the solid models of physical objects.
- The CSG method is best understood so far for creating a solid model.
- This scheme is often called as building block approach of creating a solid model.
- The building blocks for CSG model are primitives such as blocks, cylinders, spheres, cones, etc.
- The three-dimensional primitives, which are considered themselves as valid solid models, are positioned properly before combining in certain order to create the physical object, following a set of regularized Boolean operations such as Union ($\cup^*$), Intersection ($\cap^*$) and Difference ($-^*$).

CONSTRUCTIVE SOLID GEOMETRY…

CSG Primitives…

- An object is stored as a tree with Boolean operators at the internal nodes and simple primitives at the leaves.
- Each primitive is bounded by the set of surfaces usually closed and orientable.
- A CSG model does not explicitly store the boundary model primitives such as vertices, edges, faces, etc.; instead, it can be calculated via the boundary evaluation process whenever needed by the application software.

CONSTRUCTIVE SOLID GEOMETRY…



Typical CSG Primitives

UNBOUNDED HALF-SPACES

- The creation of solid model using *unbounded half-spaces* of CSG solid modeling scheme is less popular in computer graphics.
- Half-spaces are lower-level primitives and form a basic representation scheme for creating the bounded solids.
- The regularized Boolean set operations combine the half-spaces for creating various models.
- Half-spaces are usually unbounded geometric entities; each divides the representation space into two infinite portions, one side filled with *material* and other side is kept *empty*.
- The most popular unbounded half-spaces are planar, cylindrical, spherical, conical and toroidal.
- Each half-space is defined in terms of the half-space (local) coordinate system (x_H, y_H, z_H) and a point in half-space is defined as a regular point set of ordered triplets (x, y, z) .

Fig. shows few unbounded half-spaces, which are mathematically represented as

Planar half-space

$H = \{(x, y, z) : z < 0\}$, (x, y, z) is a point in half-space such that $z < 0$ (Fig. a). The corresponding surface of half-space is $z = 0$. This means that there is a *material* when $z < 0$ and the space is *empty* for $z > 0$.

Cylindrical half-space

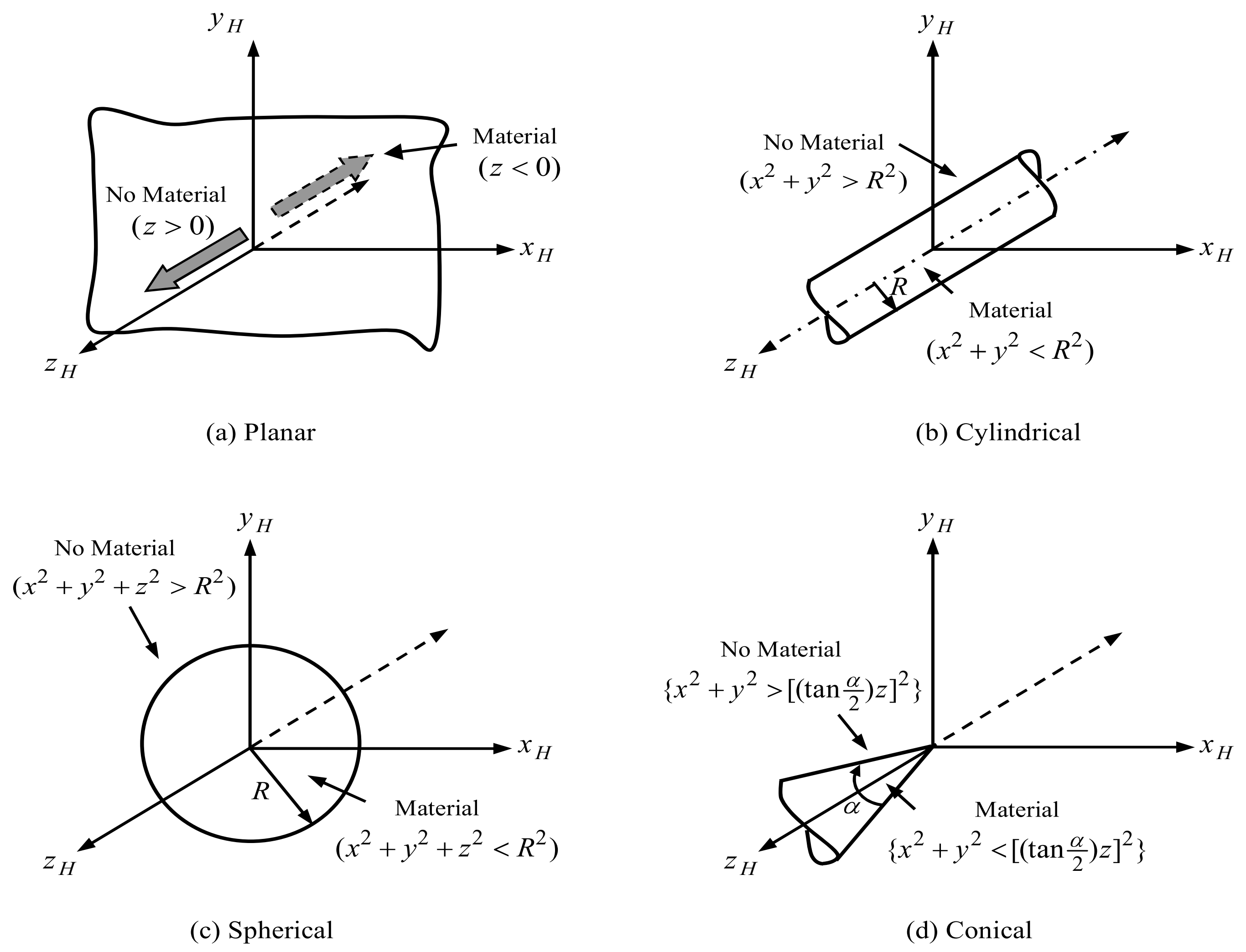
$H = \{(x, y, z) : x^2 + y^2 < R^2\}$, where $R > 0$ and (x, y, z) is a point inside the cylindrical half-space (Fig. b).

Spherical half-space

$H = \{(x, y, z) : x^2 + y^2 + z^2 < R^2\}$, where $R > 0$ and (x, y, z) is a point inside the spherical half-space (Fig. c).

Conical half-space

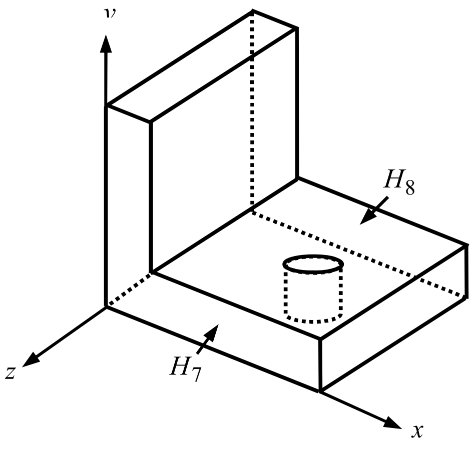
$H = \{(x, y, z) : x^2 + y^2 < [(\tan\frac{\alpha}{2})z]^2\}$, where $R > 0$, $0 < \alpha < \pi$ and (x, y, z) is a point inside the conical half-space (Fig. d).



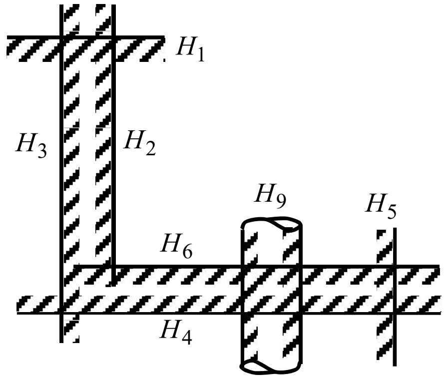
Unbounded half-spaces

HALF-SPACE REPRESENTATION OF SOLIDS

Fig. b shows the half-space representation of sold S .



(a)



(b)

- The half-space representation requires a total of 9 half spaces.
- Out of nine half-spaces, there are eight planar half-spaces represented from H_1 to H_8, and one cylindrical half space represented as H_9 .
- Using the local coordinate system, some half- spaces have to be positioned first by applying the geometrical transformations, e.g., rotate planar half space by an angle -90^o (ccw) about the x-axis and translate it up to obtain H_1 .

HALF-SPACE REPRESENTATION OF SOLIDS

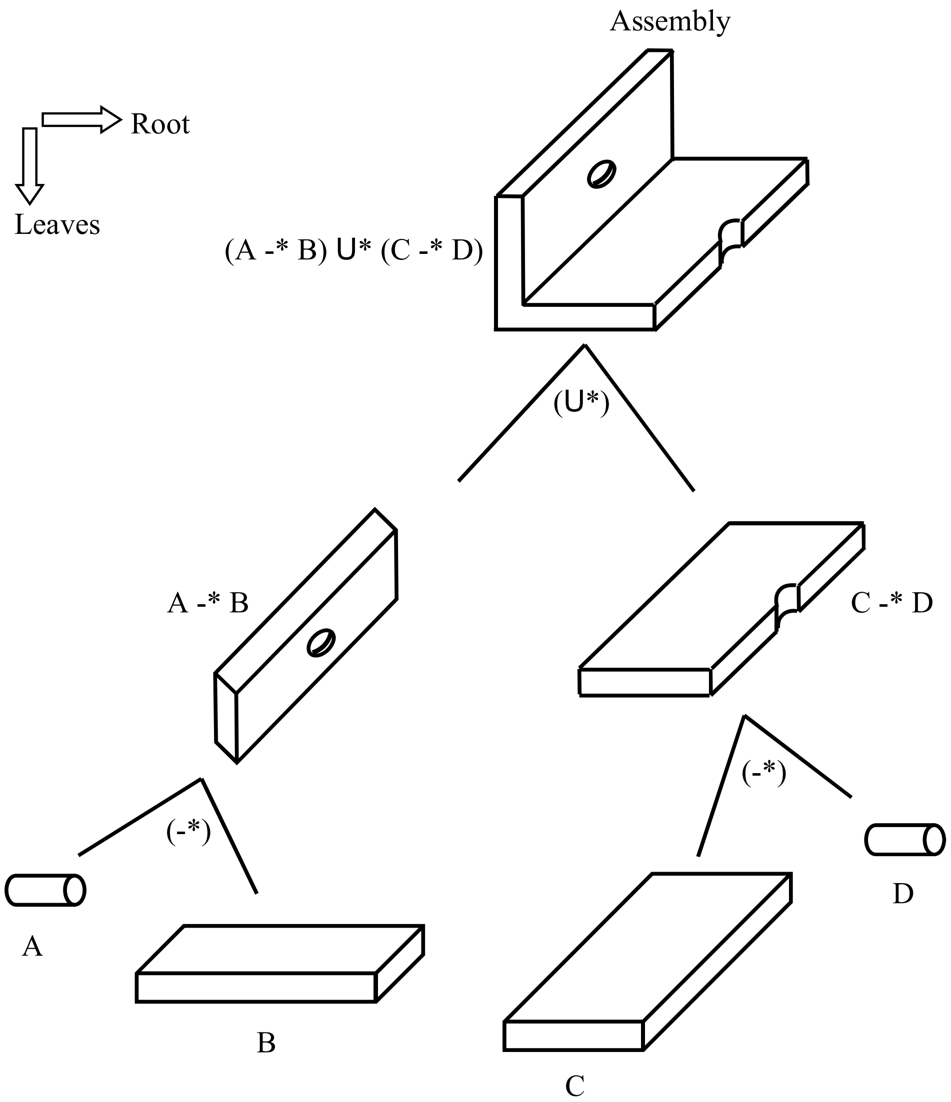
- In a similar way, the other planar half-spaces can be positioned using the proper set of geometric transformations.
- Only planar half-space H_7 does not require any positioning before applying the Boolean operations.
- The cylindrical half-space H_9 first requires translation along the x-axis, and then rotation by 90^o about the x-axis.
- Fig. a shows that desired solid S can be obtained by the union of half-spaces H_1 to H_8 followed by the subtraction of cylindrical half-space H_9.

,

BOUNDED SOLID PRIMITIVES

- In CSG scheme, simple three-dimensional bounded solid primitives combine using the regularized Boolean operators.
- An object is stored as a tree with operators at the internal nodes and simple primitives at the leaves.
- Some nodes represent Boolean operators, whereas others perform geometric transformations to correctly position the objects before modeling.
- For example, scaling for right dimensions, rotation for correct orientations, translation for correct positions, etc.
- Fig. shows an object designed with regularized Boolean set operations represented by an inverted binary tree.
- To determine the physical properties of the assembly (root) or to make the objects, we must be able to combine the physical properties of leaves objects.
- The complexity of this task depends upon the way of representation and storage of the leaf objects.

BOUNDED SOLID PRIMITIVES



Object defined by CSG using inverted binary tree

BOUNDED SOLID PRIMITIVES

- In fact, bounded solid primitives may be considered as composite half-spaces, and surfaces of primitives may be considered as the surfaces of the corresponding half-spaces.
- Mathematically, each bounded primitive is defined as a regular point set of ordered triplets (*x, y, z*).
- Fig. shows two-dimensional illustrations of the half-spaces of bounded primitives. For these primitives, the point sets are given by

Block

$H = \{(x, y, z) : 0 < x < a, 0 < y < b, \text{and } 0 < z < c\}$, (x, y, z) is a point in the block (Fig. a). A block is the regularized union of six intersecting planar half-spaces. Fig. a does not depict the two planar half-spaces along the z_L-axis. Each of these half-spaces is given by one limit of the three inequalities for x, y and z coordinates.

Cylinder

$H = \{(x, y, z) : x^2 + y^2 < R^2 \text{ and } 0 < z < L\}$, $R > 0$ and (x, y, z) is a point inside the cylinder (Fig. b). A cylinder is the regularized union of two planar half-spaces and one cylindrical half-space.

BOUNDED SOLID PRIMITIVES

COMPUTER AIDED DESIGN
(BME-42)

Unit-IV: Color Models

(7 Lectures)

- **Coloring in computer graphics,**
- **RGB, CMY and YIQ Color models**
- HSV and HLS color models

Lecture 33

Topics Covered

Coloring in Computer Graphics

Color Models

RGB Color Model

CMY Color Model

YIQ Color Model

Prepared By

Prof. S. K. SRIVASTAVA

MED, MMMUT, Gorakhpur (UP)

sksme@mmmut.ac.in

COLORING IN COMPUTER GRAPHICS

- Color graphics capability is another powerful tool, which helps designer to classify components in an assembly or highlight the sectional views and dimensions of the model.
- *Colors* and *textures* are the two main ingredients of shaded images produced by the shading algorithms.
- Colors are used for the aesthetics, realism and to highlight the associated areas in the geometry of an object.
- Colors are effectively used to distinguish various wireframe, surface or solid entities.
- The color contours display the stress distribution or heat flux variation resulting from finite element analysis.
- Black and white raster displays provide *achromatic* colors (black, various levels of gray scales and white), whereas color displays provide *chromatic* color.

COLORING IN COMPUTER GRAPHICS…

- The *intensity* and *amount of color* are the only parameters to control the achromatic light. For example, binary digits 0 and 1, respectively, refer black and white colors.
- In between the two extreme binary values, we have various *gray* scales, e.g., 0.5 is assigned for medium gray, i.e., mixture of 50% black in white.
- Multiple planes, e.g., 8 planes produce 2^8 or 256 different levels of *gray* scale for each pixel.
- *Chromatic* colors produce more pleasing effects on the human vision systems; however, they are more complex and very difficult to generate.
- A single-color image is obtained by combining three primary-colored images. Each colored image is generated by striking the electron gun on the phosphor coated glass screen.
- Red, Blue and Green primary colors are typically used to generate wide range of pleasant colors.

COLORING IN COMPUTER GRAPHICS…

There are three colors parameters namely *hue, saturation* and *brightness*.

Hue

- A light source (sun or light bulb) emits all frequency within visible range to produce the white light.
- When white light is incident upon an object, some frequencies are reflected while others are absorbed.
- The combination of frequencies of reflected light decides the color, which human eye perceives.
- For example, low frequencies dominant reflected light results into *red* color; alternatively, we say that the perceived light has a dominant frequency at the red end of color spectrum (VIBGYOR).
- Similarly, high frequency dominant reflected light results into *violet* color. The dominant frequency is also called as *hue or color* of the light.

Hue…

- Hue (*dominant frequency*) represents the color position in a color spectrum.
- Typically, color models that describe combinations of light in terms of the dominant frequency (hue), employ three primary colors to obtain wide range of colors, called as *color gamut* of that model.
- Normally, *two or three* colors are used as primary colors in a color model. For example, *Red*, *Blue* and *Green* are the hue names, i.e., three primary colors.

Saturation

- *Saturation (or purity)* describes the dilution of a color by white light. Purity describes how washed out or *how pure* the color of light appears.
- Pure red, blue and green colors are *saturated* colors whereas gray levels are *desaturated* colors.
- *Pastels* (soothing) and *pale* (whitish to being colorless) colors are specified as less pure.

Brightness

- Brightness represents *intensity level* (or lightness) of the perceived color light.
- Intensity is the radiant energy emitted per unit time, per unit solid angle, and per unit projected area of the source.
- Radiant energy is related to the *luminance* of a source.

COLOR MODELS

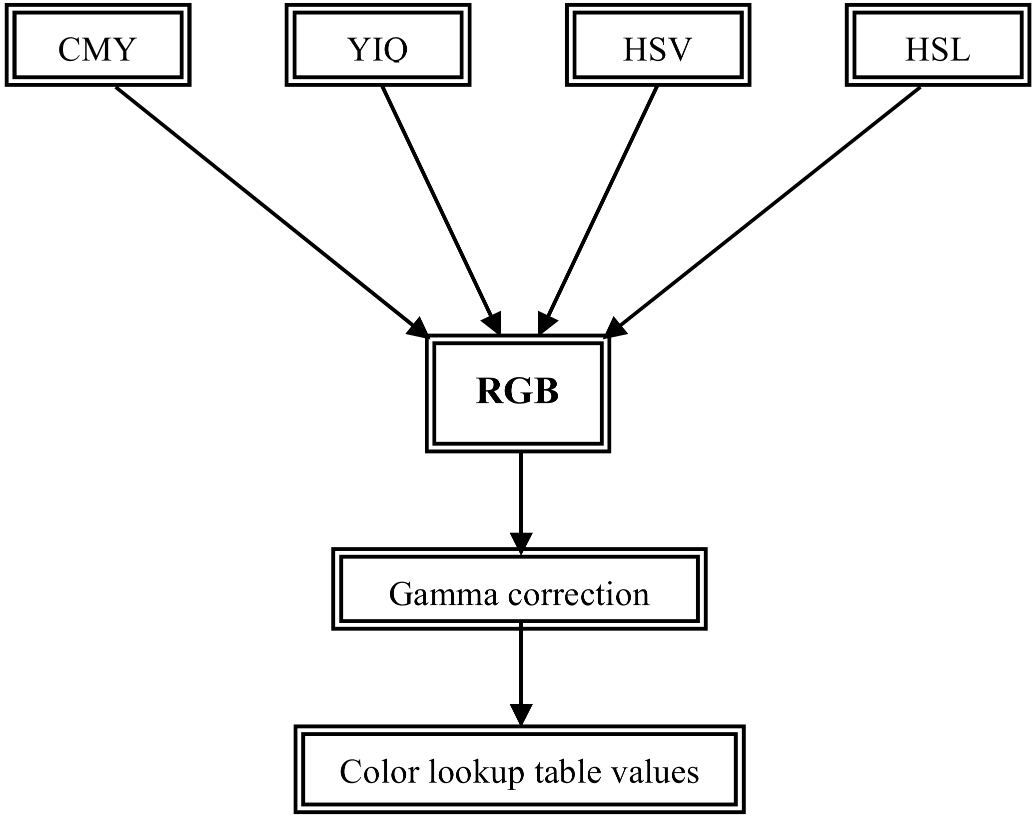
- A color model is an *orderly* system for creating wide range of colors from small range of primary colors.
- Each color is represented by a point in three-dimensional color coordinates system to permit whole range of color specifications within some color range.
- This model is helpful in understanding how color is represented on the video monitor.
- Broadly, there are two color models: *additive* and *subtractive.*
- Additive color models use *light* while subtractive color models use printing *inks* to display a color.
- Additive color models are based on the principle of *transmitted* light while subtractive color models perceive the color as a result of *reflected* light.
- A color model is a method for explaining the properties or behavior of color within some particular context. .

COLOR MODELS

- No finite set of real primary colors can be combined to produce all possible visible colors.
- No single color model can explain all aspects of color; therefore, different models describe different perceived characteristics of colors.
- There are large numbers of color models. Three primary colors are sufficient for most purposes.
- In the following, some popular color models, based on **R**ed, **B**lue and **G**reen colors (RGB models), are discussed.
- For any one of these models, the color coordinates are translated into three voltages to control the display, follow the transformation sequence as shown in Fig.

The gamma correction establishes a linear relationship between the digital RGB values, and the intensity of light emitted by the CRT monitor.

COLOR MODELS



Transformation Sequence of a color model to RGB primaries

RGB COLOR MODEL

- The RGB color model is the most popular model, based on Cartesian coordinate system, as shown in Fig. (a).
- The model can be represented with the help of a unit cube defined on Red, Blue and Green axes in which origin (0, 0 0) represents the black and the vertex coordinate (1, 1, 1) represents white.
- Thus, lowest intensity (0 for each color) produces the black color and highest intensity (1 for each color) produces the white color.
- The principal diagonal of the cube is locus of equal amount of primary colors; therefore, represents gray shades (black color corresponding to maximum gray level and white color corresponding to minimum gray level).
- A gray shade, halfway between black and white, is represented as (0.5, 0.5, 0.5).
- Vertices of the cube on color axes represent three primary colors, and the remaining vertices represent the complementary colors corresponding to each primary color.

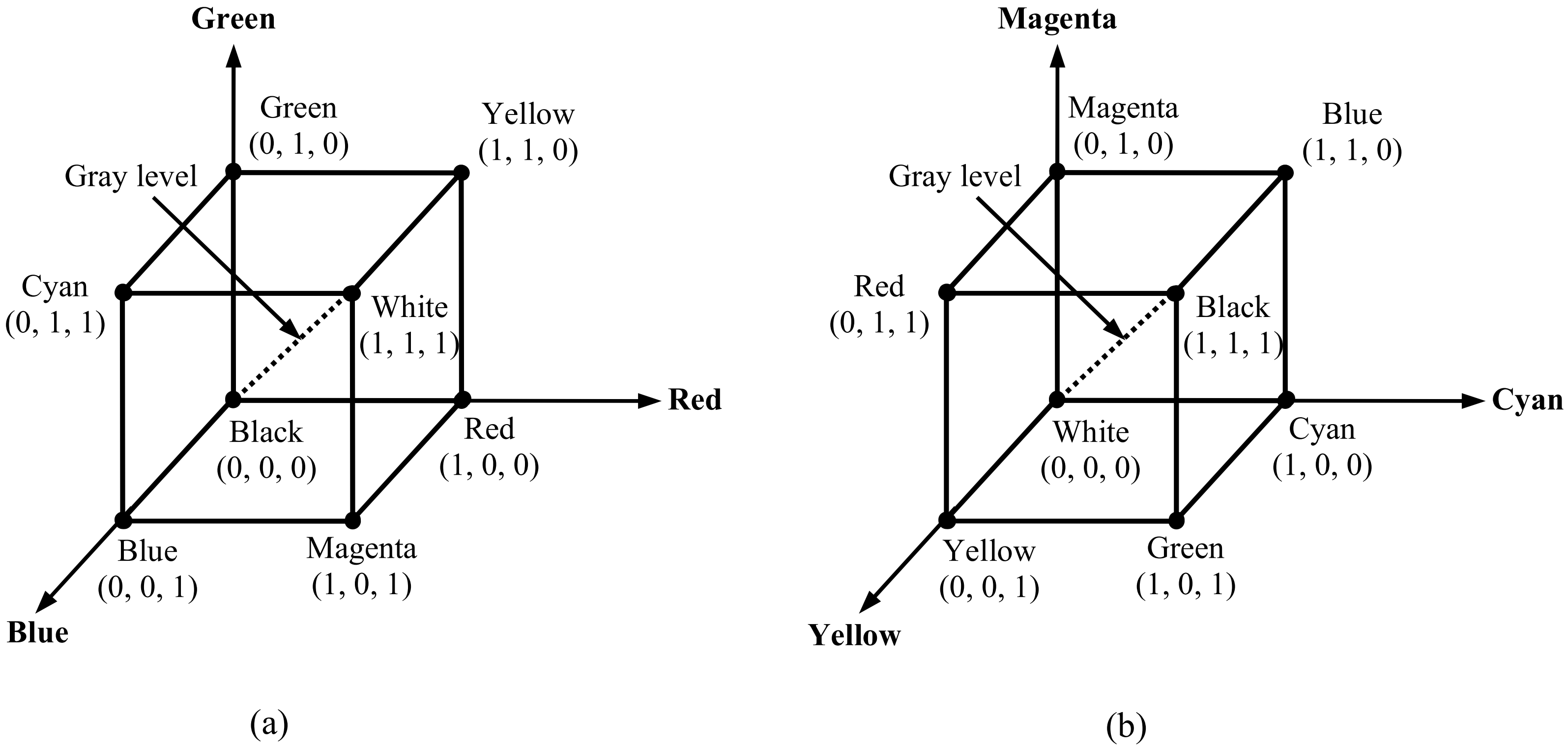
RGB COLOR MODEL

- Any point in space is obtained from the three primary colors: Red, Blue and Green; therefore, the space represents *additive* color model.
- Intensities of primary colors are added to produce other colors.
- Each color point in space is represented by color triplet (R, G, B) where each primary color is assigned a range from 0 to 1.
- The other colors are obtained by the combinations of primaries as follows:
  - *Magenta* vertex is obtained by adding *red* and *blue* colors to produce a color triplet $(1, 0, 1)$.
  - *Cyan* vertex is obtained by adding *green* and *blue* colors to produce a color triplet $(0, 1, 1)$.
  - *Yellow* vertex is obtained by adding *red* and *green* colors to produce a color triplet $(1, 1, 0)$.

Properties

- Additive color model
- Used for computer displays
- Uses light (sun or light bulb) to display the color
- Colors obtained from the *transmitted* light
- Red + Green + Blue = White

CMY COLOR MODEL



(a) RGB (additive) color model (b) CMY (subtractive) color model

CMY COLOR MODEL

- The CMY (*Cyan*, *Magenta*, *Yellow*) color model is the complement of RGB color model.
- Fig. (b) shows a unit cube representation of CMY model.
- Cyan, magenta and yellow (CMY) colors are the complements of red, green and blue (RGB) primaries, respectively.
- In CMY model, the white is at the origin (0, 0, 0) and black is at the vertex (1, 1, 1) which is opposite to the RGB color model.
- Like the RGB model, equal amounts of each primary color produces gray levels along the principal diagonal of the unit cube.
- CMY color model describes color output to hard copy devices.
- Unlike video monitors (using light), hard copy devices (using ink) such as plotters produce color images by coating a paper with the color pigments.

CMY COLOR MODEL

- Because the colors are seen by the *reflected light*; therefore, this represents a *subtractive* color model.
- For example, red color is obtained by subtracting a cyan color from the white light.
- Other colors are obtained by a similar subtractive process.
- The printing with CMY model is obtained with a collection of four ink dots: *cyan, magenta, yellow and black*.
- A separate black dot is used because combination of *cyan, magenta* and *yellow* inks typically produces dark gray color instead of a true black.

Mathematically, the conversion from CMY representation to RGB representation may be expressed in matrix form as

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where unit column vector represents white in RGB color model.

CMY COLOR MODEL

Similarly, we convert from CMY color representation to an RGB representation in matrix form as

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

where unit column vector represents black in CMY color model.

Properties

- Subtractive color model
- Used for printing colored image
- Uses inks to display the color
- Colors obtained from the reflected light
- Cyan + Magenta + Yellow = Black

YIQ COLOR MODEL

- RGB monitor requires *separate signals* for red, green and blue components of an image whereas a *television monitor* uses single composite color obtained from YIQ model.

- The y-axis of color model represents the *luminance* (brightness) information while chromaticity information (hue and purity) is represented by I and Q parameters.

- The I axis represents chrominance information (orange-cyan hue information), and occupies a bandwidth of approximately 1.5 MHz. Parameter Q carries chrominance information (green-magenta hue information) in a bandwidth of about 0.6 MHz.

- The RGB signal can be converted into YIQ values (television signal) using the matrix equation expressed as

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ COLOR MODEL

The television signal can be converted to RGB signal with the inverse matrix transformation from above eqn. as

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.620 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.108 & 1.705 \end{bmatrix} . \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

COMPUTER AIDED DESIGN
(BME-42)

Unit-IV: Color Models

(7 Lectures)

- **Coloring in computer graphics,**
- **RGB, CMY and YIQ Color models**
- HSV and HLS color models

Lecture 34

Topics Covered

Coloring in Computer Graphics

Color Models

RGB Color Model

CMY Color Model

YIQ Color Model

Prepared By

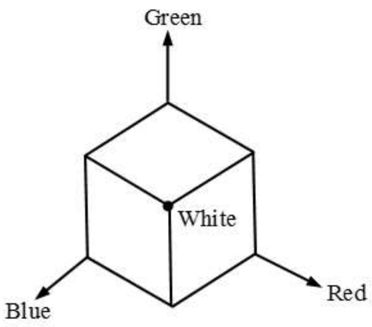
Prof. S. K. SRIVASTAVA

MED, MMMUT, Gorakhpur (UP)

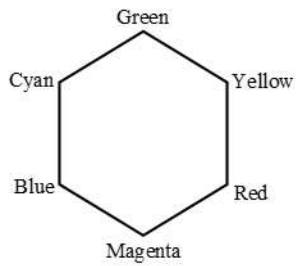
sksme@mmmut.ac.in

HSV COLOR MODEL

- The HSV color model uses color descriptions that have more intuitive appeal to the user.
- This is user oriented because it depends on the user, who produces color based on the color parameters: *Hue*, *Saturation* and *Value* (HSV).
- Different *shades* and *tones* of colors are obtained by adding white and black to the selected spectral color.
- The three-dimensional HSV color representation is derived from RGB model.
- If RGB cube is viewed along the principal diagonal from the white vertex to the black vertex (Fig. a), the outline of hexagon (Fig. b) represents the various hues, which is used as a top of hexagon (Fig. c).



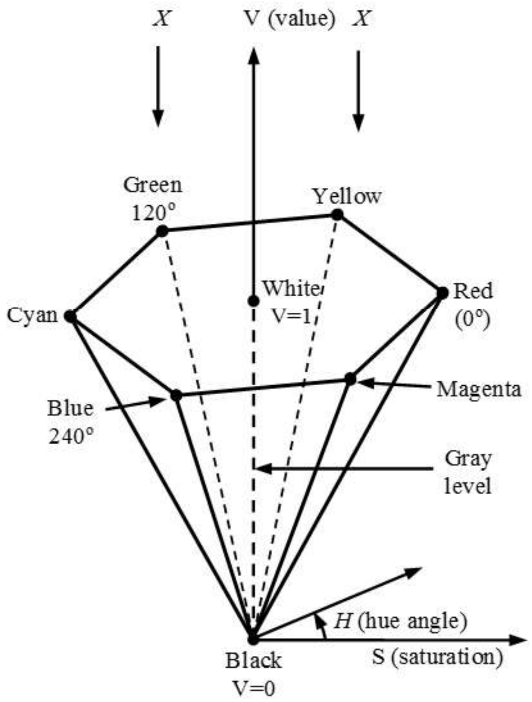
(a)



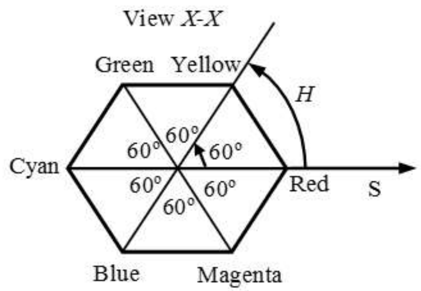
(b)

HSV COLOR MODEL

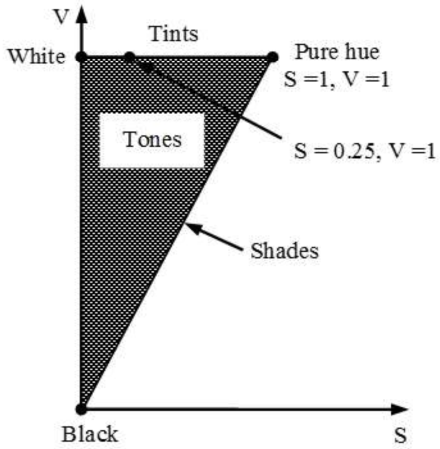
- In hexacone, saturation is measured along the horizontal axis, and value is along the vertical axis through the centre of hexacone.
- Fig. d shows the hexacone viewed from *X-X* direction. Hue (color) is represented about the vertical axis ranging from 0° (Red) through 360° at an interval of 60° with Yellow at 60°, Green at 120°, Cyan at 180°, Blue at 240° and Magenta at 300°. The complementary colors are separated by the angle 180°.



(c)



(d)

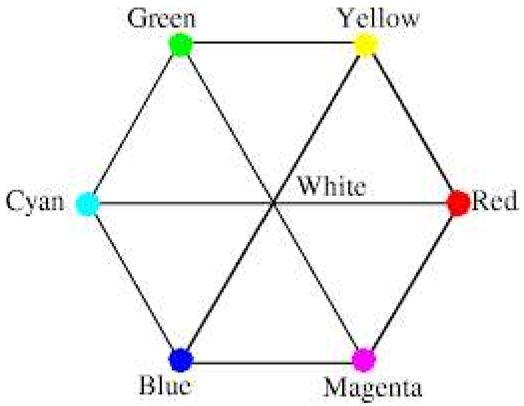


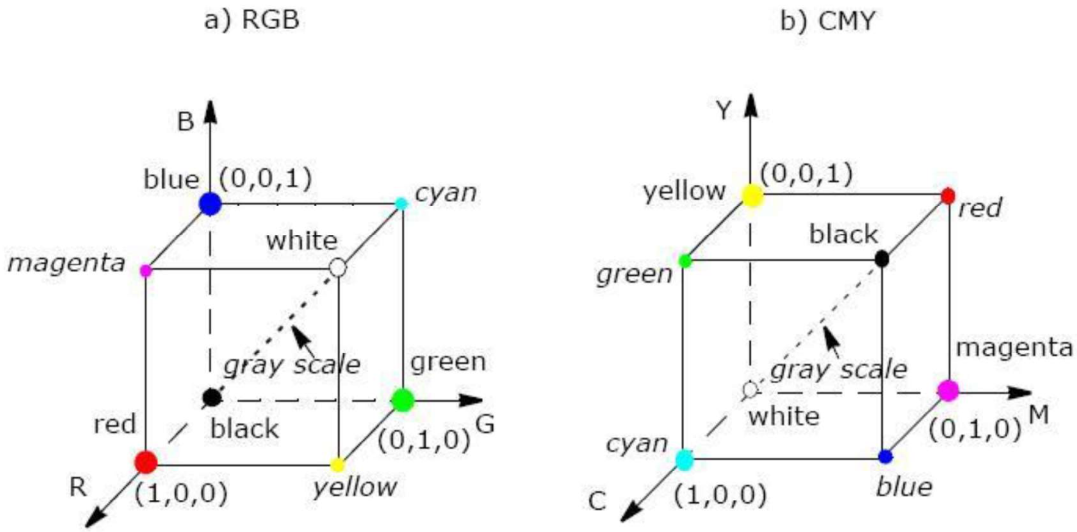
(e)

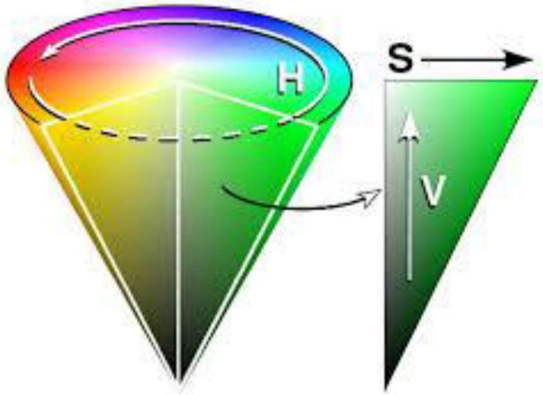
HSV COLOR MODEL

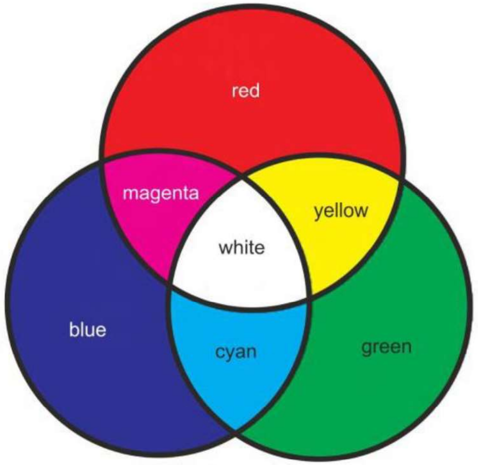
- The color concepts associated in terms of shades, tints, and tones are represented in a cross-sectional plane of the HSV hexacone (Fig. 9.51e). The human eye can distinguish about 128 different hues and about 130 different tints, i.e., saturation levels. For each combination, number of shades can be obtained corresponding to a particular hue. Saturation S, defined as the ratio of purity of a selected hue to its maximum purity, ranges from 0 to 1. The hexacone is one unit high in V, with apex at the origin. The parameter V ranges from apex (0 represents minimum intensity leading to black) to top of hexacone (1 represents maximum intensity leading to white). At V = 0, the values of H and S are irrelevant. The top of the hexacone corresponds to V = 1, which contains relatively bright colors.

HSV COLOR MODEL









RGB color model

THANK YOU