UNIT – 3

Mathematical

Contents

Important Repeated Questions:

1. **Differentiate between/Compare Constructive Solid Geometry (CSG) and B-Representation (B-Rep).** (W25 - Q2b, 04 marks) (S23 - Q4a, 03 marks) (S25 - Q2a, 03 marks)
2. **Explain Homogeneous Coordinate system and its importance/advantages.** (W25 - Q3a, 03 marks) (W22 - Q3a OR, 03 marks) (S25 - Q3a OR, 03 marks)
3. **Write/Explain transformation matrices (Rotation, Reflection, Shearing, etc.) in 2D/3D using homogeneous coordinates.** (W25 - Q3a OR, 03 marks) (W22 - Q4b, 04 marks) (S24 - Q5a, 03 marks)
4. **Differentiate between/Compare Scaling and Shearing transformation.** (W25 - Q3b, 04 marks)
5. **Find the new coordinates of a shape (triangle/rectangle) after applying a series of transformations (translation, rotation, scaling, reflection).** (S24 - Q3c OR, N/A marks) (W22 - Q3c OR, 07 marks) (W23 - Q3c, N/A marks) (W24 - Q3c OR, N/A marks)
6. **Obtain the mirror reflection of a triangle/polygon about a given line.** (W25 - Q3c, 07 marks) (S25 - Q3c, 07 marks) (S22 - Q4c, N/A marks)
7. **Prove that two successive rotations/translations/scaling are commutative.** (W25 - Q3c OR, 07 marks)
8. **Prove that differential scaling and rotation are not commutative, but uniform scaling and rotation are commutative.** (S23 - Q4c, 07 marks)
9. **What do you understand by geometry and topology in solid modelling?** (S23 - Q1c, 07 marks) (W23 - Q5a OR, 03 marks)
10. **Explain Boolean operations for Constructive Solid Geometry (CSG).** (S23 - Q4b, 04 marks) (S25 - Q2b, 04 marks)
11. **Compare Wireframe, Surface and Solid modeling techniques.** (W22 - Q3b OR, 04 marks)
12. **Explain Feature based modeling.** (S24 - Q5c, N/A marks) (W24 - Q4c, 07 marks)
13. **List methods of geometric modeling. Explain Wire frame modeling.** (S23 - Q3b OR, 04 marks)

Legends: W- Winter, S- Summer, Q- Question and 03/04/07- Marks of Question

3.1 Wire Frame Modeling

- A geometric model of an object is created by using the two-dimensional geometric entities such as: points, lines, curves, circles etc.
- Very often, designers build physical models to help in the visualization of a design. This may require the construction of skeleton model using wires to represent the edges of an object or component.
- Wire frame modeling is used in computer aided engineering techniques and also to facilitate the production of various projected views to aid visualization.
- The model appears like a frame constructed out of wire hence it is called as WIRE FRAME model.
- It is simple to construct.
- It requires less computer memory for storage compare to other types of geometric models.
- The time required to retrieve, edit or update is less for wire-frame models compare to other types of geometric models.
- It is more ambiguous to interpret the wire-frame model (see *Fig.3.1*).

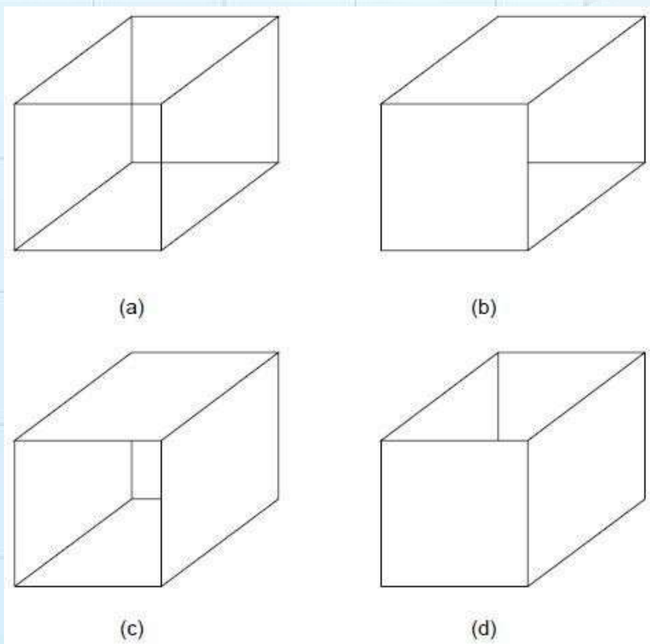


Fig.3.1 – Ambiguity in Wire frame Model

- Calculation of the properties, such as mass, volume, moment of inertia etc., is not possible.
- It is difficult and time consuming to generate wire-frame model for complicated objects.
- It requires more input data compare to others as it requires co-ordinates of each node and their connectivity.

3.2 Solid Modeling

- To eliminate all kinds of ambiguities in representation and manipulations of the objects, the solid modeler was developed. Out of the various approaches developed, one of them was an approach to the design of mechanical parts by treating them as combinations of simple building blocks like cylinders and cuboids.

- Such solid modelers or volume modelers can hold complete unambiguous representations of the geometry of a wide range of solid objects. The completeness of the information contained in a solid model allows the automatic production of realistic images of a shape and automation of the process of interference checking.
- Furthermore, interfaces can interrogate the model and extract lot of useful data. The model can also serve as a means of geometric input for finite element analysis or even manufacturing tasks such as the generation of instructions for numerically controlled machining.
- Solid modelers store more information (geometry and topology) than wireframe or surface modelers (geometry only). Both wireframe and surface models are incapable of handling spatial addressability as well as verifying that the model is well formed, the latter meaning that these models cannot verify whether two objects occupy the same space.
- Surface models provide a precise definition of surfaces and can handle complex geometries, they are slow to render, are computationally intensive and do not further CAD/CAM automation and integration goals. A shaded surface model is by no means considered a solid model.
- On the other hand, solid modeling produces accurate designs, provides complete three- dimensional definition, improves the quality of design, improves visualization and has potential for functional automation and integration.
- However solid modeling has some limitations. For example, it cannot automatically create other models from the solid definition and neither can it automatically use data created in other models to create a solid. In addition, solid modeling has not been proven for large-scale production applications.
- Other limitations such as slow rendering and computations as well as poor user interface are fading away with the rapid enhancement of both hardware and software.

3.3 Geometry and Topology

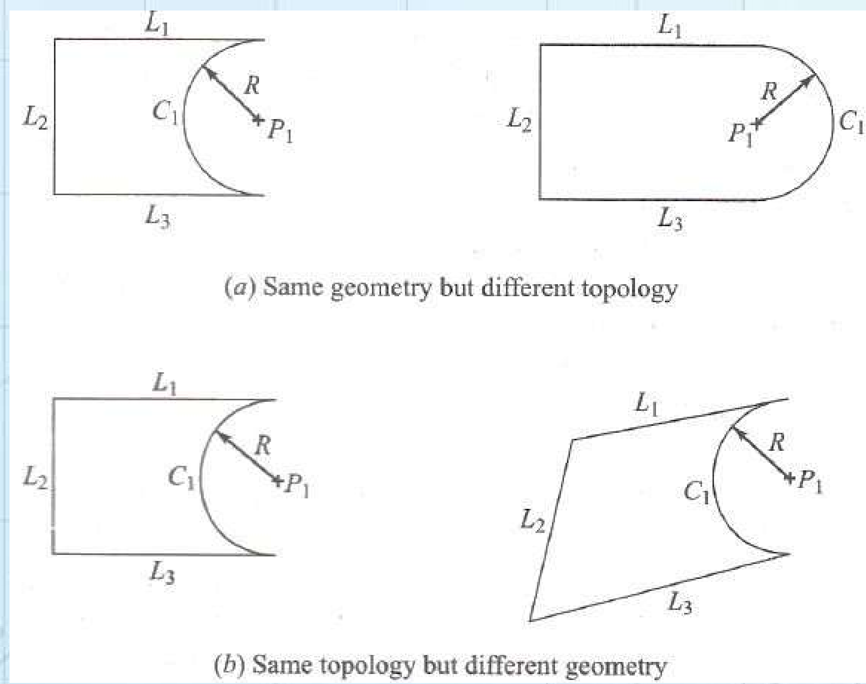


Fig.3.2 – Difference between Geometry and Topology of an Object

- The difference between geometry and topology is illustrated in *Fig.3.2*. Geometry (sometimes called metric information) is the actual dimensions that define the entities of the object.
- The geometry that defines the object shown in *Fig.3.2* is the lengths of lines L_1, L_2 and L_3, angles between the lines and the radius R and the center P_1 of the half-circle.
- Topology (sometimes called combinatorial structure), on the other hand, is the connectivity and associativity of the object entities.
- It has to do with the notion of neighborhood; that is, it determines the relational information between object entities.
- The topology of the object shown in *Fig.3.2*(b) can be stated as follows: L_1 shares a vertex (point) with L_2 and C_1, L_2 shares a vertex with L_1 and L_3, L_3 shares a vertex with L_2 and C_1, L_1 and L_3 do not overlap and P_1 lies outside the object. Based on these definitions, neither geometry nor topology alone can completely model objects.
- Wireframe and surface models deal only with geometrical information of objects and are therefore considered incomplete and ambiguous. From a user point of view, geometry is visible and topology is considered to be nongraphical relational information that is stored in solid model databases and are not visible to users.

3.4 Properties of Solid Model

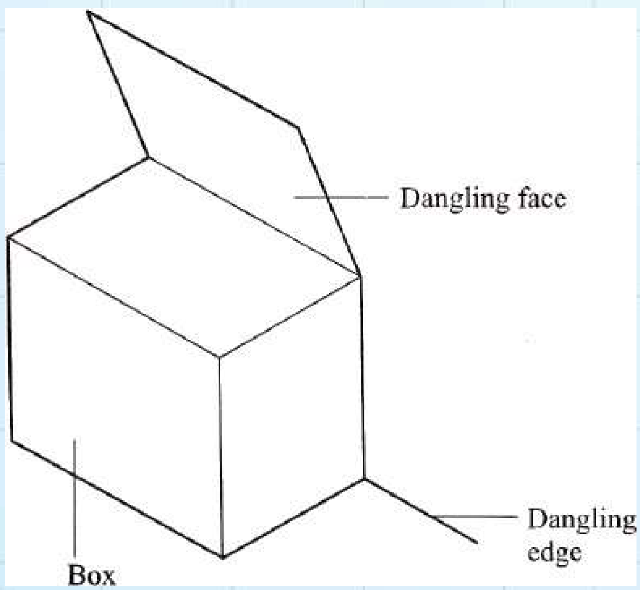


Fig.3.3

- The properties that a solid model or an abstract solid should capture mathematically can be stated as follows:
  1. **Rigidity** This implies that the shape of a solid model is invariant and does not depend on the model location or orientation in space.
  2. **Homogeneous three-dimensionality** Solid boundaries must be in contact with the interior. No isolated or dangling boundaries (see *Fig.3.3*) should be permitted.
  3. **Finiteness and finite describability** The former property means that the size of the solid is not infinite while the latter ensures that a limited amount of information can describe the

solid. The latter property is needed in order to be able to store solid models into computers whose storage space is always limited. It should be noted that the former property does not include the latter and vice versa. For example, a cylinder which may have a finite radius and length may be described by an infinite number of planar faces.

4. **Closure under rigid motion and regularized boolean operations** This property ensures that manipulation of solids by moving them in space or changing them via boolean operations must produce other valid solids.

5. **Boundary determinism** The boundary of a solid must contain the solid and hence must determine distinctively the interior of the solid.

3.5 Properties of Representation Scheme

The formal properties of representation schemes which determine their usefulness in geometric modeling can be stated as follows:

(1) **Domain:** The domain of a representation scheme is the class of objects that the scheme can represent or it is the geometric coverage of the scheme.

(2) **Validity**: The validity of a representation scheme is determined by its range, that is, the set of valid representations or models it can produce. If a scheme produces an invalid model, the CAD/CAM system in use may crash or the model database may be lost or corrupted if an algorithm is invoked on the model database. Validity checks can be achieved in three ways: test the resulting databases via a given algorithm, build checks into the scheme generator itself, or design scheme elements (such as primitives) that can be manipulated via a given syntax.

(3) **Completeness or Unambiguousness:** This property determines the ability of the scheme to support analysis and other engineering applications. A complete scheme must provide models with sufficient data for any geometric calculation to be performed on them.

(4) **Uniqueness** This property is useful to determine object equality. It is a custom in algebra to check for uniqueness but it is rare to do so in geometry. This is because it is difficult to develop algorithms to detect the equivalence of two objects and it is computationally expensive to implement these algorithms if they exist. Positional and permutational nonuniqueness are two simple cases shown in *Fig.3.4*. *Fig.3.4*(a) shows a two dimensional rectangular solid (of side lengths a and b) in two different positions and orientations. The two-dimensional solid S shown in *Fig.3.4*(b) is divided into three blocks A, B and C that can be unioned in a different order.

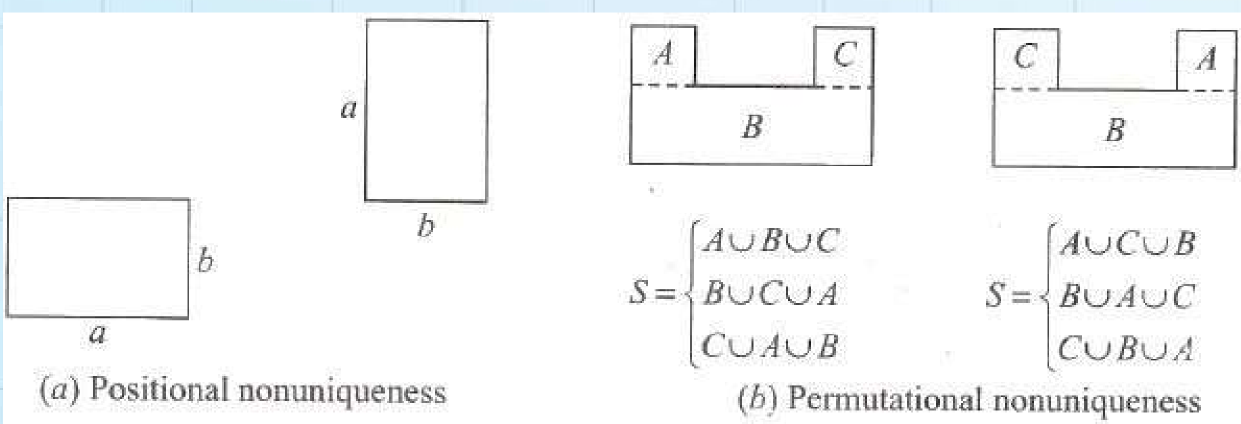


Fig.3.4

The informal properties of representation schemes for solid models

The primary considerations for storing the model in a computer are:

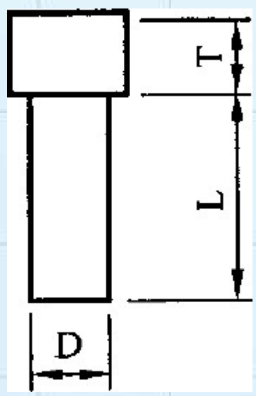
i. **Conciseness** (i.e. how much computer memory is occupied by the model of a shape).

ii. **Efficiency** (i.e. how much computer time is used in creating, interrogating or modifying the model of a shape.

- In general, there is a trade-off between the usage of memory and usage of time. For example, if we use up storage for 'remembering' the properties of a shape, then we do not need to spend a lot of time recalculating these properties every time one needs the information.
- Representation schemes where the elements of a shape are explicitly held in the model are termed as evaluated representations. Conversely, those where the elements must be calculated from implicit instructions for construction of the shape are known as unevaluated representations.
- Depending on the application, either of the above approaches can be used. Representation of solid in a computer can be divided into 6 general classes as follows:

1. Pure Primitive Instancing

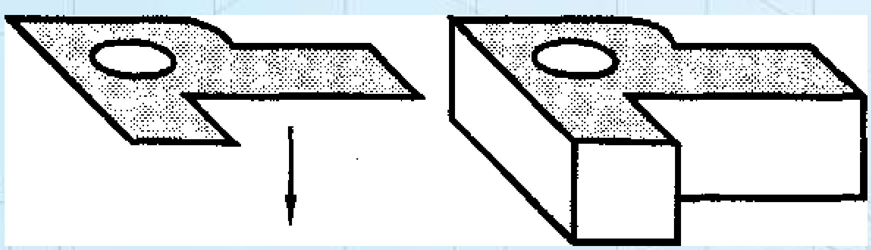
– This technique is used to define numerous families of objects, each defined parametrically as shown in *Fig.3.5*. A particular solid is specified completely by giving the family to which it belongs, together with a limited set of parameter values. However, such systems can only define a restricted range of objects which have been predefined in the system.



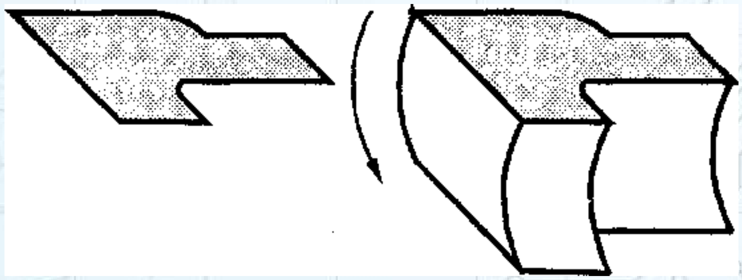
Pin parameters (D, T, and L)
Fig.3.5 – Pure Primitive Instancing

2. Generalized Sweeps

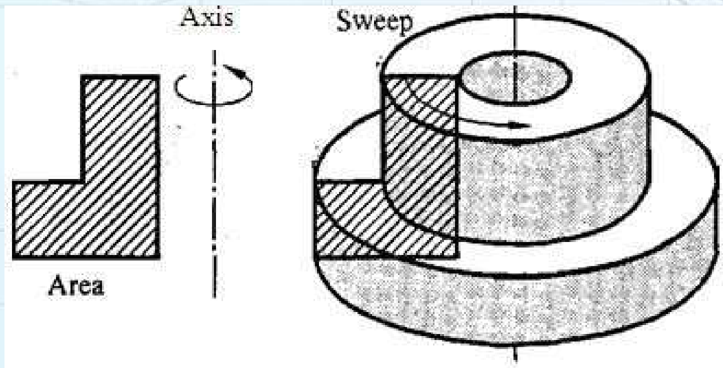
– A solid is defined in terms of volumes swept out by 2D or 3D lamina as they move along a curve. Some of the typical objects generated by this method are shown in *Fig.3.6*.



(a) Translational Sweep



(b) *Translational Sweep over Path*

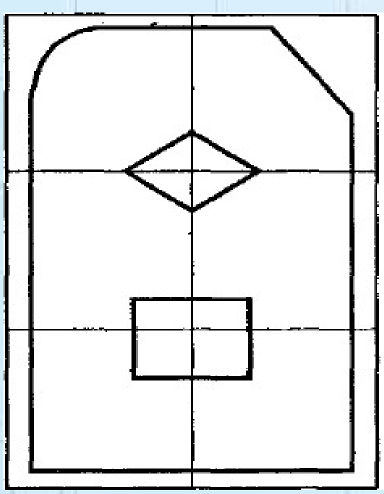


(c) *Rotational Sweep*

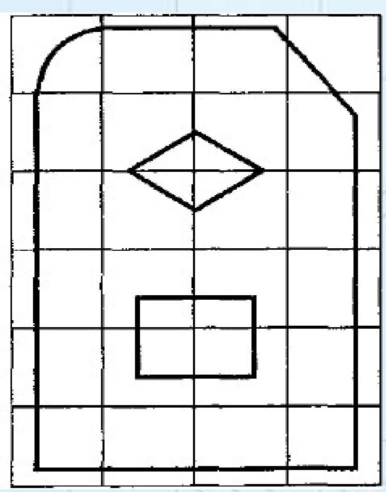
Fig.3.6 – Generalised Sweep

3. **Spatial Occupancy Enumeration**

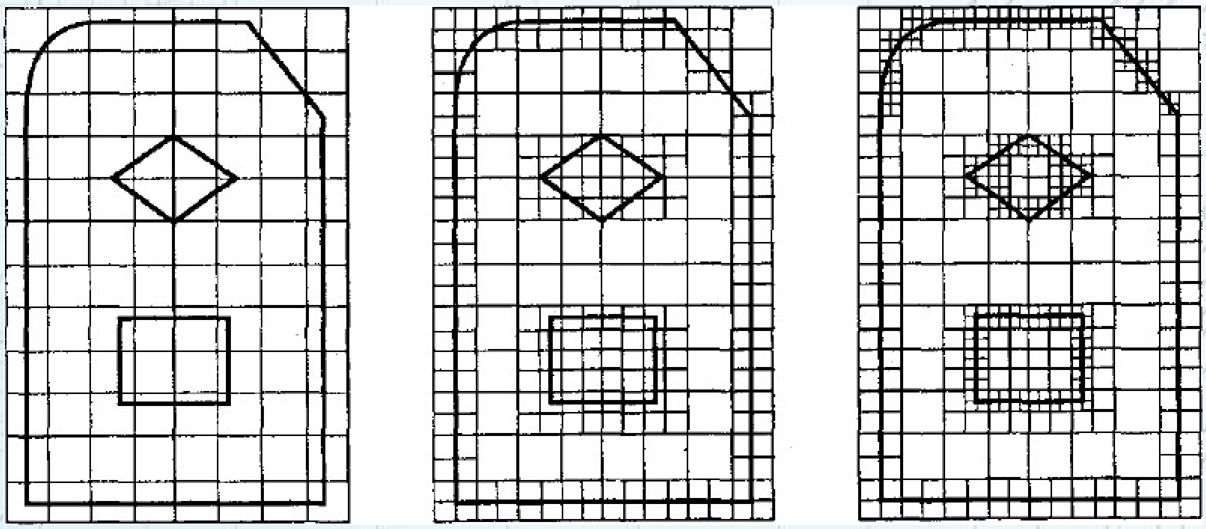
- Under this technique 3D objects are divided up into cubical cells at a particular resolution and objects are modeled by listing the cells that they occupy as shown in *Fig.3.7* (a) to (c).
- Smaller the size of the cube, more accurate would be the representation. This representation scheme requires large amounts of storage for reasonable resolution and thus has not been favored in practical systems.
- However, this problem has been reduced in a development of a scheme known as Octree decomposition [Refer *Fig.3.7* (d) & (e)]. In this scheme only those cells that are occupied at each level are stored and sub-division is only necessary for partially occupied cells.



(a) First Level



(b) Second Level



(c) Third Level (d) Fourth Level (e) Fifth Level

Fig.3.7 – Spatial Occupancy Enumeration and Octree Decomposition

4. Cellular Decomposition

– An object is represented in this scheme by a list of cells it occupies, but the cells are not necessarily cubes, nor are they necessarily identical (see *Fig.3.8*).

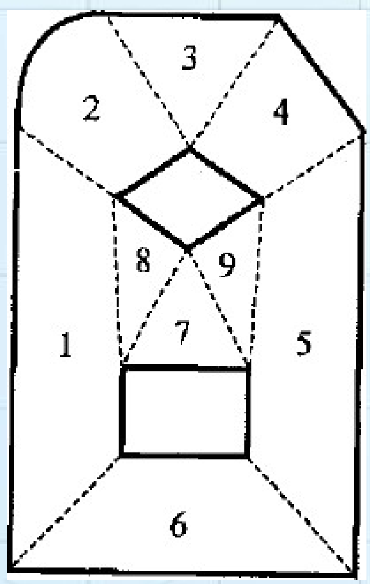


Fig.3.8 – Cellular Decomposition (9 Cell Decomposition)

5. Constructive Solid Geometry (CSG or C-rep)

– This method is also called as the Building Block Approach. The CSG system allows the user to build the model out of solid graphic primitives, such as rectangular blocks, spheres, cylinders, cones, wedges and torus. These shapes and the mathematical representation of these blocks are shown below in *Fig.3.9*.

– This is one of the most popular method of representing and building complex solids. ACSG model is based on the principle that any physical object can be divided into a set of bounded primitives which can be combined in a certain manner to form the physical object.

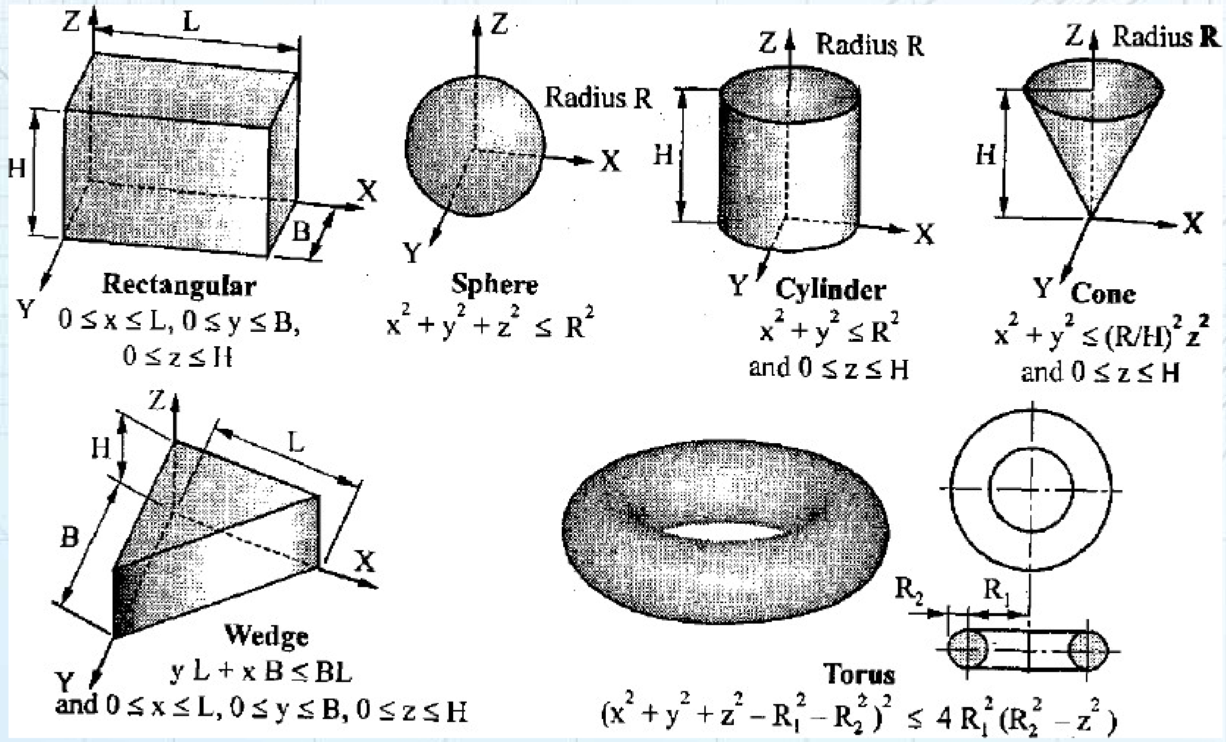


Fig.3.9 – Solid Primitives

- Structuring and combining the primitives of the solid model in the graphics database, is achieved by the use of Boolean operations. Consider two solids A and B, sharing a common volume of space as shown in *Fig.3.10* (a). The Boolean operations that can be performed on them and the resultant solid are shown in *Fig.3.10* (b) to (d).

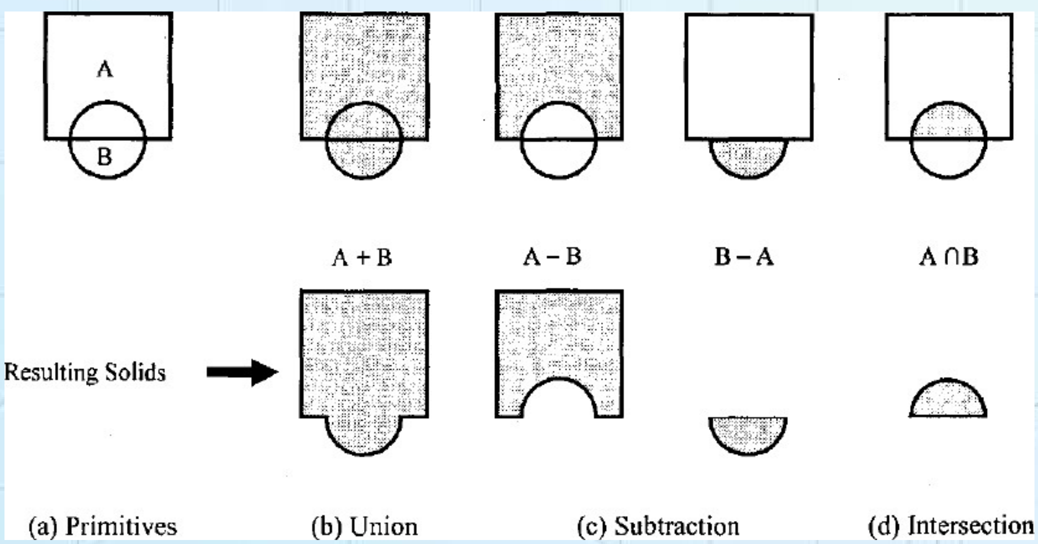


Fig.3.10 – Boolean Operations

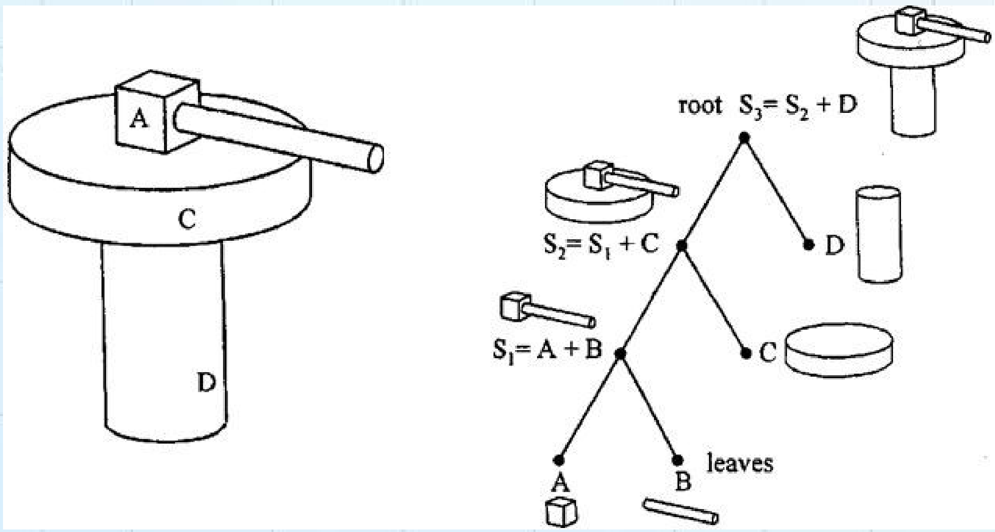
- CSG method has a procedural advantage in the initial formulation of the model. It is relatively easy to construct a precise solid model out of regular solid primitives by adding, subtracting and intersecting the components. CSG uses the unevaluated representation format which results in compact file of the model in the database. However, this results in more computations to evaluate the model.

– The data representation of CSG objects is represented by a binary tree. The binary tree gives the complete information of how individual primitives are combined to represent the object. The number of primitives thus decides the number of Boolean operations required to construct the binary tree.

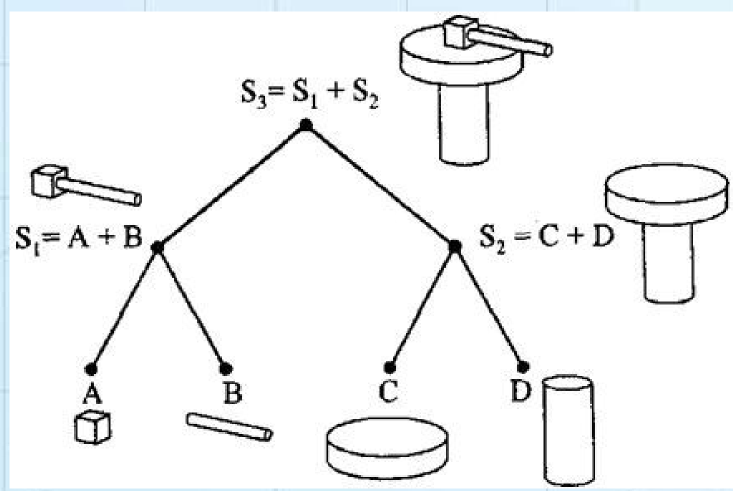
– The balanced distribution of the tree is desired to achieve minimum computations while modifying or interrogating a model. A balanced tree can be defined as a tree whose left and right subtrees have almost an equal number of nodes.

– The creation of balanced or an unbalanced tree is solely dependent on the user and is related to how the primitives are combined. Ideally the model should be built from an almost central position and branch out in two opposite directions.

– For example, while doing a union operation, it is preferred to combine all the unions together, rather than achieving unions of objects two at a time. A balanced and an unbalanced method of building the same object is shown in *Fig.3.11*.



(a) Unbalanced Tree



(b) *Balanced Tree*

Fig.3.11 – CSG Tree

– The CSG tree is organized upside down, with the root representing the composite object at the top and primitives called as leaves at the bottom.

6. **Boundary Representation (B-rep)**

- The boundary representation approach or B-rep is a popular method of solid representation. It can be considered as an extension of the wire frame model with additional face information added.
- In a B-rep, the solid is considered to be bounded by its surface and has its interior and exterior. The surface consists of a set of well organized faces. The two important areas for B-rep models are the topological and geometrical information.
- Topological information provides the relationship about vertices, edges and faces similar to that used in a wire frame model. In addition to connectivity, topological information also includes orientation of edges and faces. Geometric information is usually in terms of equations of the edges and the faces.
- In a B-rep model, the information related to the orientation of the face is important. Generally external vertices of the face are numbered in anti-clockwise fashion. The normal vector of the face as determined by the right hand rule must point to the exterior of the surface.
- Consider the three faces as shown in *Fig.3.12*. One of the methods to describe the top face would be 4-3-5-6. On similar lines the other two faces could be described as 1-2-3-4 and 1- 4-6-3.

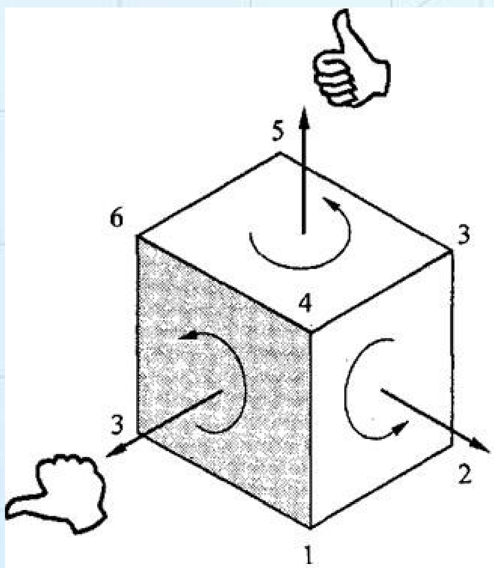
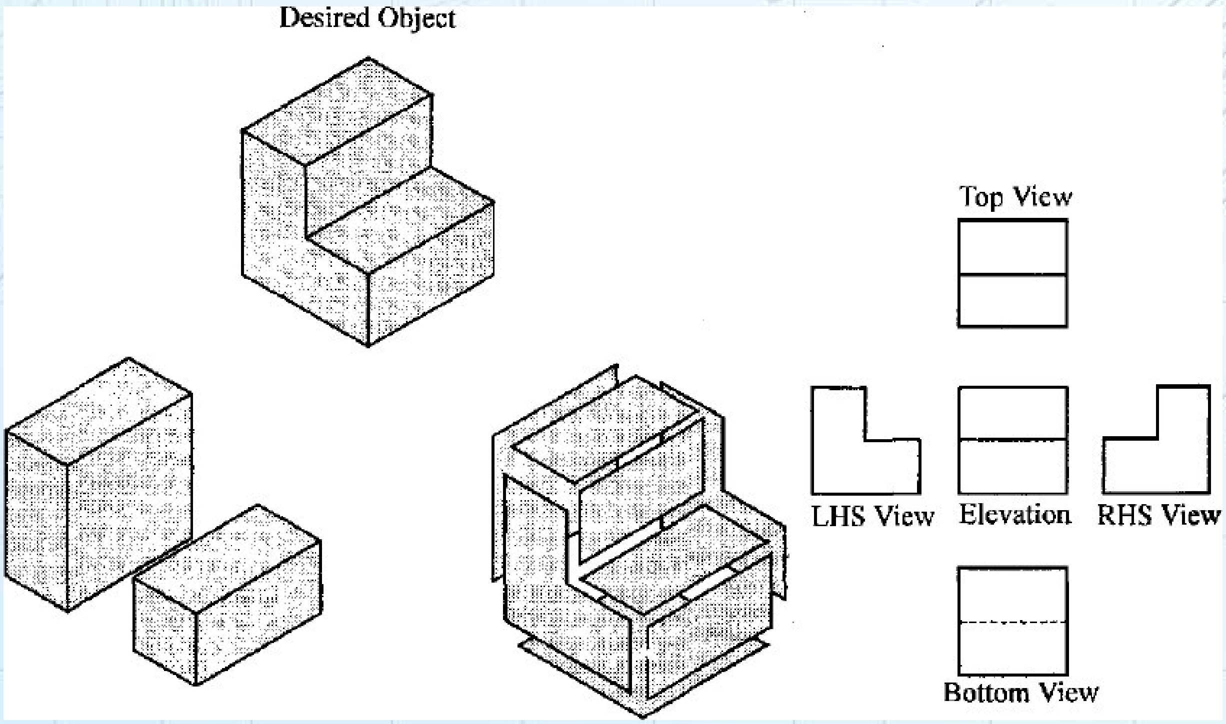


Fig.3.12 – Orientation of External Face

- However, not all surfaces can be oriented in this way. If the surface of the solid can be oriented, it is called orientable; otherwise it is non-orientable.
- The CSG and the B-rep information of the same solid are shown in *Fig.3.13*. The CSG model is achieved by the union of the two primitives whereas the facial information for B-rep is as shown.
- It is seen that B-rep systems store only the bounding surfaces of the solid, however it is still possible to compute geometrical and mass properties by virtue of numerical methods such as Gauss Divergence Theorem, which relates volume integrals to surface ones. The speed and accuracy of computations vary as per the kind of surfaces used.

- B-rep systems are extremely useful when unusual shapes are encountered, which would not be possible with CSG. This kind of situation occurs in aircraft fuselage, wing shapes and in automobile body styling. Such shapes would be extremely difficult to build by CSG.
- In B-rep models, the solid is represented as an evaluated data structure. This requires more storage space but does not demand nearly the same computation as would be required for CSG to reconstruct the image. This makes it more efficient.



(a) CSG Information *(b) B-rep Information*

Fig.3.13 – Comparison of CSG and B-rep Information

- Another advantage of B-rep is that, it is relatively simple to convert back and forth between a boundary representation and a corresponding wire-frame model. The reason is that the model's boundary definition is similar to the wire-frame definition which facilitates conversion of one form to another.

*Table **3.1** - Comparison of C-rep and B-rep*

C-rep	B-rep
It does not store only the bounding surfaces of the solid.	It stores only the bounding surfaces of the solid.
It is easy to create precise solid model using standard primitives.	It does not use standard primitives.
It is difficult to create unusual shapes.	It is easy to create unusual shapes.
Less storage space.	More storage space.
More computation time required.	Less computation time required.
It is relatively difficult to convert back and forth between a C-rep and a corresponding wire-frame model.	It is relatively simple to convert back and forth between a boundary representation and a corresponding wire-frame model.

7. Skinning or Lofting

- This method is uses a number of two dimensional profiles for generating a three dimensional object. The two dimensional profiles are arranged as desired and a three dimensional space curve is made to pass through these profiles.
- A skinning operation is then done for generating a solid. A typical example is shown in *Fig.3.14*. This method is useful in modeling turbine blades, aerofoil shapes, manifolds, volute chambers and other such surfaces.

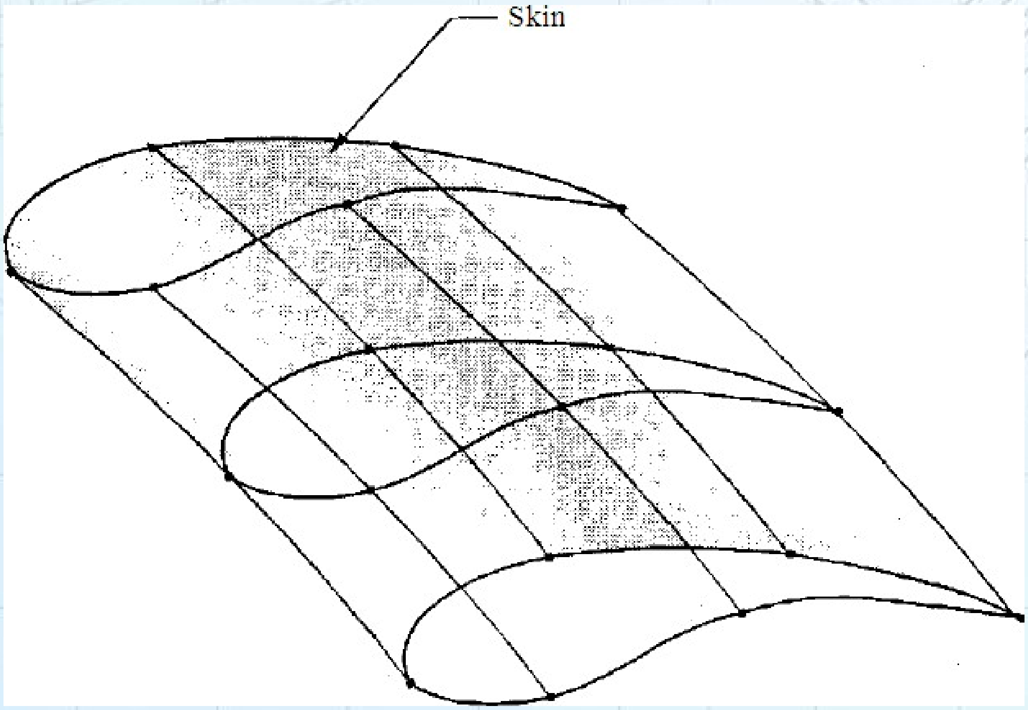


Fig.3.14 – Skinning

8. Miscellaneous Modeling Techniques:

- A complete modeler should have certain additional features which would simplify the modeling procedures. A few of them are described below:

(a) Tweaking :

- A model can be created by using a combination of methods as described previously. The ability available to the user to modify or move a vertex or a face and to see the effect of that on the model is referred to as tweaking. This feature is useful in styling automobile bodies and other consumer products.

(b) Filleting :

- Most engineering and consumer components are provided with smooth edges or generous fillets. This not only improves the stress distribution, but also prevents injury while handling them.
- If filleting procedure is done through CSG method, it would require a lot of data input for achieving it. Instead direct procedures which allow the user to input the properties of a fillet are necessary. This would result in a quick filleting operation, without a large database.
- A typical filleting operation in wire frame and rendered modes are shown in *Fig.3.15*.

- Each of the solid modeling representation discussed above has its advantages and disadvantages. The CSG and B-rep approaches are widely used in major CAD softwares.
- In a single representation modeler, the representation is usually B-rep, with certain commands such as sweep and a few CSG like inputs are accepted. However, the storing of data is in B-rep format only.
- As compared to this, a dual representation modeler has a primary representation scheme as CSG. (e. g. PADL-2). Here the modeler has both B-rep and CSG representations. However, in this case, B-rep is derived internally and the user has no control over it.
- In a truly hybrid modeler there are two independent internal representations of CSG and B- rep. With these systems, the users have the capability to construct the geometric model by either approach (CSG or B-rep), whichever is more appropriate to the particular problem.

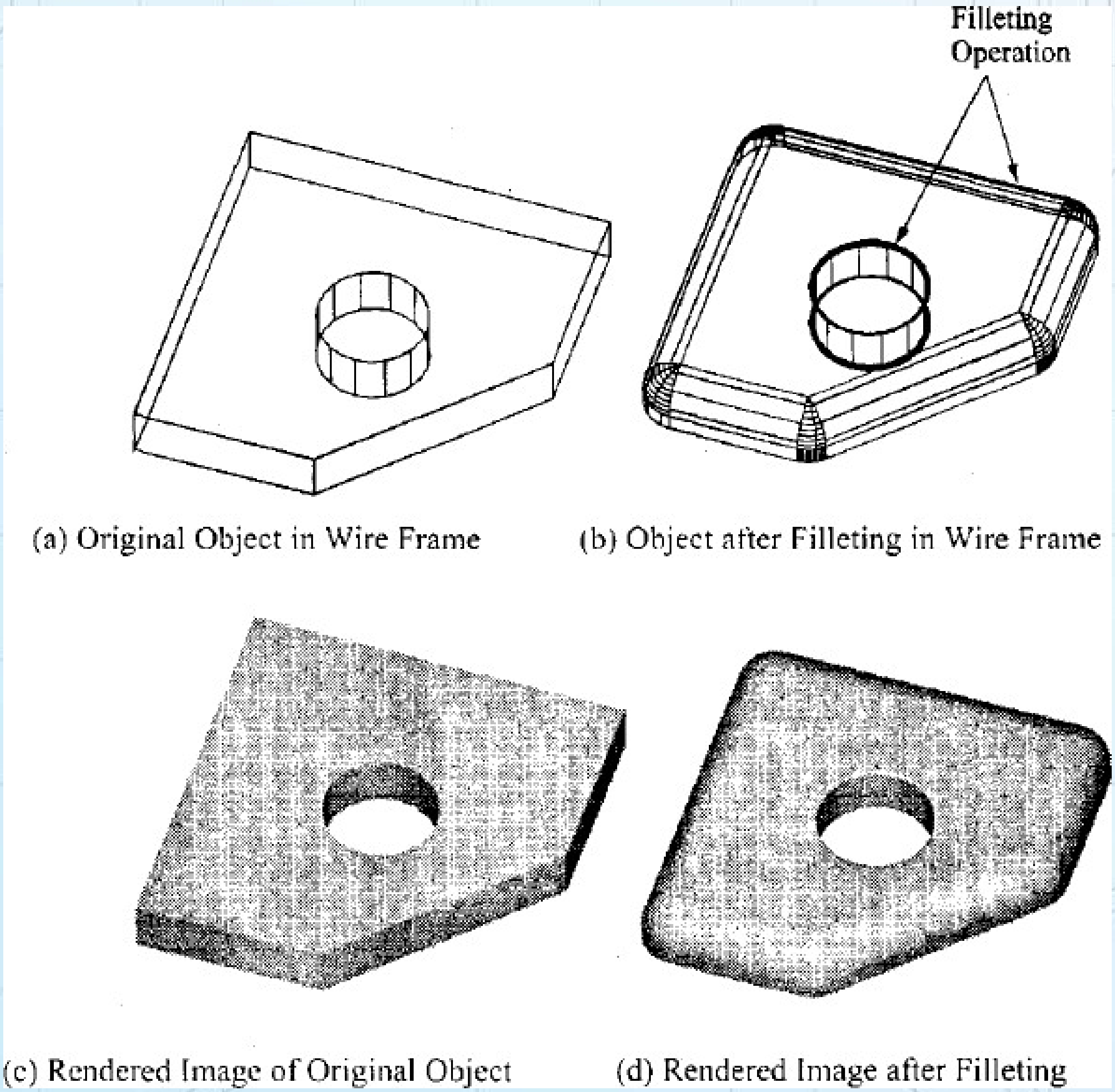


Fig.3.15 – Filleting Operation

3.6 Half Spaces

- Half-spaces form a basic representation scheme for bounded solids. By combining half-spaces (using set operations) in a building block fashion, various solids can be constructed. Half-spaces are usually unbounded geometric entities; each one of them divides the representation space into two infinite portions, one filled with material and the other empty. Surfaces can be considered half- space boundaries and half-spaces can be considered directed surfaces.
- A half-space is defined as a regular point set in three dimensional coordinate system (E^3) as follows:

$$H = \{P: P \in E^3 \text{ and } f(P) < 0\}$$

- where P is a point in E^3 and $f(P) = 0$ defines the surface equation of the half-space boundaries. Half-spaces can be combined together using set operations to create complex objects.
- The most used half spaces are:
- Planar half-space: $H=\{(x, y, z): z < 0\}$
- Cylindrical half-space: $H=\{x, y, z): x^2 + y^2 < R^2\}$
- Spherical half-space: $H = \{(x, y, z): x^2 + y^2 + z^2 < R^2\}$
- Conical half-space: $H = \{(x, y, z): x^2 + y^2 < [(\tan □/2)z]^2]\}$
- Toroidal half-space: $H = \{(x, y, z): (x^2 + y^2 + z^2 - R_2^2 - R_1^2)^2 < 4 R\ ^2(R_1^2 - z^2)\}$

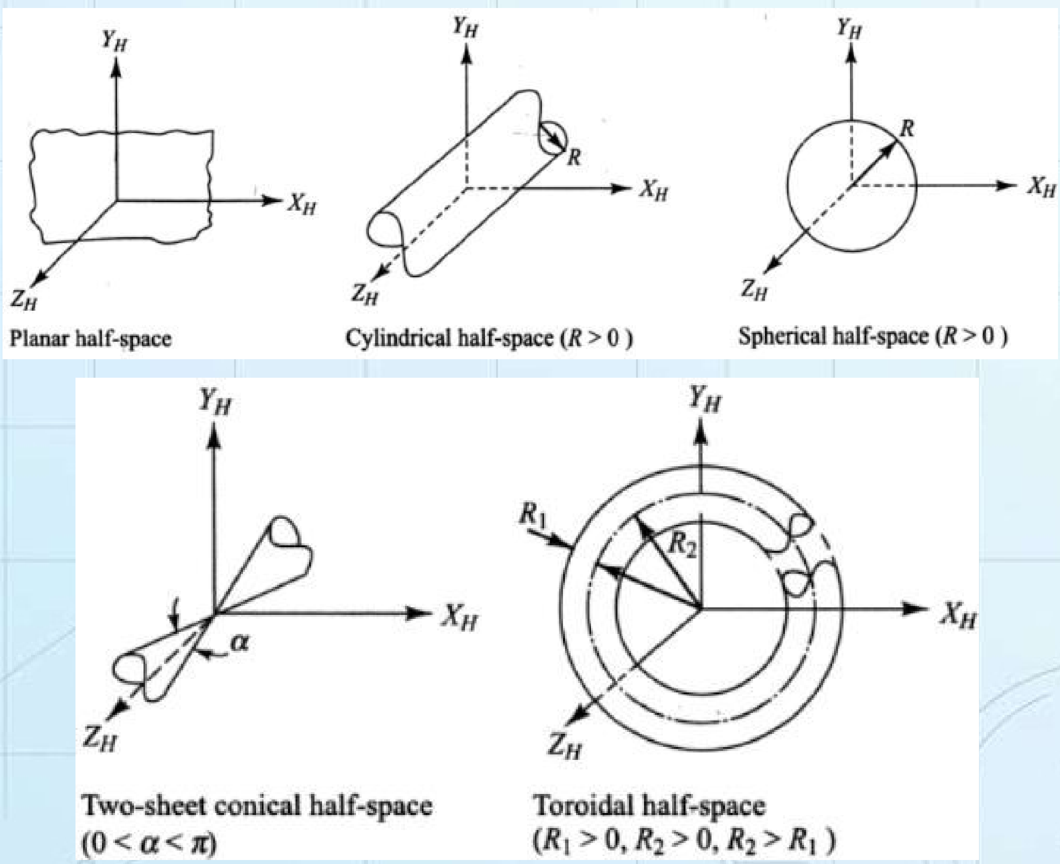


Fig.3.16

3.7 References

1) CAD/CAM Principles and Applications by P. N. Rao, Tata McGraw-Hill Publication.
2) Mastering CAD/CAM by Zeid,Ibrahim, Tata McGraw-Hill Publication.
3) CAD/CAM/CIM by Radhakrishnan,P;Subramanyan,S;Raju,V, New Age International (P) Limited.
4) CAD/CAM: Theory and Practice by Zeid,Ibrahim;Subramanian,R, Tata McGraw-Hill Publication.