Rhino Basics

Intro

This is an introduction to surface modeling with focus on smoothness because this is were NURBS modeling has one of its strengths.
Although we use examples from Rhino modeling, it will in general apply to most CAD programs. To fully make use of NURBS based CAD programs helps to have an understanding of how this type of modeling works.

As with any tool your thinking will be influenced by it. In computer modeling you will be influenced by the work flow and the technology of the program.
NURBS modeling is an intuitive way of working with smoothly curved shapes, and lets you do it with high precision. You can create almost all imaginable shapes, and also analyze their curvatures, section lines, volume distribution and so on.

The command names are highlighted with **bold** in the text.
This manual will be updated continuously, and the latest version will be available from our website. Try the tutorials there too. If you need to get more hands on feeling for the concepts described here, these tutorials should be to the point and short.

Fredrik Wenstøp | 20.03.08

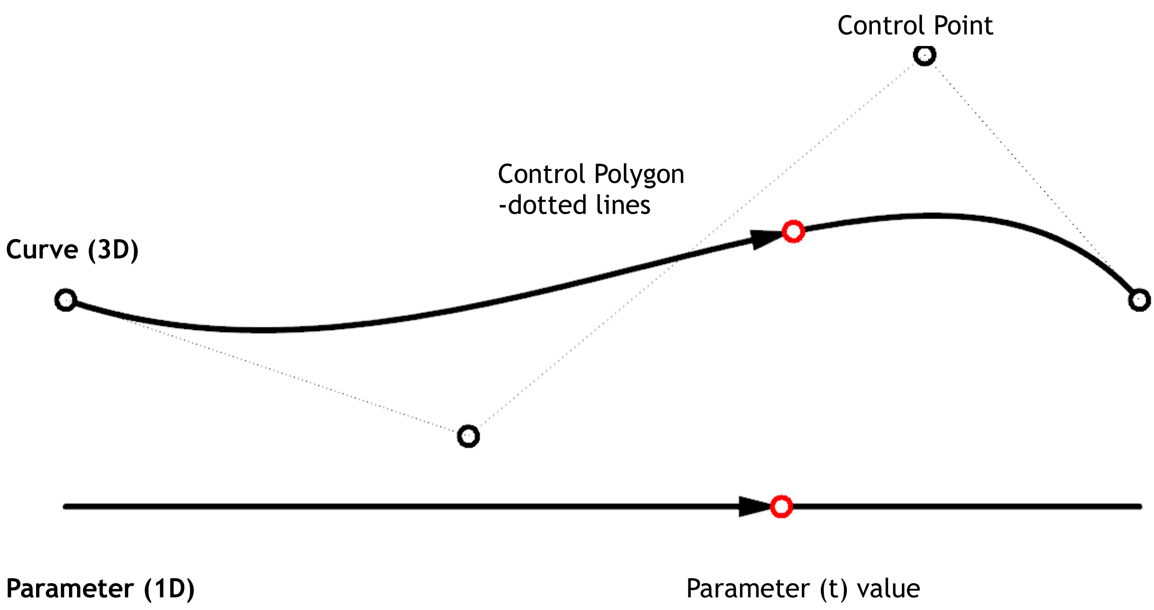
What is NURBS?

So what does this mystical word actually mean?

Below is a breakdown of the word, and it is explained in some more detail later on.

NURBS is short for “Non-Uniform Rational Basis- Spline”.
Non-**U**niform means that the portions of a curve that are affected by individual controls points, are not necessarily uniformly distributed along the curve.
Rational means that the pull of each individual control points can be adjusted.
Basis- **S**pline functions define how much each control point influences the curve at any given parameter value.

A NURBS curve is the result of a shape-blending between control points.
The curve on the below, is calculated by a blending function that uses all 4 control points in an equation to draw out the curve.
The blending between the control points changes continuously as you move along the curve.
At the start of the curve, the first point is pulling 100%, and the other ones are pulling 0%.
This is why the curve touches the first point.
As you move towards the middle of the curve, the influence is more an average of all four points, then shifting to only the 4th point at the very end.
The shape-blending in a curve happens along a one-dimensional space where a parameter value(t) corresponds to a distance along the three-dimensional curve:



Curve and surface characteristics are <u>mainly</u> controlled by these two factors:

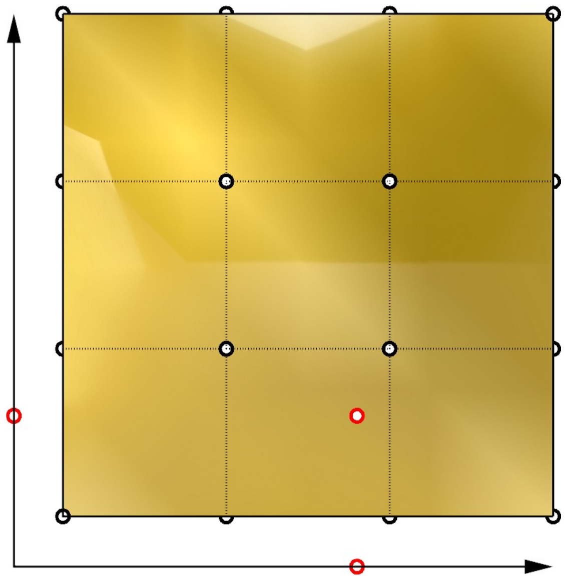
1. number of control points
2. degree

So all NURBS curves and surfaces have control points that act like magnets and pull the curve towards themselves: In Rhino you can draw curves by placing control points directly, and you can change the shape of a curve by moving its control points.

The degree of a curve determines the influence each control point has on the curve. A higher degree curve gives less local influence and more global influence.

The number of Control Points in a curve must be one number higher than the Degree or more. For example, if you want to draw a degree 5 curve with as few points as possible, you will need at least 6 control points.
The degree number is usually 1, 2, 3 or 5, but can be any positive whole number.
Rhinos lines are defined by 2 endpoints and are degree 1.
Circles arcs and conic sections are degree 2, and use weighted control points.

NURBS **surfaces** are a controlled by a set of control points and two parameters (u and v). The u and v direction is the internal two-dimensional coordinate system of the surface. The red point seen on the surfaces below is a uv-coordinate that was extracted with the command **PointsFromUV**.

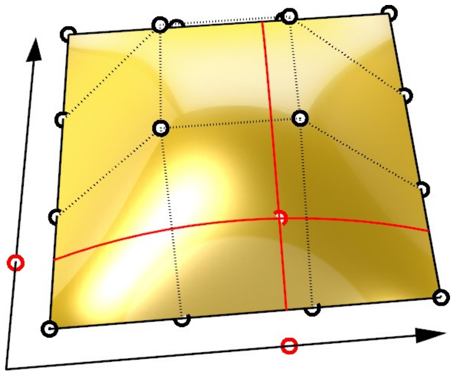


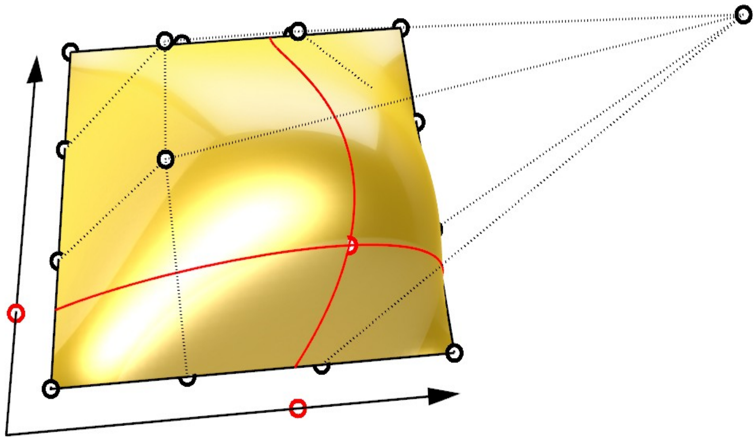
You can think of the NURBS surface as a rectangular rubber sheet, except that the NURBS surface can be deformed almost infinitely:

One edge can be collapsed into a single point by placing all the control points of the edge at the same place. Note that doing this can create tiny wrinkles, and problems when exporting to some programs.

The red lines in the images below are called isocurves (isoparametric curves). These are curves of constant u or v-value on the surface. The isocurves here were extracted with the command **ExtractIsocurve**. Isocurves can be extracted at any point on a surface.

When moving the control points of a NURBS surface, the uv lines will change position in 3d space; But their uv coordinate values do not change.





A surface can have different degrees in the u and v directions. The isocurves have the same number of control points and degree as the surface in that direction.

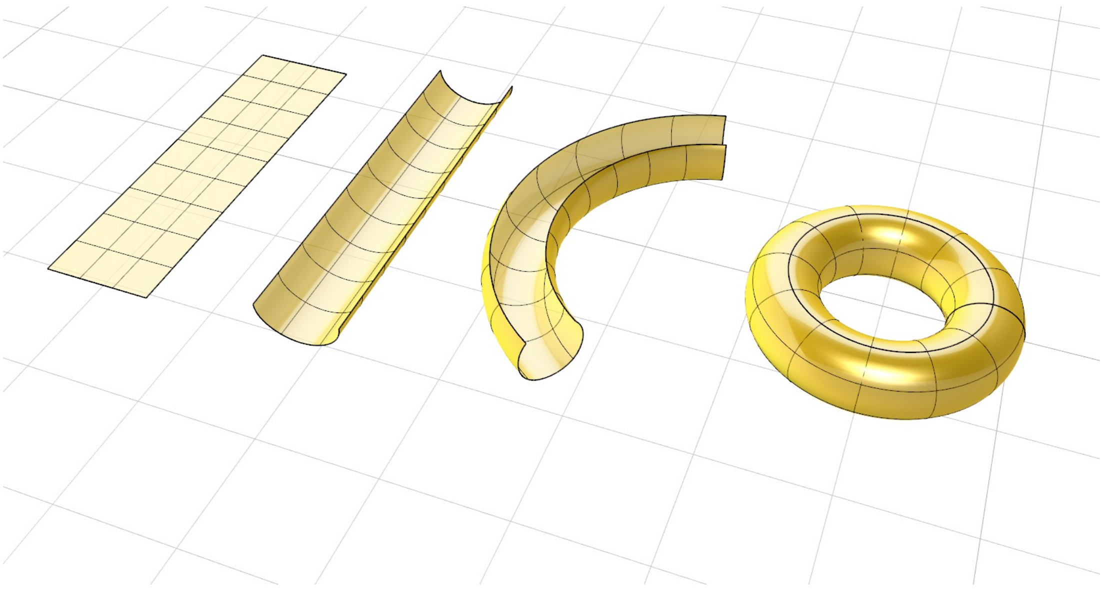
in Rhino, isocurves are drawn at knot locations by default. If the surface is a single knot-span surface, isocurves are drawn in the middle of the surface.

Solids

Solids are surfaces or polysurfaces that form a closed volume. They should in other words not have open edges, and you can check this with the **ShowEdges** command (Mode=NakedEdges).

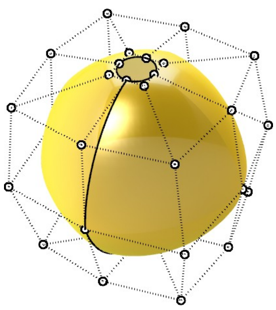
Solids define thickness distribution, which makes it easier to calculate volumes, do strength analysis, or 3D print CAD models (these are just a few examples). Solids resemble physical objects better than open surface collections because you can imagine filling them with a material. If you print out this manual, the pages will have a thickness filled with paper and ink, even if it is very thin. If you don't print it, its like an open NURBS surface that has no thickness (it's just light coming out of your screen).

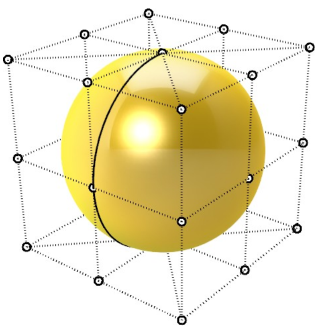
You can form solid models with many joined surfaces that constitute a closed polysurface, or with just one single surface. The Torus and Sphere are examples of single surfaces that form closed volumes.



The sphere has two poles just like the the globe. The surface still has four edges, but as mentioned they can be collapsed into one point called a singularity.

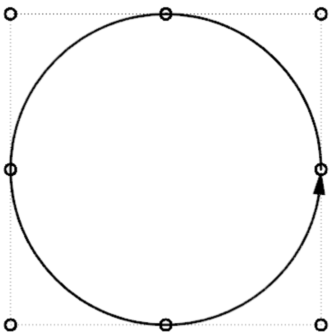
Edges are by default drawn with thick lines in Rhino, therefore you see a thick line where the edges match up. These are the seams of the surfaces. In toruses and spheres made by the **Torus** and **Sphere** commands, edge points at the seam are glued together. When you move them, they will stick together. There are also other commands which create such objects.

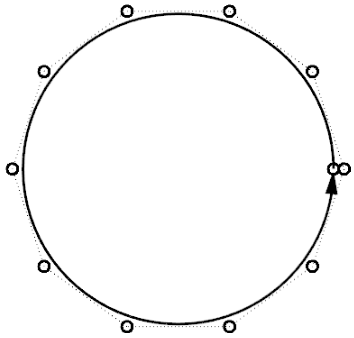




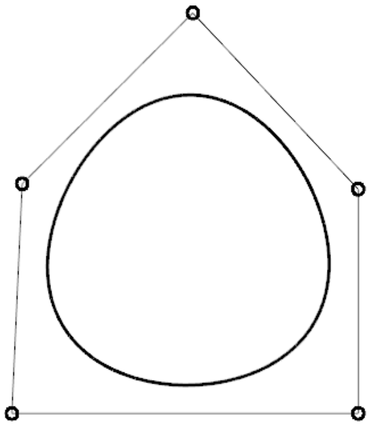
Periodic curves and surfaces

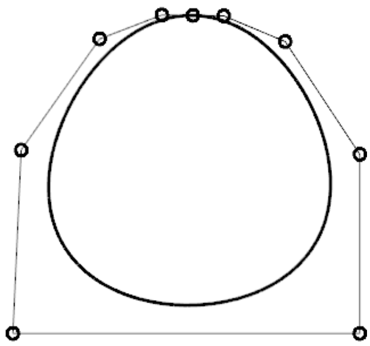
A periodic curve is a smooth closed curve, that is defined in a special way so that the seam stays smooth when you edit them. The circle seam is just the point where the ends meet. You can move curve seams with the **CrvSeam** command.
In the **Circle** and **Sphere** command there is an option called **Deformable**. This gives a NURBS approximation of the circle or ellipse, where you can specify the degree and number of points. This can make then easier to work with in some cases.



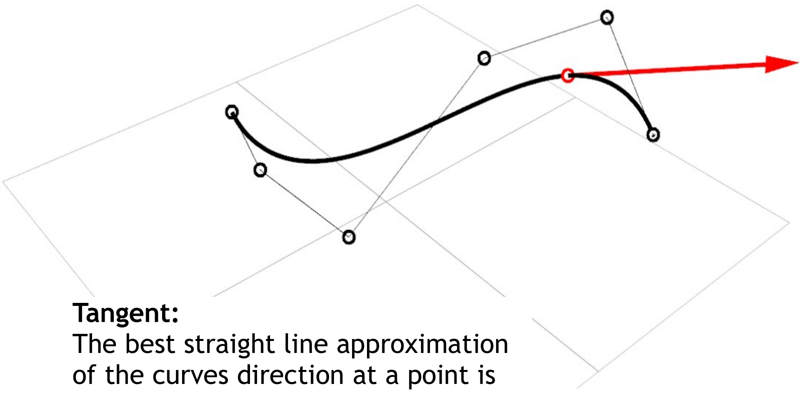


You can make a closed curve periodic with the **MakePeriodic** command, and you can make a periodic curve non periodic with the command...**MakeNonPeriodic**.



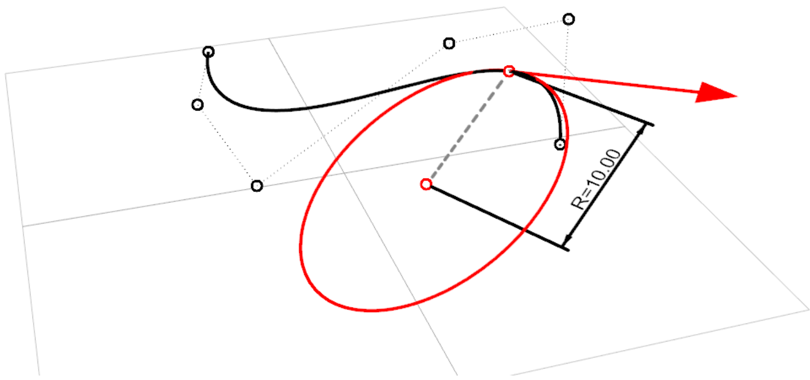


A way to think of curvature



Tangent:
The best straight line approximation of the curves direction at a point is called a tangent.

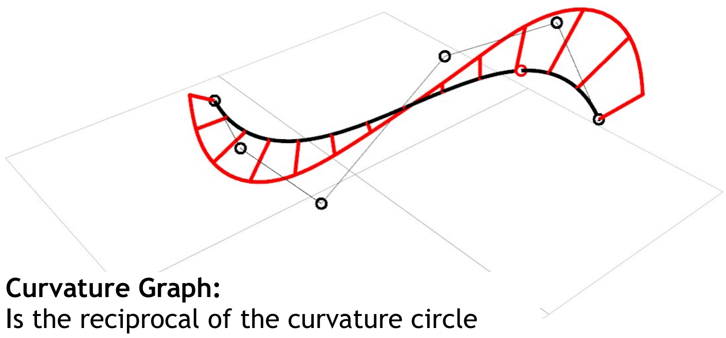
The tangent is the instantaneous direction of the curve at a point, whereas the curvature measures the radius by which the tangent vector rotates at the point.



Curvature Circle:
The circle which most closely approximates the curve at a point.

Try this in Rhino:
Draw some arbitrary s-shaped curves in 3D and use the CurvatureCircle command to see how the curvature changes. Notice how it also twists along the curve if your curve curves in three dimensions.

You may also notice how the circle plane twists through the inflection point where the curve goes from concave to convex.

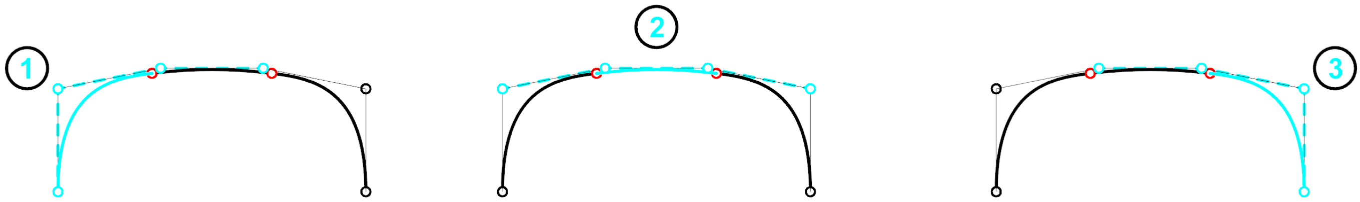


Curvature Graph:
Is the reciprocal of the curvature circle along the curve, making it high in areas where the curvature circle is small.

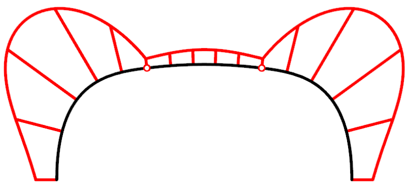
Knots

Curves that have more control points than 1+the degree are internally piecewise, that is they consist of more than one polynomial curve span. The points of transition between these internal curve spans are called knots.

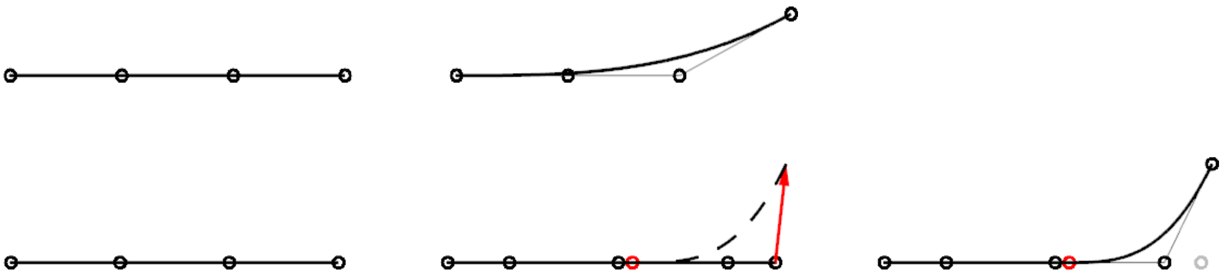
The degree of a curve determines how many points are used to calculate one span of the curve. For example, a degree 3 curve with six control points will have three spans, that are each drawn by of four overlaping sets of points:



Going over the knots from one span to another, curves of different degrees have different levels of smoothness. This depends on how many overlapping control points are used to calculate the individual spans. The degree 3 curve above has three overlapping points for each span, which highlighted with cyan color. Turning on the curvature graph will reveal the curvature continuity at the transitions:

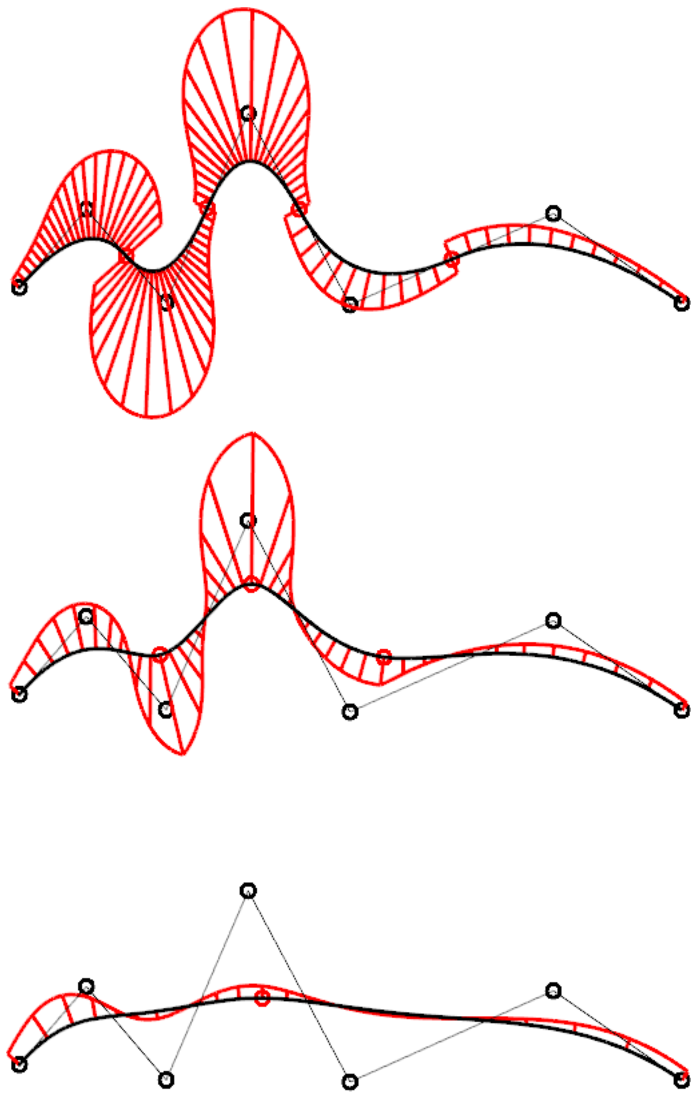


By inserting knots you can constrain a part of a curve or surface. The example below is a degree 3 curve with four control points that is constrained by inserting a knot.
Inserting knots rearranges the control point structure and introduces new control points.
To insert or delete knots, use the **InsertKnot** and **RemoveKnot** commands.



Internal Smoothness of Curves

More on interpreting the curvature graphs below will come in the next section.



As mentioned in the previous section, curves of different degrees have different internal smoothness at the transitions between curve spans. On the left are three curves of varying degrees drawn with the same control points. Knots are marked with red points.

Degree 2 curves have internally continuous tangency. The curvature graph indicates this: The steps go in straight lines which means the tangent direction stays the same across the transition.

Degree 3 curves have continuous curvature. The graph does not have steps, but may show sharp changes in curvature in the transitions.

Degree 5 curves have a totally smooth graph. The rate of change in curvature is smooth.

You can change the curve degree with the **ChangeDegree** command. It keeps the knot structure of the curves the same, but it adds or subtracts control points between each knot span.

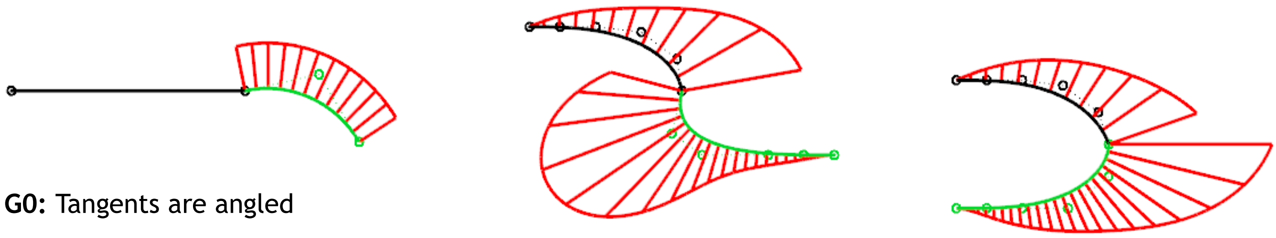
Smoothness between curves

Because models usually consist of many connected surfaces and curves, knowing how to control the smoothness in the transitions is important. Three types of transitions commonly are used in design.

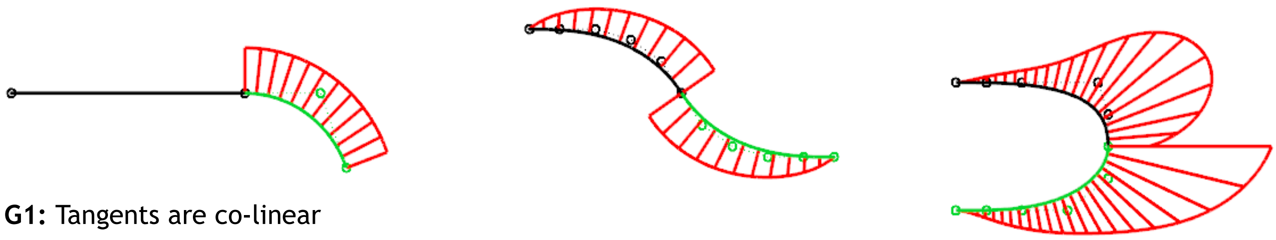
1. Position, G0 – the curve ends touch within tolerance.

2. Tangency, G1 – the tangent direction stays the same across the transition, but the curvature circle radius may change abruptly. The tangency direction is determined by the first and second point on each curve. If these all fall on a line then two curves are tangent.
3. Curvature, G2 – The radius of curvature is the same at the common end point. In G2 surface transitions, light reflections will not break at the transition. The surfaces should appear as one smooth surface, as long as the distance in which the transition takes place is sufficient.

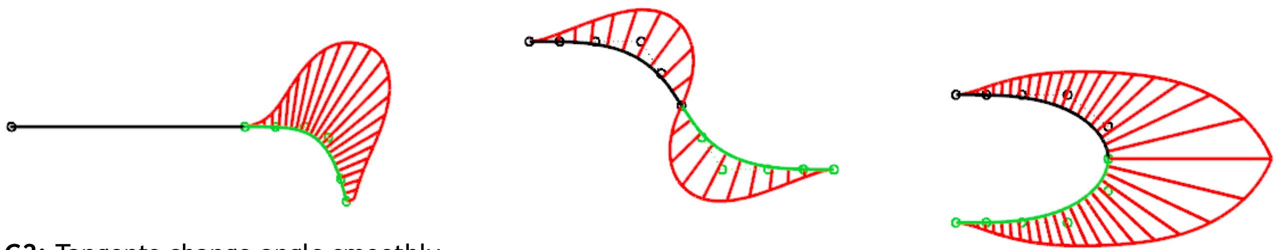
The curvature graphs of three transition types in three different situations:



G0: Tangents are angled



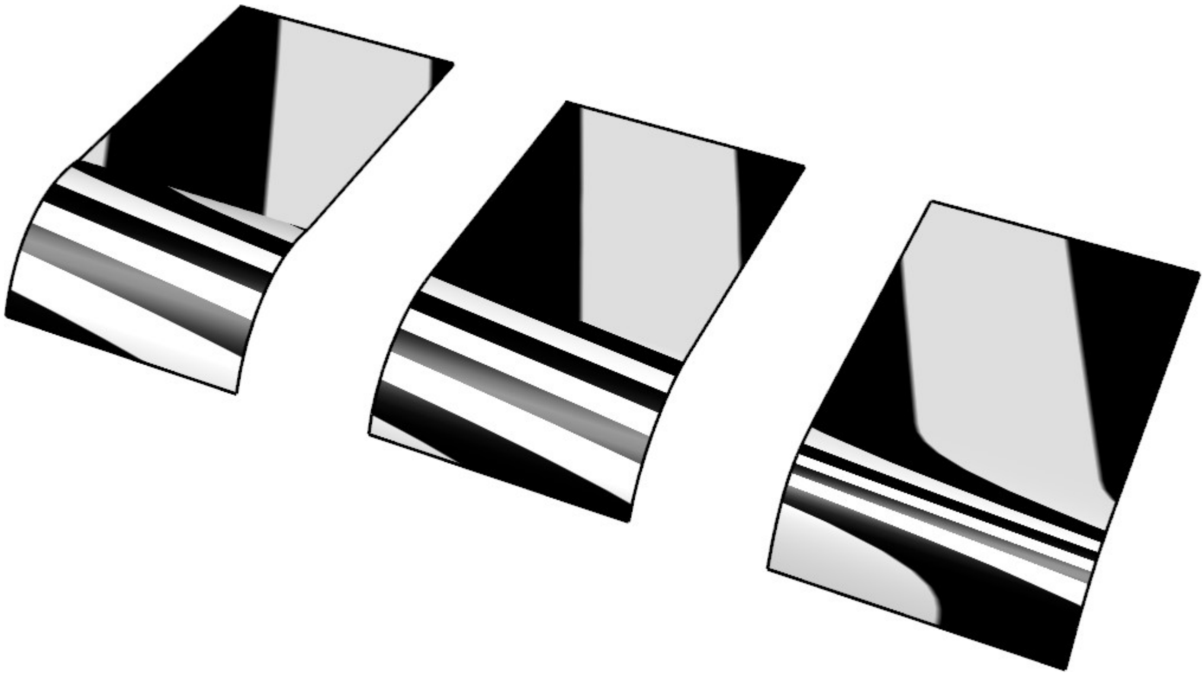
G1: Tangents are co-linear



G2: Tangents change angle smoothly

The **Zebra** command lets you visually analyze surface smoothness and curvatures.
The stripe-behavior can be interpreted as follows:

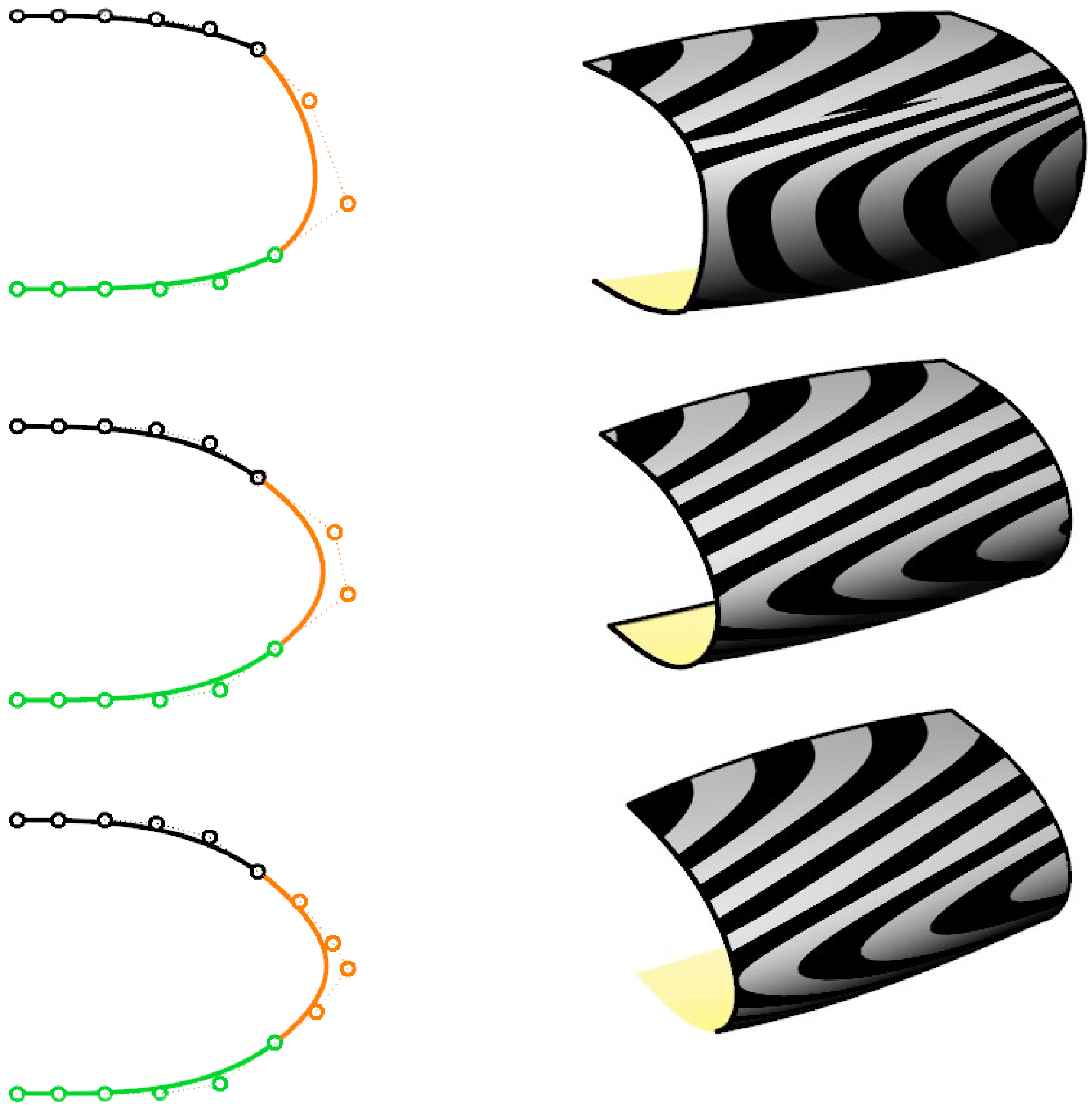
Position	G0	-The stripes don't line up but jump sideways as they cross the transition
Tangent	G1	-The stripes line up as they cross but turn sharply at the transition
Curvature	G2	-The stripes line up and continue smoothly over the transition



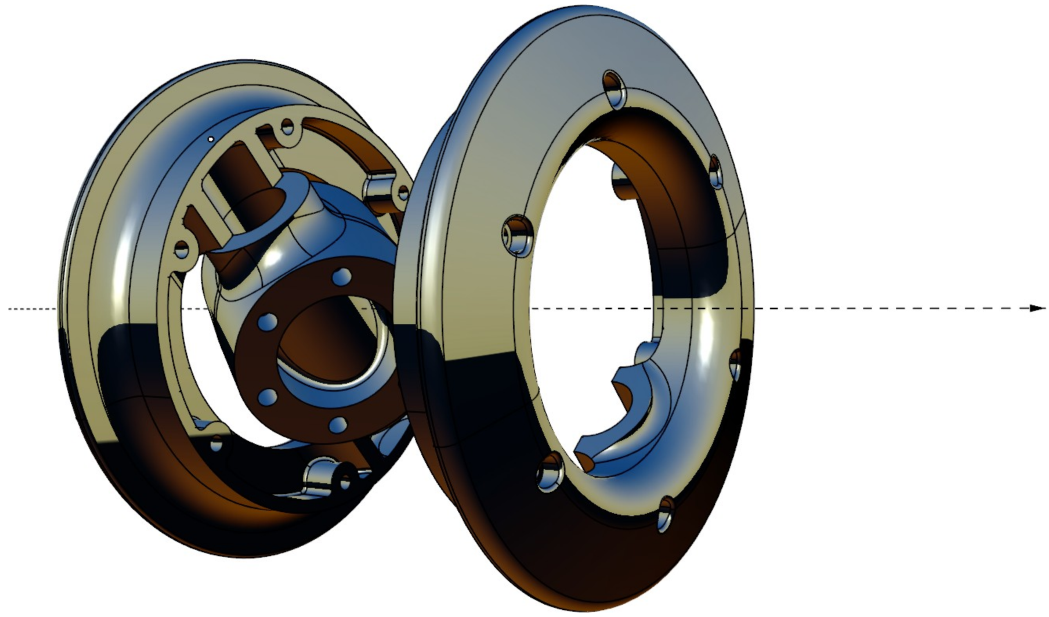
Blends

Below are the three transition types in a curve blend and a surface blend.

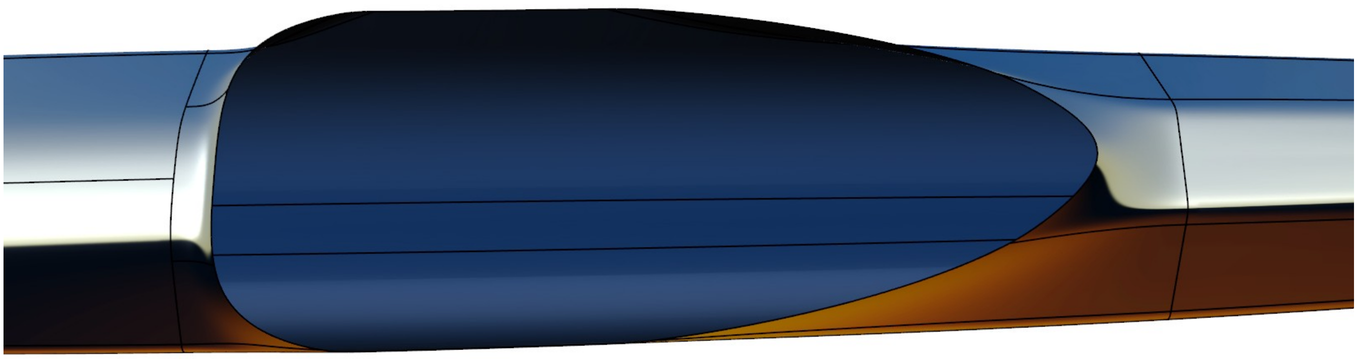
Notice: The number of points in the blend increases from four to six to allow for curvature control in the G2 blend. It can sometimes be difficult distinguish between the G1 and G2 blend using Zebra. In this particular case there is just a small shift in stripe curvature across the connection:



G1 transitions are typically used in machine parts that have circular fillets and straight surface segments.



G2 transitions are used in places where smoothness is of importance:

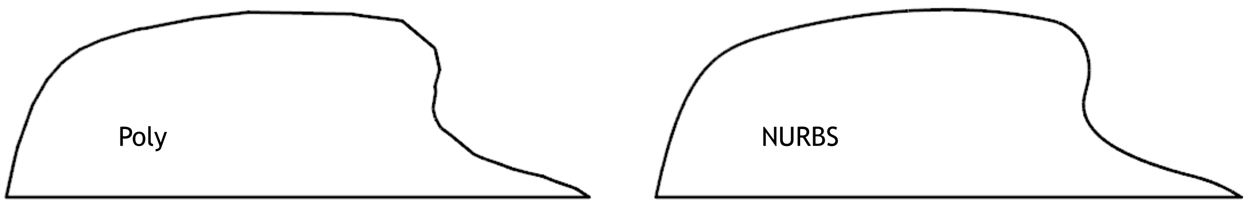


Polygon models compared to NURBS models

I have said a little about what nurbs are and how they function. It's about time to describe the difference between nurbs and polygon models. Polygon modeling Is used a lot in gaming and the movie industry, because there is usually no need for extreme precision in those applications.

This is a polygon (triangle and quad) version of the mouse:

It's really just xyz coordinates with two-dimensional planes between them, usually triangles and four-sided. The light reflection is calculated from these planes.
To the right is a shading of the mesh mouse with smoothing. It looks quite all right, but the looks are deceiving. Making sections through a mesh surface often gives quite horrible results, while NURBS sections are precisely defined down to 0.01 or 0.001 mm or more if you need it.



Smoothness and precision is the reason why NURBS is the standard in computer aided manufacturing; A NURBS model <u>is</u> the actual geomerty of the physical object you are making, whereas a polygon model in most cases serves only as a representation of it.

The advantage of polygons is that they are can be treated in a uniform way; It's almost like pushing on sand when you make a sand castle.They are not described by mathematical formulas and don't follow rectangular point grids. Therefore they and can be smoothed, pushed and stretched in the same way regardless of the objects shape.

Modeling tips

1. keep it Precise

Use object snaps and grid snap: Two lines that look like they intersect on screen need not do so in reality. Common file tolerances are 0.01 -0.001 units. If a gap is larger than the file tolerance, curves and surfaces may not want to join. Such a small gap can be impossible to see on screen, but it's still there. One big advantage of computer modeling is precision, right?

Continuously check your work by joining surfaces and curves as you model. If they join, you are within the tolerance limits. If they don't join, you should fix it at once. This way you avoid ending up with a useless model with inaccuracies and gaps everywhere. If you want to use the model later to make a prototype or 3D print you will be glad you followed this advice.

2. to use as little data as you can to describe the shape

This makes things more smooth and gives better control of the curvatures. It also makes the model easier to edit.

Even though surfaces can be created directly from a set of points, they are usually created by surface commands that use shape defining curves as input. For a Surface command to result in a nice editable surface, the input curves should be as simple as possible:
- Introduce new points only when they describe new features in the curves.
- Introduce only new points or curves when they describe new features in surfaces.
- Avoid many small patches where one large surface can naturally describes the shape.

3. use oversized surfaces with four corners as much as you can

Usually it is better to make surfaces that have four corners rather than to collapse edges or to squeeze the edges into continuous lines.

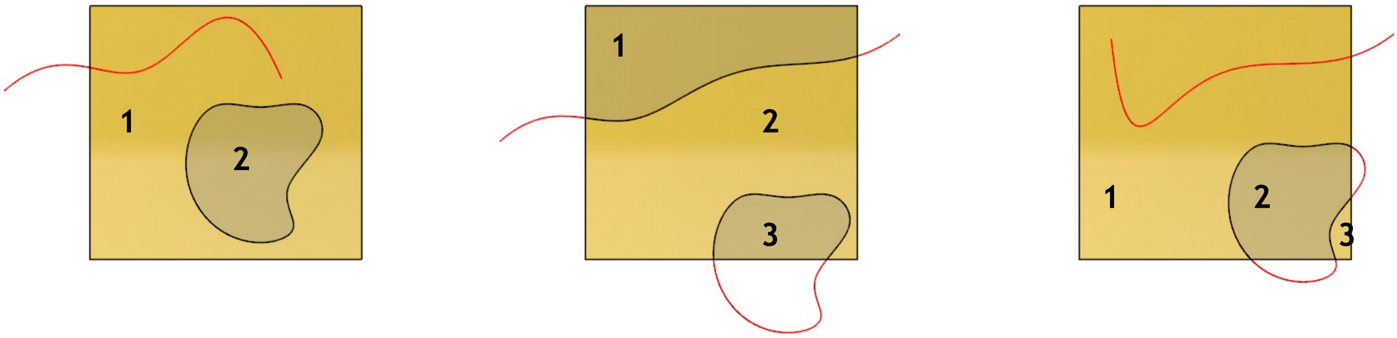
Often, it can be a good idea to make surfaces that extend beyond what you need for the end result and then use trimming to create the final shape.

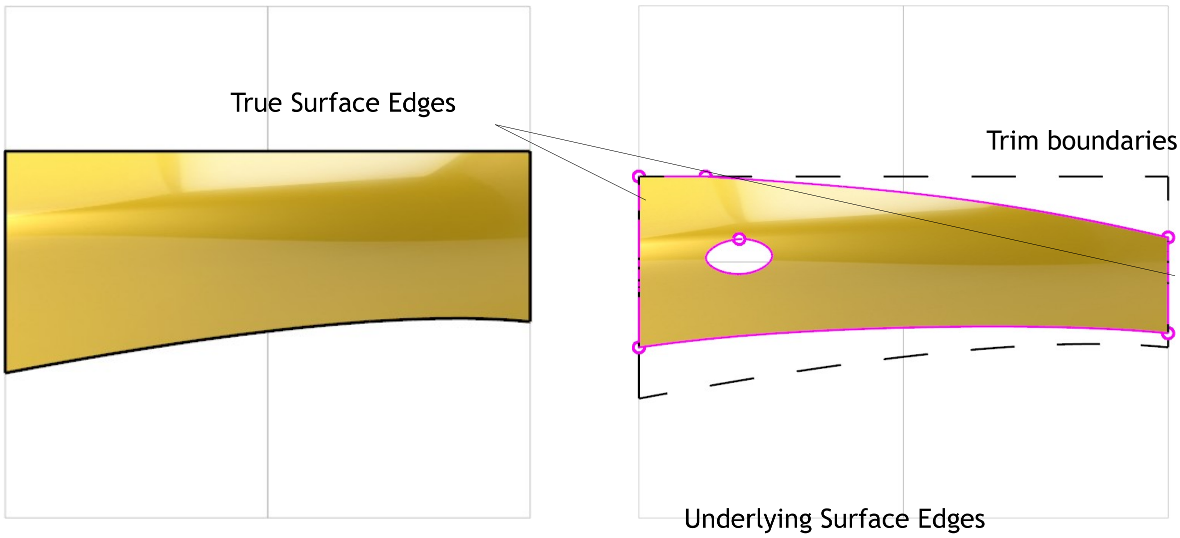
However, if you need precise dimensions at the edge itself, it's best to use the true surface edge as the final boundary in those particular places. You will also need to do this if you have to match the surface edge to another surface.

Trimmed surfaces

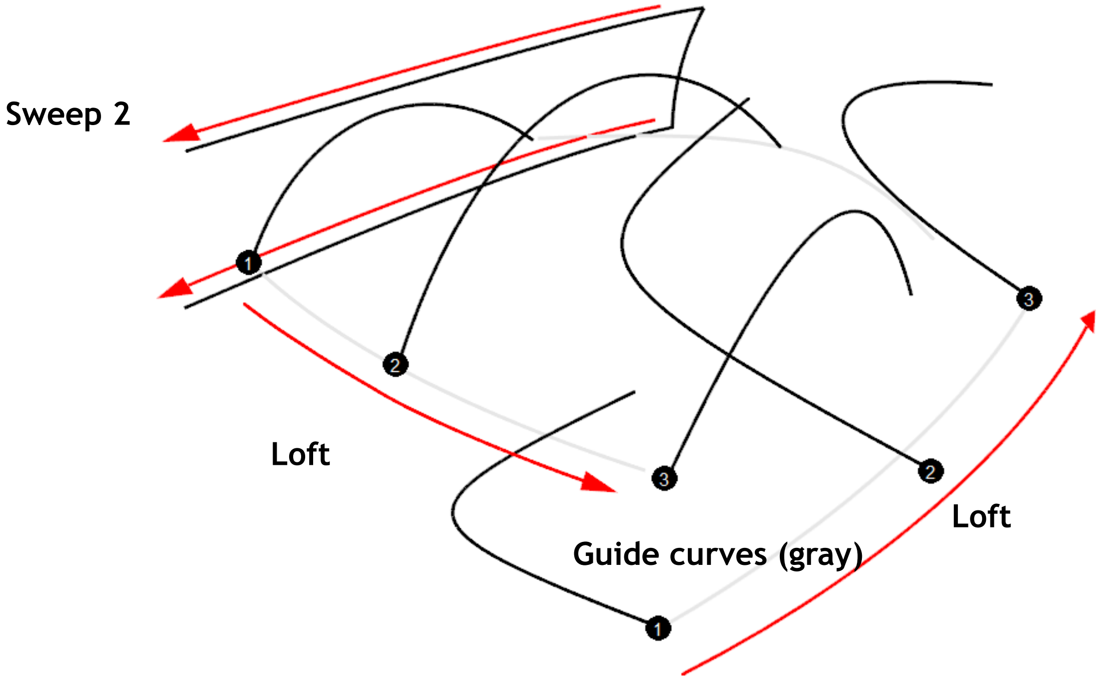
It's quite similar to trimming paper with scissors:

To divide it you either have to cut completely through its edges, or cut inside it with a closed "circular" cut:

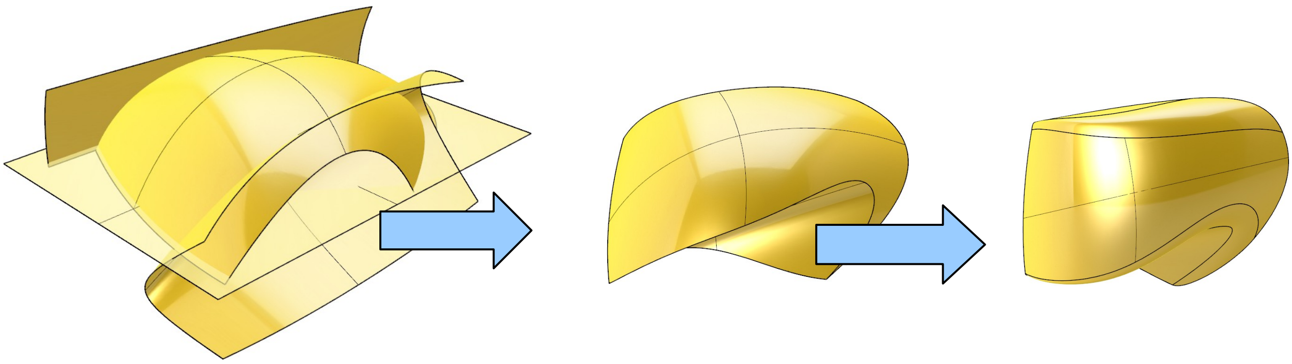




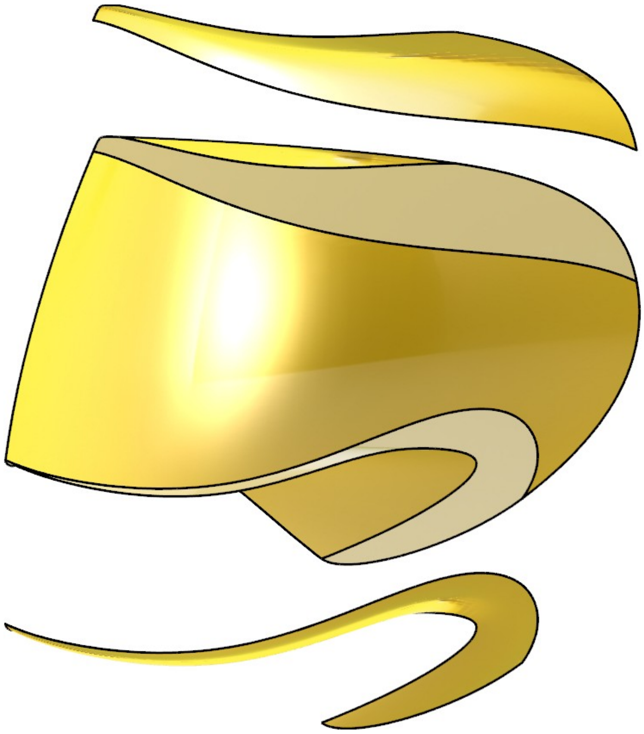
Crossing surfaces Example



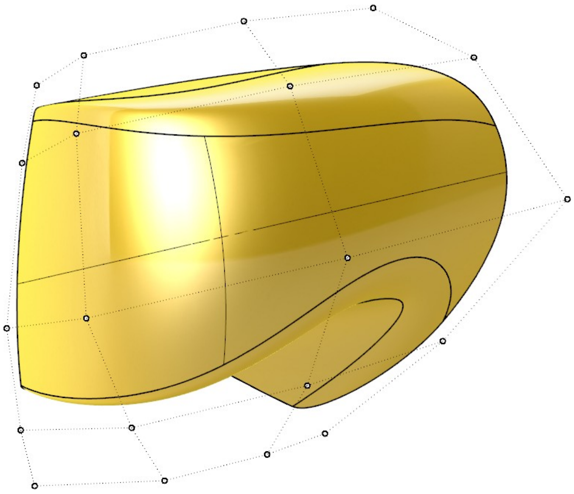
Several surfaces that intersect, and are joined together like this:

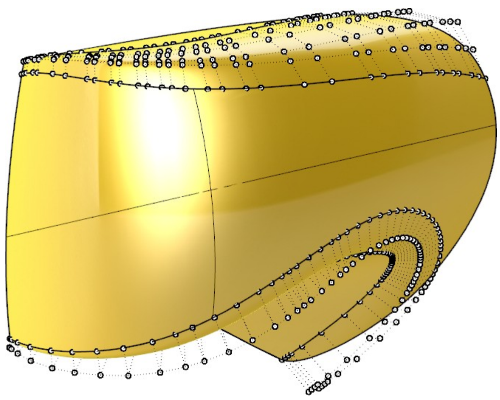


Even though we don't see it in the final model, the original surfaces boundaries and control points are still there. They are just hidden from view by something called trimming curves. Only the underlying surfaces define the actual geometry. The trim curves only mark which part of the surface is to be considered trimmed away (invisible).



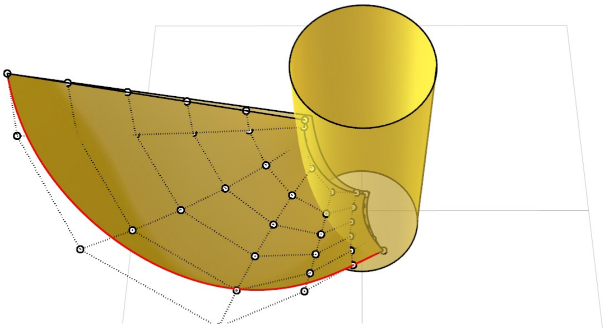
If you type the **Explode** command to separate all the surfaces, it is possible to turn on the control points on each surface:



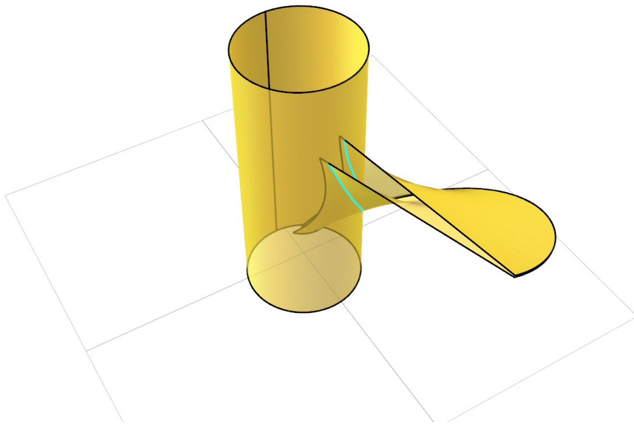


Using the actual surface edges vs. using trimmed edges

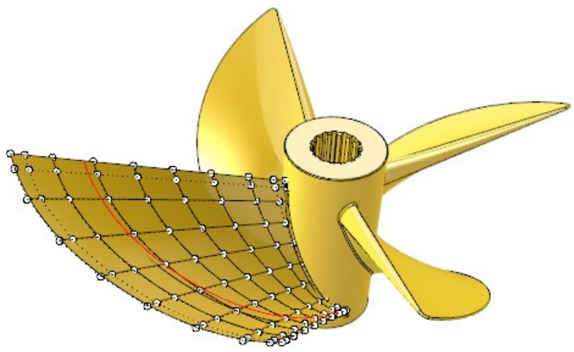
Let's say that one requirement for a propeller is that the the front ridge of the blade to have a thickness of exactly 1.00 millimeter. The ridge is already given to us as curves that are placed correctly and have the right distance (marked with red). The blades thickness at the base is not so important, as long as the surfaces are smooth and the intersection curve between the base and blade curve in the right angles. The cyan lines in the image to the right are created by the **Intersect** command, and can be made to update interactively by using **History.**



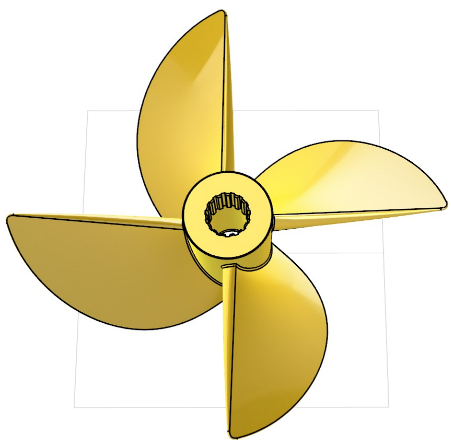
True Edge on curve gives exact fit



Always ask your self which edges need to be exact, and which can be made by intersecting surfaces. Because of the inherent rectangular structure of NURBS surfaces, we sometimes get problems when trying to fit them into triangular boundaries.
If we tried to make the blade surfaces pass exactly through the ridge lines and extending beyond them, it would have been difficult to be precise. It would also have been difficult to make the three edge surfaces by collapsing one edge into a single point, because this would create problems with filleting later on. Luckily we could make rectangular blade surfaces and hide the rest of the rectangular corners inside the base.



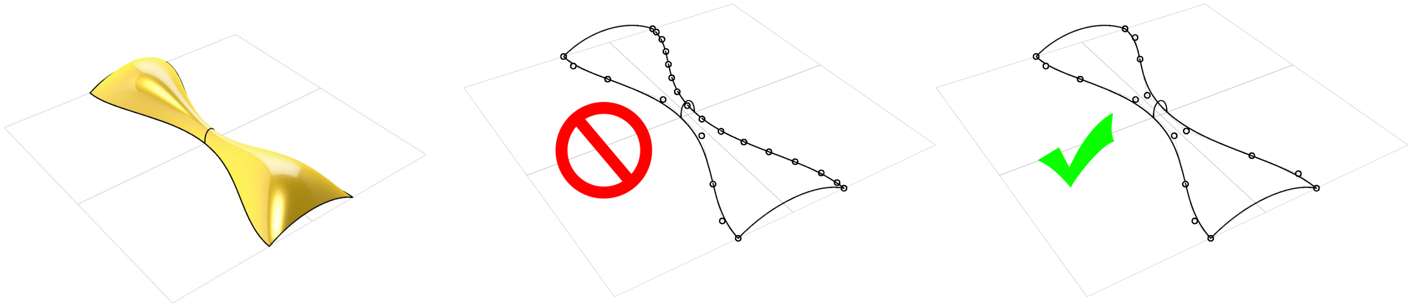
It is more difficult to fit the surface through the curve



Surface Commands

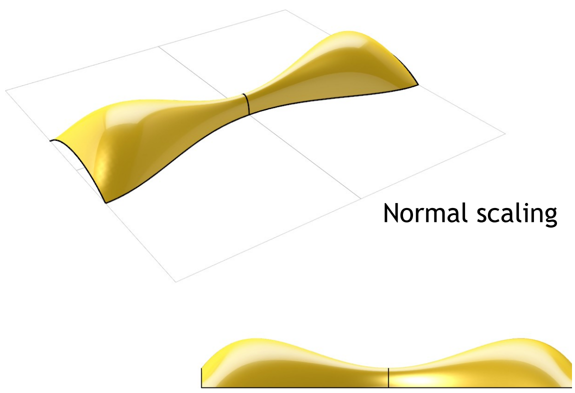
There are many surface commands in Rhino. I will not go into detail about them, so it is highly recommended that you read about them in Rhinos help file.
Loft, Sweep and **EdgeSrf** are simple but effective commands for creating surfaces.

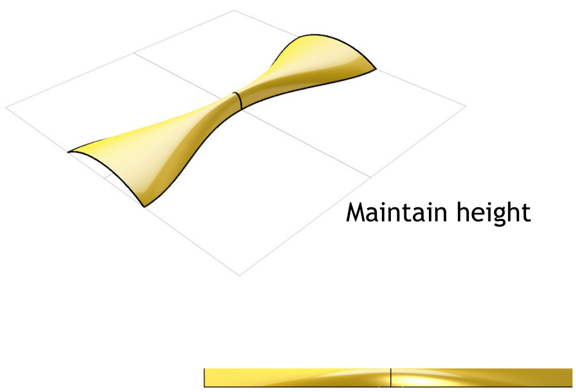
Keep the same number of control points equal for all stations or guide curves going in the same direction. This will reduce the amount of points and knots in your models:



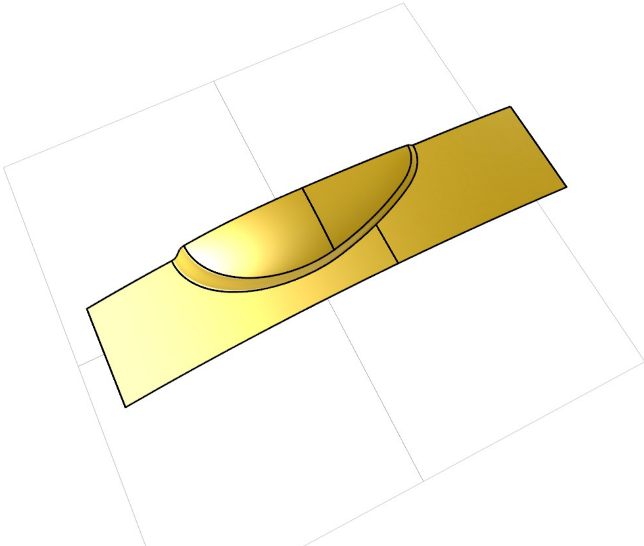
Scaling in Sweeps

By default, shape curves normally scale in both the height and width dimensions. Maintain height removes the association between the height and width scaling. It is s an option in **Sweep2** and some other surface commands.



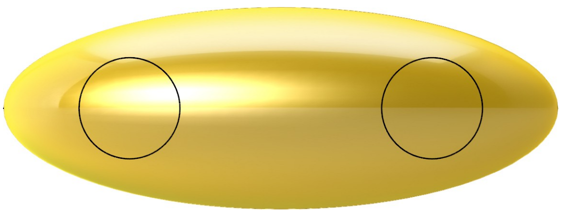


Sweep across several surface edges:

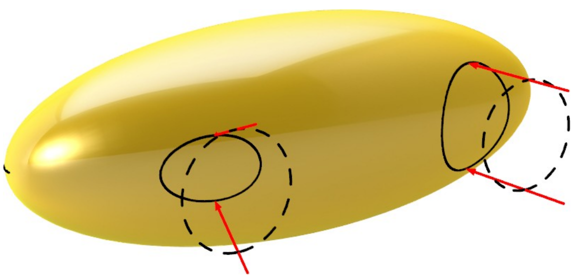


Project versus pull

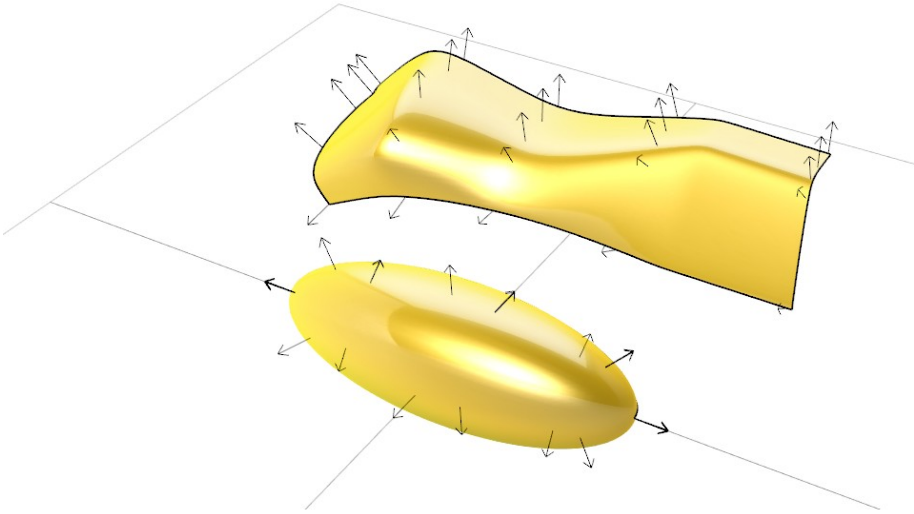
This is going to be a submarine, and you want to cut out some holes for windows using the circles. We use **Project** on one window, and **Pull** on the other:



Project projects the circle onto the surface from one of the construction planes.
But when we pull, the curve is sent onto the surface following the surface normals.



The normal is the perpendicular to the surfaces at a given point. For closed polysurfaces (solids), and single-surface solids (sphere, torus), the normal always points "out". On an open surface or polysurface, the normal can point out or in, it depends on how it was created.
In boolean operations the normal direction is used to tell what's the inside or outside of the solid. If boolean operations behave opposite of what you expect, you might need to use the **Dir** command to **Flip** the direction of some polysurfaces or surfaces involved.



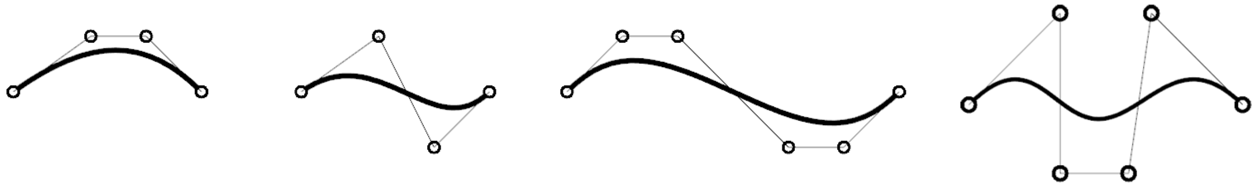
More on Curves
Drawing free-form curves you will typically use degree 3 or 5. Curves of degree higher than 5 are usually not practical or necessary to use. These curves will deviate a lot from the shape of the control polygon and are more difficult to control precisely.

To create simple curves, start off trying to find the minimum amount of points you will need:

If the curve is c-shaped (convex in only one direction), and you only need to control where and how much that curve changes, you will need a degree 3 curve with four-control points. For a s and m-shapes you can use a 5-degree curve with six control points, or degree 3 with 4 control points.

Degree 3, four points

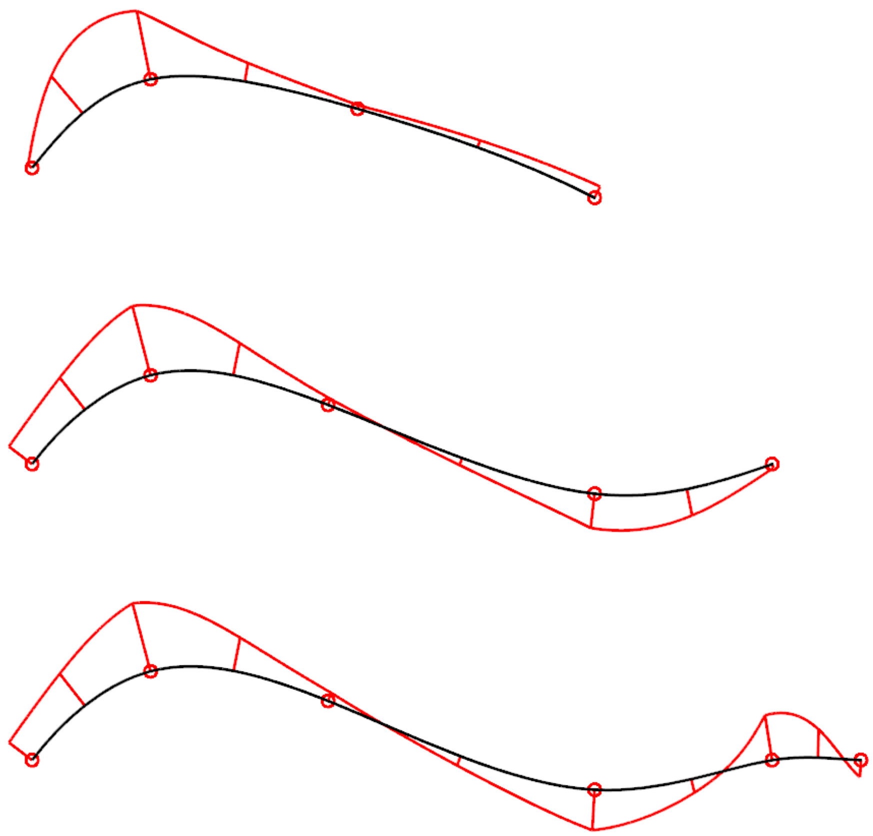
Degree 5, six points



It is also possible to put a weighting magnitude on individual control points to adjust their influence, with the command **Weight**.

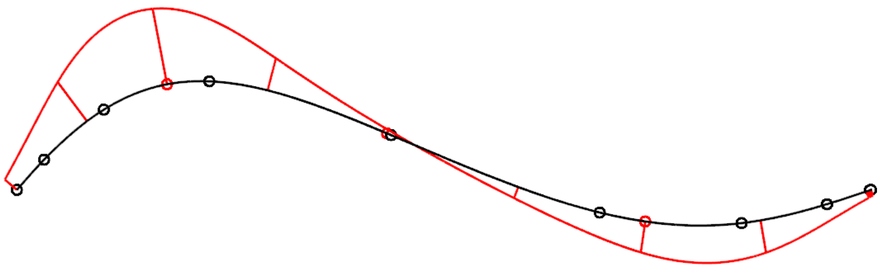
Curve through points

Sometimes we want the curve to pass exactly through a set of points. With a NURBS curve, this is quite similar to a steel band or “spline” that is forced through fixed points. NURBS was developed as a digital counterpart the real steel bands or “splines” that were used to draw smooth curves in ship design.
Like real splines, NURBS curves also have a in-built tension that can make it difficult for it to pass through points the way we want.
When you have many points to go through, this tension can make the spline squeeze out other places and create wiggles. It gets more difficult to make smooth curves the more points you have to go through. With 3 or 4 points it's usually easy, but the more points the more difficult it gets to keep the curves smoothness:

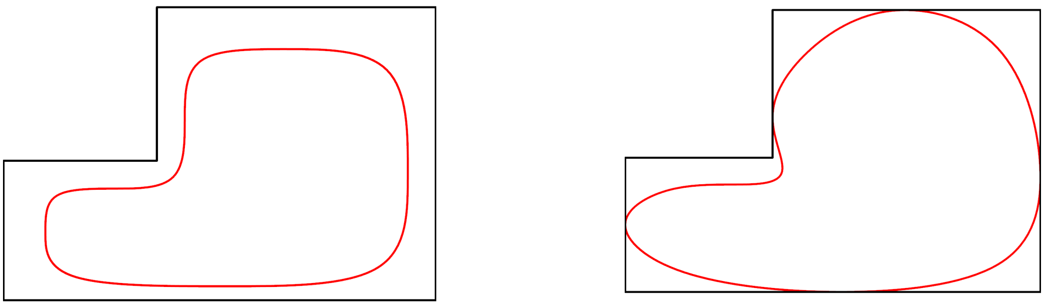


In Rhino, we can use the **CurveThroughPt** command to draw interpolate curves through a set of points. There are different settings for the knot spacing, Uniform, Chord and SqtChord. Use Uniform if your points are roughly uniformly spaced, and Chord or SqtChord if they have varying distances. You can check the result interactively in the command.

If you want to make an existing curve pass fixed points, you can use command the **Hbar** or turn on the Edit points. The edit points in Rhino are not the same as knots, although they are associated.

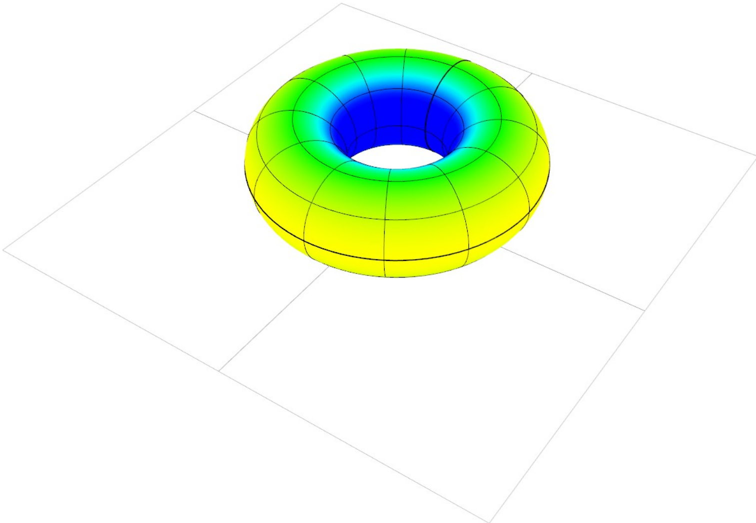


Using the **Hbar** command to make a smooth curve touch the lines of a polygon:



Analyzing Shape

Gaussian curvature
The Gaussian Curvature analysis is used to find how the surface curves.



The torus is convex on the outside, saddle-shaped in the center. The outside surface of a torus is not concave at any point; it only switches from saddle to convex. The green area is where it switches, and this area approaches single curvature at the very top.
Imagine that you are an ant walking around on the torus: At a convex point, the surface curves downwards in all directions. At a convex point, the surface curves downwards in all directions. At a saddle point the surface curves downwards in one principal direction, and up in the other principal direction.

Surfaces can curve in three basic ways:

Convex/Concave	Saddle	Single Curved
GC=Positive	GC=Negative	GC=Zero