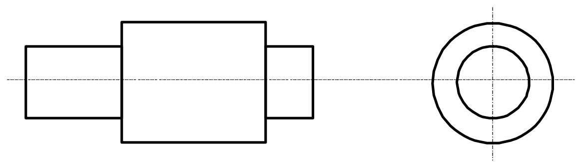
Representation of Part Geometry

2D Projections

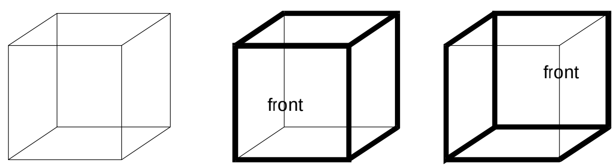
Multiple two dimensional (2D) projections in the form of Orthographic Views (top, front, right and section) of a modeled object uses electronic drafting to display engineering drawings and to represent the geometry of a model object.



3D Wireframe

In a three-dimensional wireframe model, an object is not recorded as a solid. Instead the vertices that define the boundary of the object, or the intersections of the edges of the object boundary are recorded as a collection of points and their connectivity.

A projection of these points and the line segments that connect these points onto a 2D viewing plane present the 3D shape of the modeled object. However, without removing the hidden lines, the representation may produce ambiguity. It is very difficult to remove the hidden lines based upon a wireframe model.

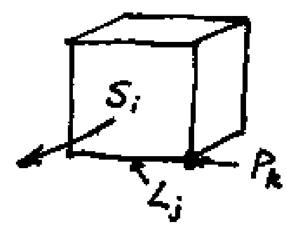


Mathematical Representation of A Plane, Its Edges and Points

(a) Plane, S_i:

Each surface is a polygon that is represented by the plane equation.

$$Ax + By + Cz + D = 0$$



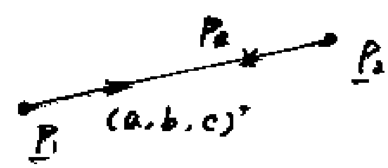
where. A, B & C are direction cosines of a line perpendicular to the plane (called surface normal). D – location of the plane.

(b) Edge, L_i, and Point, P_k:

We also need to define the edges, using a parametric representation:

$$\frac{x - x_1}{a} = \frac{y - y_1}{b} = \frac{z - z_1}{c} = P_k$$

$$0 \le P_k \le 1$$



where,

$a = x_2 - x_1$, $b = y_2 - y_1$ $c = z_2 - z_1$ and the edge goes from point P_1 $(x_1, y_1, z_1)^T$ to point P_2 $(x_2, y_2, z_2)^T$

starting point $(x_1\ y_1\ z_1)$; direction $(a\ b\ c)$ length: P_k.

Vector form representation of an edge

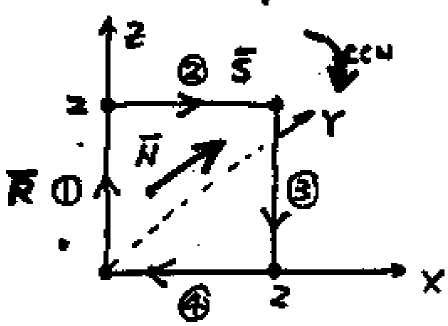
$$\vec{R} = a P_R \vec{i} + b P_R \vec{j} + c P_R \vec{k}$$

$$= \underbrace{P_R}_{\text{Length}} \underbrace{(a\vec{i} + b\vec{j} + c\vec{k})}_{\text{Direction, start/end point (implicit)}}$$

$$\begin{cases} a = x_2 - x_1 \\ b = y_2 - y_1 \\ c = z_2 - z_1 \end{cases}$$

The polygons are defined in the Model (or World) Coordinate System.

An Example



△ Plane equation

$y = 0$

$A = 0, \; B = 1, \; C = 0, \; D = 0$

△ Surface Normal : $\vec{j}$

△ Edges

		a	b	c	x_1	y_1	z_1
	①	0	0	2	0	0	0
CCW:	②	2	0	0	0	0	2
	③	0	0	-2	2	0	2
	④	-2	0	0	2	0	0

The direction cosines of a plane can be found from any two intersecting edges, $\bar{R}$ and $\bar{S}$

$$\bar{N} = \vec{R} \times \vec{S}$$

$$\vec{①} \times \vec{②} \Rightarrow \vec{j} \qquad \text{(CCW for most systems)}$$

$$\vec{R} = R_x \vec{i} + R_y \vec{j} + R_z \vec{k}$$

$$= (x_2 - x_1)P_k \vec{i} + (y_2 - y_1)P_k \vec{j} + (z_2 - z_1)P_k \vec{k}$$

$$\vec{S} = S_x \vec{i} + S_y \vec{j} + S_z \vec{k} = \cdots$$

$$\vec{N} = \vec{R} \times \vec{S} = \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ R_x & R_y & R_z \\ S_x & S_y & S_z \end{bmatrix}$$

$$= \underbrace{[R_y S_z - S_y R_z]}_{A} \vec{i} + \underbrace{[R_z S_x - R_x S_z]}_{B} \vec{j} + \underbrace{[R_x S_y - R_y S_x]}_{C} \vec{k}$$

All parallel planes have same direction cosines, (A, B, C).

If we specify one point on the plane, we can define D.

<u>An Example</u>

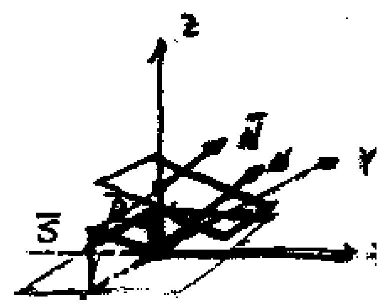
Given $\vec{R} = -\vec{i} + \vec{j}$, $\vec{S} = -\vec{j} + \vec{k}$ and a through point $(0, 0, 1)^T$, find plane equation.

① $$\vec{N} = \vec{R} \times \vec{S} = \begin{bmatrix} i & j & k \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} = (1\times1)i + (0\times0)j + [(-1)\times(-1)]k - (0\times1)k - [(-1)\times0]i - [(-1)\times1]j$$

$$= \vec{i} + \vec{j} + \vec{k}$$

② $x + y + z + D = 0$ go through $(0, 0, 1)^T$

$\therefore\ 1 + D = 0,\ D = -1$



$\therefore$ Plane eq: $x + y + z - 1 = 0$

Solid Models

The solid modeling technique is based upon the "half-space" concept. The boundary of the model separates the interior and exterior of the modeled object. The object is defined by the volume space contained within the defined boundary of the object. In general speaking, a closed boundary is needed to define a solid object.
The approach facilitates many important operations, including

- weight or volume calculation
- centroids, moments of inertia calculation
- stress analysis (finite elements analysis)
- heat conduction calculations
- dynamic analysis
- generation of CNC codes
- robotic and assembly simulation

In modeling a solid part, one needs more than just a description of the model surfaces. We need to specify unambiguously which points in space are inside the object and which are outside. The solid model of a mechanical part can be formed using a number of solid modeling techniques. These include:

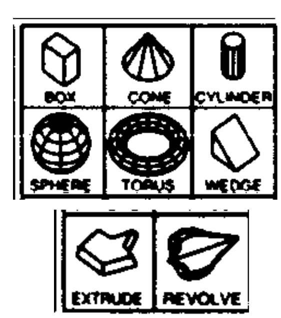
- Constructive Solid Geometry (CSG).
- Sweeping.
- Boundary Representation (B-Rep).
- Octree - for irregular objects (uses cubes of varying size).
- Primitive Instancing - uses large numbers (200 - 300) of "Primitives" to build object - used for programming NC machine tools.
- Feature-Based Modeling - uses feature-based primitives to conduct a design.

■ Constructive Solid Geometry (CSG)

Constructive solid geometry forms the model of a part using a collection of pre-defined geometric primitives and a sequence of Boolean operations applied to these geometric primitives. For uncommon shapes, sweeping of a 2D cross-section in the form of extrusion and revolving are used to define the 3D shape.

Geometric Primitives

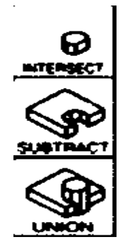
Take the example of AutoCAD, typical geometric primitives include box, cone, cylinder, sphere, torus and wedge. Additional primitives can be introduced by extrusion and revolving of user defined 2D cross-sections.



Boolean Operations

Boolean operation includes

- *union, U*
- *difference, or subtraction,* -
- *intersection,*



∩

These operations are well illustrated by J. Bolluyt in the Pre/E Design Modeling book:

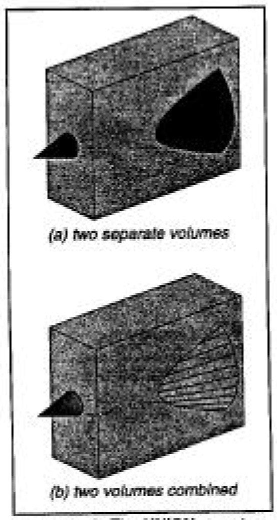


Figure 1-11 *The* ***UNION*** *operator combines two volumes or solids as in (a) into a single volume or solid as in (b).*

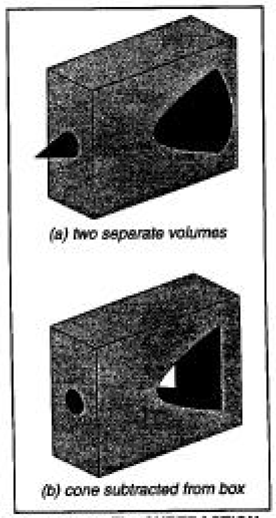


Figure 1-12 *The* ***SUBTRACTION*** *operator subtracts the volume of one solid from that of a another.*

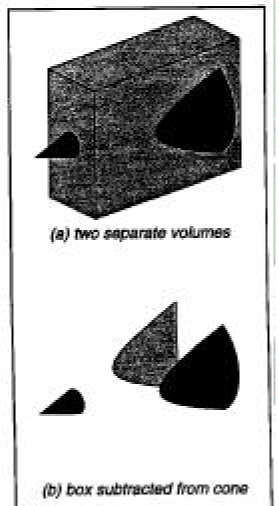


Figure 1-13 *For the* ***SUBTRACTION*** *operator, order of operands is important to the resulting geometry.*

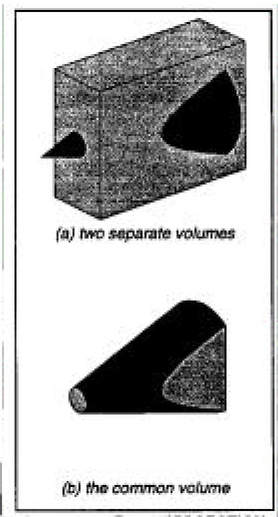


Figure 1-14 *The* ***INTERSECTION*** *operator finds the volume common to both of the initial volumes.*

The Constructive Solid Geometry combines simple geometric primitives to form more complex solid objects using the stated Boolean operations. Based upon the sequence of these Boolean operations, a hierarchical object composition tree is formed. This **CSG tree** is used to store the modeled object geometry and the sequence of model formation. In some solid modeling based CAD systems, the CSG tree can be edited later on to alter the logical relations among various geometric elements.

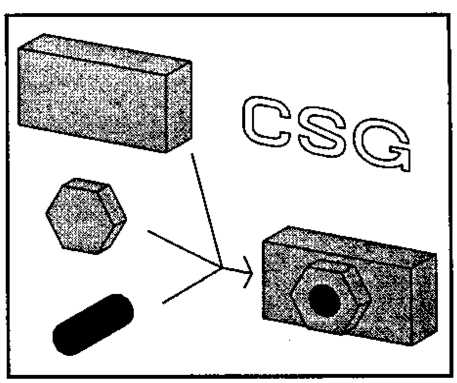


Figure 1-8 *Constructive Solid Geometry involves the combining of simple solid shapes to produce more complex solid shapes.*

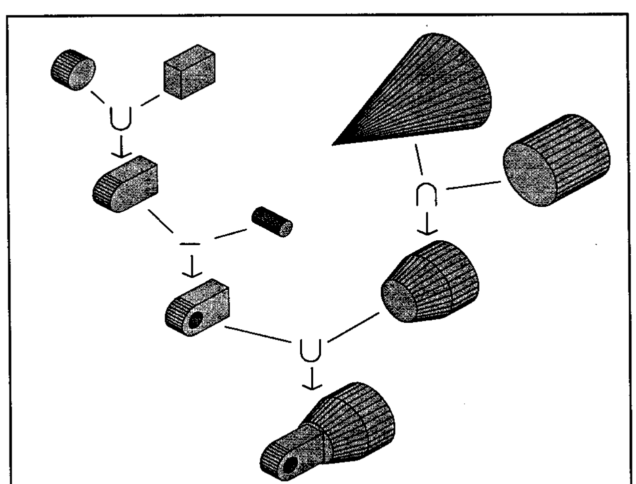


Figure 1-10 *If a computer model is created using Constructive Solid Geometry techniques, it can be represented and stored as a* ***CSG tree.***

■ Sweeping

The geometric shape that can be directly built using the given geometric primitives is limited. For more complex object geometry or objects of certain types, it is more efficient to generate the "new" primitives using **Sweeping**.

Sweeping can be carried out in different forms. The most commonly used include:

- *extrusion*
  Extrusion is used to produce an object model from a 2D cross-section shape, the direction of extrusion, and a given depth. Advanced applications include curved extrusion guideline and varying cross-sections.
- *revolving*
  Revolving is used to produce a rotation part, either in solid or in shell shape. Revolving a 2D cross-section that is specified by a closed curve around the axis of symmetry forms the model of an axially symmetric object.

Sweeping is most convenient for solids with translational or rotational symmetry. Sweeping also has the capability to guarantee a closed object.

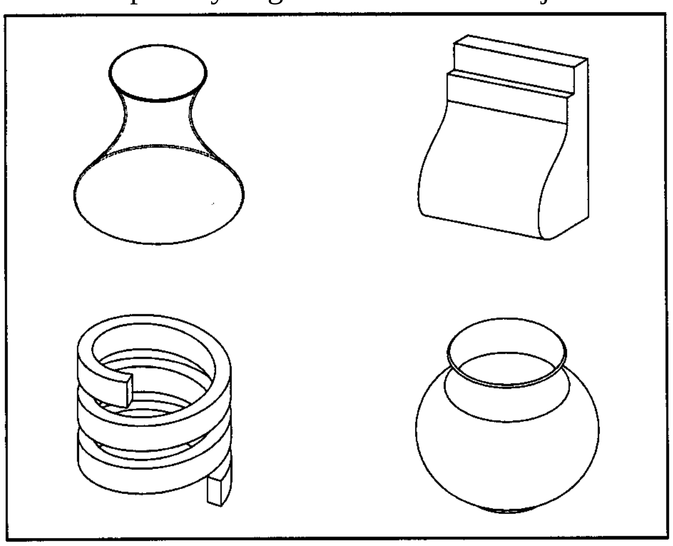


Figure 1-15 *Some shapes are difficult if not impossible to create using CSG techniques but quite easy to create with sweeping techniques.*

■ Boundary Representation (B-rep)

The boundary representation method represents a solid as a collection of boundary surfaces. The database records both of the surface geometry and the topological relations among these surfaces.

The method is can be conveniently used for hidden surface removal and rendering due to the directly available surface information of the modeled objects. The solid is defined by specifying that all points on one side of the boundary are outside and those on the other side are inside of the object. This inside and outside information is represented using the surface normals of all surface patches. Points along outward normal $\overline{N}$ (N > o) are outside; points for N < o are inside.

Boundary representation does not guarantee that a group of boundary surfaces (often polygons) form a closed solid. The data are also not in the ideal form for model calculations. This representation is used mainly for graphical displays. Many CAD systems have a hybrid data structure, using CSG and B-rep at the same time.

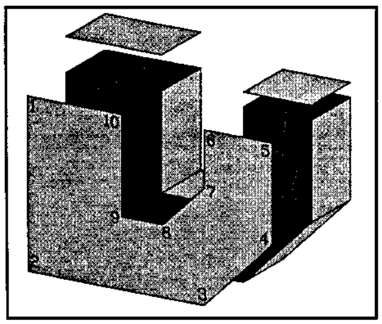


Figure 1-4 *In a surface representation, edges are connected such that they enclose one or more surfaces. In the model above, for example, the "front" surface could be represented as the "connectivity list" 1-2-3-4-5-6-7-8-9-10.*

In B-rep surface geometry and topology (or connectivity) are recorded in a hierchical data structure as follows:

Object List -- giving object name, a list of all its boundary surfaces, and the relation to other objects of the model.
Surface List -- giving surface name, a list of all its component polygons, and the relation to other surfaces of the object.
Polygon List -- giving polygon name, a list of all boundary segments that form this polygon, and the relation to other polygons of the surface.
Boundary List -- giving boundary name, a list of all line segments that for this boundary, and the relation to other boundary lines of the polygon.
Line List -- giving line name, the name of its two end points, and the relation to other lines of the boundary line.
Point List -- giving point name, the X, Y and Z coordinates of the point and, and the relation to other end point of the line.

■ Feature-based, Parametric Models

Feature-based, Parametric Solid Modeling system represents the recent advance of computer geometric modeling. It is used as the foundation of Pro/ENGINEER and AutoCAD Designer.

Feature-based, parametric solid modeling eliminated the direct use of common geometric primitives such as *cone, cylinder, sphere*, etc. The reason is that these pure geometry-sense primitives only represent low-level geometric entities. In designing and manufacturing a mechanical part, one would always refer to mechanical features, rather than these geometric primitives.

The modeling approach uses **sweeping** to form the main shape of the part, and the build-in mechanical features to specify the detailed geometry of the model. These features include *holes* (through, blind, sink), *rounds, chamfers, slots*, etc. Operations to solid model, such as *cut* and *shell* (change a solid model into a hollow shell) are also supported.

To create the 2D cross-section for *sweeping*, a 2D sketch needs to be generated in the **2D Sketcher**. A user can sketch the rough shape of the closed shape. The system will automatically assign a dimension value of the sketched feature. The **dimensions** of the sketched feature can be changed at any time by simply entering the desired value, or kept as a variable, allowing even more convenient change of its value. The user has to provide all necessary dimensions to pass the section of cross-section generation. Under-dimensioned and over-dimensioned design will be pointed out.

Primary Shape Definition

- **Drawing 2D Cross-section in A 2D Sketcher**
- **Defining the Precise Geometry**
- **Building Solid Objects**

    <u>Extrusion to Form Depth</u>

    <u>Revolving to Form Rotational Features</u>

    <u>Sweeps and Blends</u>

Detailed Geometry

- **Holes and Cuts**
- **Rounds, Chamfers, Slots, and Shells**