

Title of Seminar: Non-organic Surface Modeling
Presenter: Peter Bakic

I. TECHNIQUES FOR FASTER AND EFFICIENT MODELING

Creating a good interface for modeling purposes

Since modeling is largely intuitive process, it is essential to have a good working space that enables us to work quickly and without too much looking for tools. For this, marking menus are essential. Personally, I assign all quadrants of marking menus the same menus, so that I know what I will get every time. This can hold 24 different tools, and while it is much less than if each quadrant had a different menu, it is also a lot more intuitive.

The rest of frequently used commands should be assigned as hotkeys, and only those rarely used should be either on a shelf or in the menus.

Many good functions can be put in marking menus: pick nothing, pick CV, move CV normal, etc.

"Must have" marking menu buttons:

(Note: These should be entered in the script editor and dragged onto a shelf/marketing menu)

```
//Pick nothing
select -all;
select -cl;

//Pick CV
changeSelectMode -component;
setComponentPickMask "All" 0;
selectType -cv 1;

//Pick Object
changeSelectMode -object;
updateSelectionModeIcons;
setObjectPickMask "All" 1;

//Pick Template
changeSelectMode -hierarchical;
updateSelectionModeIcons;
setHierSelectMode -template;
```

Also, transforms should be easily accessible:

```
//Move in world space
manipMoveContext -e -mode 2 Move;
setToolTo $gMove;

//Move along normal
manipMoveContext -e -mode 3 Move;
setToolTo $gMove;

//Rotate
setToolTo $gRotate

//Scale
setToolTo $gScale
```

Custom shelf buttons

For modeling with b-splines, a "uniform rebuild" button is essential. Open the "Rebuild Surfaces" window; in it set these parameters:

- **Rebuild Type to Uniform**
- **Parameter Range to 0-1, or 0 to number of spans**
- **Direction to U and V**
- **Keep Corners to ON**
- **Keep CVs to ON**
- **Degree U and Degree V should be 3**
- **Output Geometry should be NURBS.**

Once these parameters are set, this window can be copied onto a shelf by holding Shift+Alt+Ctrl and clicking on the "Rebuild Surface" menu item. The shelf button will "remember" all these settings for future sessions.

This MEL command does the same as above:

```
rebuildSurface -ch 1 -rpo 1 -rt 0 -end 1 -kr 2 -kcp 1
-kc 1 -su 3 -du 3 -sv 8 -dv 3 -tol 0.01 -dir 2;
```

Hard surface models are more sensitive to minor surface curvature variations; this is especially true for shiny surfaces. Because of that, we

need a shaded display setting higher than the highest one available with the hotkey:

```
displaySmoothness -divisionsU 3 -divisionsV 3 -  
pointsWire 16 -pointsShaded 8;
```

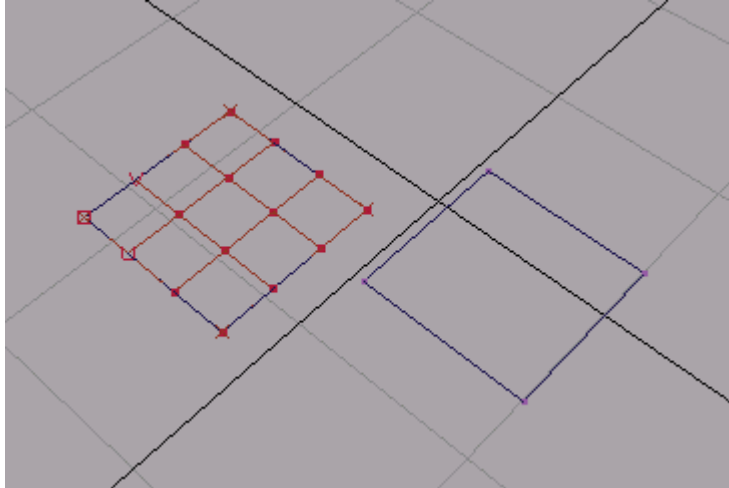
This command will also help with irregular display of trimmed edges. Note that this shading setting will slow down the interface, so it should be used only periodically and not left on for the entire session.

II. ANALYZING THE NEEDS OF A MODEL

The first decision a modeler has to make is the type of geometry that will be used to create a model. This is mostly determined by the pipeline requirements and modeler's preference. Typically, chosen geometry falls into one of these two categories: polygonal and b-splines. B-splines are widely used for complex hard surface models as they give the highest quality surfaces, and are well suited for curved surfaces, as curvature continuity is inherent to third degree splines. Polygonal models are typically used for video games, and a lot of times for more specific applications like low-resolution models, models for dynamic simulations, and so on. It is possible to create a poly model of high complexity, but polygonal geometry has some limitations that make b-splines more attractive.

Advantages of polygonal models

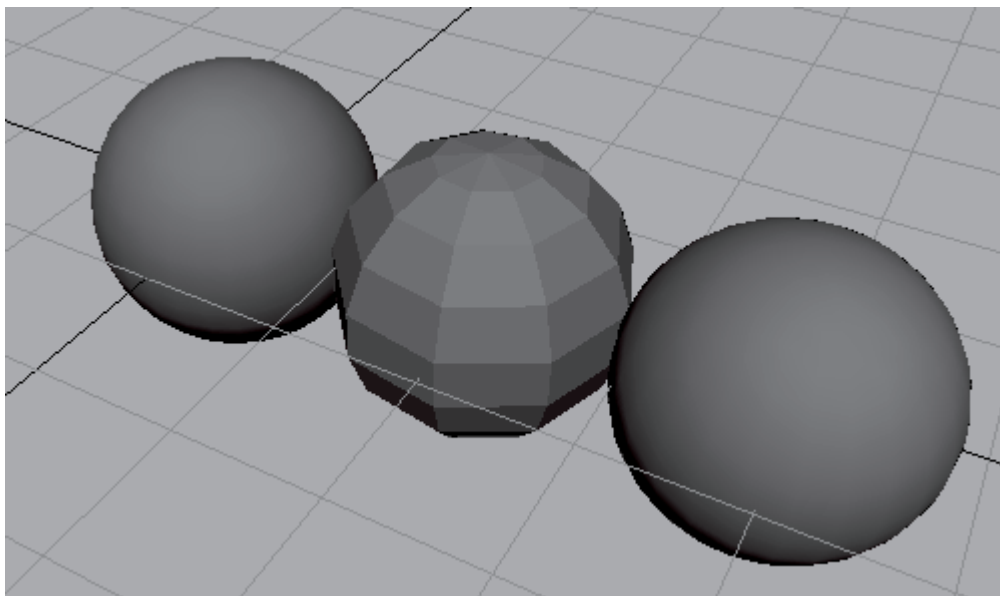
- Polygons are the oldest type of geometry used in 3D computer graphics; therefore there is a universal familiarity with these types of models.
- Polygons are simple type of geometry, therefore it is easy to exchange files between different platforms.
- It takes less data to describe simple surfaces, like a flat plane, using a four-point polygon rather than a planar b-spline patch, which would need 16 points (CVs) minimum.



Polygons use less CVs

Disadvantages of polygonal models

- As each polygon is planar, the silhouette of a poly model is always faceted. The only way to avoid this problem would be to ensure that each poly face is smaller than a pixel in the final rendered image; this calls for a prohibitively high number of CVs, defeating the purpose of using a poly model.
- Changing model resolution usually borders with impossible. Decreasing the number of CVs may give good results, with minimum "clean-up", but increasing resolution and model detail is not as straightforward, as with b-splines.

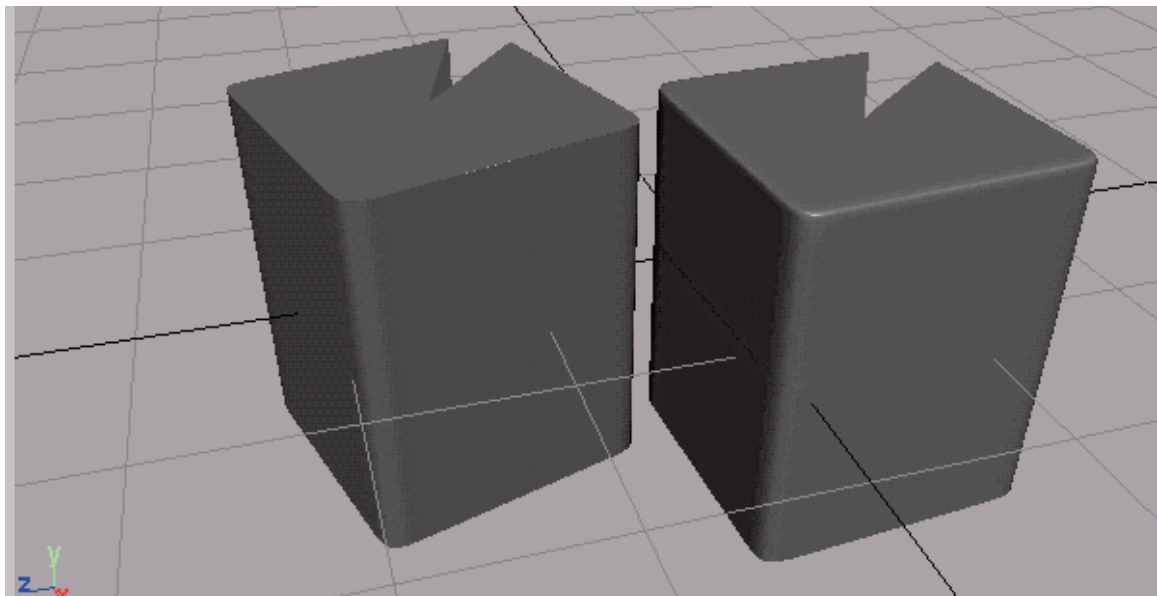


Three spheres

The above image shows three spheres: the one on the left is a NURBS sphere and has 56 CVs, the one in the middle is a polygonal sphere and has 70 vertices, the one on the right is also polygonal sphere and needs over 600 vertices to look as smooth as the NURBS sphere.

Resolution

The resolution of a NURBS model is determined by the relative distance of a camera to the model and the resolution of the final rendered image. A good rule of thumb is that any detail that is less than half a pixel large in the final render may be taken out. However, some details may be just as small; like long thin ropes, or tiny rounded edges. They should still be left in, as due to anti-aliasing they will be visible, and will add to the richness of the final image. The most important detail on any model is its play of dark and light areas. This is most obvious in the highlights, as they attract our eyes attention first (our eyes are more sensitive to light and dark than to color).



The edge makes a difference

The above image shows two cube-like objects where the only difference is that the right one has round top edge. The materials, lights, size, and everything else is identical. This edge and its highlight are less than one pixel in size, yet it makes an important qualitative difference.

Pipeline limitations and requirements

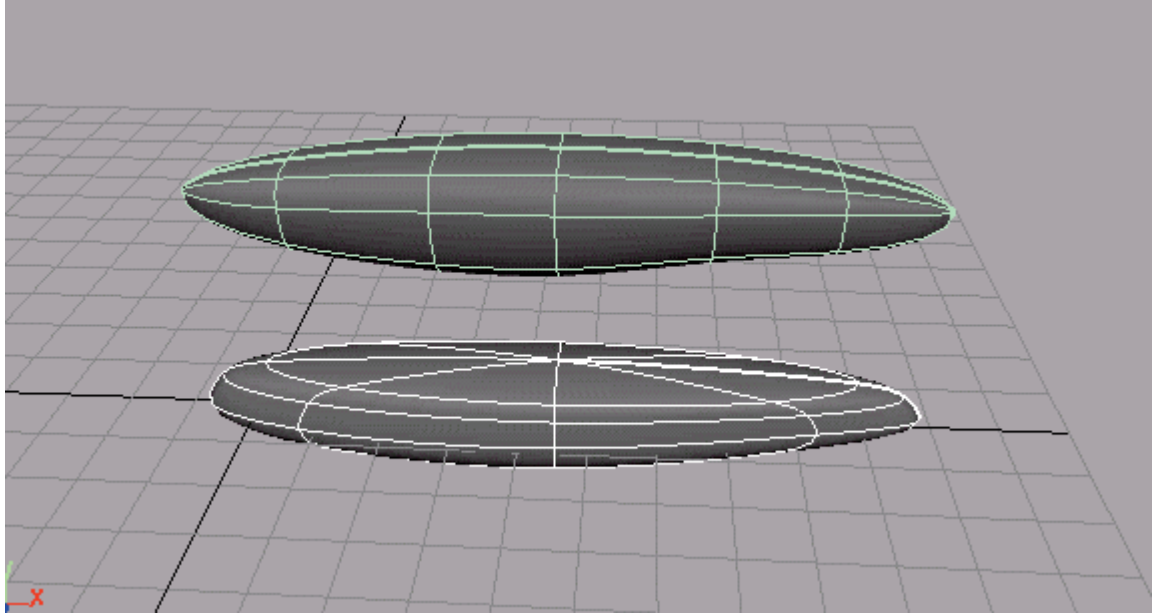
Depending on the rendering engine and possible need for exchange of models between different platforms and the type of animation that will be applied on the model, we may choose to use non-uniform rational b-splines, or just uniform b-splines. Maya offers both options, and for many reasons that will be mentioned later, uniform b-splines have certain advantages over

non-uniform b-splines. These advantages are particularly obvious when it comes to "stitching" a model, or even just trying to achieve a smooth surface as uniform b-splines have more evenly spaced isoparameters and CVs. To achieve uniform parameterization, use the uniform rebuild button that was created earlier.

III. MODEL TOPOLOGY AND NURBS CHARACTERISTICS

Influence of model shape on model topology

Model topology is determined by the way isoparms "flow" along the model. For instance one of the things to avoid in model topology are "degenerate points", where all CVs from one edge of the surface are in the same point (as in the poles of a sphere). Creating a model with the best possible topology is probably the most crucial step in the modeling process. Badly created topology, most of the time, cannot be compensated for later. Topology will have influence on the model's smoothness or roughness, animation, and texturing. Ideally, isoparms on the model will form a square-like pattern where CVs are uniformly spaced. Also, when determining the model topology, the most characteristic lines of the object being reproduced should be used as guidelines for the shape. In the example below, there are two different topologies for the same shape. They were both created from a NURBS sphere, but the bottom model has badly planned topology because it has a degenerate point on the top and also because its isoparms are not really following the flow of the surface. The model on the top has isoparms running lengthwise along the cockpit, and also around the cockpit, describing its circumference, which is correct topology for this case. Its degenerate points are in the smallest areas of the model, and they can easily be changed.

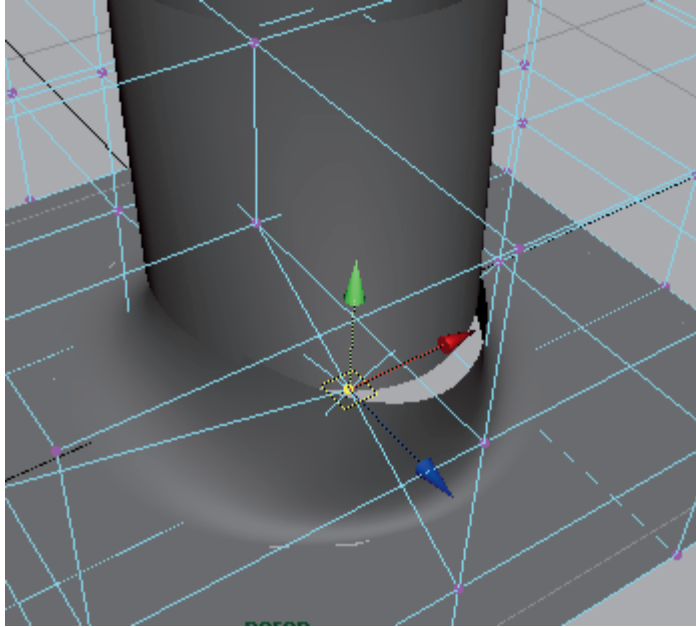


Topology example

IV. APPROACHES TO HARD SURFACE MODELS

Hard surface models with trims

Hard surface models typically represent man made objects like cars, airplanes, ships, robots, and so on. These types of objects often have very complex topology, and require trimming. However, trims add to the complexity of the model, and place many requirements on the model, which further increase its complexity. A trimmed edge will often need a "round" surface to avoid the sharp edge. Trims basically make a part of the surface that is discarded - invisible. Since the invisible part of the geometry is still there, any texture applied to the model will only be partially visible. Also, trimmed edge doesn't have the same characteristics as an edge of the b-spline edge, and we will not be able to manipulate it in the same way - with attach, or to make it easily continuous with another surface. Whether the model will have trims or not is decided at the very beginning of the modeling process, and is usually decided based on the pipeline requirements. If the model will not be deformed in animation, and if the file size is of no concern, then using trimmed models should work. However, if there are any deformations in the animation process, trims may prove to be more trouble than they are worth. In the image below, deformations are applied with a lattice and we can see a gap that forms during deformation, between a trim edge and adjacent surface.

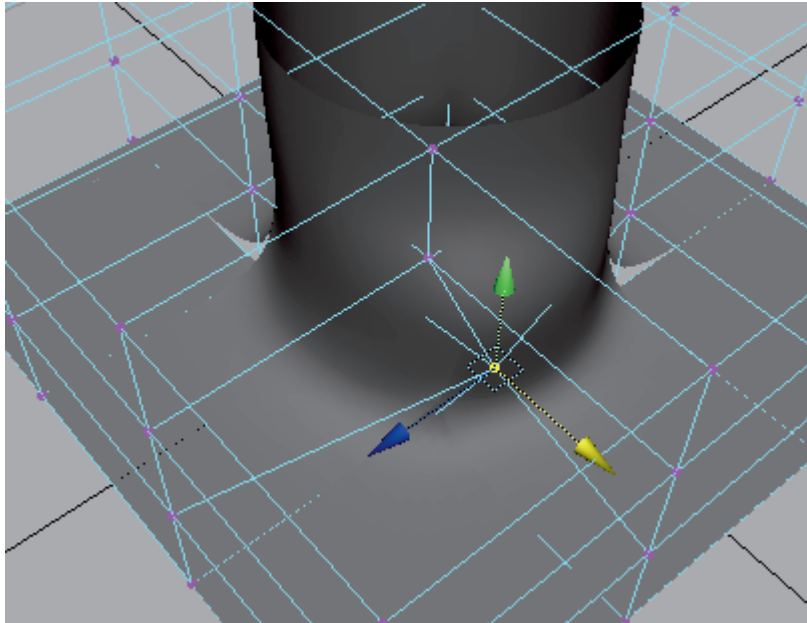


Deformation gap

Hard surface models with no trims

It is possible to create hard surface models without using trims, but it can be a challenge sometimes, depending on the model. In the above example we can see that the same model was created without the use of trims. This type of model always requires more modeling time, but also saves a lot of time in other steps of production. One other advantage of non-trimmed models is that if needed they are easily converted to polygonal models. Trim edges tend to produce a high poly count.

All trims usually happen around holes in the model or around areas where one piece attaches to another. However, it is possible to avoid trims as in the example below:



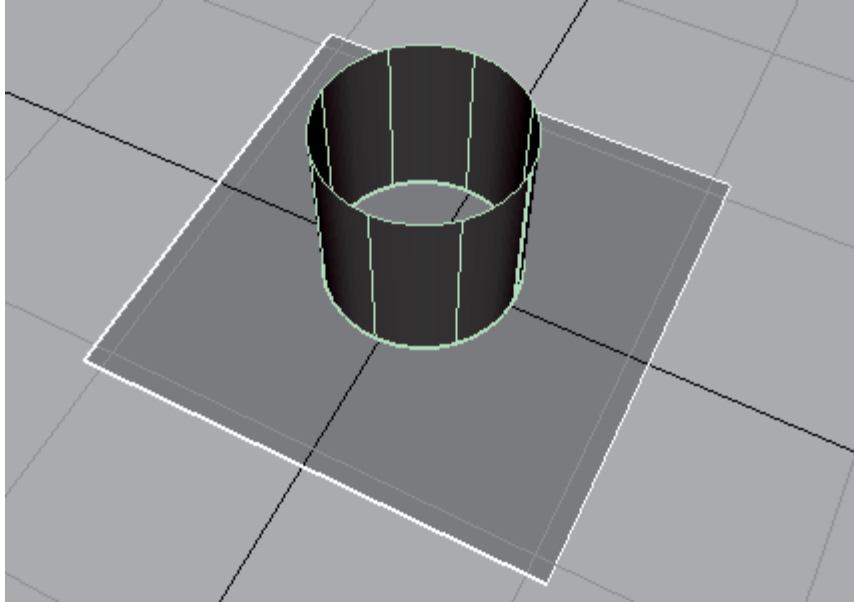
No gap problem

V. CREATING A MULTI PATCH HARD SURFACE MODEL

Going through steps of creating this shape without trims is basis for all models with complex topology.

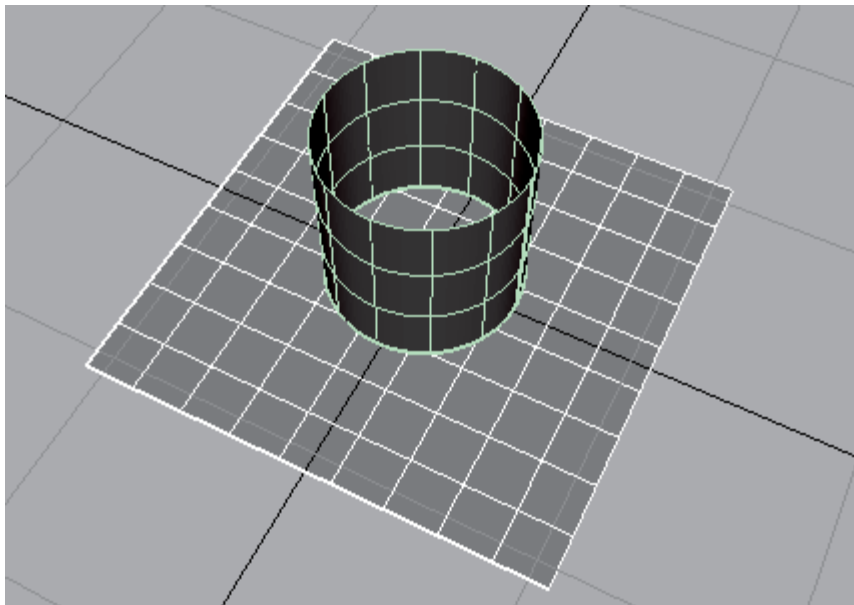
An example using a NURBS Plane and Cylinder

First create a NURBS plane and a NURBS cylinder. Place them at the origin, with the cylinder's bottom resting on the plane.



Proper placement

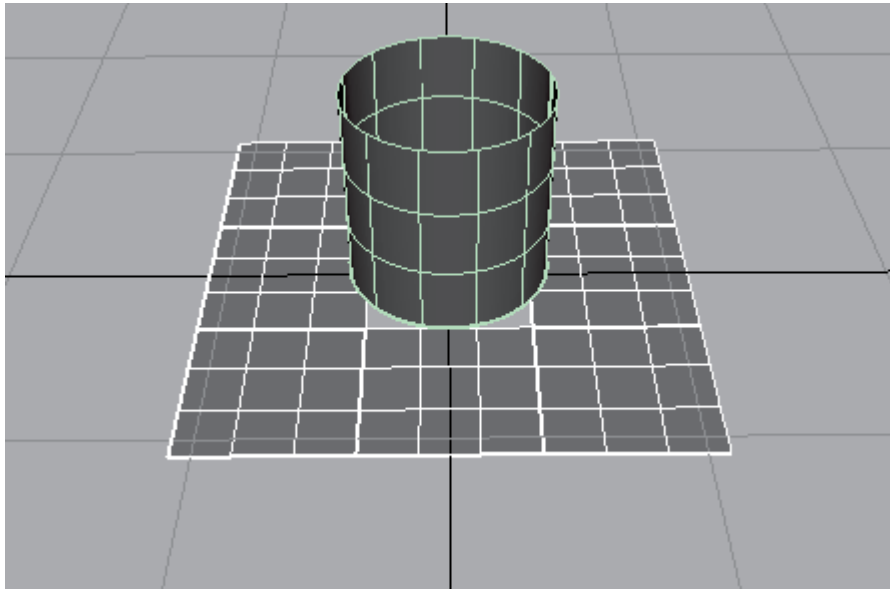
To be able to cut these surfaces, we need to increase their resolution. Typically, it is good to have a minimum of three spans on each patch. Since we have a cylinder that will be cut into four sides, the cylinder will need 12 spans around its circumference and 3 spans lengthwise. The plane will need 9 spans in both U and V.



Changed span count

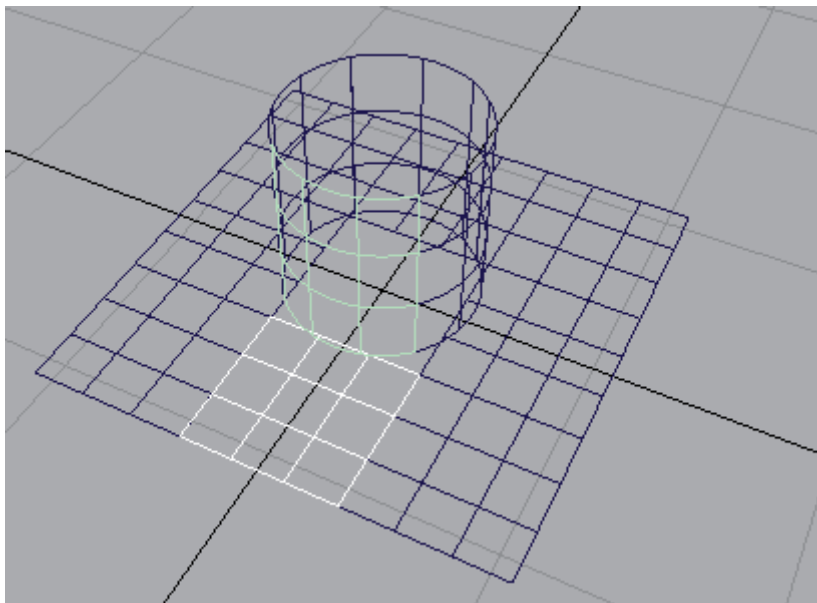
To change the span count as in the above image, use the "Rebuild Surface" window and set the desired number of spans, while keeping uniform option ON.

Once the spans are there, we can align the isoparms from two surfaces by rotating the cylinder 15 degrees around Y-axis.



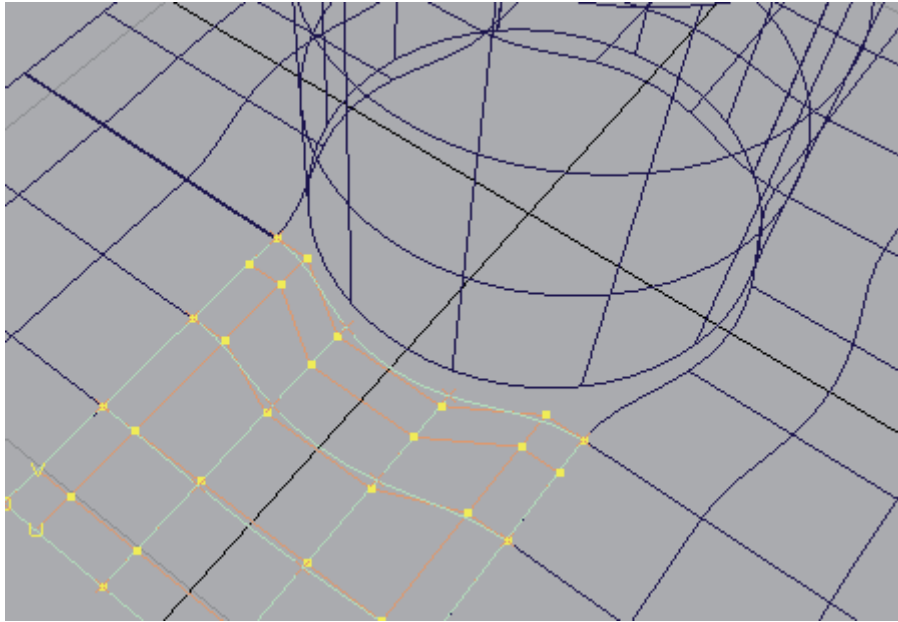
Patched models

Now the most crucial part: recognizing where to cut the model into patches. Most of the problem areas in any model can be represented as this cylinder and plane example. The cylinder should be cut into four pieces, which will correspond with four sides of the plane. The plane should be cut into nine pieces, with the middle piece taken out.



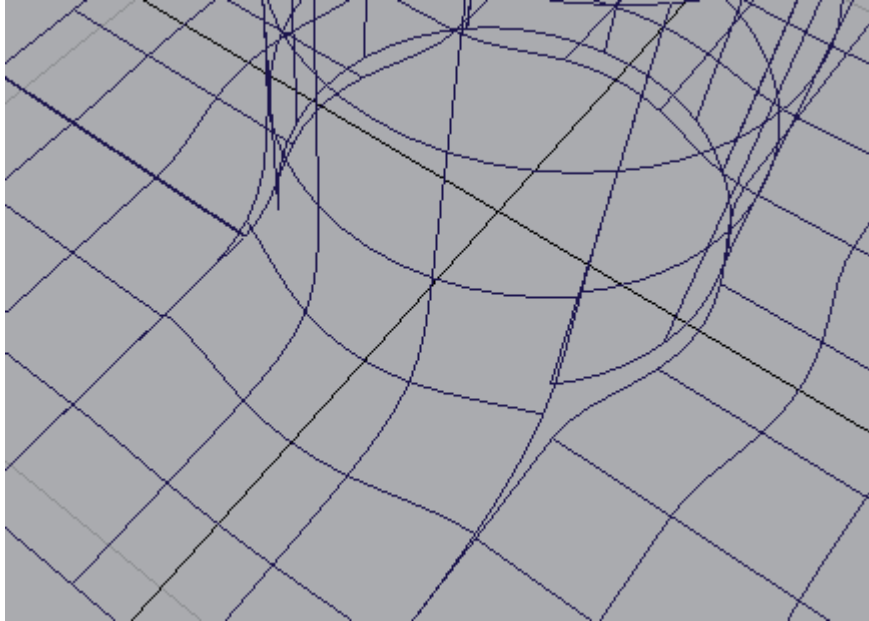
Corresponding patches

To achieve the same look with this setup as with a trim, we need to create continuity between adjacent surfaces. First adjust the CVs on the plane as in the picture:



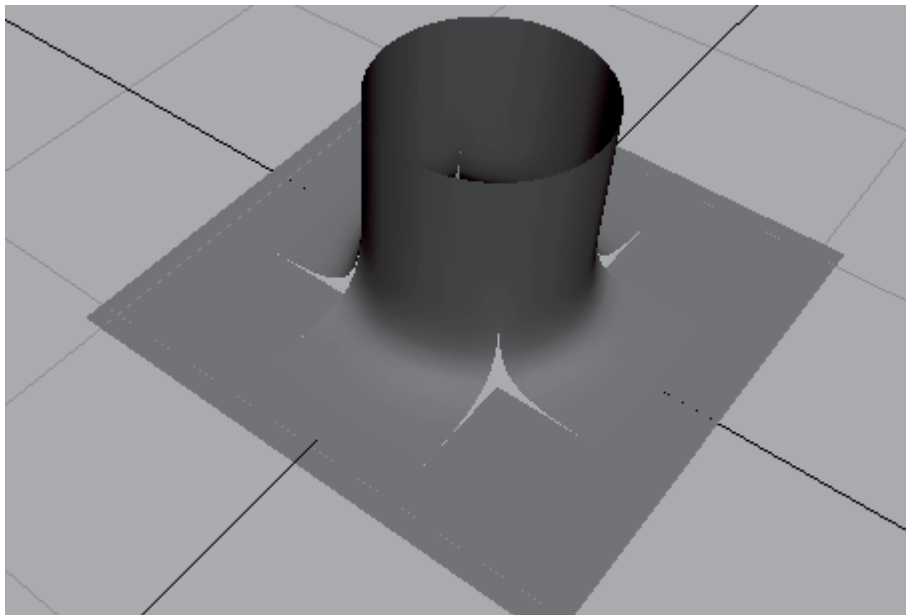
Proper CV adjustment

Now we can proceed with creating smooth transition from one surface to another. There are many ways to achieve this; one would be to use "global stitch". However, with hard surface models, highlights don't permit any deviation in the surface, so it's best to achieve surface continuity (curvature continuity in this case), by doing Surface Edit -> Attach. In the Attach options "Blend" option should be on, and bias should be set at 0.5.



Properly attached

And this is what it looks like with all four sides attached in the same way:

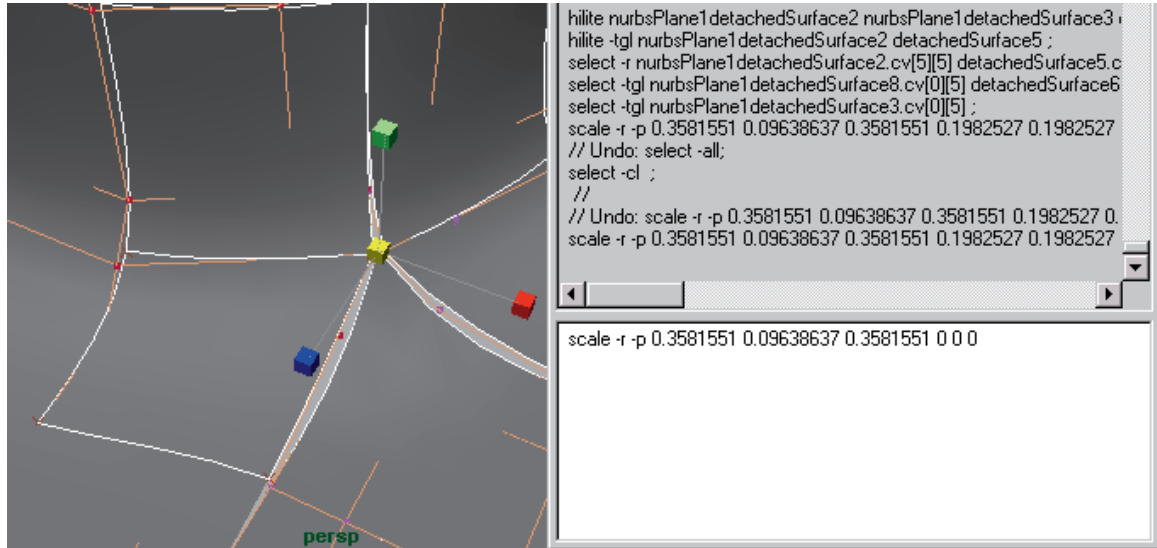


Four sides attached

The area that was attached should be detached using Edit Surface->Detach. This will not change the shape of the model. The model still looks exactly the same as in the previous image.

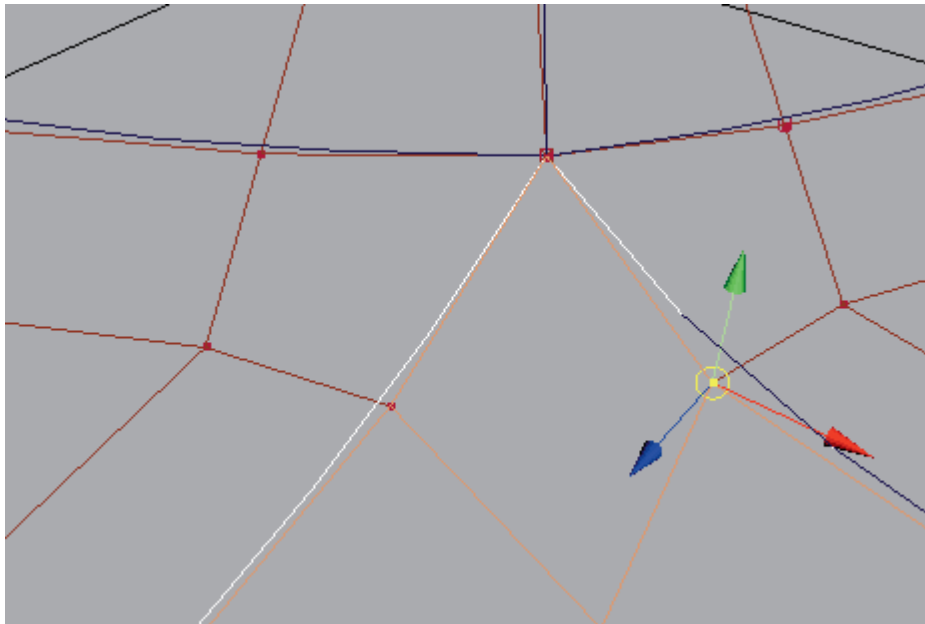
There are still "holes" or "gaps" in the four corners, and we can start closing those using "scale". Select the five CVs that are on the five corners of the five surfaces around the gap and scale them to zero. However, scale can

be negative when done interactively, and we need scale value of zero. This is easily done by doing scale first, undoing it, copying the command from the top of the script editor into the bottom part, and then changing the last three numbers to 0, 0, 0.



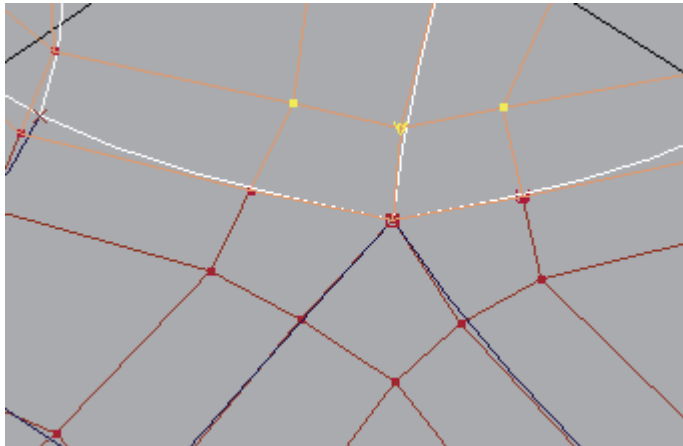
Scale to 0,0,0

Next step is snapping all the CVs that are close together. Most of the CVs are already snapped, except for the two on the bottom patch. To snap CVs to neighboring CVs on the neighboring surface, use the move tool together with "snap to point" button or hotkey.



Snap CVs

At this point the process is almost finished. To ensure the "smooth" look between adjacent surfaces, we need to achieve either curvature continuity or tangency between those two surfaces. You can use this script to achieve tangency continuity between two surfaces. To achieve tangency continuity it is necessary for the two surfaces to have equal number of CVs as well as equal parameterization. It is best if parameterization is UNIFORM. If we look at the snapped CVs at the edges, we see that each of them has one CV as a neighbor. Those neighboring CVs should be selected first (There are two of them - one on each surface), then the two snapped CVs should be selected last - this because of the way the script works.



Achieving tangency

Using a script for creating tangency between adjacent surfaces

```
// This script is for creating tangency between
//adjacent surfaces. The order of selecting CVs is
//important: the two CVs on the edge should be
//selected LAST. The two neighboring CVs should be
//selected first. This script will move the two CVs
//on the edge, in such a way that all four CVs will
//lie on the straight line, creating a tangency
//continuity at that point. Note that the script is
//smart enough to move the middle two CVs by only the
//minimum necessary to create tangency; it will not
//arbitrarily snap them to a mid point between two
//other CVs. This helps preserve the shape of the
//model.
//Make sure FOUR CVs are selected when using this
script.

//ftov and vtof proc:

proc vector ftov (float $f3[])
{
    vector $v;
```

```

        float $f3[];
        $v = <<$f3[0], $f3[1], $f3[2]>>;
        return $v;
    }

proc float[] vtof (vector $v3)
{
    vector $v3;
    float $f[];
    $f[0] = $v3.x;
    $f[1] = $v3.y;
    $f[2] = $v3.z;
    return $f;
}

//procedure for distance

proc float dist1 (vector $v1, vector $v2)
{
    float $dist;
    vector $dif;
    $dif = ($v1 - $v2);
    $dist = sqrt ($dif.x * $dif.x + $dif.y * $dif.y +
    $dif.z * $dif.z);
    return $dist;
}

global proc propTangent ()
{
    //get a list of four CVs, select two in the middle
    //last
    string $cvs[4];
    $cvs = `filterExpand -sm 28`;

    float $end1[3], $end2[3];
    vector $end1v, $end2v;

    $end1 = `xform -q -ws -t $cvs[0]`;
    $end2 = `xform -q -ws -t $cvs[1]`;

    $end1v = ftoV ($end1);
    $end2v = ftoV ($end2);

    float $midPos1[3], $midPos2[3];
    $midPos1 = `xform -q -ws -t $cvs[2]`;
    $midPos2 = `xform -q -ws -t $cvs[3]`;

    vector $midPos1v;
    $midPos1v = ftoV ($midPos1);
    vector $midPos2v;

```

```

$midPos2v = ftov ($midPos2);

vector $midPosRes;
$midPosRes = ($midPos2v + $midPos1v) / 2;
float $distance1;
$distance1 = dist1 ($midPosRes, $end1v);
$distance2 = dist1 ($midPosRes, $end2v);

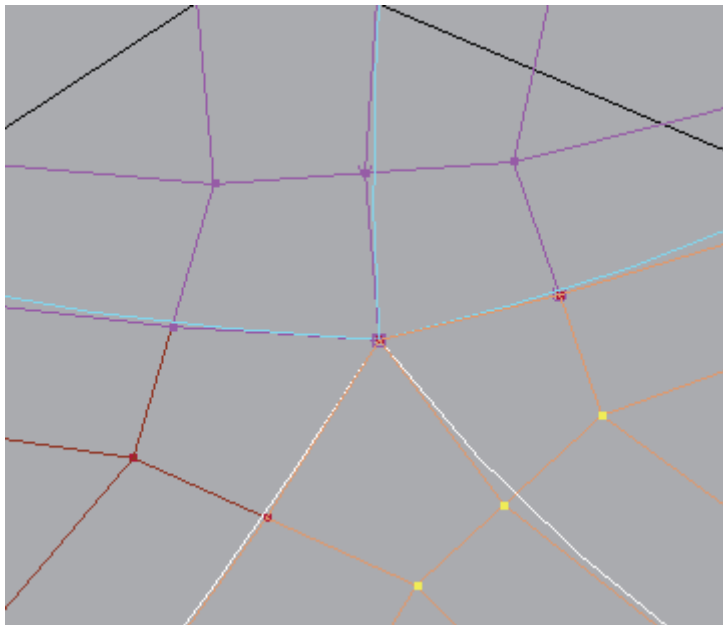
float $ratio;
$ratio = $distance1 / ($distance2 + $distance1) ;
print "Ratio is";
print $ratio;

vector $newPosition;
$newPosition = (($end2v - $end1v) * $ratio) + $end1v;
float $newPositionF1[3];
$newPositionF1 = vtof ($newPosition);
print $newPositionF1;
xform -ws -t $newPositionF1[0] $newPositionF1[1]
$newPositionF1[2] $cvs[2] $cvs[3] ;
}
propTangent;

//End of the script

```

This is what the five corners should look like now, after being made tangent:



Finished corner

The only thing left at this point is to adjust the point where five CVs from five surfaces are snapped together. This is done by hand, and should be minimal.

Final result:

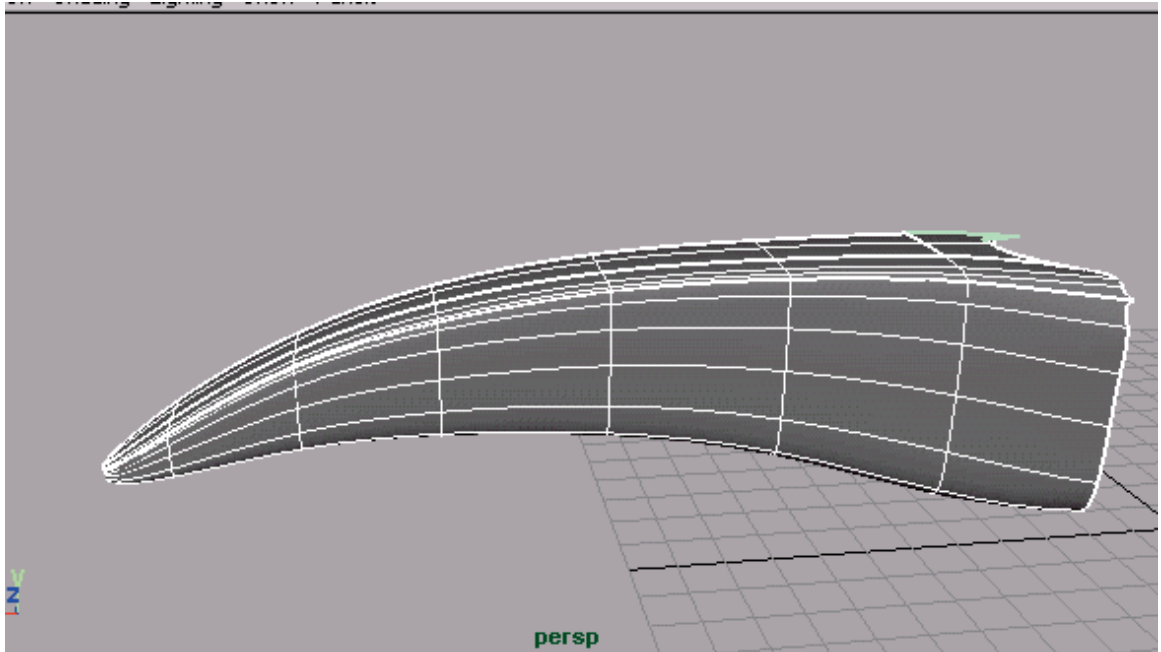


Looking good

The principle explained above is the same for all other situations where more than 4 surfaces meet at one point.

An example with a Formula 1 car

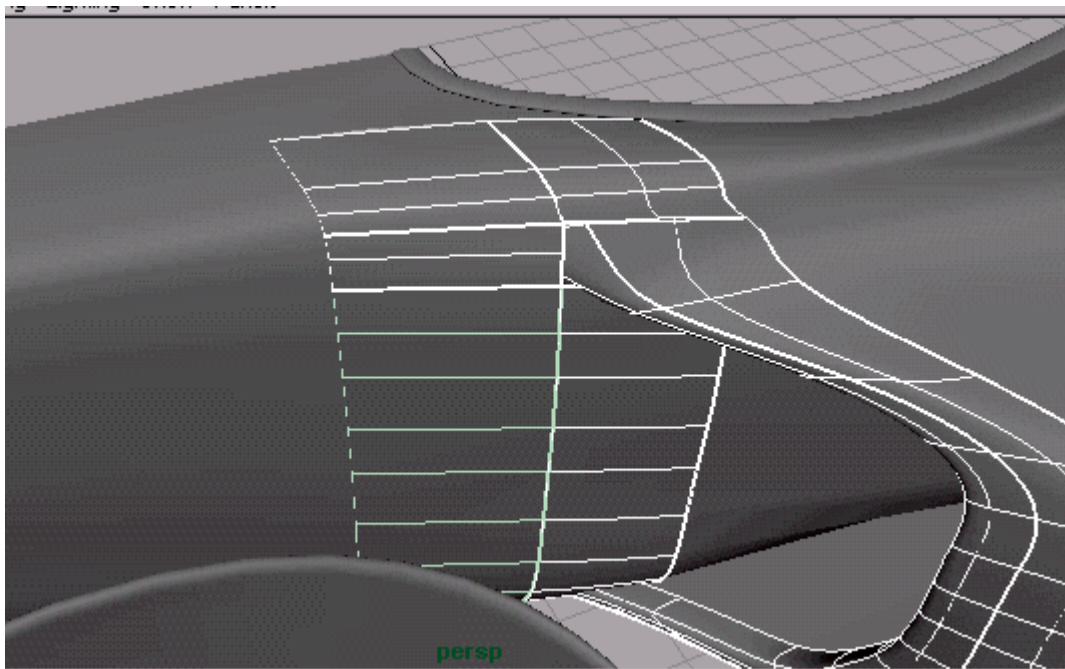
The most important thing when creating hard surface models, which need to be smooth, is to start with as low a span count as possible. This helps preserve the smoothness later on. If we increase the span count before we are finished with the final shape of the model, then we will have to move hundreds of CVs to achieve the same smoothness.



Nose of an F1 car

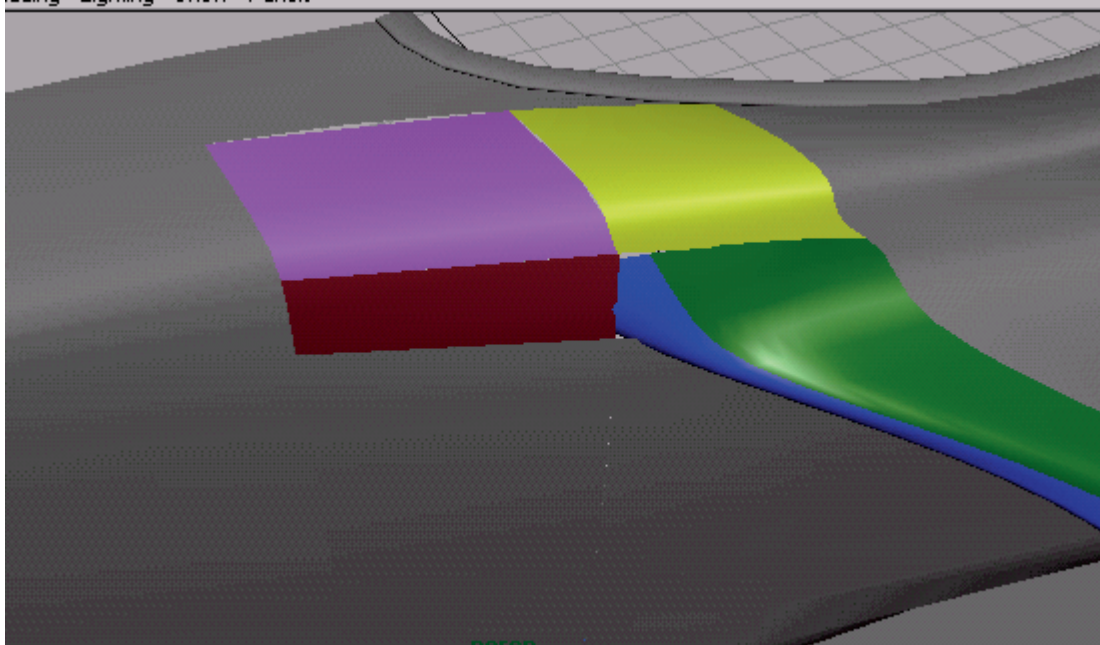
The above image is of a nose of an F-1 car, and was made by lofting through profile curves.

The area just next to the cockpit on an F-1 car, where an air intake meets the main body is very complex, and at first it may seem that it cannot be done without a trim or some sort of a blend surface to connect the two areas.



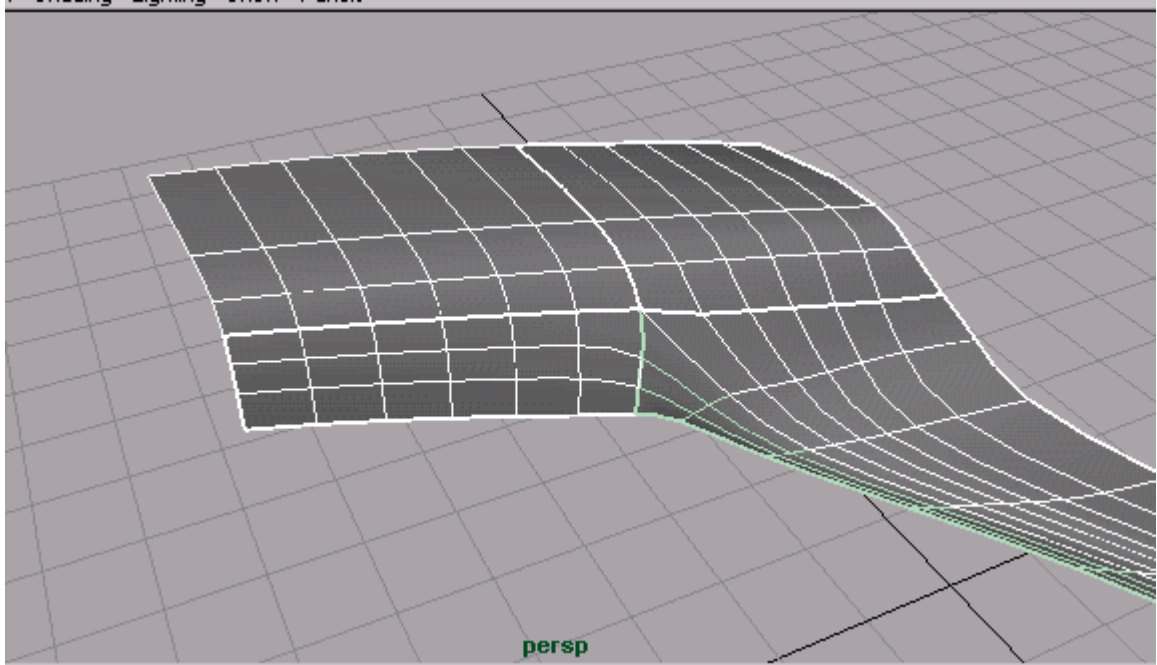
Air intake near the cockpit

We can start by first splitting the model into patches in the same way we did for the cylinder and plane example. Here, the blue and red surfaces will have to be continuous, as will be the case with green and yellow surface. The purple surface will be the fifth one just like in the example with the cylinder and plane:



Colored patches

The next image shows the same five surfaces after they have been aligned using first attach then detach and then the propTangent.mel script, the same way it was all used for the cylinder and plane example. Note that when five surfaces are coming together to a point, if they are not planar, it is impossible to achieve perfect continuity between all of their CVs, simply because there are five of them and these are bi-cubic patches, meaning they have two directions, or four edges, and not five. The more pronounced the curvature in the area where five patches meet, the more challenging it is to achieve continuity with uniform b-spline patches.

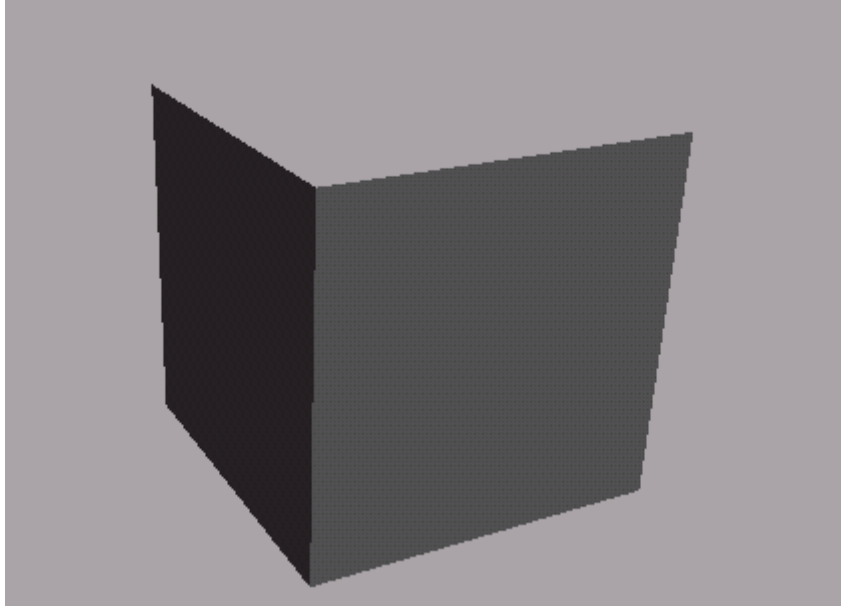


Aligned spans

VI. VARIOUS SURFACE CONTINUITY TOOLS AND METHODS USED IN PRODUCTION

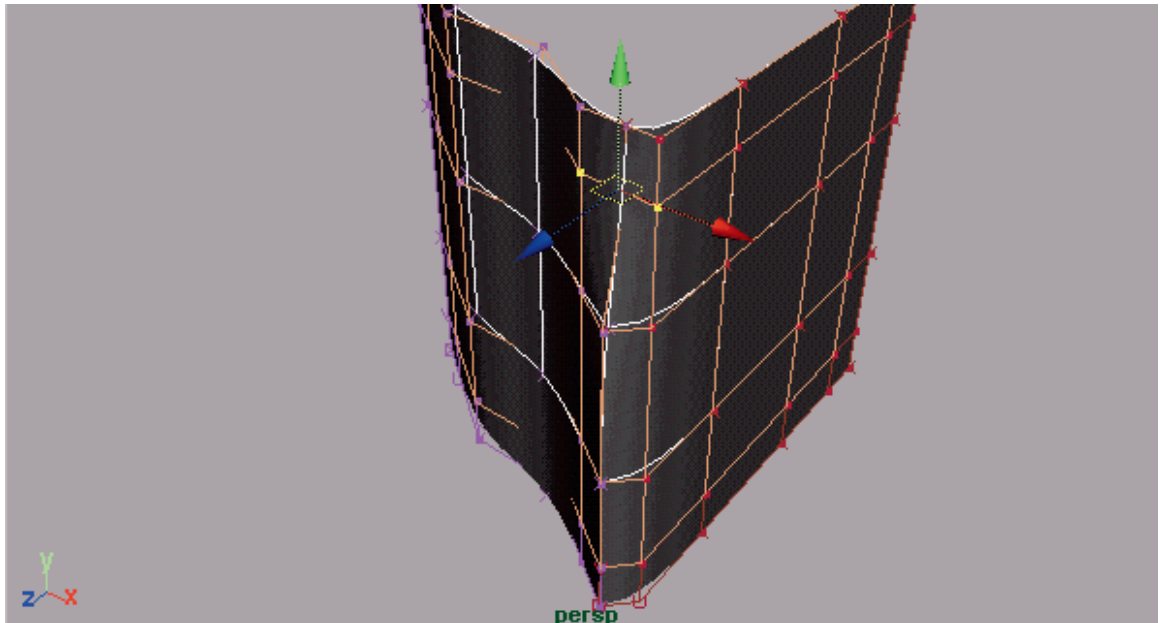
Positional, tangential and curvature continuity with Maya NURBS

(Please note that it is assumed that surfaces are UNIFORMLY parameterized and have equal number of spans. If they are not, these examples will not work)



Positional continuity

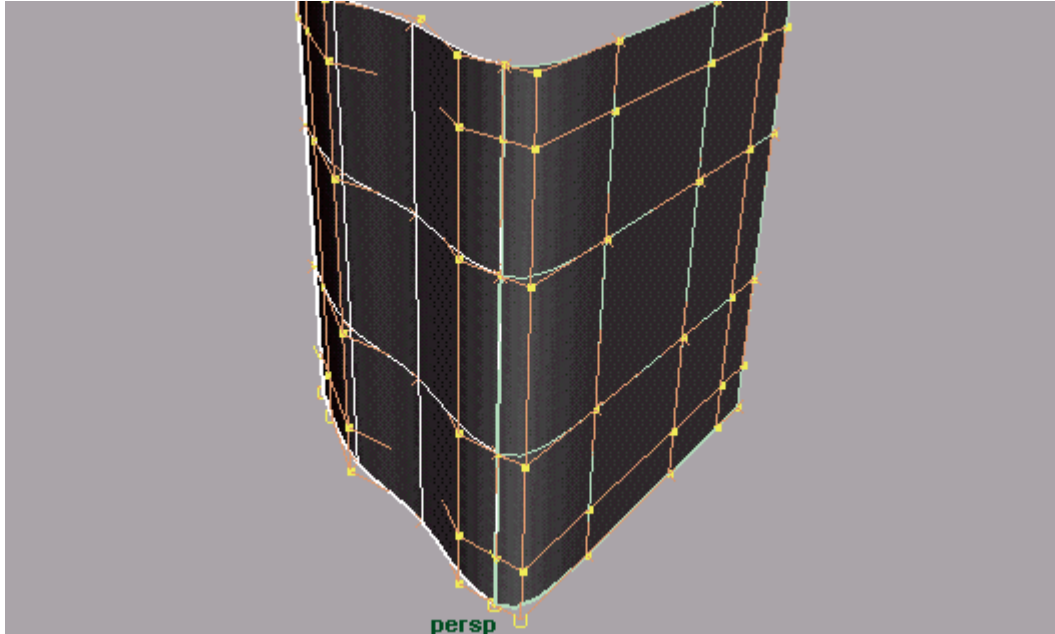
Here we can see that positional continuity creates a sharp edge between two surfaces. To obtain positional continuity, edge CVs of two neighboring surfaces have to be "snapped" together, i.e. edge CVs have to be coincident as in the image above. This type of continuity is rarely used in modeling as it creates an unnaturally sharp edge with no highlight.



Tangency continuity needed

When we snap edge CVs onto an imaginary point that lies on a straight line between two neighboring CVs, then we get tangency between those surfaces. In this example it is achieved using the same `propTangent.mel` script, which is supplied here. The same effect can be achieved if one

creates a curve of first degree, and then snaps two edge pivots to the curve. The problem with using the curve is that it is only "precise" when snapping to the middle of the curve; otherwise an uneven edge may result. Using the supplied script solves that problem and speeds up the process as well.

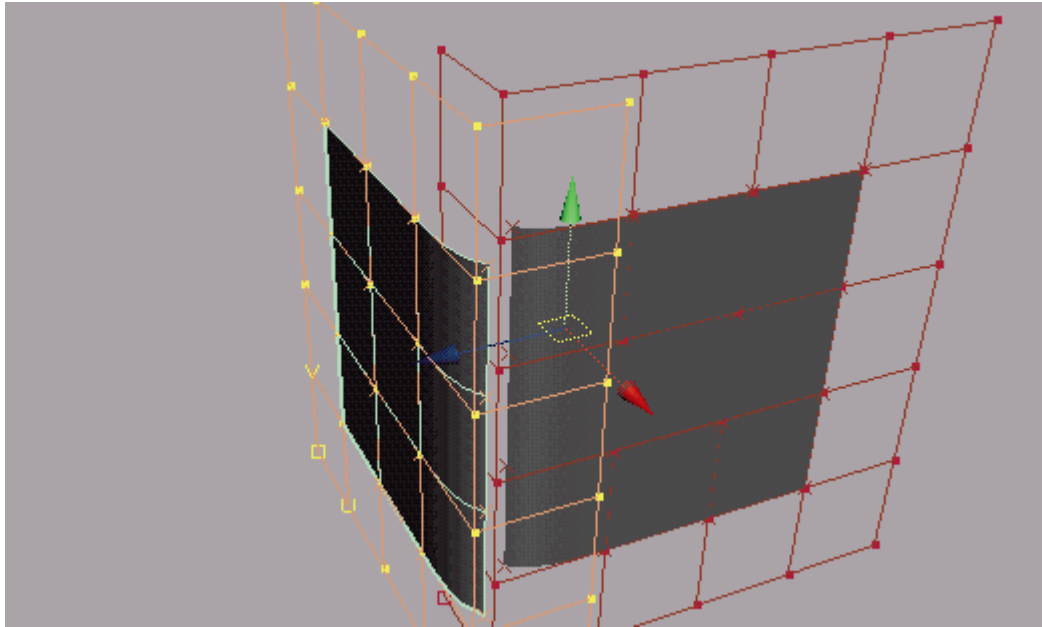


Tangency continuity achieved

This shows the final result after using the propTangent procedure. The edge between two surfaces has "smooth" look to it. However, it doesn't have curvature continuity. For curvature continuity, one needs to move three CVs on each surface, i.e. to snap them together. In Maya we can set up open or closed end conditions, which is done in the "Rebuild Surface" window by checking on:

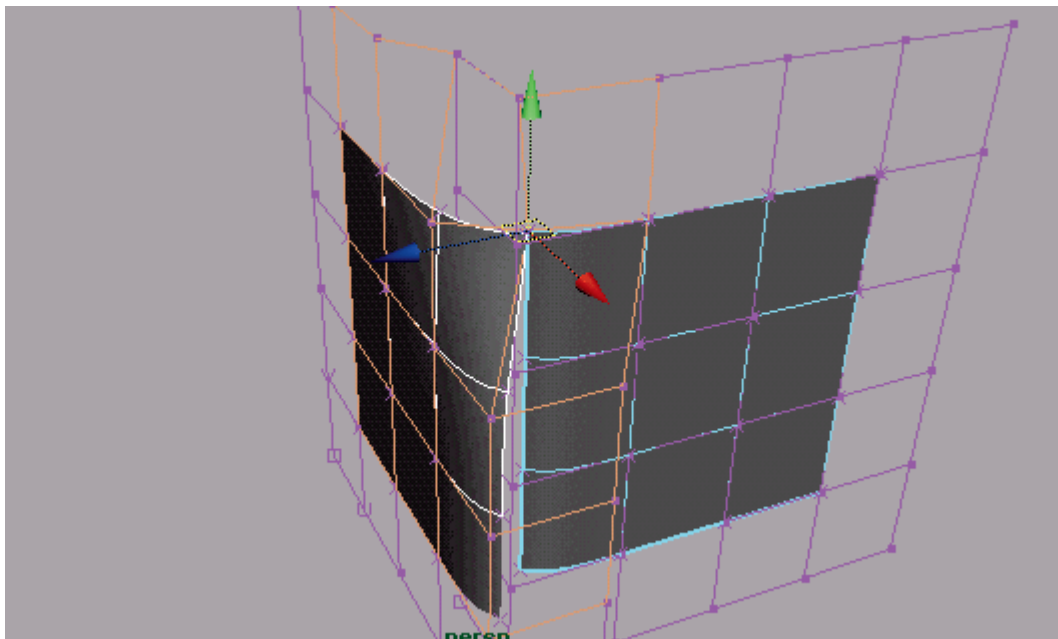
- End Conditions
- No multiple knots

This will "unpile" end knots of our surface, and make it appear as if CVs are hanging over the edge of the surface. This is actually what a real b-spline surface looks like. However, it is often hard to work with, as end CVs are not coincident with the edge of the surface or curve. To achieve curvature continuity, however, this is exactly what we need.



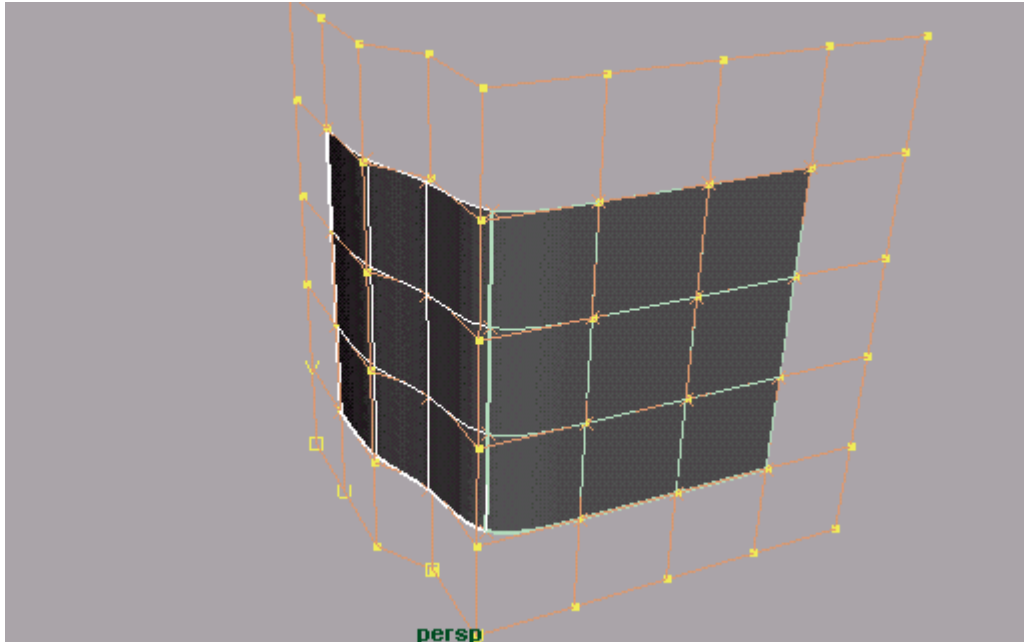
No continuity

This image shows two surfaces that have no continuity at all. By snapping the last three CVs on one surface to the last three CVs on the other surface, we achieve curvature continuity.



Snapping

This image shows the process of snapping the last three CVs from one surface to the other.



Snapping achieved

And this is what the surfaces look like once the snapping is done. The appearance is that of a single surface. Therefore, when modeling highly smooth surfaces curvature continuity may be a better option than tangency continuity.

To return to regular or "multi knot" surface, as it was before, use Rebuild Surface with:

- **End Conditions** to **ON**
- **Multiple knots** to **ON**

It is quite hard to achieve a 5-patch corner using open-end knots, so we can either achieve tangency in those places, or use global stitch.

Stitch and Global stitch

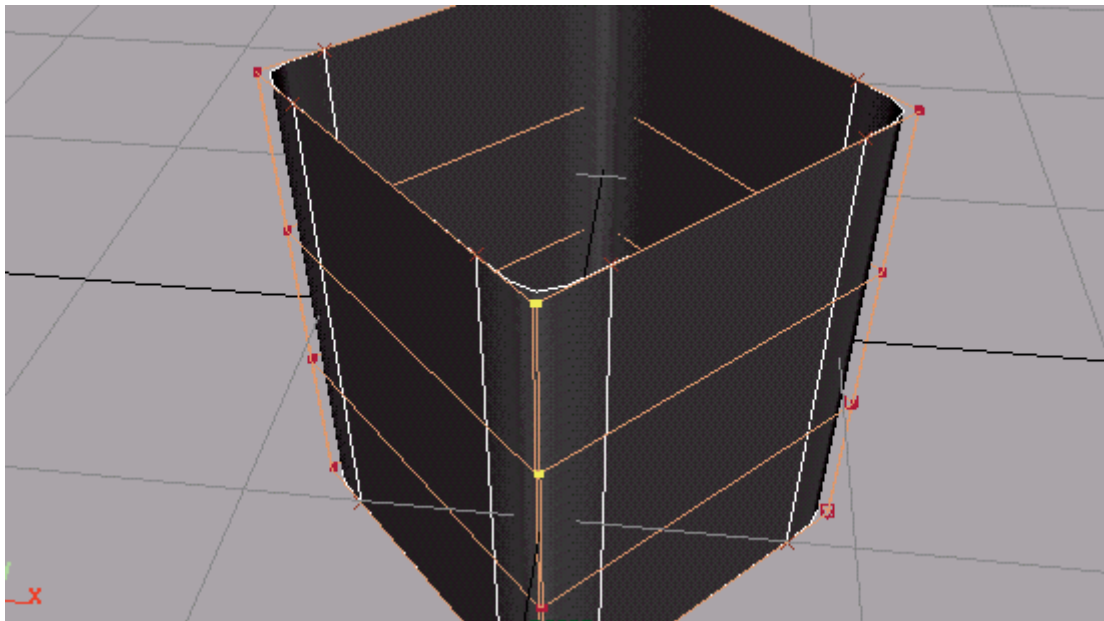
Global stitch saves a lot of sweat in achieving surface continuity on patch models. If patches are well prepared, meaning they are uniform, and close to the final look, one click of the global stitch button may be all it takes to achieve continuity. The problem is that most hard surface models don't allow for too much change, so this step is a lot of times best performed by hand rather than by a tool. However, if the surfaces are already continuous, then Global stitch will not introduce too much change. Therefore it is better to create your own continuity the way you want it than to let computer decide for you.

VII. MODEL DETAIL LEVEL, COMPLEXITY AND SMOOTHNESS

Model detail level depends on the requirements of the final render. Typically it's the resolution of the final image. The limit to what we can see on the final image is one pixel, therefore it is unnecessary to model anything that is smaller than a pixel in the final image. Sometimes, due to anti-aliasing and animation, it may be possible to perceive even sub pixel detail; however, usually those are painted in textures rather than modeled. Without firm knowledge of the closest position of the camera in relationship to the model and without knowledge of the final render resolution, it is impossible to tell how much detail is needed.

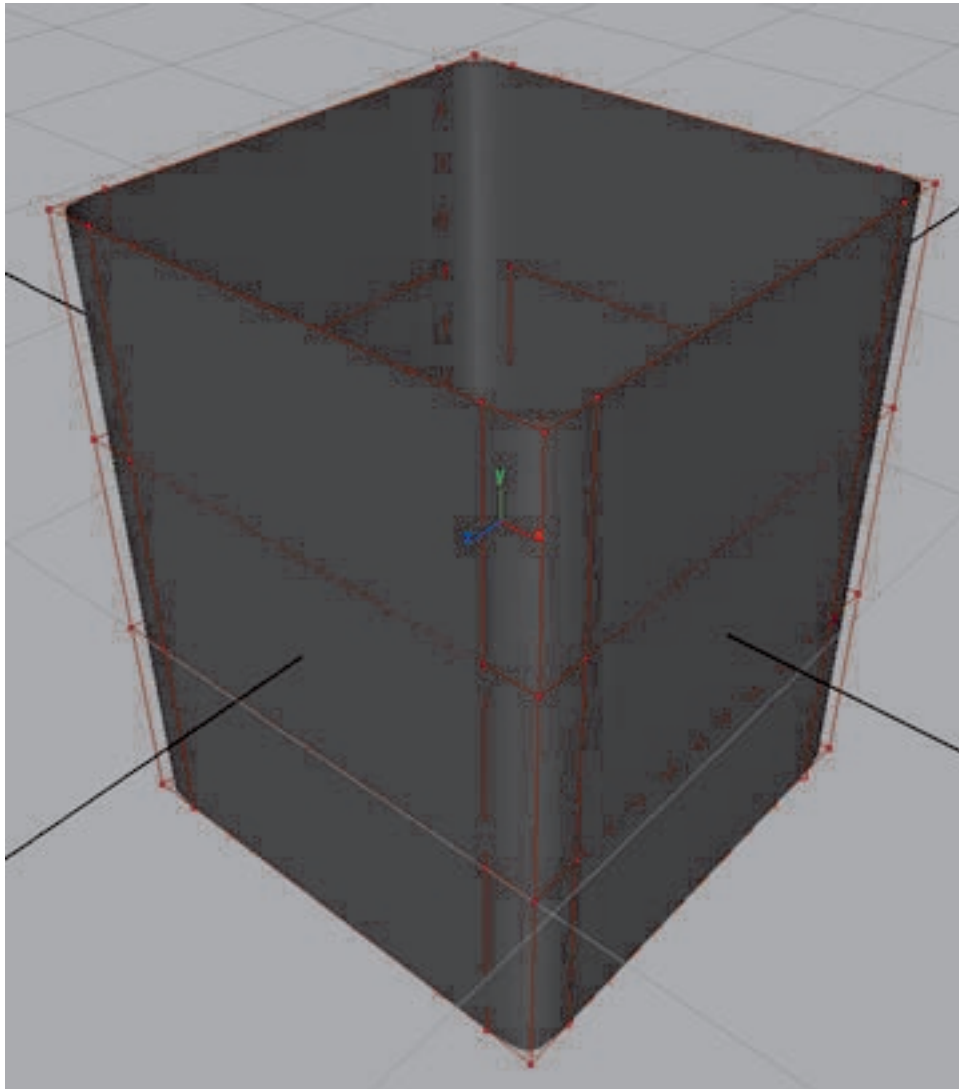
Achieving model detail with minimum geometry

The single most important type of detail on hard surface model is edges. Very few surfaces around us have no highlights on the edge. Vast majority of them do have a highlight at the edge, and the size of that highlight is determined by the roundness of the edge and specular component of the surface. Just by achieving the correct roundness of the edge, we give the illusion of the model detail. Edge roundness is best achieved by grouping CVs close to the round edge.



Two CVs close to the edge

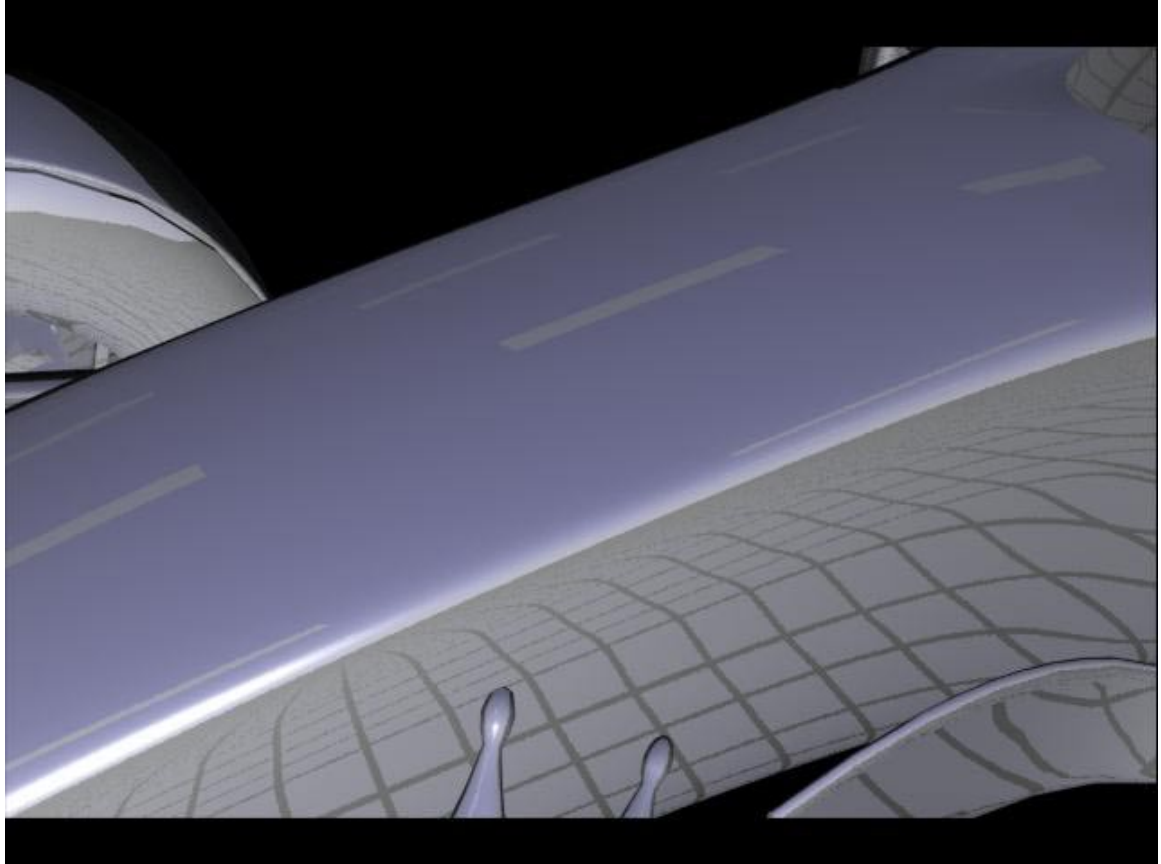
Above image shows a round edge where two CVs overlap. This is the simplest form of a round edge, and doesn't give much control to the modeler. However, for small objects this is usually sufficient and has an advantage of a very low CV count.



Three CVs close to the edge

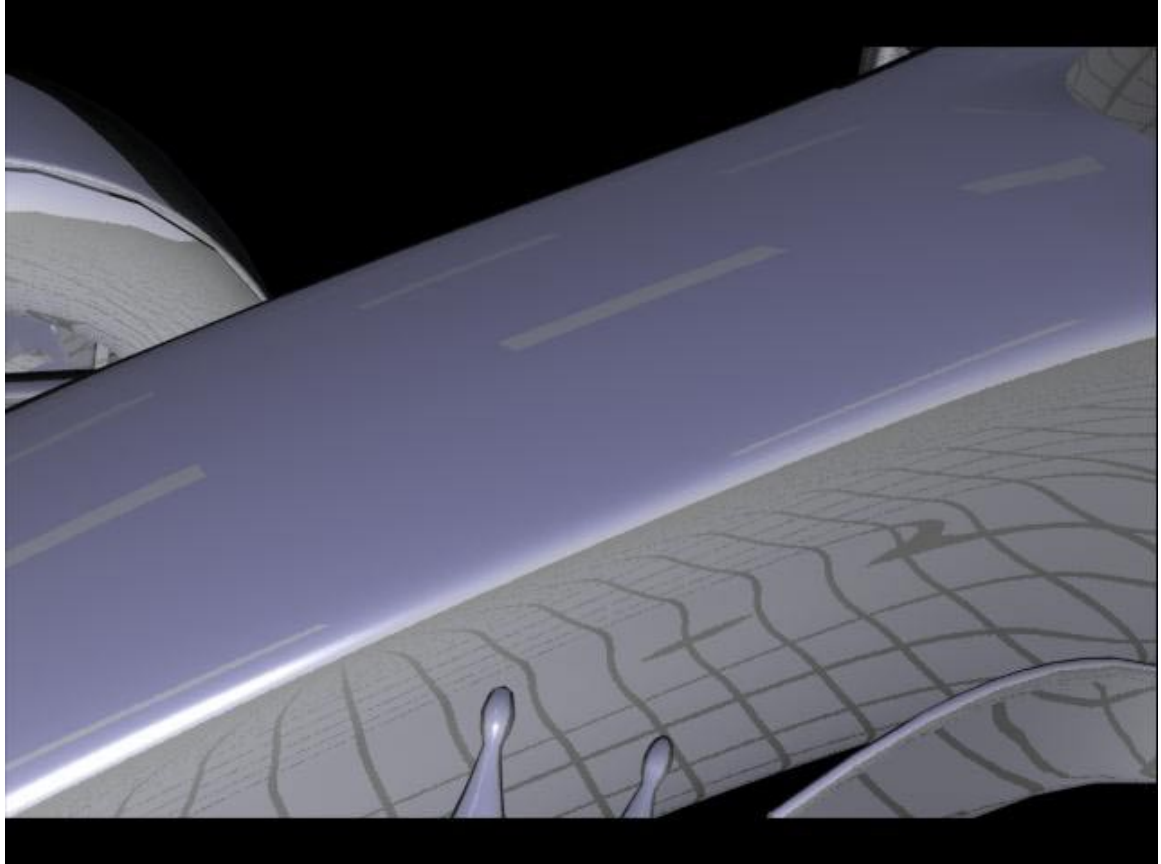
Above image shows round edge with three CVs close together. The edge is sharper, and the model will have more CVs, but there is more control over the look of the edge.

Achieving model smoothness

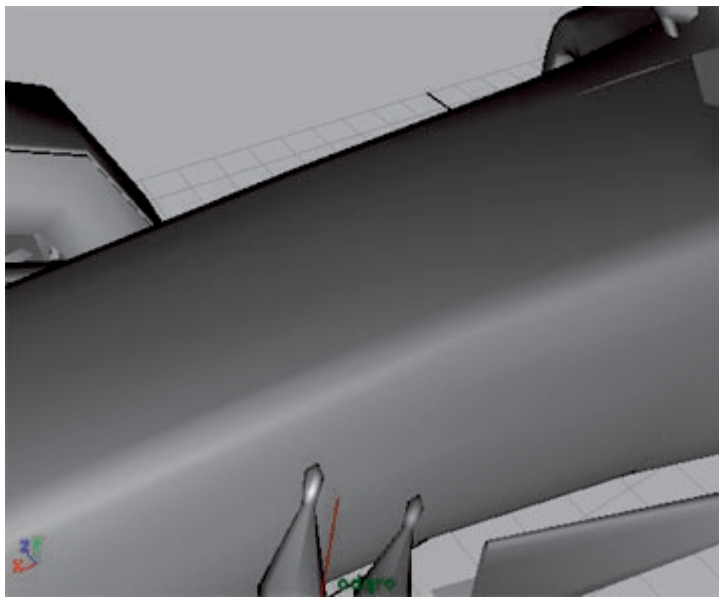


Smooth surfaces

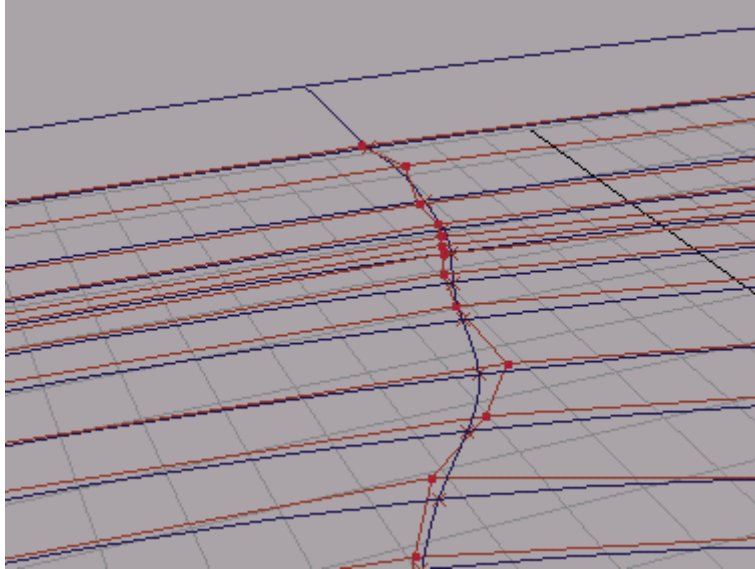
This is one of the greatest challenges. Regardless of how hard we try to preserve the original smoothness of the surfaces we created, we will always have to adjust position of some CVs, and that requires not only a good feel for modeling, but also good understanding of surface smoothness. The first thing to look for when trying to smooth out the model is how well lined up CVs are. By using edit->move->normal we can adjust those CVs that were away from the surface. However, we also need to visually inspect the surface and make sure that all the CVs are nicely lined up. The difference this makes can be seen if we raytrace the model with it reflecting some grid pattern. These types of renders are often needed to define problem areas.



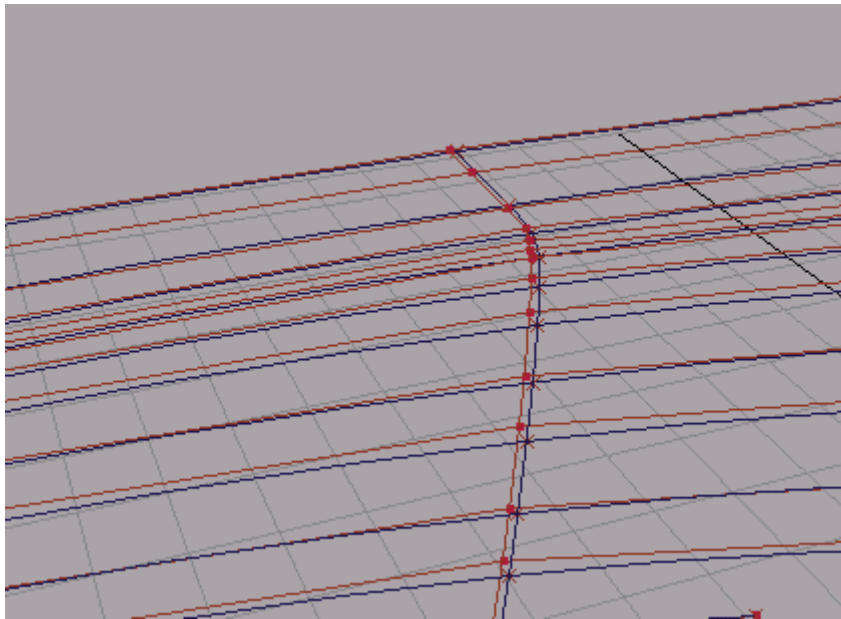
Not so smooth



Take a closer look at the CVs



CVs not lined up



CVs lined up

Note that in the image where CVs are not lined up, they are still on the surface; they haven't been moved along the normal, only along U and V space. However, this still creates small but important variations in the surface curvature. Therefore MoveNormal is an essential tool for modelers.

VIII.ALTERNATIVE MODELING APPROACHES

Using MEL

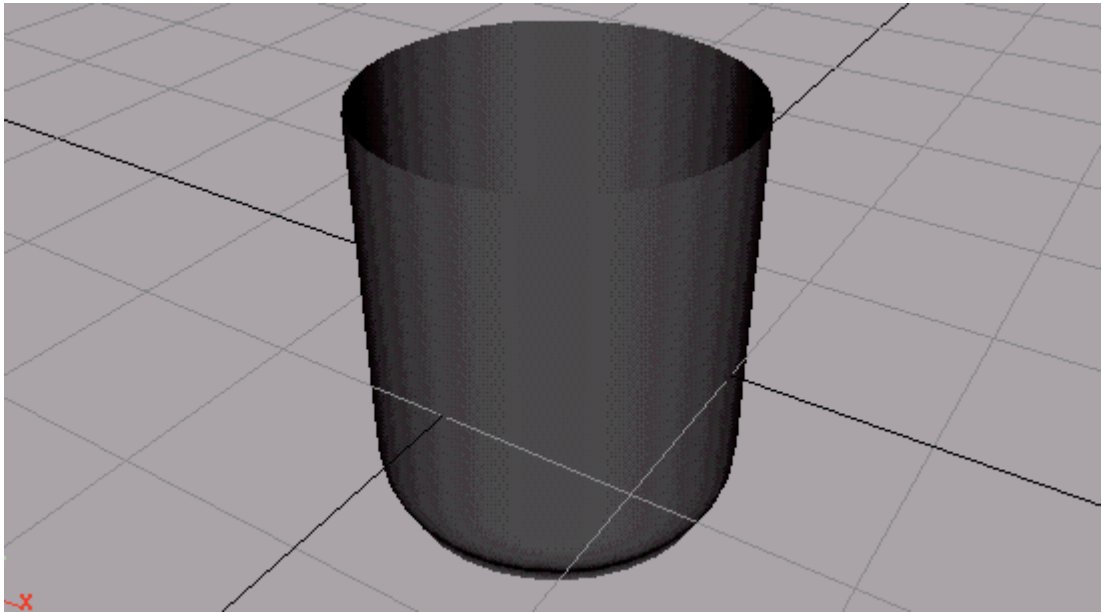
The best way to get familiar with MEL is to read the commands that are echoed in the script editor. Turn off "echo all commands" under edit menu in the Script Editor if it looks too complicated. If you want to learn about interface, then it is a good thing to have "Echo all commands" ON, as it will give you an idea about how some interface commands work.

Tip: you can skip all the lines in Script Editor that have a "|" (pipe) in them.

Using Animation and Dynamics

Here's an example of quick result with dynamics:

- Create a default cylinder from "Create" menu as in this image:



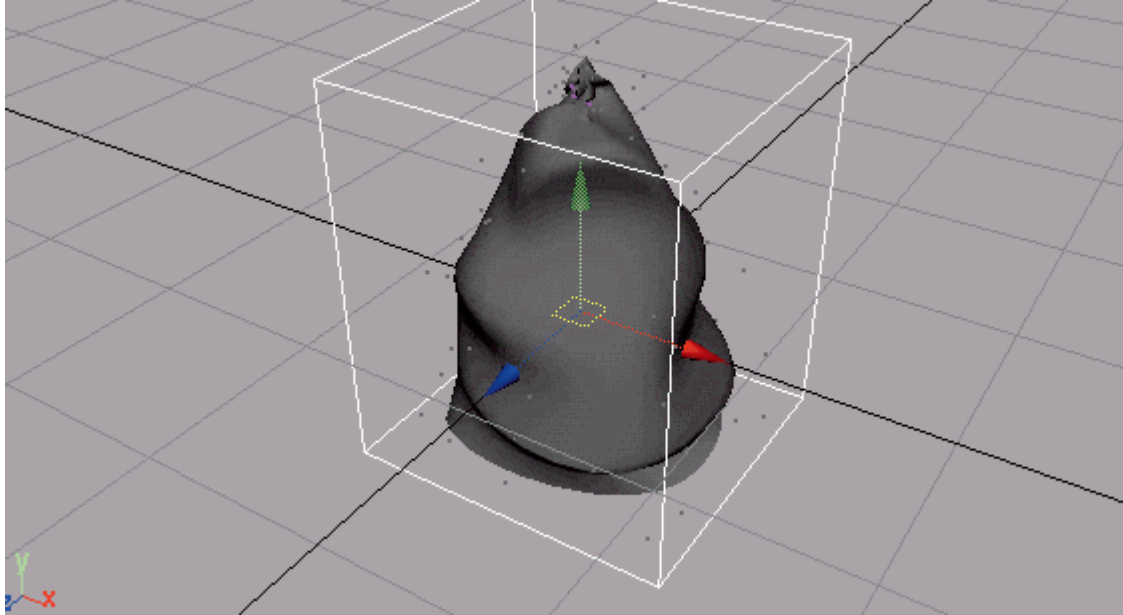
Default cylinder

- Change the number of spans to 8 by 16.

Select the cylinder, and press F4 to go to dynamics menu. There, click on "create soft body". Default settings should be fine.

While the soft body is still active, in the Fields menu click "Newton Field". Move the Newton Field to the top of the cylinder. Select the cylinder again and click on the "Turbulence field". In the channel box on the right, change the value of the turbulence field's frequency to 50.

Press play, and watch the cylinder turn into a duffel bag.



Looking more like a bag

This kind of trick can be applied on the lattice as well to give the model a bit of a warped look.

IX. MEL IN MODELING

Introduction to MEL for modelers

Custom tools with MEL

All the commands and actions we perform in Maya are printed out in the script editor, and there is no better way to learn MEL than to look at the script editor. A beginner can just copy and paste some commands and later try to adjust some parameters. By combining a few different commands, we get a script that can be very useful in a marking menu or on the shelf as it saves many clicks of the mouse.

The above script - PropTangent is a good example of how a lot of time can be saved with MEL for something that takes a lot of time when done "by hand".

One of the essential commands for a modeler to know in MEL is "xform". To get a world space position of a CV, xform is the best way.

Example:

```
xform -q -ws -t "sphere1.[0] [0]";
```

This will return three floats (a float array) for X Y and Z position of the CV.
The same command can be used to move a CV.

Note that the supplied script "propTangent" also has three other procedures in it. Two of those convert float array to vector and vice versa, and third one calculates the distance between two vectors.

X. Q&A