

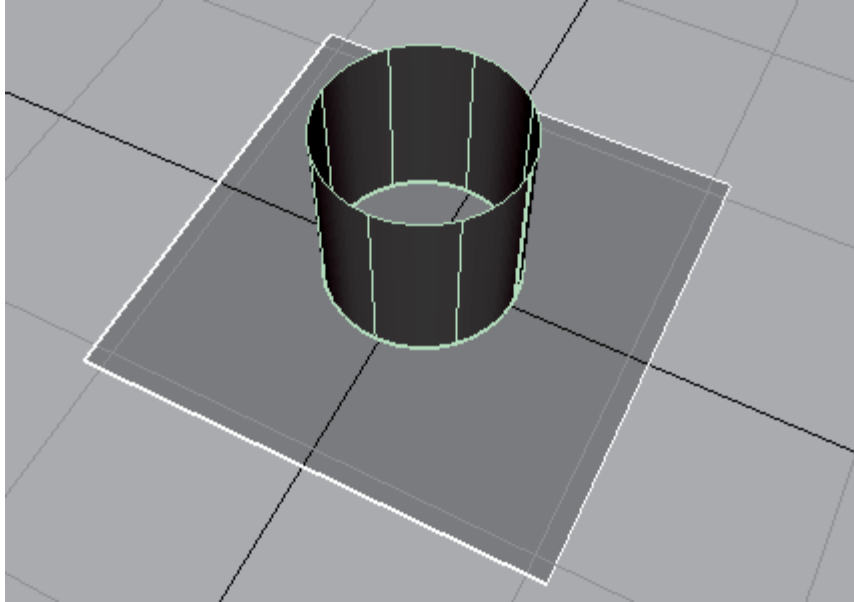
No gap problem

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.

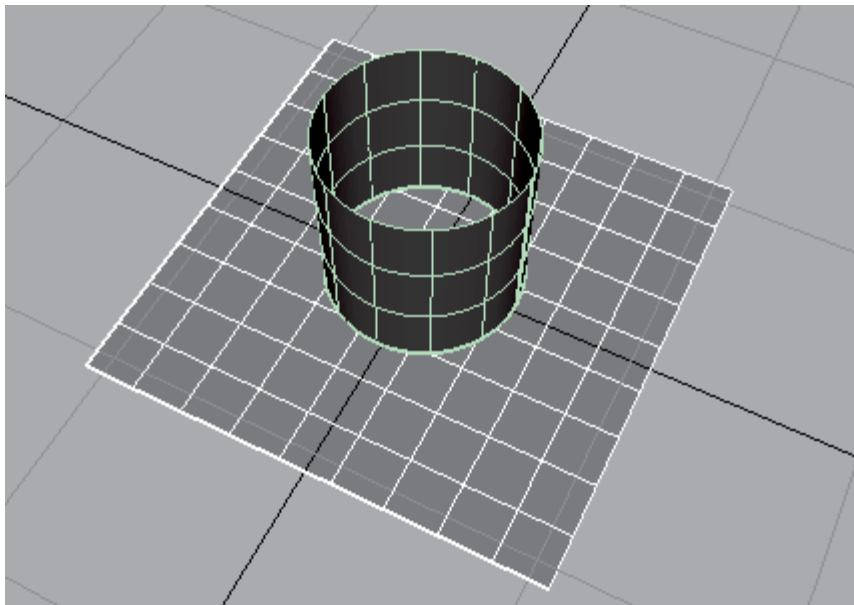
An example using a NURBS Plane and Cylinder

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

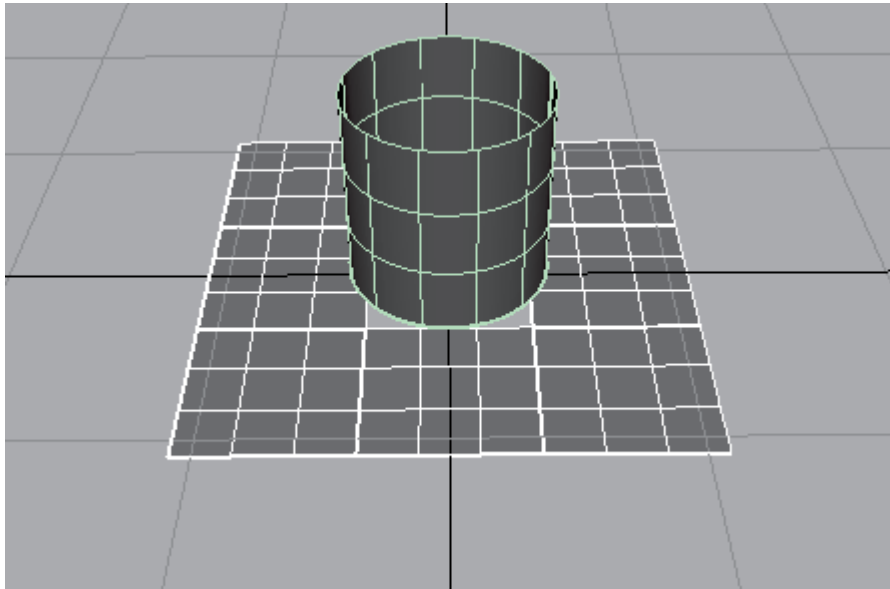
To be able to cut these surfaces, we need to increase their resolution. Typically, it is good to have a minimum of three spans on each patch. Since we have a cylinder that will be cut into four sides, the cylinder will need 12 spans around its circumference and 3 spans lengthwise. The plane will need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



Changed span count

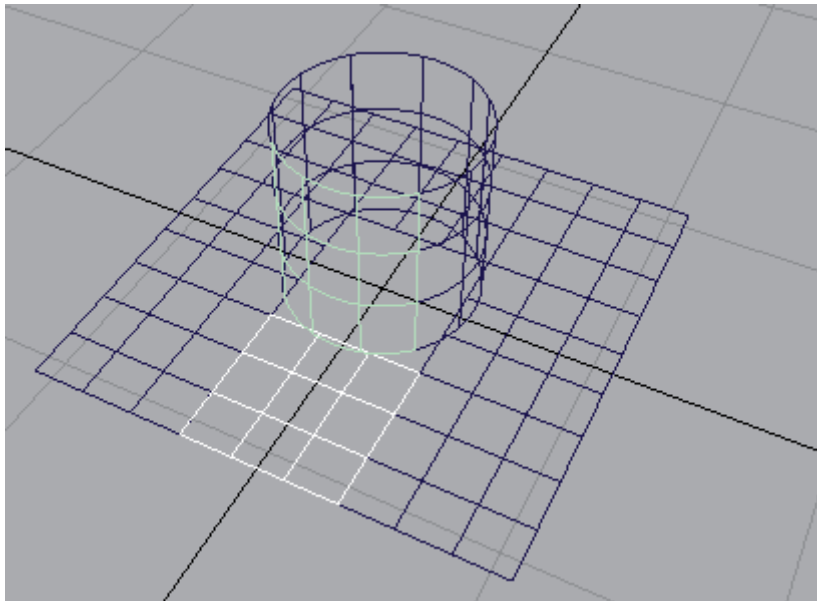
To change the span count as in the above image, use the "Rebuild Surface" window and set the desired number of spans, while keeping uniform option ON.

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



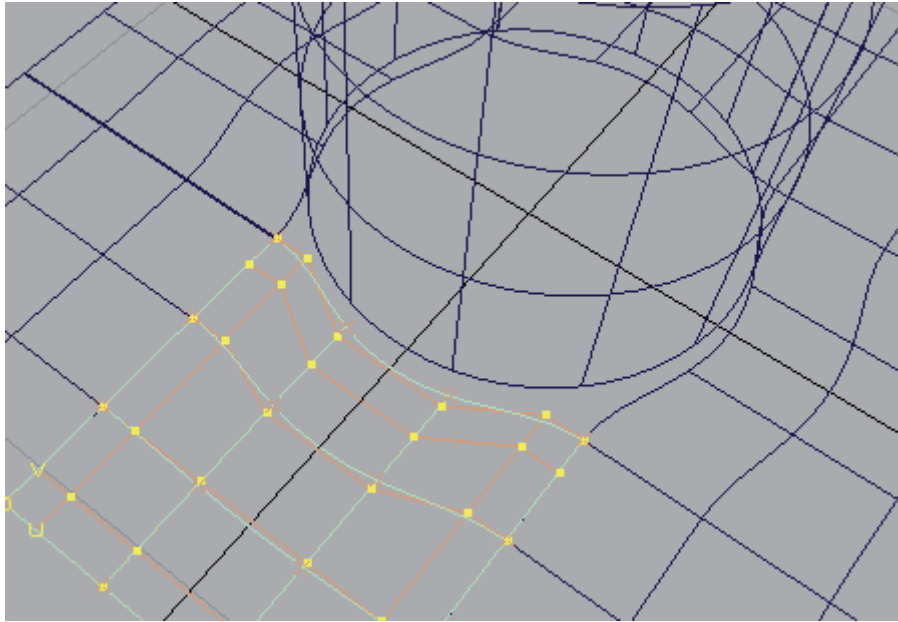
Patched models

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.



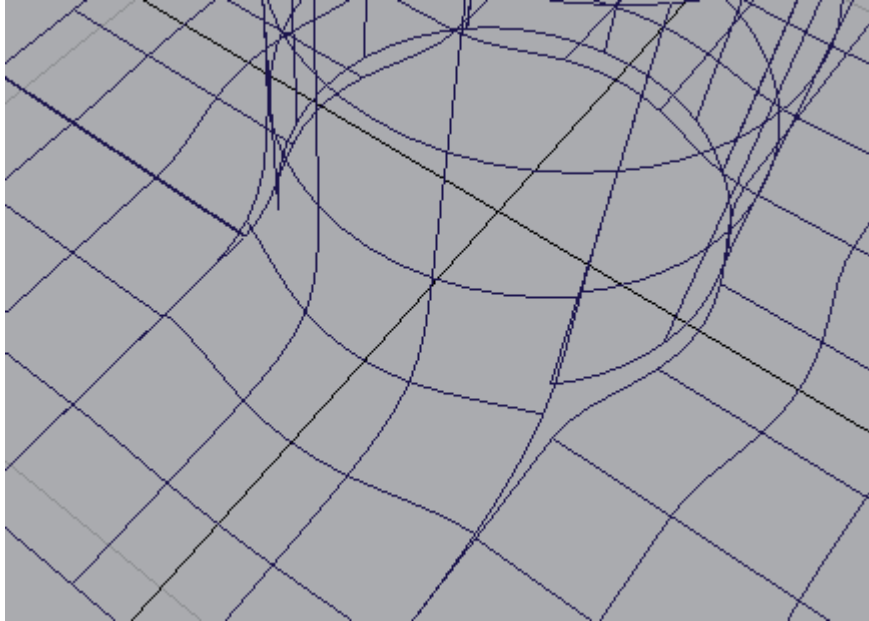
Corresponding patches

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:



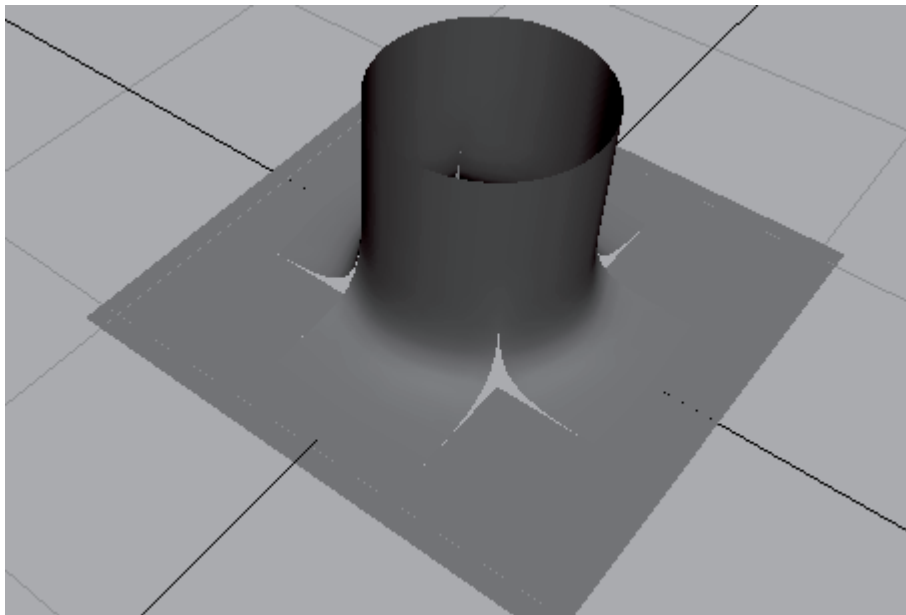
Proper CV adjustment

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:

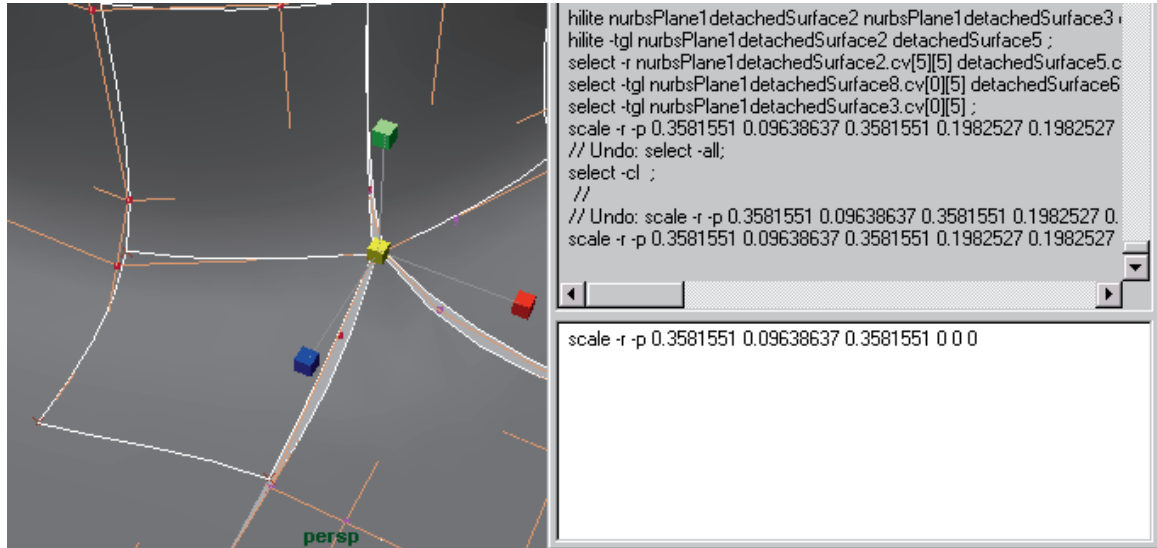


Four sides attached

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.

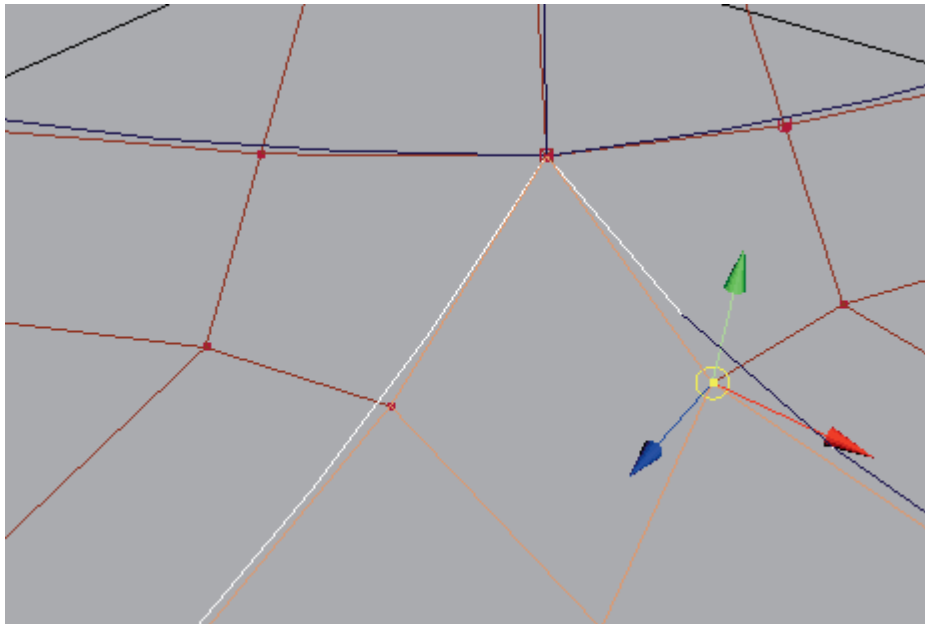
There are still "holes" or "gaps" in the four corners, and we can start closing those using "scale". Select the five CVs that are on the five corners of the five surfaces around the gap and scale them to zero. However, scale can

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.



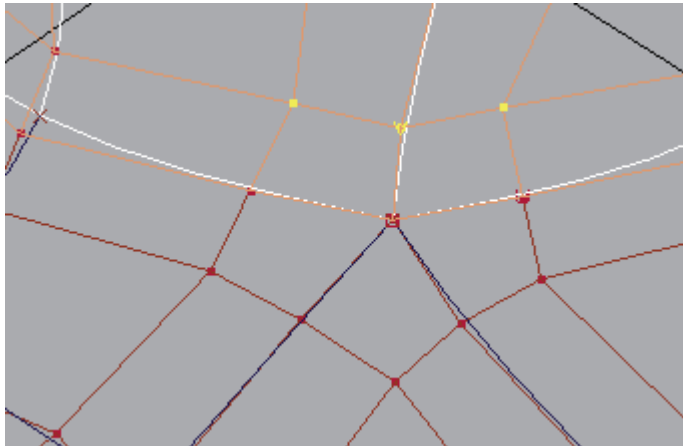
Scale to 0,0,0

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);

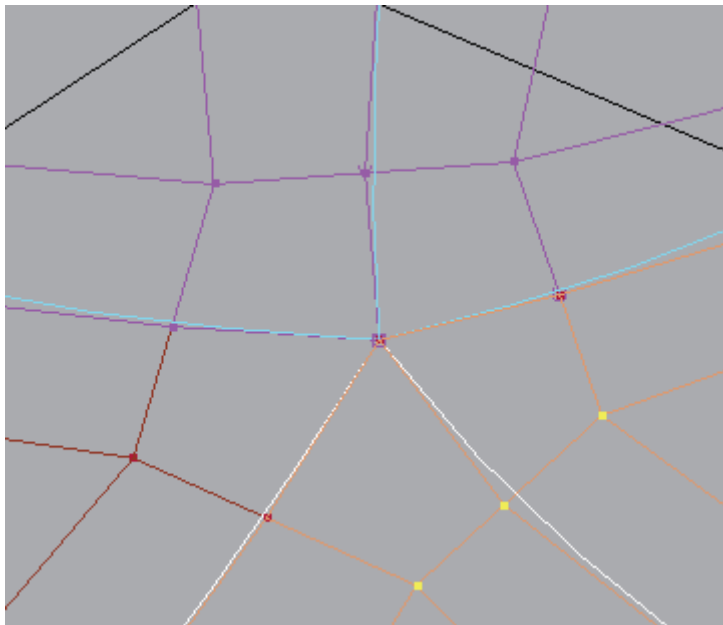
float $ratio;
$ratio = $distance1 / ($distance2 + $distance1) ;
print "Ratio is";
print $ratio;

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.