

## 14. Surface Modeling

A wire frame model consists only of edges of surfaces or lines of intersection of surfaces but does not contain information about the areas between the lines. For a 3D surface model, you must define the surfaces mathematically. Several other surfaces, such as cylinders and spheres, have mathematical expressions that can be written to describe them. Most real surfaces cannot be defined by a single mathematical expression. In some cases, it may be adequate to break the non-plane surface into a large number of plane facets. You can establish the location of surface points as precisely as necessary by this method.

### 14.1. HISTORICAL BACKGROUND

Surface modeling was essentially the situation in the early 1940s. The pressures of wartime production, particularly in the aircraft industry, led to changes in the way the geometry was represented. It was found possible to avoid working to full size, and to represent fuselage cross-sections in terms of mathematically defined curves such as conics. Subsequently, mathematical techniques were devised allowing these cross-sections to be blended or interpolated into a smooth surface. Thus the 'network of curves' approach was gradually superseded, becoming replaced by methods, which actually described a complete surface. A book by Liming [1944] explains many of the techniques used at this time.

The transition to a more mathematical procedure naturally brought with it a demand for increased computation. The use of mechanical or electro-mechanical desk calculators initially met this, but with the advent of powerful electronic digital computers the methods became increasingly automated. Lidbro [1956] describes a system used by Saab-Scania in Sweden in the 1950s, a developed version of which was still in use on mainframe computers in the 1980s, based closely on the approach described.

The use of parametric techniques demands greater computing resources but has numerous advantages, as will be described. Such methods became popular in the 1960s, largely due to the pioneering work of Coons [1964]. One of their most important features is that the mathematical formulation of curves and surfaces defined parametrically does not change under transformations such as translation, rotation or reflection.

In the 1960s surface generation was essentially a batch process, and the output had to be judged by interpretation of tables of numbers. Nowadays, however, computers are fast enough for surface design to be an interactive process, and the graphics terminal allows designers to judge the acceptability of their results as they proceed.

Although the earliest surface systems were developed within the aircraft industry and demanded very large and expensive computer installations, modern systems are used in a wide spectrum of industrial applications and now have comparatively modest hardware requirements by present-day standards. Their use is widespread in shipbuilding, automotive manufacture and the shoe industry, and they are rapidly spreading into numerous small and medium sized companies manufacturing forgings, castings and molded products.

### 14.2. WIRE-FRAME GEOMETRY

The first of the 3D schemes, and computationally the most straightforward, is the wire-frame scheme. In this the geometry is defined as a series of lines and curves representing the edges of, and perhaps sections through, the object. The name of the scheme arises from the wire-like appearance of the models when viewed on a computer-screen or hard copy.

Wire-frame representation may be regarded as an extension into a third dimension of the techniques used for drawing. The entities used are generally the same as those used for drawing, although the data stored to define the entities must be extended. For points and lines this simply means adding a Z value to the coordinate data, but for arcs and other planar curves the plane in which the curve lies has to be defined. This might involve, for example, referencing the work coordinate system in which an arc is defined.

The construction techniques used for the definition of wire-frame geometry are again broadly similar to those for drawing, but with certain extensions. One that has already been mentioned is the use of movable work coordinate systems, and in general a number of ways of defining WCS from existing points or from existing coordinate systems are provided. Some examples of methods offered by one CAD system. Associated with the WCS are the concepts of the work plane and the working depth. Often, planar entities such as arcs and conic sections are constructed by default in a plane (the work plane) parallel to the X-Y plane of the WCS at a Z-axis value equal to the working depth. The work plane is also used for projected intersections and points: it is often useful to be able to use apparent intersections of entities when viewed along the z-axis, even though the entities may not physically intersect in space. In such cases the intersection points are projected into the work plane.

#### **14.2.1. Wire-frame modeling**

Very often designers also build physical models to help in the visualization of a design. This may require the construction of 'skeleton' models using wires to represent the edges of an object or component. *Wire-frame modeling*, as used currently in computer-aided engineering techniques, is the computer-based analogue of this process, and is used not only to model objects but also to facilitate the production of various projected views to aid visualization.

Essentially, a wire-frame model consists of a finite set of points together with the edges connecting various pairs of these points, so that a drawing of these points and edges aids the visualization of the object. The usefulness and ubiquity of wire-frames in computer-aided engineering drawing systems from the computer's ability to calculate quickly and accurately the positions of the points (and hence also the edges) on the screen or plotter-paper, when a view of a wire-frame object is required from some viewpoint. Objects represented by point (and associated edge) data can be manipulated to produce any required orthographic, isometric or perspective projection with equal ease. Since these projections make visualization easier, and since point and edge data are always available for even the most rudimentary manual drawing of a three-dimensional object, wire-frame models have entered the standard tool kit of the computer-aided designer.

Although in some ways a drawing system is a two-dimensional wire-frame modeler, the actual geometric information implicitly used by designers when drawing, or when producing orthographic projections (say), is usually greater than that used to define a wire-frame model, since (for drawing) they use their knowledge of the 'solidity' of the object and of its bounding surfaces. Thus a designer may draw a dotted line on a front view of an object to show that a particular (rear) edge cannot be seen, but only a human reader could interpret this as a hidden line. The computer drawing system has no three-dimensional model of the object with which to make any inference of this kind. Even a wire-frame model does not usually embody this type of additional information about the 'real' object. It is a representation which lies somewhere between that of a typical engineering drawing (whether produced manually or by a drawing system) and the more complex object representations which embody knowledge of surfaces and underlying solidity.

There are two important aspects to the use of wire-frame models in computer-aided design. The first is the computer representation of an object, and this is concerned with the structure needed to encode a wire-

frame model. The second is concerned with the computational procedures needed to produce and manipulate the viewing or visualization of this representation. We will concentrate here entirely on the first of these aspects.

### 14.2.2. Wire-frame representation

A wire-frame representation of a three-dimensional object consists of a finite set of points and connecting edges, which define the object ‘adequately’ and facilitate subsequent visualization. The term ‘edge’ does not necessarily imply that it is a straight-line segment. An edge may well be an arc of a circle or any other well-defined space curve, which is required for a ‘good’ wire-frame representation.

The geometric complexity of the representation of an object is dependent on a trade-off between what can be represented and computed easily and quickly, and what is required for a reasonably faithful and useful representation. The wire-frame offers a relatively simple three-dimensional modeling technique for relatively small computing resources and overheads. However, there are certain disadvantages associated with wire-frame modeling techniques (such as *ambiguity* and *lack of validity*), which make this type of representation unreliable in general. We will return to a discussion of these disadvantages later, but first we discuss the details of a wire-frame representation using several simple objects.

A computer representation of a wire-frame structure consists essentially of two types of information. The first is termed ‘metric’ or ‘geometric’ data which relate to the three-dimensional coordinate positions of the wire-frame node’ points in space. The second is concerned with the ‘connectivity’ or ‘topological’ data, which relate pairs of points together as edges.

### 14.2.3. Real objects and wire-frame models

Wire-frame structures can be created and manipulated as entities without any consideration for the object(s) they may be used to represent. However, in most cases three-dimensional wire-frames are used to model objects in the real world, providing (amongst other things) a tool to aid object visualization. In general this modeling process can be likened to the increasing use of constraints, which limit the allowable data structures available to represent true objects. A general wire-frame such as each of those given in the previous examples usually has associated low-level constraints. Thus each point must have three coordinate values, and each edge must be associated with only two points, which must both be defined in the point list. Additionally, real object representation effectively involves further constraints on the wire-frame structure. These constraints may be left to the user to maintain (often leading to error), or, more commonly, they may be embodied in checking algorithms, which process a wire-frame structure in order to determine its suitability in representing an object. A third possibility is that they can be built into the interactive tools used to create the wire-frame.

Even supposing that all of these requirements are satisfied, three-of-a sphere is dimensional wire-frames still have failings in two major areas: *lack of validity* and *ambiguity*. Consider the object shown in Figure 4.1. This can best be represented by the type of structure described above, consisting of simple (vertex) points and linear edges. Each point is well defined in three-dimensional space; each edge is associated with just two end-points; the face edges all form closed loops; and no faces are self-intersecting. However, despite these constraints, this wire-frame is essentially nonsense. We would never be able to manufacture such an object. In other words, the wire-frame does not represent a valid solid entity. We could argue that imposing further constraints such as ‘faces form closed volumes’ would not allow such objects to be defined, but this is effectively moving away from what we may term pure wire-frames into the area of solid modeling, where the requirement from the outset is the creation of representations of real solid objects within computers.

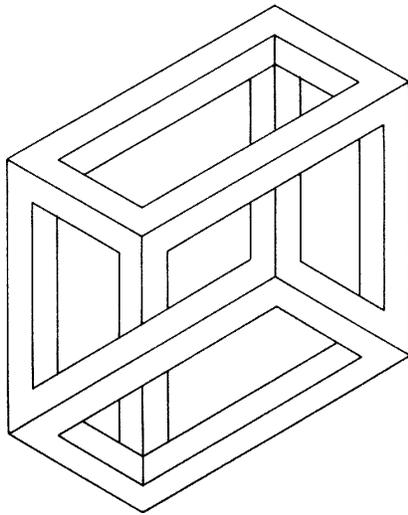


Figure 4.1. A wire-frame nonsense object

The second failing is that of ambiguity. Essentially it is possible to produce a general wire-frame, which can be interpreted to represent more than one real object. An example is shown in Figure 4.2. Again this wire-frame conforms to the constraints mentioned earlier, and in addition we can argue that the faces formed by edge-loops do in fact form closed volumes, but which volumes are they? The wire-frame is ambiguous and can represent any of the other real objects shown in Figure 4.2. Again we can argue further that more sophisticated constraints can be built into the wire-frame structure which eliminate this type of ambiguity, but this takes us into the realm of solid modeling theory which was originally driven by such problems in wire-frame modeling for real object representation.

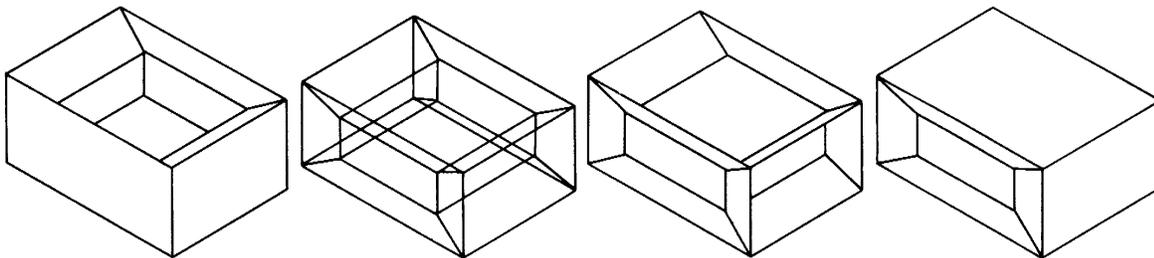


Figure 4.2. An ambiguous wire-frame object

One final point to be made is that a simple wire-frame model incorporating curvilinear edges produces several difficulties. The most serious is the problem of its *silhouette*, *horizon*, or *profile curve*. Generally this will not be the same as one of the edges and it must be calculated separately from knowledge of the wire-frame edges and the proposed viewpoint.

### 14.3. SURFACE MODELLING

Surface modeling defines a component with greater mathematical integrity as it models the surfaces to give more definitive spatial boundaries to the design. It is particularly useful for modeling objects, which can be modeled as shells, such as car body panels, aircraft fuselages or fan blades. Figure 4.3 is an example of the kind of modeling application for which surface modeling is ideally suited. A complex surface can be very difficult to visualize without a physical model and surface modeling using a computer eases this process considerably. Before computer-aided surface modeling existed, automobile profiles had

to design by making a clay model of the car bodywork and by trial and error methods this model was shaped until aesthetically and functionally satisfactory. The geometry of the body panels was then determined by physical measurement of the clay. With surface modeling and the facilities which exist within a good surface modeling package, the aesthetic shape and the three-dimensional geometry are defined in one process and can be altered or sculpted interactively on the graphics screen. The basis of a surface is the definition of a curve in two dimensions in a given plane. This can then be *swept* along another curve defined in another plane.

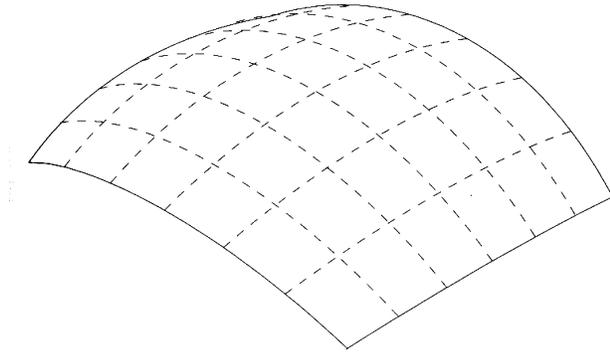


Figure 4.3. A surface modeled shell

It is the definition of the sweeping and swept curves which determine the characteristics of the *surface patch*. The surface modeling software should contain curve-fitting techniques such as Bezier curves, cubic splines and B-splines. The method of operation curve fitting techniques is to define a number of control points in the required plane and fit the desired curve type through the points. The shape of the curve can then be accurately controlled by adjusting the positions of the control points, which provides the interactive sculpting facility.

As a surface model defines adequate data on a component's surface geometry hidden lines and surfaces are readily and automatically removed as required. This gives rise to non-ambiguous visualization of the object when viewed from any direction. It is also possible to render or shade the view, which is a valuable visualization tool available to a designer as it enables an almost real image to be displayed. The principle of shading is to create an imaginary light source to shine on a three-dimensional image. The amount of reflectivity from the object's surface can be adjusted for greater realism. The software calculates the amount of light reflected back to the user from different areas on the surface and each area is color filled with varying shades accordingly.

Many of the ambiguities of wire-frame models are overcome by using the second of the three main 3D representation schemes — surface modeling. As the name implies, this scheme involves representing the model by specifying some or all of the surfaces on the component. Once again the representation generally involves a series of geometric entities, with each surface forming a single entity. These are often constructed from surface edges and curves on the surface, and so surface representations are often mixed with, or developed from, wire-frame representations.

The most elementary of the surface types is the flat plane, which may be defined between two parallel straight lines, through three points or through a line and a point. Figure 4.4a shows an example of a plane as displayed by a commercial CAD system.

Figures 4.4b – Figure 4.4g show other examples of surface types commonly used in CAD, which are:

- A tabulated cylinder, which is defined by projecting a generating curve along a line or a vector.
- A ruled surface, which is produced by linear interpolation between two different normal, generating or edge curves. The effect is of a surface generated by moving that straight line with its end points resting on the edge curves.
- A surface of revolution, which has been generated by revolving a generating curve about a centerline or vector. This surface is particularly useful when modeling turned parts, or parts which possess axial symmetry.
- A swept surface — in a sense an extension of the surface of revolution — (Figure 4.4d) the defining curve is swept along an arbitrary curve instead of a circular arc.
- A sculptured or curve-mesh surface. This is among the most general of the surface types and is defined using a family of generating curves, or two families intersecting in a crisscross fashion so creating a network of interconnected surface patches.
- A fillet surface, which is analogous to the fillet arc in curve construction, and is defined as a surface connecting two other surfaces in a smooth transition (generally of a constant, or smoothly changing, radius of curvature).

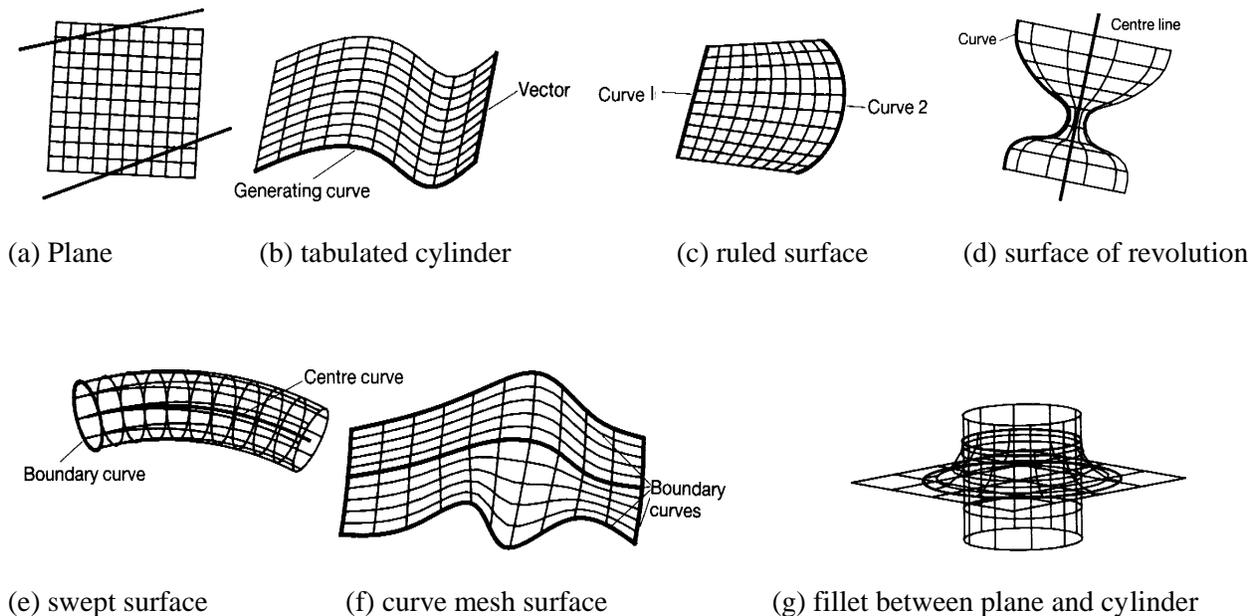


Figure 4.4. Examples of surface types.

The ‘core’ surface types are the ruled surface, surface of revolution and sculptured surface, as the others may be defined using these. In each case it should be noted that the surface is drawn as a mesh of intersecting curves on the surface. This is only for display purposes — the surfaces are continuous, with every point on the surface defined by the mathematical relationship used in its definition.

### 14.3.1. Surfacing Methods

Surfacing methods are divided into two categories – solid based creation techniques and manual surface creation tools. These techniques create solid parts and features directly, much like Extrude command, only using more than one section in the definition.

The second category of surfacing tools is manual techniques to create surfaces from wire frame geometry. These tools will require more steps to create solid parts from the bottom up, starting with wire frame geometry to create surfaces, and using surfaces to create solids.

### 14.3.2. Solids-Based Surfacing Tools

The techniques of sweep and loft work very much like extrude and revolve. They use wire frame sections to directly create new solid parts or features protruded or cut from other parts. The major difference is that these commands use multiple sections to define the feature. A part such as the boat hull below is conveniently described by a series of 2D cross sections (see Figure 4.5).

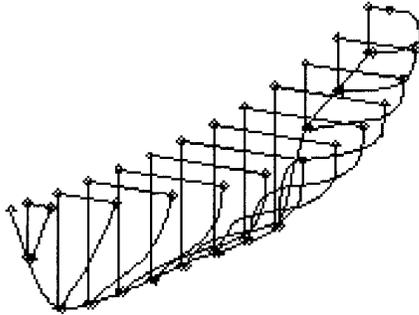


Figure 4.5. Solid-based surfacing technique

Like *Extrude* and *Revolve*, sections can be built first or created “on the fly” by the command that uses them. They can be thought of as just a grouping of curves. In most cases where the software asks for a section, you can either pick an existing section or pick the curves directly.

### 14.3.3. Manual Surface Creation Tools

Although it is usually preferable to create parts using solids-based tools, there are applications where surfaces must be created and manipulated manually. For example, parts like the surface of the fan blade in Figure 4.6a. are not easily defined by flat cross sections. This fan blade is more easily defined by a mesh of points or curves in 3D space generated in cylindrical coordinates.

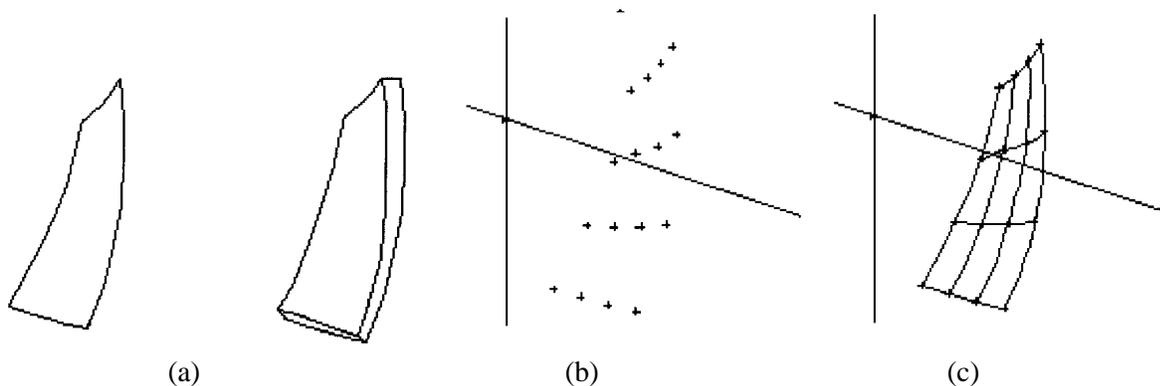


Figure 4.6. Manual surface creation

Creating parts from surfaces is a multi-step process. First, the wire frame curves are created and manipulated. Next, these curves are used to define surfaces and/or solid parts. One more step is necessary

if you want to generate a solid part from the resulting open surface. This might be a shelling operation to give the surface thickness, or stitching multiple surfaces together to form a closed solid.

You might start with a grid of points as shown in Figure 4.6b. These points could be used to define a grid of curves in two directions along the surface as shown on the right. The curves do not actually have to touch, because the program will average between them if they don't. Parts, which converge to a point, can be defined by picking the same end points for each column of points, or by selecting curves that converge on the same end points.

## 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. O. Ostrowsky, **Engineering Drawing with CAD Applications**, ELBS Pub., 1993.
13. S.J. Ethier, C.A.Ethier, **Instant AutoCAD Mechanical Desktop 5.0**, Prentice Hall, 2002.
14. R. Pizza, **Getting Started with Pro/Engineer**, Prentice Hall, 2002.