

## 7. Creating parts in 3D drafting

### 7.1. CONSTRUCTIVE SOLID GEOMETRY(CSG)

Constructive Solid Geometry (CSG) modeling is a powerful technique that allows flexibility in both the way primitives are defined and the way they are combined. The relationships between the primitives are defined with Boolean operations. There are three types of Boolean operations: union ( $\cup$ ), difference ( $-$ ), and intersection ( $\cap$ ). The critical area is the place where two objects overlap. This is where the differences between the Boolean operations are evident. The union operation is essentially additive, with the two primitives being combined. However, in the final form, the volume where the two primitives overlap is only represented once. Otherwise there would be twice as much material in the area of overlap, which is not possible in a real object. With a difference operation, the area of overlap is not represented at all. The final form resembles one of the original primitives with the area of overlap removed. With the intersection operation, *only* the area of overlap remains; the rest of the primitive volume is removed.

#### 7.1.1. CSG Primitives

Primitives are standard geometric shapes such as blocks, cylinders, cones, and spheres. They are an older traditional way to create features, now less used, since these shapes can all be created using extrude or revolve. To create primitive shapes, get them out of the standard parts catalog and specify their characteristic parameters (dimensions) in a form. They are initially located with their center positioned at the origin of the workplane.

CSG describes a method of creating a solid model that parallels the thought process you use to create the object from primitives. These are indeed primitives, but as a practical matter, any solid you are modifying to create a new one could also be thought of as a primitive for that operation. See Figure 7.1 for some common primitives that are available in “model libraries” in CAD modelers. These objects can be combined in a variety of ways to create a model of the object you are designing. This process is handled through the modeler by means of solid (3D) Boolean operations. Thus we must examine Boolean operations before proceeding with solid models.

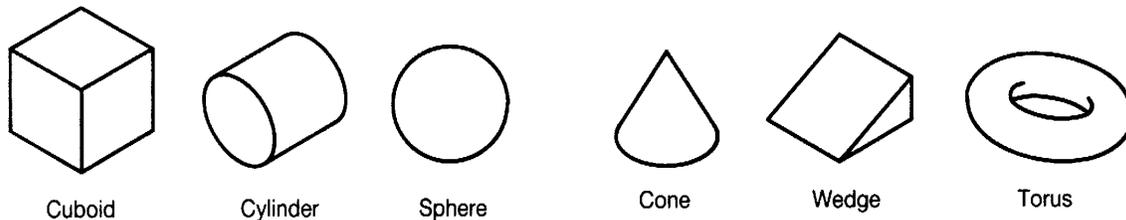


Figure 7.1. Typical solid primitives available in a model library.

The CSG process can be represented as a binary tree that shows the series of steps that lead from a set of primitives to a valid complex solid. Note that each primitive is itself a valid solid. As each step is completed, it is necessary to check to see if the newly created solid is valid. This avoids the problem that would exist, for example, if primitives were combined in such a way that a zero-thickness wall was formed. Figure 7.2 illustrates a series of steps that will produce the solid labeled V.

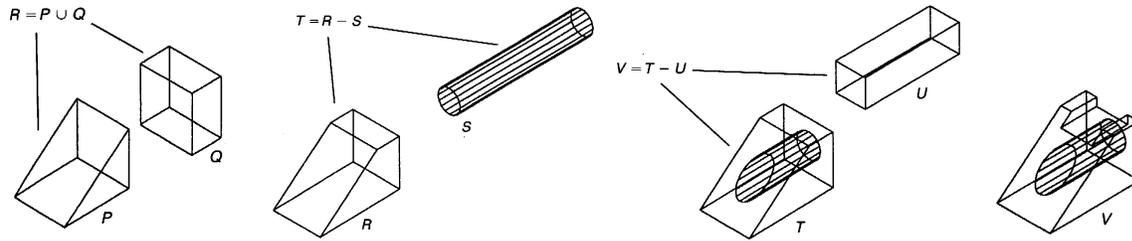


Figure 7.2. CSG Tree for object V

**Step 1:** Primitives P (wedge) and Q (box) are brought together and fused into one by the union operation  $P \cup Q$  resulting in object R. Note that object R could also have been produced by using a larger box and removing a wedge from it with a subtraction operation. As this suggests, there are often alternative approaches in CSG that will produce the same outcome.

**Step 2:** Object R is considered a primitive for this step since it is combined with primitive S with a subtraction operation to give object T.

**Step 3:** Primitive U is now removed from object T with a subtraction operation to produce the desired object V.

Figure 7.3 shows another object created by the CSG procedure. In this case, two preliminary objects, C and T, were created independently prior to them being combined into object D. It is clear that the subtraction operation with object E could have been performed with object C prior to the union operation with T. The same solid would have resulted.

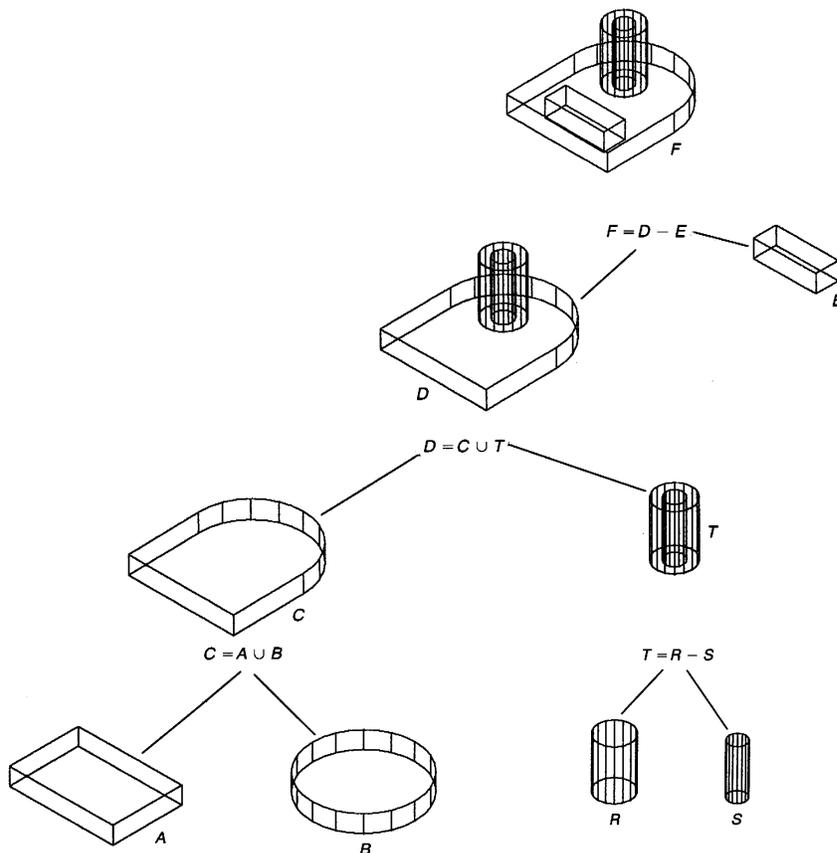


Figure 7.3. CSG tree for object F

Special primitives can be created with the intersection operation as seen in Figure 7.4. Here rounded corners are produced on an object. Intersection operation can be used to produce special features such as the rounds on object N in Figure 7.4.

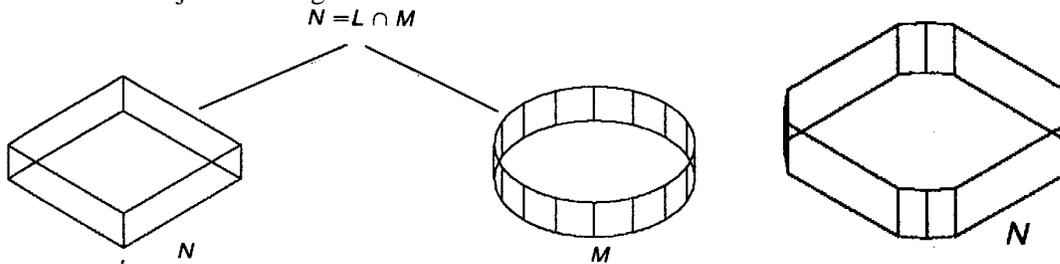


Figure 7.4. Intersection operation

### 7.1.2. Two-dimensional Boolean Operations

Two-dimensional Boolean operations can be used to advantage in producing the 2D image prior to applying sweeping operations. These Boolean operations are analogous to their 3D counterparts except that the 2D operations result in a 2D profile or region rather than a solid. The 2D Boolean operations are shown in Figure 7.5.

- Union ( $\cup$ ): Operation places all of the areas enclosed by the original primitives into a new primitive.
- Intersection ( $\cap$ ): Operation retains the area that was common to the original primitives.
- Subtraction ( $-$ ): Operation removes the common area from the first-named primitive.

	$L \cup M$ 2-D union	
	$L \cap M$ 2-D intersection	
	$L - M$ 2-D subtraction	
	$M - L$ 2-D subtraction	

Figure 7.5. Two-dimensional Boolean operations

### 7.1.3. Three-dimensional Boolean Operations

When working with a traditional 3-D modeling system, defining the size or shape of a geometric feature is typically done independently of any other feature. It is completely up to the user of the CAD system to make sure the resulting model reflects the real-world design constraints the final product will have to face. With a simple part consisting of three overall dimensions and one or two features (e.g. holes or slots), making sure these features are properly placed as the part is constructed is not a major task. On the other hand, with complex parts or large assemblies, the total number of features to track can be a major task (see Figure 7.6)

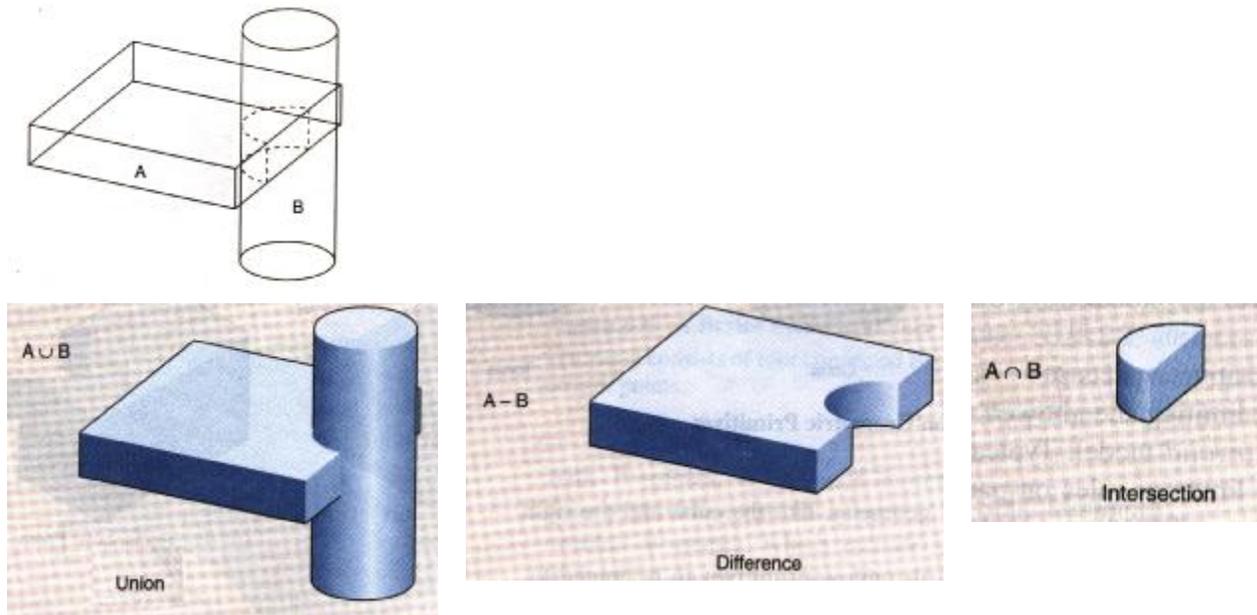


Figure 7.6. The Three Boolean Operations: Union, Difference, and Intersection

Solid Boolean operations combine two solids into a new solid according to defined rules. The basic operations are

1. Union ( $\cup$ ): This operation places all of the original solids into the new solid. The symbol is  $A \cup B$ . The union corresponds to the logical operator OR. (See Figure 7.7)
2. Intersection ( $\cap$ ): This operation places the volume that is common to each of the original solids into the new solid. The symbol is  $A \cap B$ . The intersection corresponds to the logical operator AND. (See Figure 7.7)
3. Subtraction (also called difference) ( $-$ ): This operation places all of the first-named original solid into the new solid except for the volume that is common with the second-named original solid. The symbol is  $A - B$  or  $B - A$ . Note that the order of the subtraction operation is important in that different solids are created by  $A - B$  than by  $B - A$ . For an example of each. The subtraction operator corresponds to the logical NOT. (See Fig. 7.7)

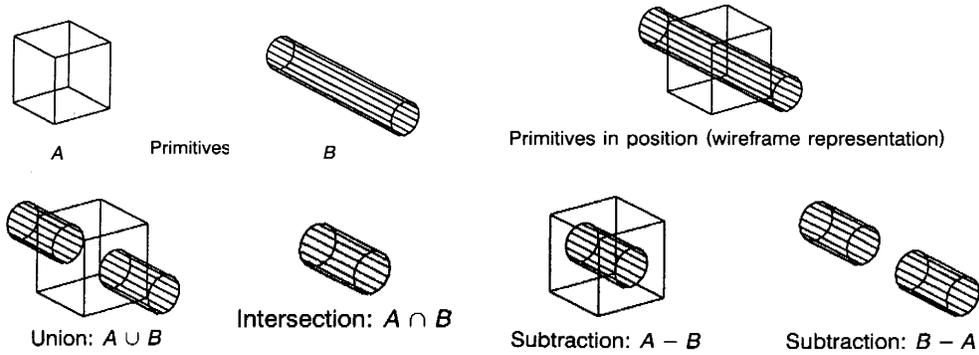
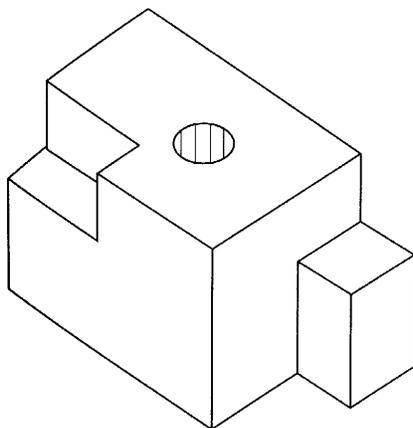


Figure 7.7. Interpretation of solid Boolean operation

### 7.1.4. Differences Between CSF and B-rep

The main differences between pure CSG and B-Rep modeling systems, however, are in the ways in which a model is represented and stored in the geometric database and how that data is processed for graphical output. A CSG model is defined as an ordered binary tree in which the primitives are defined by their position, size and orientation and objects are represented as Boolean expressions between the primitives. Further construction of an object is stored as Boolean operations between additional primitives and an existing object or between two objects. Figure 7.8(a) shows a simple object modeled with a CSG system and Figure 7.8(b) shows how such a model would be typically represented within the geometric database. The final object, OBJ3, is constructed from three blocks or cuboids and one cylinder and the modeler defines primitives according to how they are defined by the user.



BLK1	=	BLCFN( 100,80,70,0,0,0,0,0)
BLK2	=	BLCNR(45, 30,20,—50,—40,35,0,0,0)
OBJi	=	BLK1 - BLK2
CYL1	=	CYLEN(90,10,10,0,0,90,0,0)
OBJ2	=	OBJi - CYL1
BLK3	=	BLCNR(25,50,30,50,—40,35,0,0,0)
OBJ3	=	OBJ2 + BLK3

Figure 7.8. CSG modeling example and model representation

With this pseudo-modeler, a block defined with its center as a reference point is specified as BLCEN and one referenced by its corner as BLCNR. Therefore the block BLK1 is defined by its center: its dimensions are 100mm long by 80 mm wide by 70mm high, its center point is at the origin (0, 0, 0) and it is not rotated about any of the three Cartesian axes. The cylinder, CYL1, is defined as being 90mm long with a 10mm radius, its center point is at x 10, y = 0, z 0, and rotated about the x axis by 90 degrees. In order to display a pure CSG model, the software must be able to evaluate the positions of all the vertices and edges of the model after evaluating the Boolean expressions, which go to make it up.

The majority of commercial solid modelers available nowadays use a combination of B-Rep and CSG modeling techniques and representations. Objects can be created using the sweep methods of revolutions and extrusion and the results stored as B-Rep data, but identifiable as primitives so that they can be combined with other objects and primitives to realize the final design. The procedures for converting a CSG model into B-Rep are well established.

## 7.2. OBJECT EDGE APPEARANCE

There are many different methods for displaying solid objects on the computer screen, but the primary differences among them are concerned with how the surfaces and edges defining the object appear to the observer. Edges of an object usually appear as lines or curves and differ mainly in how the *hidden edges* appear on the screen. Hidden edges are defined as those edges that are “behind” the object relative to the current setting for the observer view angle. The most common choices for displaying hidden lines are to show them as solid lines or as dashed lines, or not to show them at all (in effect, to show them as invisible lines). When hidden lines are shown as solid lines, the display is sometimes referred to as the *wireframe view* of the object. In a wireframe view, the object appears to be hollow and made up of thick wires in a framework along its edges. Figure 7.9 shows an object with the hidden edges displayed as either solid, dashed, or invisible lines.

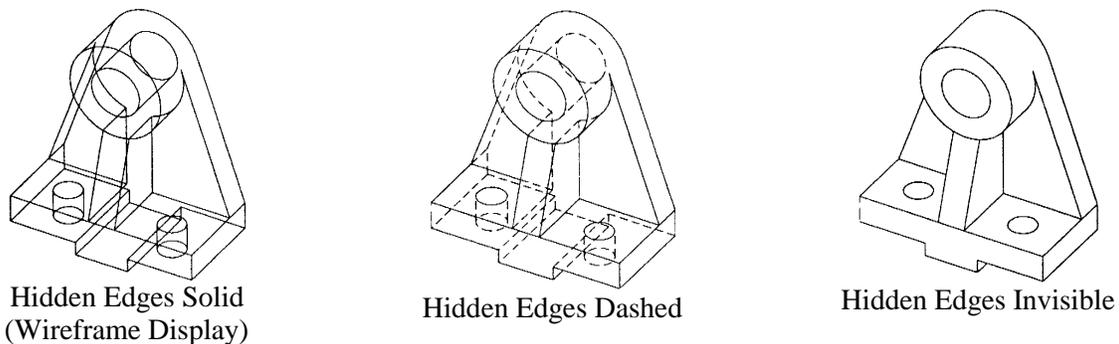


Figure 7.9. Various Hidden-Edge Appearances

Another difference in edge display for computer models appears when you show the object with or without perspective. Perspective display shows all of the lines on the object receding to the same point on the horizon.

## 7.3. OBJECT SURFACE APPEARANCE

Surfaces are usually displayed as either shaded or not shaded. Some software packages call this feature rendering. Depending on the software, you will probably have the option of showing your shaded image of the object with the edges either visible or invisible. Displaying the shaded image with the edges invisible sometimes results in a hard copy that is unclear, because the edges may not be well defined due to the light source and color of the object.

Another surface display setting is visible only when there are curved surfaces on the object. Curved surfaces exist on many objects; for example, the cylindrical surface in a circular hole is a curved surface. Sometimes, you may wish to show the curved surface with ruling lines on it. Although these lines do not appear on the actual surface of the object, their display often gives a better definition to the surface.

## 7.4. CREATING 3-D PARTS FROM 2-D GEOMETRY

Features can be defined in different ways, such as using wireframe geometry or by getting standard parts out of a catalog. Some of the basic feature types includes extrude, revolve, primitive shapes, fillet, chamfer, and shell.

Some of the most common modeling techniques used in creating 3-D parts involve first sketching or drawing 2-D geometry and then either *extruding* or *revolving* the geometry to form a solid. The most common feature creation methods are extruding or revolving sections to get shapes as shown in Figure 7.10.

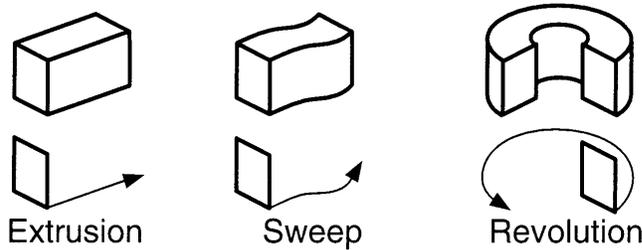


Figure 7.10. Feature creation methods

To create these features, you must first create wireframe construction geometry to define the sections to be extruded or revolved. The wireframe geometry may contain geometric constraints and dimensions. A part is a collection of features. The first feature created on a part may be referred to as the base part, but there is really no difference between this first feature and other features. The process of modeling parts is the process of creating features.

Features can be created as separate parts and then applied using construction operations such as cut and join. Features created by sketching in place are automatically associated to the sketch face. This means that if the underlying part geometry is modified later, the feature will remain attached to the face. The position of the feature on the face is maintained using dimensions to place the section relative to known edges of the sketch face.

### 7.4.1. Construction Operations

Parts created as separate parts on the workbench can be used as features using the various construction operations. For example, an offset block to create the part in Figure 7.11(b), and then cut with a cylinder to produce the part on Figure 7.11(a) could cut the extrusion on Figure 7.11(c). One consideration is which method gets to the desired result with the minimum number of steps, which is not only faster, but will also make the part easier to modify later. Another consideration is which method most closely matches the manufacturing operation that will be used to produce the part. The same part could be made more easily by extruding the shape including the holes as shown in Figure 7.11(d).

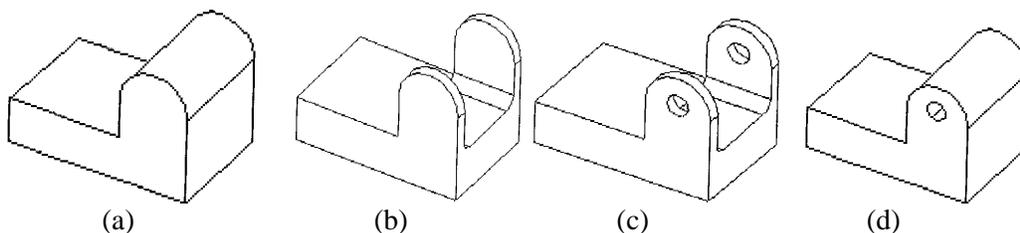


Figure 7.11. Construction operations

### 7.4.2. Extrude

Your wireframe curves may be a nice neat closed section with each curve starting at the end of the previous curve, or you may pick curve segments out of a jumbled “haystack” of intersecting lines and curves. You may describe the section in a separate command and then extrude or revolve this section, or you may build the section “on the fly” when you execute the *Extrude* or *Revolve* command.

Reasons for defining a section as a separate step before the *Extrude* command might be to extract a section from another part or to modify an existing section before extruding. An example would be to extract a section from one part, offsetting it to allow for a clearance, and then use this section to cut a mating part (see Figure 7.12).

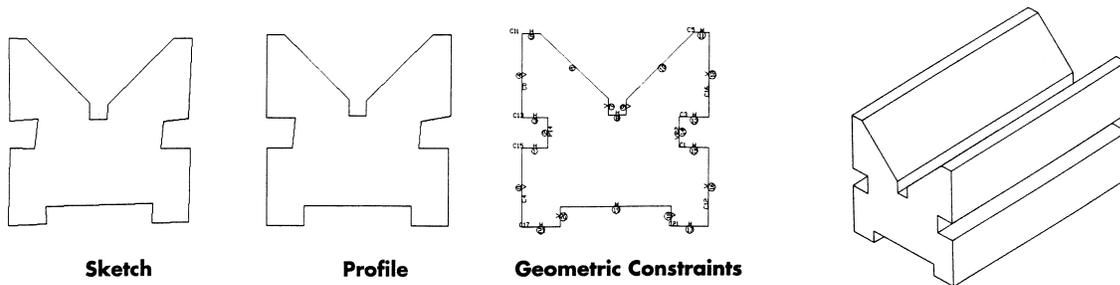


Figure 7.12. Object Created by Extrusion

### 7.4.3. Revolve

Revolution is another technique used to create a 3-D part from a two-dimensional shape. With this technique, a given 2-D shape is revolved about an axis, sweeping a solid. Figure 7.13 shows a 2-D shape, together with the 3-D part created by revolving that shape. The axis of revolution is also indicated in Figure 7.13.

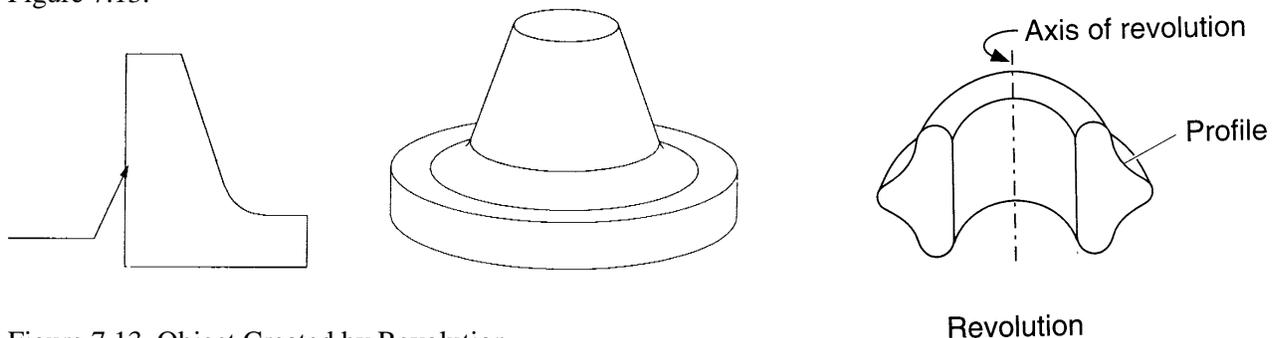


Figure 7.13. Object Created by Revolution

### 7.4.4. Shell

The *Shell* command takes surfaces and gives them thickness. You can use this command both to start with an open surface model and create a solid, or to start with a complete solid and thicken selected surfaces, deleting others. The example shown in Figure 7.14 using a shell operation to create a plastic wheel. During the shell operation, the flat face was deleted, resulting in the part in the center. After the shell operation, other features were added inside and a fillet added to the outside edges.

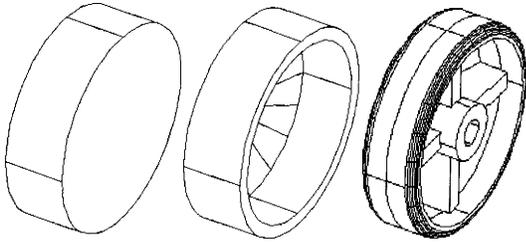


Figure 7.14. Shell operation to create the shell

## 7.5. SWEPT AND BLENDED FEATURES

The extrude and revolve options can be used to create either positive space or negative space features. The sweep option can be compared to the extrude option; whereas the extrude option creates a feature by protruding a section along a straight trajectory; the sweep option creates a section along a user-defined trajectory. This trajectory can be either user-sketched or selected on the work screen. The blend option can also be compared to the extrude option. Primarily, the blend option creates a feature by protruding along a straight trajectory between two or more user-defined sections. A partially revolved blend can be created also.

### 7.5.1. Swept Features

Protruding a section along a user-defined trajectory creates sweeps. Figure 7.15 shows an example of a sweep. The section in this feature is protruded along a curved trajectory. Also in Figure 7.15, the same section is shown protruded along a straight trajectory with the extrude option. Like the extrude and revolve options, the sweep option is available with the protrusion and cut commands.

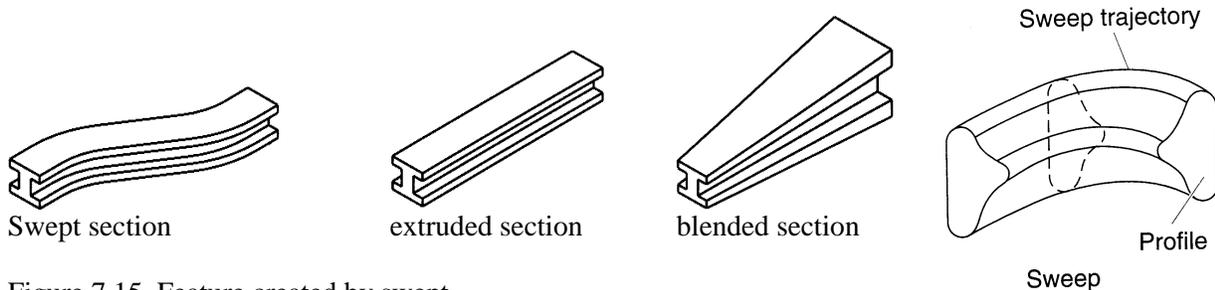


Figure 7.15. Feature created by sweep

The sweep option requires the definition of one section and one trajectory. The order of operation is to create the trajectory first followed by the section. A sweep's trajectory can be either sketched or selected. For many situations, sketching the trajectory is the preferred method. When sketching the trajectory, normal sketching tools are used. The trajectory can be opened or closed. Any planar surface or datum plane can be used as the sketching plane. Due to the nature of the sketching environment, a sketched trajectory is two-dimensional only. A selected trajectory can be used to define a three-dimensional path. A part edge or datum curve can be used as the selected trajectory. If the section should overlap in the bend of the trajectory, you get a failed regeneration. There are three ways to resolve this problem:

- Create a smaller section.
- Create a larger trajectory.
- Make larger radii arcs.

When a swept feature's open trajectory meets with an existing feature or features, the user is provided with the option of either merging the ends of the trajectory or leaving the ends free. The merge ends

option smoothly merges the new swept feature with adjacent solid features. The trajectory has to be aligned with the existing features.

### 7.5.2. Blended Features

A blend is a feature created from two or more planar sections. Blended sections are joined together at their vertex points to form one feature. Three types of blends are available:

- parallel
- rotational
- general.

An example of a parallel blend is shown in Figure 7.16. With this type of blend, each section of the feature is sketched in the same sketching environment. The sections defining the feature are shown in the figure.

The second blend type is a rotational blend. Sketching two or more sections with a rotational angle defined between each section creates rotational blends. Within each section, the user must create a coordinate system. This coordinate system defines the pivot point for each section with sections revolved around the Y-axis of the coordinate system.

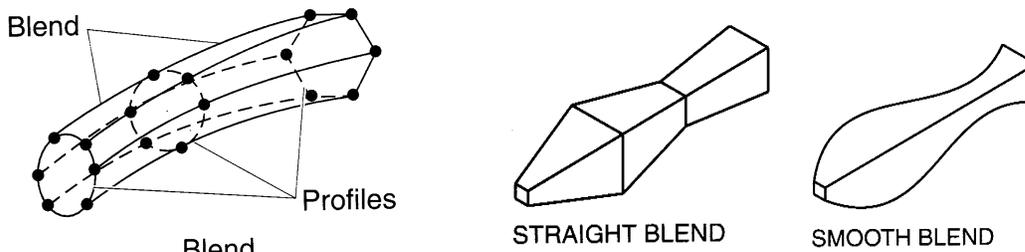


Figure 7.16. Feature created by blended

### 7.5.3. Swept Blend Option

A swept blend is a combination of a sweep and a blend. A swept feature is a section protruded along a defined trajectory. This trajectory can be either sketched or selected. A parallel blended feature is a feature protruded along a straight trajectory between two or more user-defined sections. A Swept Blend feature is two or more sections protruded along a user-defined trajectory (see Figure 7.17). Like a swept feature, a swept blend's trajectory can be either sketched or selected. In addition, sections can be either sketched or selected.

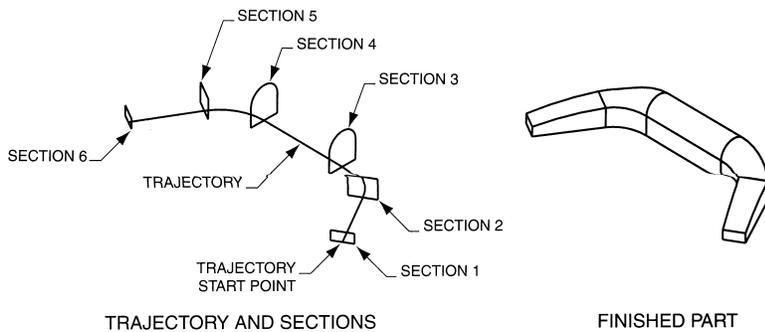


Figure 7.17. Swept blended feature

#### 7.5.4. Helical Sweeps

As its name implies, the Helical sweep option is useful for creating parts that consists of helical features. Two features often created with the helical sweep options are spring and threads. A spring feature is created with the protrusion command and a thread feature is created with the cut command.

#### 7.6. LOFTING

Lofted parts are defined by a number of sections positioned in 3D space. This is similar to sweeping, but a path curve is not used. Lofting is typically used to create objects of varying cross section; however, lofting could also be used to create objects of constant cross section in which the cross section is copied and oriented in space along something other than a straight-line path. (Note that an object that has a constant cross section along a straight-line path is best created by extrusion.) Figure 7.18 shows the initial cross sections and the final objects for two different objects created by lofting. With the first object, the cross sections were varied from one location to the other and were located along a straight-line path. With the second object, the cross section was held constant, but copies of it were oriented in space along something other than a straight-line path.

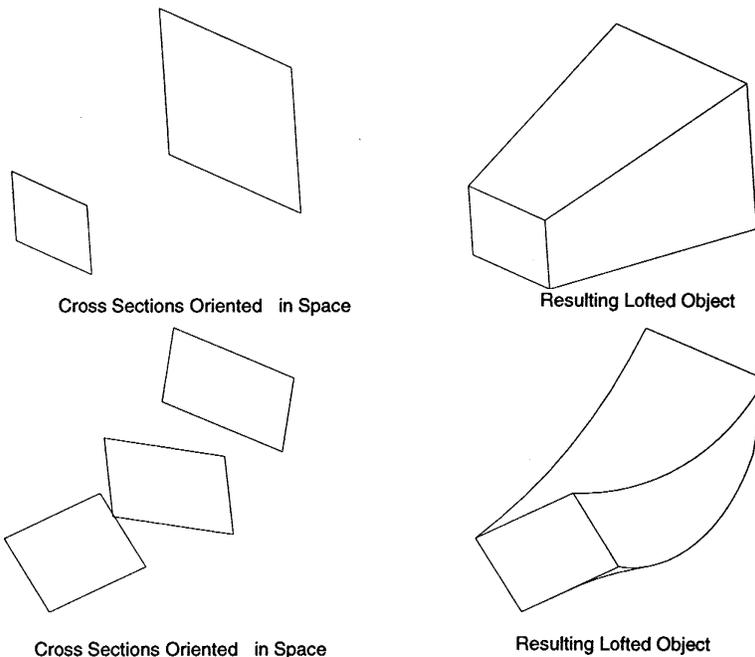


Figure 7.18. Objects created by lofting

When creating lofted objects, you must be careful to make sure that you do not inadvertently create “twisted” parts. Twisted parts will sometimes occur, depending on *how* the cross sections were selected for inclusion in the lofted part. In selecting sections for the creation of a lofted part, you must take care when using the cursor to pick a “similar” location on each of the sections that make up the part.

For this reason, if you select the sections at dissimilar locations, you may end up with a twisted part. Figure 7.19(a) shows two sections oriented in space. Figure 7.19(b) shows the lofted part that results when the first section is selected for lofting with the cursor near location *A* on its perimeter and the second section is selected with the cursor near location *B* on its perimeter. Figure 7.19(c) shows the lofted part that results when both sections are selected for lofting with the cursor near location *A* on their respective perimeters. Note the twisting of the part when the sections are selected at dissimilar locations on their perimeters.

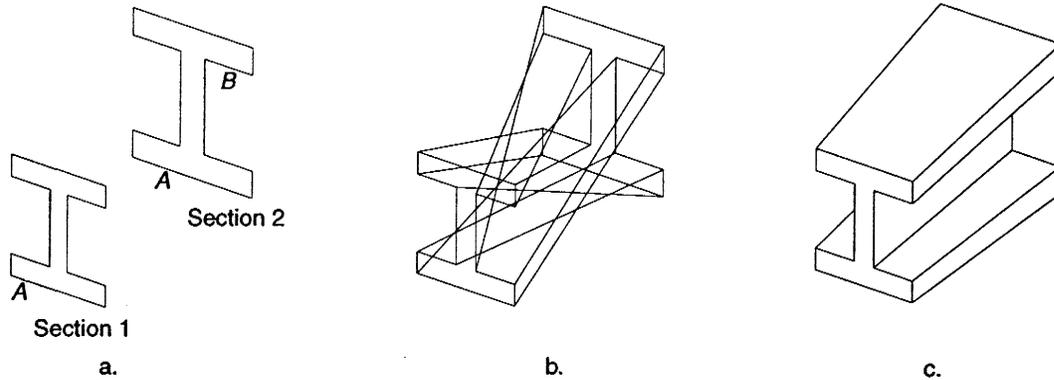


Figure 7.19. Twisted and non-twisted lofted objects

### 7.7. FILLETS AND CHAMFERS IN 3D DRAWING

The *Fillet* and *Chamfer* commands work with part edges. To fillet edges, either select edges directly, or select associated vertices or surfaces to indirectly select edges to fillet. These operations are kept as part of the part history, so you will be able to go back and change the radius values later.

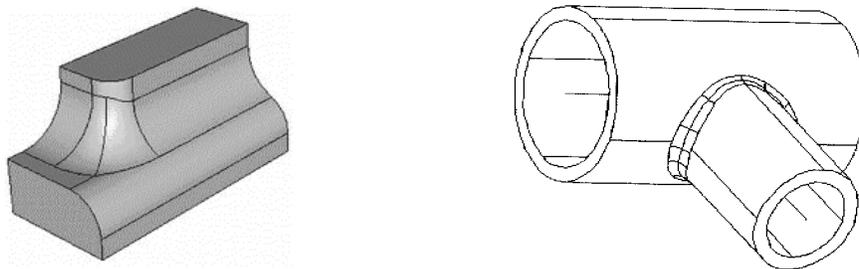


Figure 7.20. Fillets in 3D Drawing

You can pick groups of edges with different fillet radii in the same fillet operation, resulting in one feature. There are also options for setting variable fillet radii at different locations, and changing the “conic parameter” which makes the fillet a conic rather than an arc to make smoother blends.

Fillet operations are order dependent. For example, if you fillet two edges that touch at a corner, you get a beveled intersection. Generally, fillets should be added as one of the last features on a part, unless there are order dependencies, such as when you want to drill a hole through a fillet. The Chamfer command is very similar to Fillet. The difference is that the chamfer cuts straight across the edge.

### 7.8. 2½-D OBJECTS

The term 2½-D is used for those parts which are either flat objects with a constant depth (extruded parts) or for solids of revolution. This limited class of 3-D entities can be generated by an extension of 2-D techniques and was, until recently, the only type of 3-D object that could be generated on smaller CAD systems. Flat objects can be considered as 2-D shapes, which are translated, swept or extruded, into the z dimension. Figure 7.21(a) shows a typical example. Revolute objects can also be produced by sweeping a 2-D profile through 360° as shown in Figure 7.21(b). The amount of storage needed for 2½-D objects can be minimized by displaying in a wire frame or a faceted form.

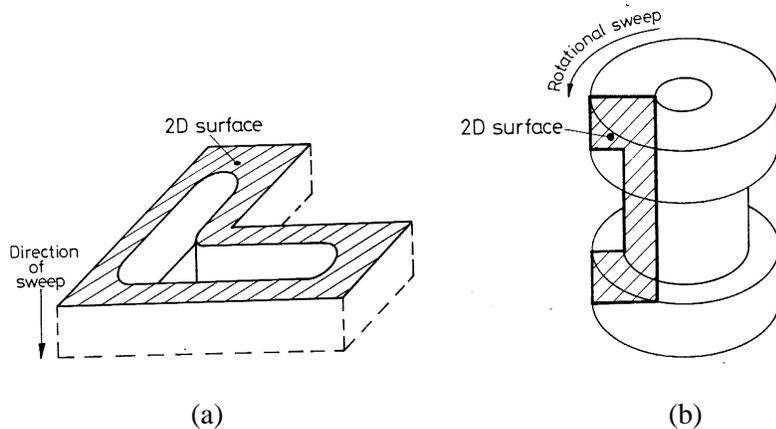


Figure 7.21. 2½-D rotational shape

### Questions

1. Define the three types of Boolean operations and sketch examples of each one. Can you derive the same final object using different Boolean operations and/ or primitives?
2. If the system has a solid modeling capability is it of the CSG, B-rep or dual representation variety?
3. Describe different ways a profile could be swept about an axis to form a solid. Will all such profiles be valid objects in all modelers?
4. Describe how parametric and instancing speed up model construction. Can both be used together in constructing a model?
5. Describe two different ways a tool solid can be defined for use in Boolean operations. Can more than one tool solid be defined at one time?
6. Describe the basic principles behind a swept blend feature.
7. Name two methods for creating the trajectory for a swept blend feature.
8. What two geometric features are often created with the helical sweep feature?
9. What two methods are available for defining a sweep's trajectory? Describe appropriate situations for using each method.
10. Describe the three types of blends available within the blend option.

### References

1. M.H. Lawry, **I-DEAS Student Guide**, McGraw-Hill Higher Education, 2001
2. D.S. Kelley, **Pro/Engineer Instructor**, McGraw-Hill, 2001
3. R.W.Lueptow, M.T.Snyder, J.Steger, **Graphics Concepts with Pro/Engineer**, Prentice Hall, 2001
4. S.A. Sorby, **Solid Modeling with I-DEAS**, Prentice Hall, 2000.
5. G.R.Bertoline, et.al., **Technical Graphics Communication**, WCB McGraw-Hill, 1997
6. J.Rooney, P.Steadman, **Principles of Computer-aided Design**, UCL Press, 1997
7. C.McMahon, J.Browne, **CAD/CAM From Principle to Practice**, Addison-Wesley, 1995.
8. A.Tizzard, **An Introduction to Computer-aided Engineering**, McGraw-Hill, 1944.
9. D.A. Madsen, T.M.Shumaker, J.L. Turpin, **Fundamentals of Drawing Technology**, Delmar Publ., 1994.
10. F.E. Giesecke, et.al., **Engineering Graphics**, Prentice Hall, 2000.
11. F.E. Giesecke, et.al., **Modern Graphics Communication**, Prentice Hall, 2001.
12. N.N. **I-DEAS Master Series Student Guide**, SDRC, 1998.
13. D.E. LaCourse, **Handbook of Solid Modeling**, McGraw-Hill, Inc. 1995.