GPH 259 Design Geometry
Instructor: Stephen Luecking

# Surfaces

Fields other than engineering and mechanical applications use surface modeling programs. Surface modelers represent 3D solids as surfaces enclosing a space. The solid is hollow as if it were constructed of paper sheet or from rubber like a balloon. For this reason this chapter will include solids as it looks at surfaces. Solid modelers treat 3D forms as solid chunks of space as if the solid were made of metal. This is useful for fields where the final objects will be machined or extruded metal.

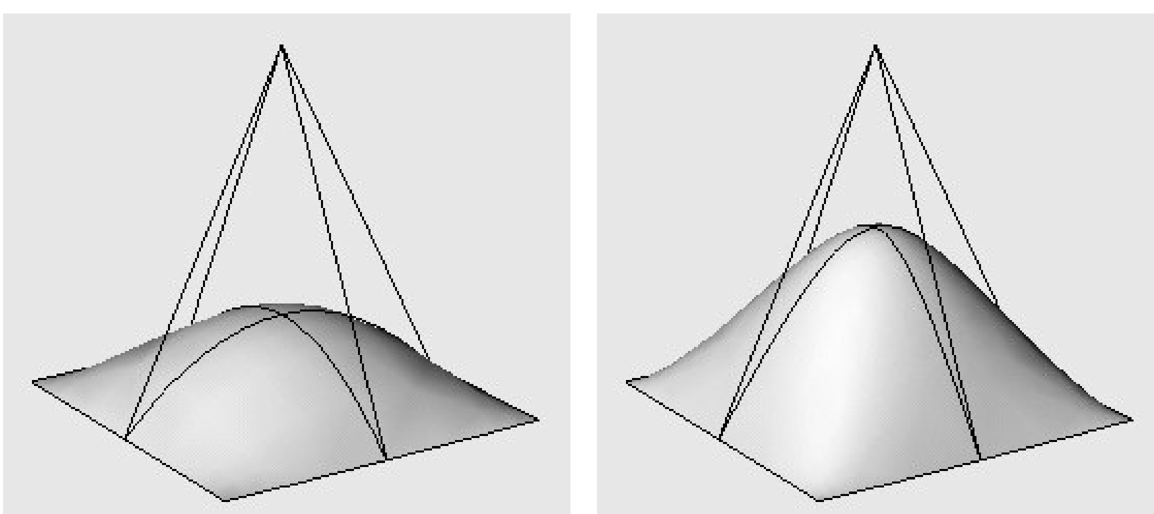
Surface modelers come in two categories: NURBS modelers and polygonal modelers. NURBS modelers build their surfaces by interpolating then through a series of spline curves. Polygonal modelers represent surfaces by constructing it from a mesh of small polygons. Polygonal surfaces are thus faceted and it is the number of polygons employed that condition the detail and smoothness of the surface.

## I. NURBS Surfaces

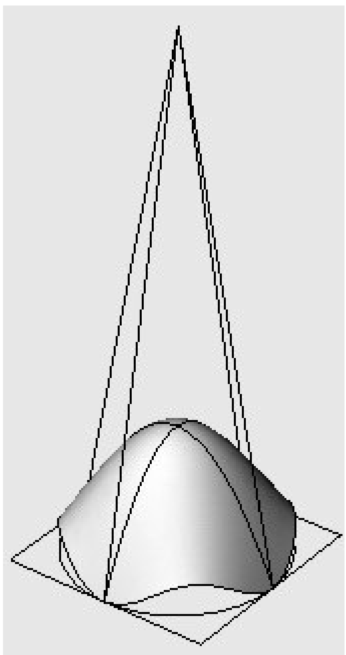
Interpolating a surface through a series of lines is called lofting. Like other NURBS terminology this, too, goes back to shipbuilding. After the lofters drew out the full size spline curves of a boat’s hull these curves were cut out, raised and positioned in the shipbuilding loft. Builders then laid wood planks across a series of these curves, thus creating the form of the hull. The term lofting grew to refer to the process of generating the external geometry of ships and, later, airplanes and cars. In NURBS modeling this term encompasses any surface generation from curves.

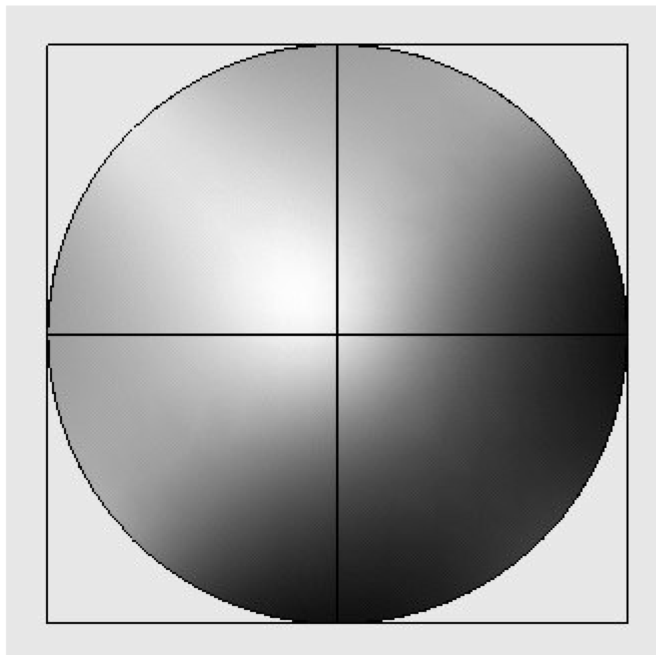
The development of NURBS surfaces was one of the most important advances in computer graphics. Computers could represent surfaces like that of a balloon or human skin or the streamlining of birds and fish without the subtle stoppages of tiny angular elements. And they could do so efficiently.

### *Anatomy of a NURBS Surface*

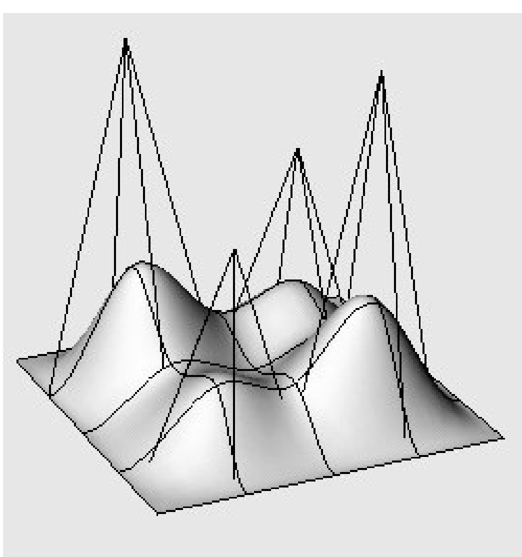


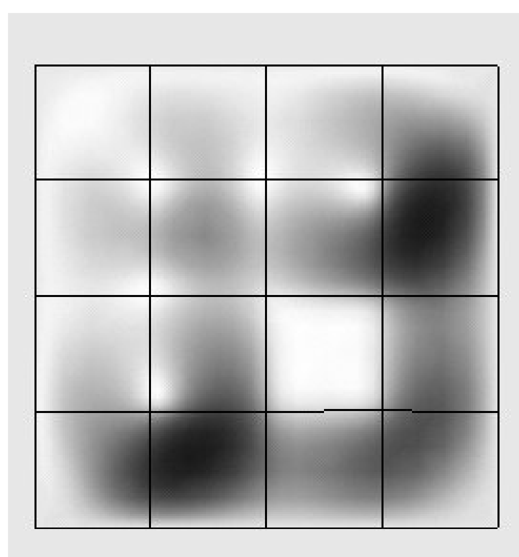
Control points shape NURBS surfaces as they do NURBS curves. Instead of two tangent lines connecting the external point to the on-surface points, there are four, positioned like the declining edges of a pyramid. In the images above the off-surface point connects to four surface points. In addition there are four more points that pin down the corners of the surface. As with NURBS curves the control points are weighted. In the illustrations above the first control point is valued at 1.0 and the second at 3.0. Note that it takes more weight to deflect a surface as much as a curve.



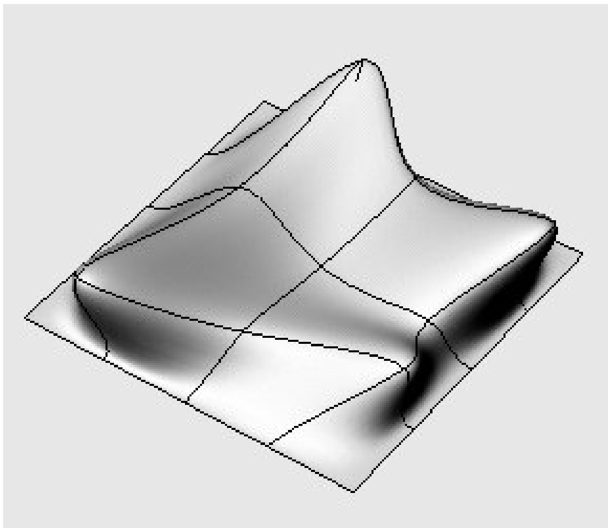


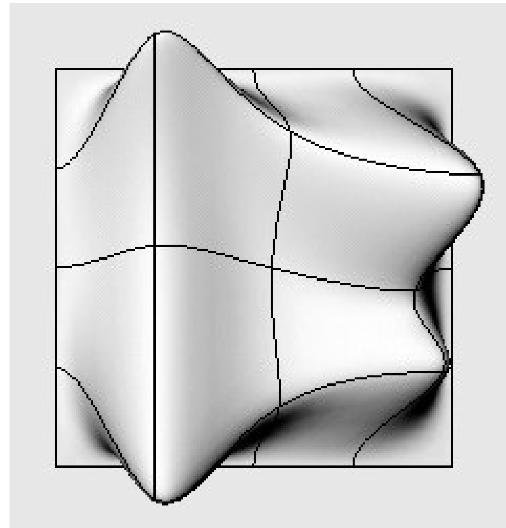
This “square” geometry is holds for other geometric shapes as well. A circular surface distends toward the control point as if it were a portion of a square surface. (Note in the second rendering above how the surface is still circular in plan.) As a result the four control points at the corners still condition the surface.





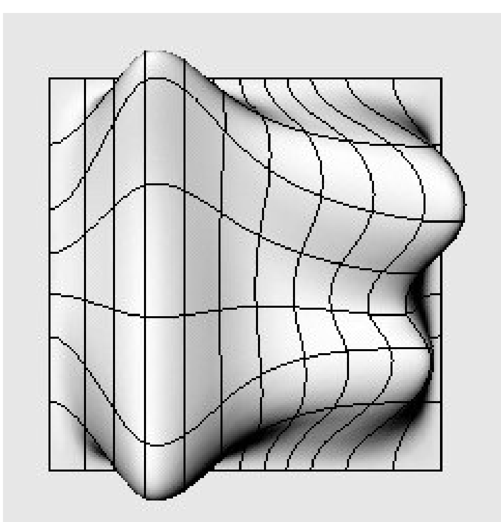
This geometry allows the NURBS surface to develop piece-wise along a grid-like network. (Like NURBS curves NURBS surfaces build piece-wise.) The surface follows the curves of this linear net, whose lines will deflect and tilt toward the control points.

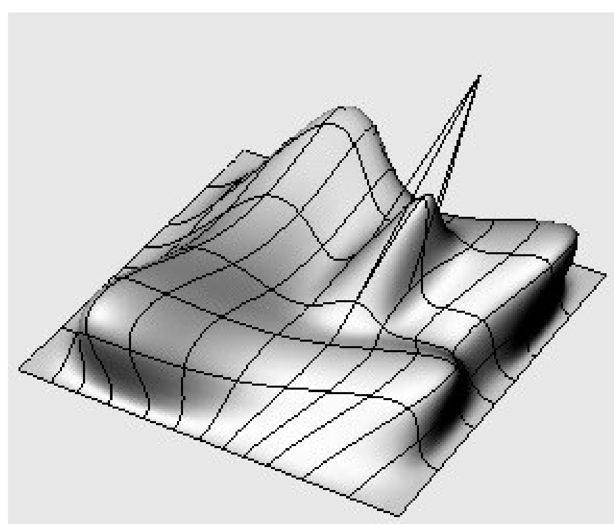




***NURBS Topology***

The directions of this NURBS "grid" are labeled U and V. Any line that runs in the U or V directions is called an isoparm. There are an infinite number of these on any NURBS surface and all are spline curves. This two coordinate method of demarcating a NURBS surface points out its essentially 2D character, something like the X,Y coordinate plane. (The study of the differences and relationships between the X,Y organization of a plane and the U,V organization of a surface belongs to a field of mathematics called *topology*.) Despite this 2D character it needs a 3D space in which to exist. The third dimension is required to deflect it from a plane into a curved surface, just as two dimensions are needed to deflect a straight line into a curve. For this reason topologists refer to 3D space as a two-dimensional manifold, that is, a space in which a 2D object is free to fold, twist, drape and the like.

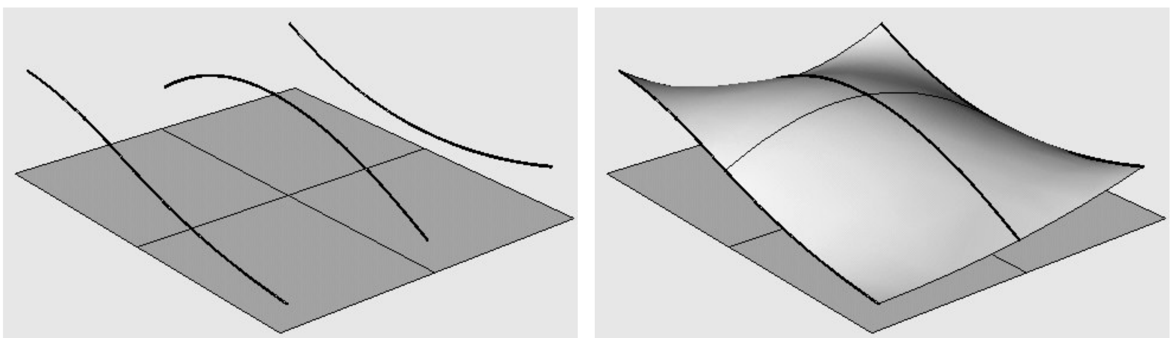




Many programs allow us to *rebuild* a surface by increasing lines in the U and V directions. The surface above is a rebuilt version of the previous surface. Eight lines now appear in the V direction and 12 in the U direction. Along with the increased lines comes an off-surface control point for each line crossing. In effect, rebuilding is the process by which the program selects new isoparms to assign control points. This permits increased and more detailed edit options. In the second example one of the new control points has been drawn outward to add a smaller, sharper protrusion on the surface.

### *Lofting*

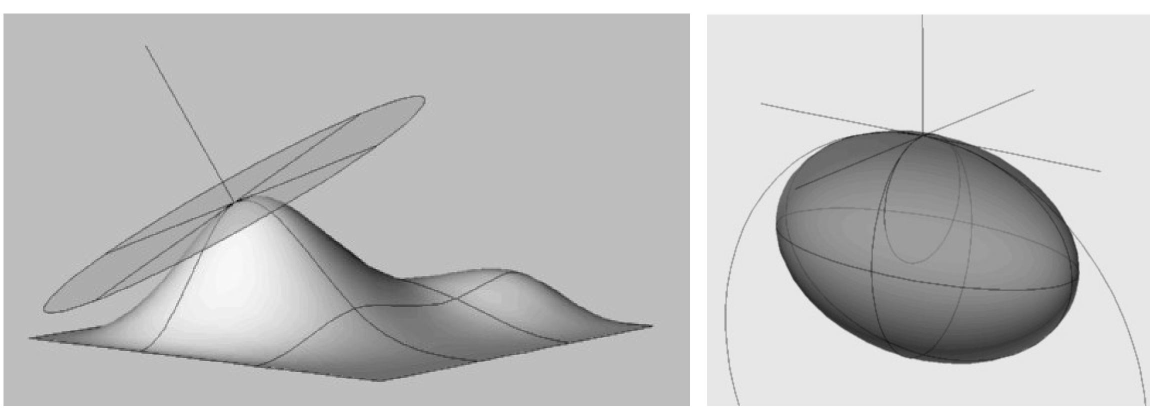
This coordinate method of piecing is elegant in conception, but editing it with control points tends toward lumpy and unnatural appearing surfaces. Despite its mathematical necessary any designers find this topology to be frustrating. Consequently the vast majority of modeled surfaces result, in practice, by lofting between curves.



Most surface generating tools in NURBS programs are computed as lofting. Sweeps, revolves and extrusions are lofts in which the lofted curves are moving versions of the original, or on which constraints are placed on the path of the loft. A revolve, for example can be regarded as a loft through rotated versions of the generating curve.

### *Tangent, Normals and Curvature*

Normals can be drawn from a curved surface, much as they are drawn from a curve. They are perpendicular to a line tangent to the surface. The exception is that a curved surface can possess an infinite number of tangent lines through a given point. All of these lines will lie on the same plane – the tangent plane. The normal of a surface is the line perpendicular to this plane at the point of tangency.



Because NURBS surfaces feature constantly changing radius, the normal is also the unique radial of the circle of curvature for that point. If we assign a circle of curvature to each of the infinite tangents, we then get an infinite set of circles that share a common radial – the normal line. Curvature moves in all of those directions. This property of curving through multiple planes of space simultaneously is called double curvature.

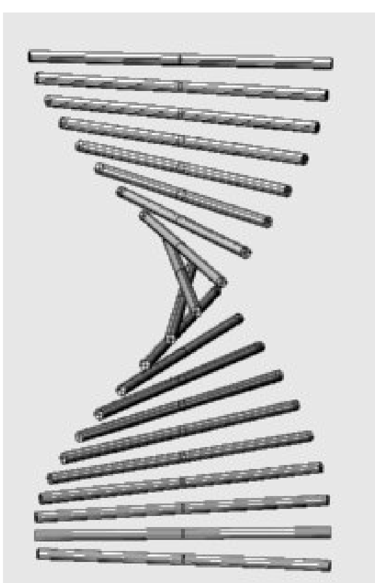
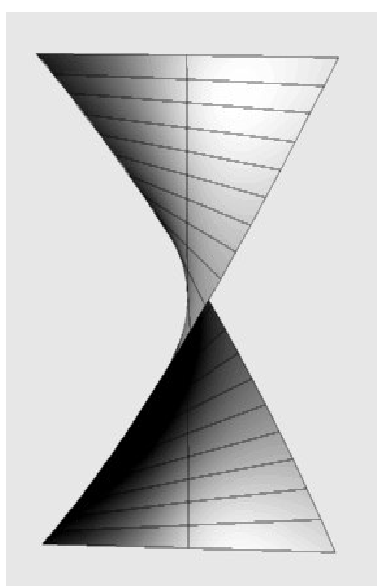
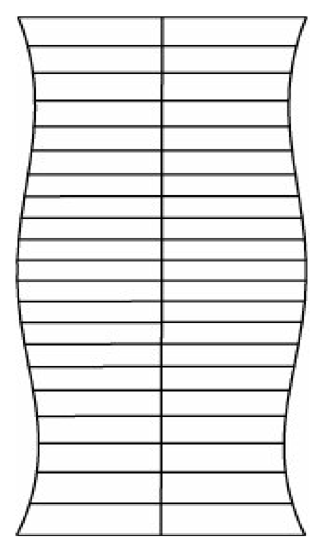
It is remarkable that a single point on a surface can possess different curvature in different directions, up to an infinite number of radii. On a sphere the curvature circles at a given point are all the same. They are great circles, or *geodesic* lines, of the sphere. Re-scale the sphere into an ellipsoid, though, and the geodesic lines take on elliptical curvature of infinitely varying degrees. In the ellipsoid above two curvature circles are indicated for a point on the equator of the solid. The *latitudinal* (sideways) curvature displays a very tight radius in contrast to the *longitudinal* (length-wise) curvature. Rotating the angle of curvature between these two directions will yield an infinite number of gradually enlarged circles.

## II. Generated Surfaces

Prior to NURBS modeling the chief means of creating curved surfaces was to generate them. This means that each began as a line moving at cross-angles through space and sweeping out a surface as it moved. Generated surfaces were common in computer modeling prior to NURBS geometry, though the variety of these surfaces has increased with its inception.

### *Ruled Surfaces*

One classic example is a ruled surface. This category of surface generates by moving a straight line through space. In the ruled surface pictured below the line moved upward as it rotated thereby sweeping out a twisted surface.

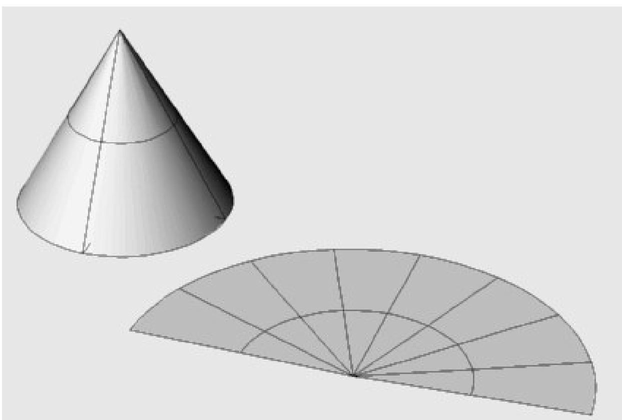




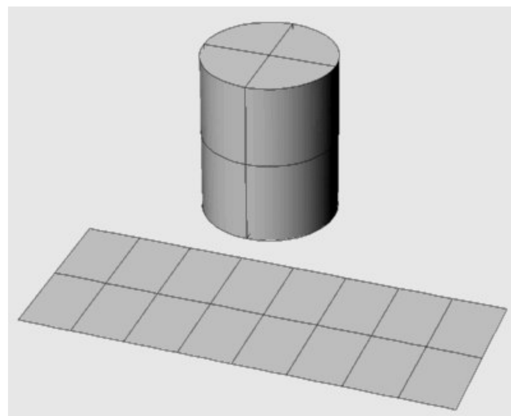
A NURBS modeler would create this surface by lofting through a sequence of straight curves. The fact that there is no off-surface control and the fact that it can be defined by a sequence of straight lines marks its *degree of curvature* as 1 in NURBS terminology. This relationship is reciprocal: all degree one curvature surfaces are ruled surfaces.

### *Developable Surfaces*

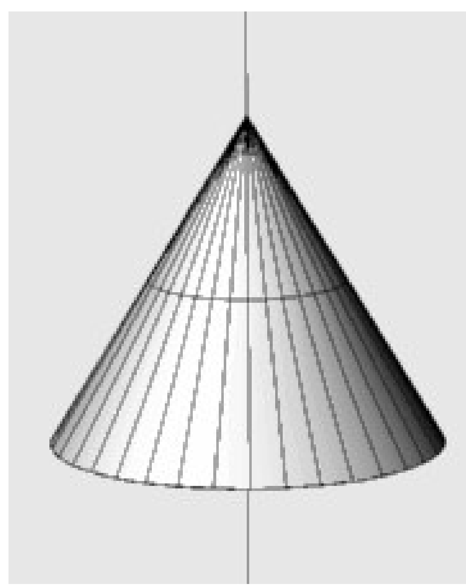
Ruled surfaces are also characterized as *developable*. This means that they can unroll flat with no distortion of their shape. These surfaces possess single curvature and can be rolled from a sheet of material like paper or sheet metal. The third image above depicts the unrolled version, or pattern, of the ruled surface shown in the first two images.

The curved surfaces of cylinders and cones are developable. These roll out into the patterns below:



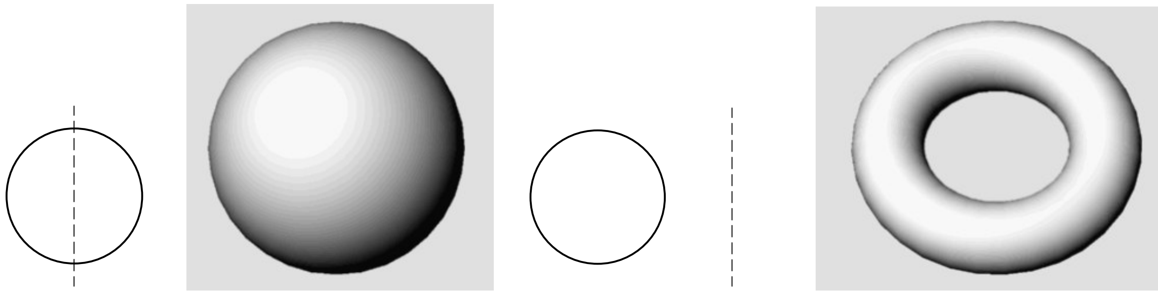


### *Surfaces of Revolution*

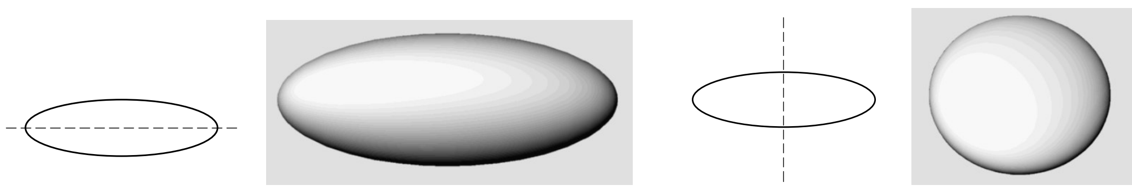


Rotating a straight line around a central axis can generate the surfaces of the cone and cylinder. So these, too, are ruled surfaces. They also belong to a large class of 3D figures called surfaces of rotation. Many common shapes belong to this category: vases, bottles, spinning tops and any other form that can be generated by rotating a profile curve around an axis.

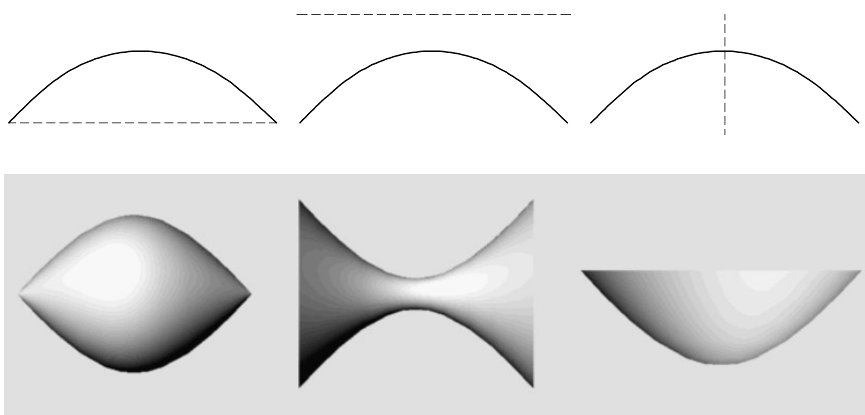
The big tunas in this category are the spheroids. This is the category of geometric solids generated by revolving one of the conic sections around an axis. Revolve a circle around its diameter to generate a sphere. Revolve that same circle around an axis outside of its perimeter to generate a torus, the figure made famous by donuts and inner tubes.



Also featured in this category are ellipsoids:



And hyperboloids along with their close cousins, the paraboloids:



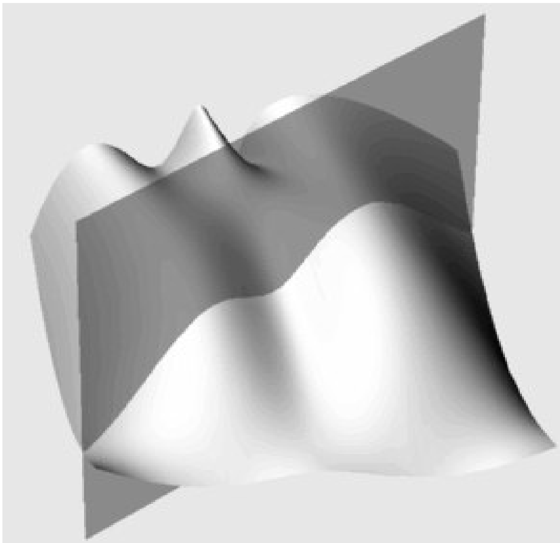
A surfaces of revolution generated from a curved line was the genesis of the term double curvature. Since the path of generation is curved the surface possesses the curvature of the generating line as well as the curvature of the path.

***Surface Curves***

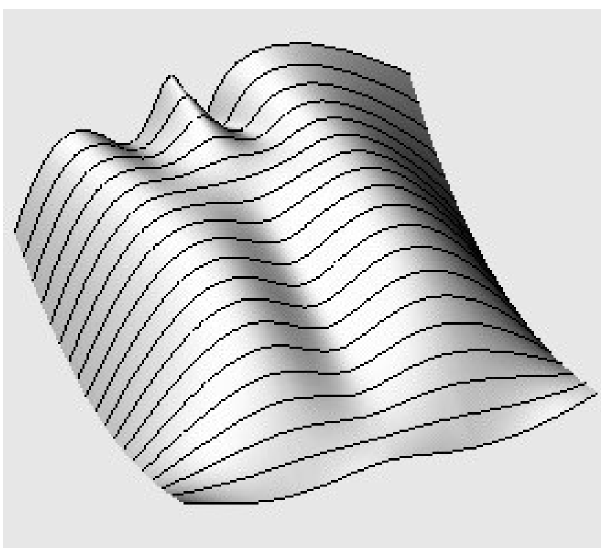
Geodesics, aka great circles, of a sphere are an example of surface curves as are the isoparms of a NURBS surface. Other curves of great use to designers are edge curves, section curves and contours.

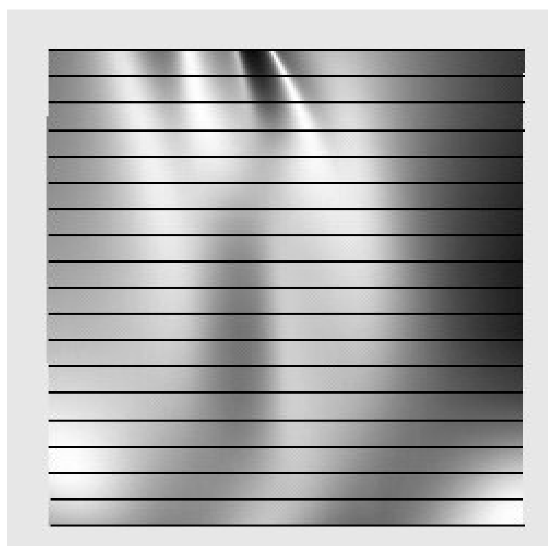
Edge curves are pretty self-explanatory. These are the curves that lie along the edge of a surface. They are most useful when it comes to joining two or more surfaces into larger more complex surfaces. For example, with these the designer can loft a surface between the edges of two other surfaces, creating an intermediary patch.

Section curves are the curves that result from the intersection of a plane with a surface. The conic sections are good examples, as they are the class of curves defined by the intersection of a plane with a conical surface.

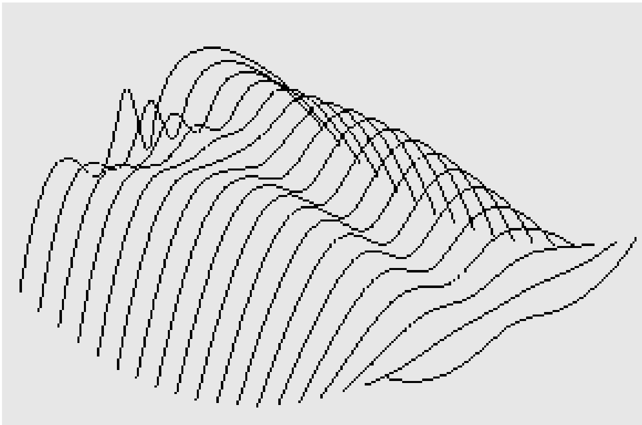
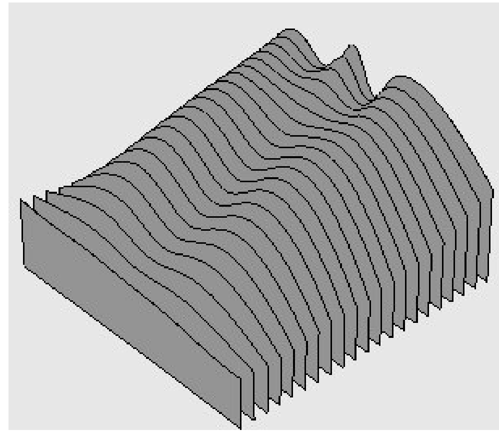


Contour curves are a special case of section curves. They comprise a series of closely space, parallel section curves that in sum give a pretty good visual description of a surface. The section curves are most often set orthogonal to a construction plane – normally one of the coordinate planes and most often the X,Y plane – and run laterally across the surface.





Because they are planar curves, contours are used to determine shapes that can be cut from rigid material and assembled to provide a physical replication of the surface. CAM (Computer-Aided Manufacture) programs can calculate contour curves spaced approximately .005 inches apart in order to drive 3D printers or cutting machines. The series of contours could be viewed as the stages of a single line moving through space as it continuously changes its curvature. In this light it is possible to view all surfaces as generated.

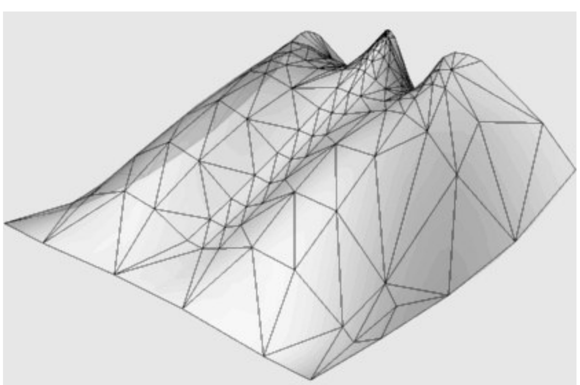
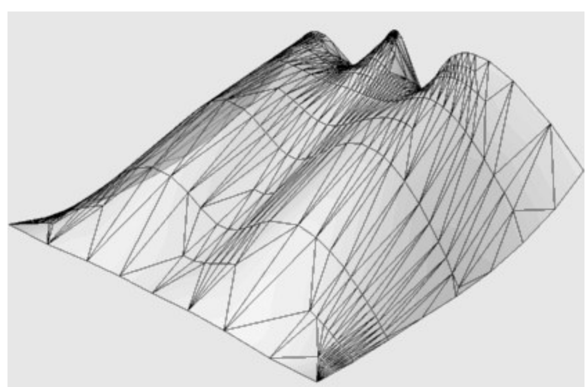
 

## III. Polygonal Surfaces

Though they do not provide the huge range and flexibility of curved surfaces available in NURBS modeling, vector-based programs nevertheless provide many curved surfaces to designers. Their descriptions of the surfaces, however, were limited to straight lines. Just as curves are represented by sequences of their vector chords, so too are surfaces. In order to define a surface in a vector space these chords must assemble into plane figures, or polygons.

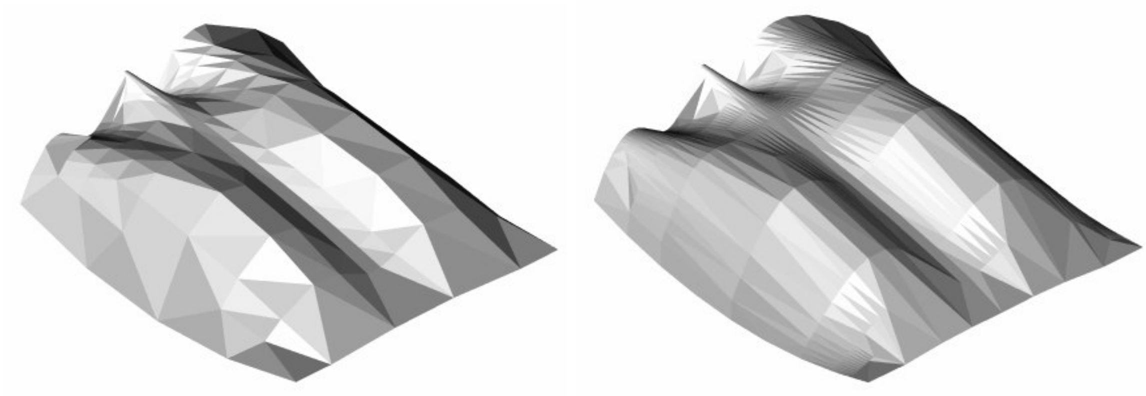
### *Meshes*

This is most easily accomplished by triangulating the chords. Any three chords that meet one another will define a triangle, a plane figure. This is because any three points are always co-planar. Four or more chords, though, will not automatically guarantee a plane figure. In a 3D quadrilateral of surface chords any three of the vertices will be co-planar, but the fourth will not necessarily fall on that plane. This is why table and chairs may wobble, while tripods are stable.

Consequently, it is most common to see free-form curved surfaces represented by a network of triangles, although quadrilaterals frequently appear when they are the more efficient or visually elegant way to define a surface portion. This network of can be computed such that the common vertices of neighboring polygons are assigned edit points. This is called a *mesh*.

Just as the frequency of vector chords affects the visual smoothness of a curve represented by vectors, so the frequency of polygons in a mesh will affect the smoothness of a surface. In the examples pictured here the frequency increases by a factor of about 3.



Not all polygonal surfaces are computed as meshes. In the case of common solids they are often constructed as polyhedrons, that are subject to transformations and Boolean edits, but not point editing.

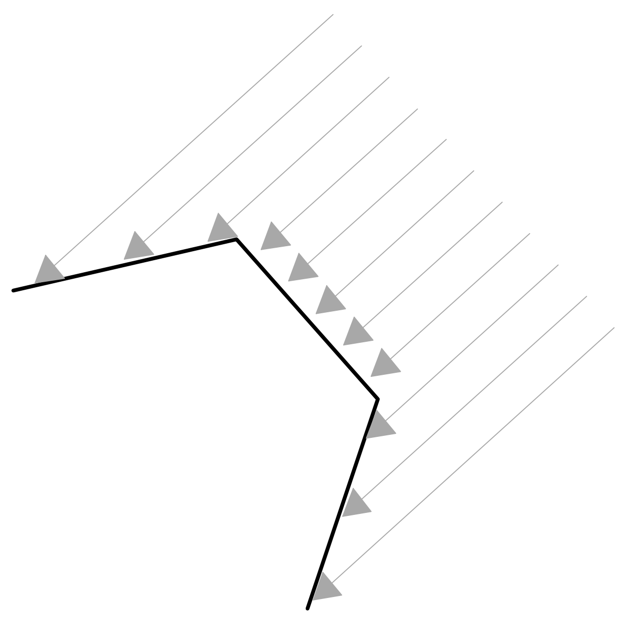
*Render Meshes*

All current rendering programs use mesh representations of surfaces. This is true for NURBS as well as vector programs. Before rendering a surface the NURBS program translates the surface into a mesh.

To understand why this is it is worth looking at the geometry of surface rendering.

Rendering works by computing the position, angle and intensity of a light source and then factoring in the angle of the reflecting surface relative to the light. From this information plus the reflectivity and opacity of the surface the computer can figure how much light is reflecting and in which direction. Those parts of a surface most directly facing the light will reflect the most light and will appear brighter, while those parts tilted away from the light will reflect less light and will appear dimmer.

The computer also calculates the relative position and direction of the camera, i.e., the viewer's eye, to determine which reflections will be viewable.



**As surfaces tilt away from a light source, they reflect fewer and fewer rays**.

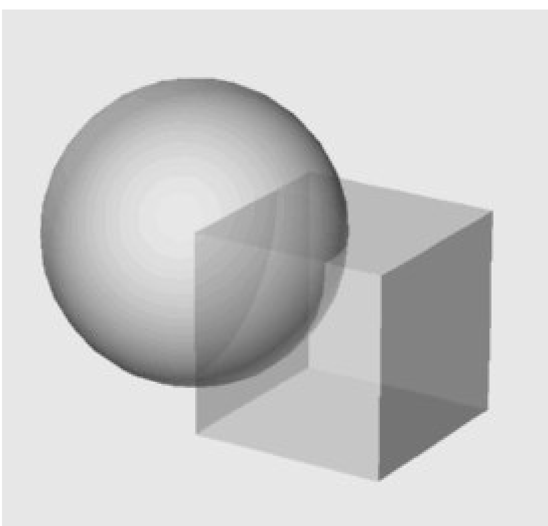
This computing process is finite in the case of meshes, but infinite in the case of curved surfaces. This is because in the former case there are a relatively limited number of surface segments and therefore a manageable number of angles to calculate the reflected light. In the latter case, though, the number of potential angles is infinite and therefore impossible to compute. The problem is that rendering flat planes on a supposedly smoothly curved surface will produce a choppy, broken effect by revealing the edges of the polygon faces of the mesh.

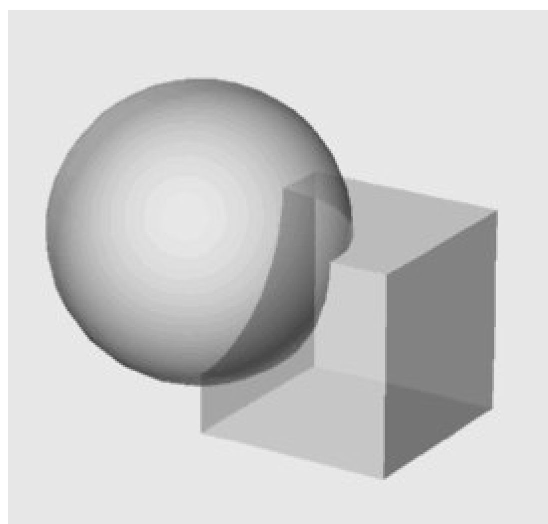
Rendering programs solve this problem by running a smoothing algorithm to blend out the edges of the polygons and provide even transitions across the surface.

***Solids***

In surface modeling a solid is simply a closed surface, resulting when a single surface closes in around a volume of space or when groups of surfaces join to fully enclose a volume. Curved geometric solids like a sphere, an ellipsoid or a torus exemplify the former. Most modeled solids of complex natural objects are assembled from smaller surfaces joined into one.

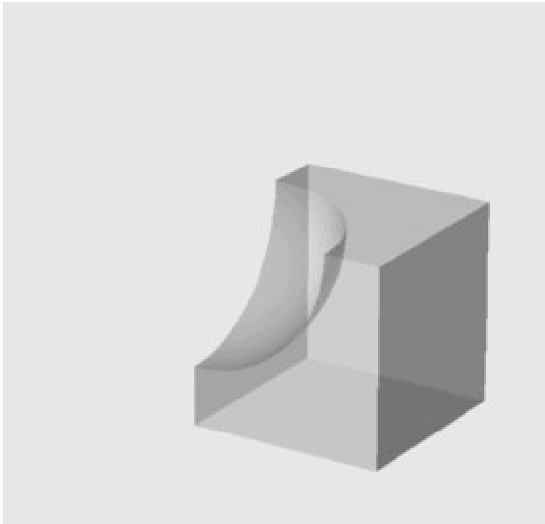
Once a surface is closed and joined the surface modeler recognizes it as a solid. It can then interact with other solids using the three Boolean operators of addition (or union) subtraction (or difference), and intersection (or common). All of these operations apply to two or more interpenetrating solids. All of these operators work by trimming and discarding portions of the closed surfaces, and saving and joining the remaining portions.

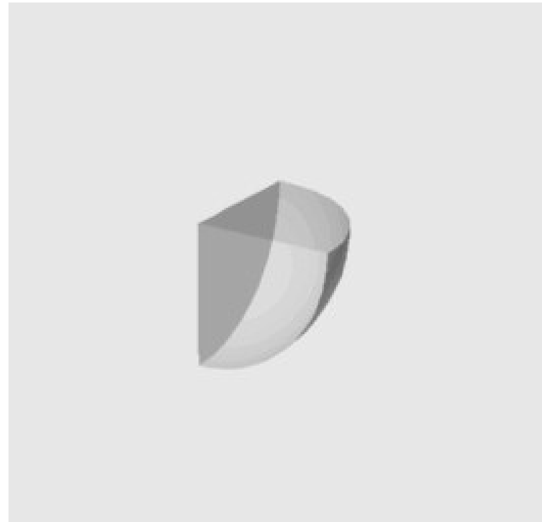




The addition operator unites the two interpenetrating solids by trimming and discarding those portions of the surface enclosed by the interpenetration. The remaining surface portions join to form the hull of a new, more complex solid.

The subtraction operator discards one of the interpenetrating solids along with the enclosed surface portion of the other solid. In the meantime the discarded solid leaves behind that portion of it enclosed by the saved solid. The remaining surfaces join to form a solid with a volume (equivalent to the discarded solid) cut away.





The intersection operator discards the exterior of both solids, but saves the enclosed surface portions and joins these into a solid. What remains is a solid equivalent to the volume of space held in common by both solids.