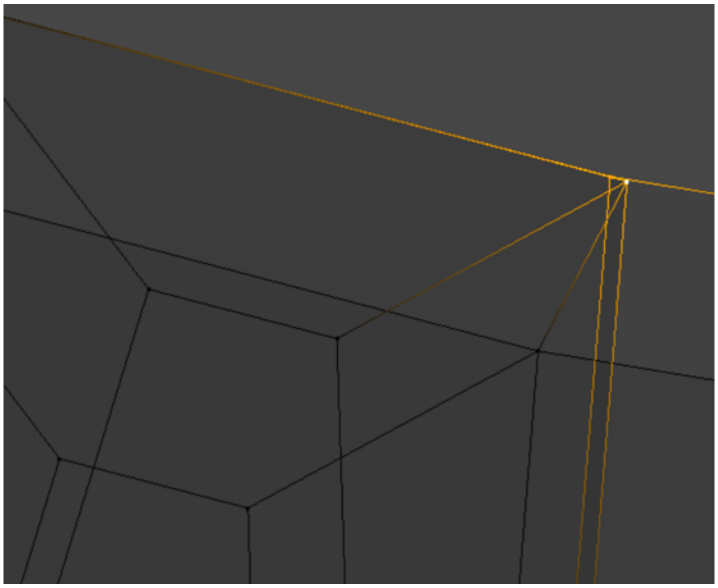
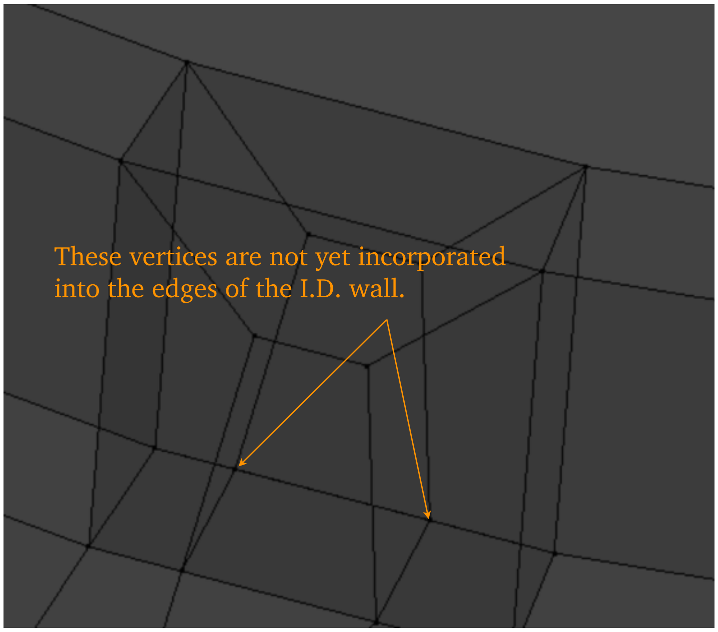
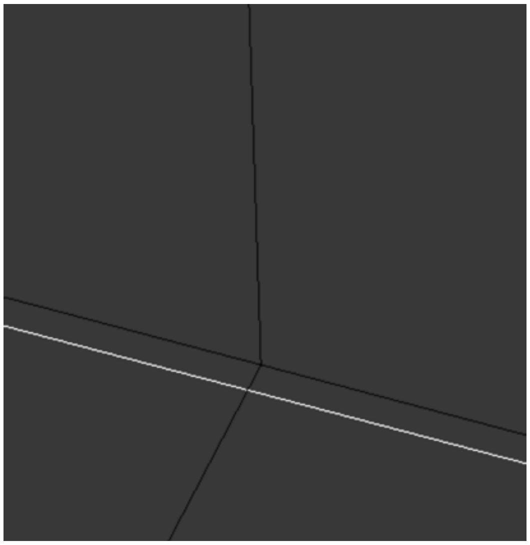
Step 5: Working at the I.D. wall, we select the original face corner vertex, then the projected corner vertex, then Alt + ‘m’, then ‘a’, to merge at the original face vertex. This is repeated for all four projected corner vertices, and brings the projected face width to the desired final width.



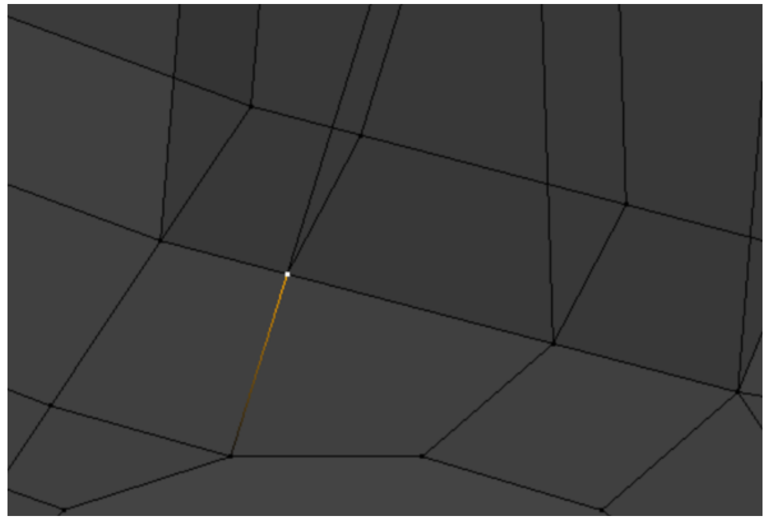
Step 6: The last step took care of the corner vertices, but there are two others which belong to edges exiting the working plane, and which must be integrated. Using the arrows in the picture at right, these can be referred to as ‘right-hand’ and ‘left-hand’.



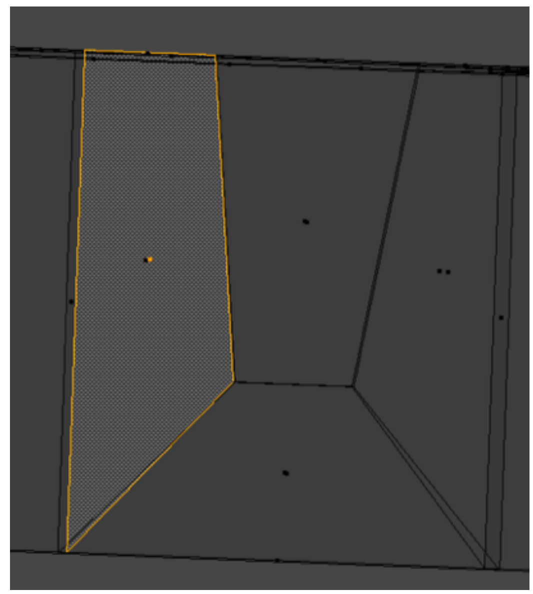
The highly zoomed view at right shows the right-hand (projected) vertex, from the previous step, with the target edge selected. We split the target edge, then drag the split point, with Shift + ‘v’, to the location of closest approach to the projected vertex. Then, the dragged vertex remaining selected, we select the projected vertex, then press Alt + ‘m’, then ‘a’, to incorporate the projected vertex. This procedure is repeated for the left-hand vertex.



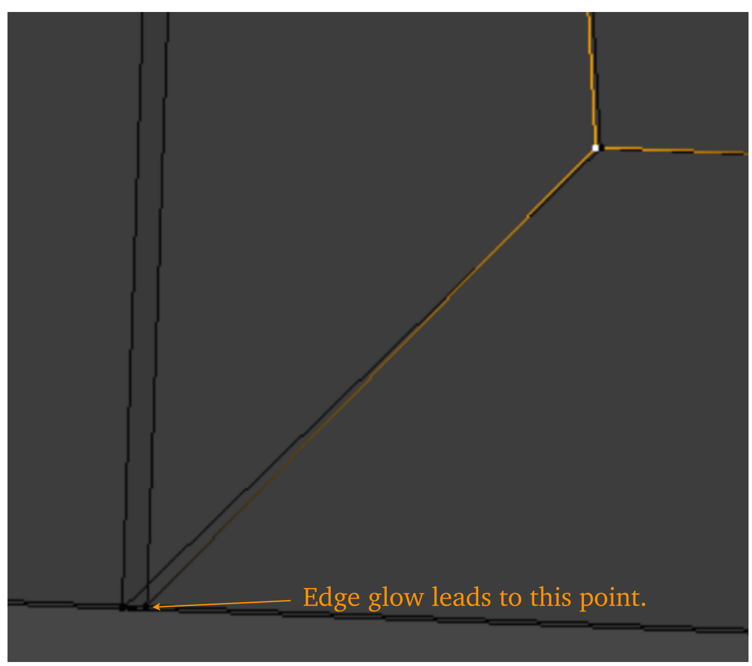
Step 7: Next we go into point select mode and check all the relevant vertices one by one to see whether they are fully integrated. The one shown right is not, because edges meeting at its location, which should be glowing in five directions, are only glowing in one. For each such vertex, the vertex remaining selected, we press ‘c’ (circle), and select a small circle of space surrounding the vertex. This captures any vertices which may exist within a cylinder defined by the selection circle. Then we press Alt + ‘m’, then ‘a’, to incorporate all into one single vertex.

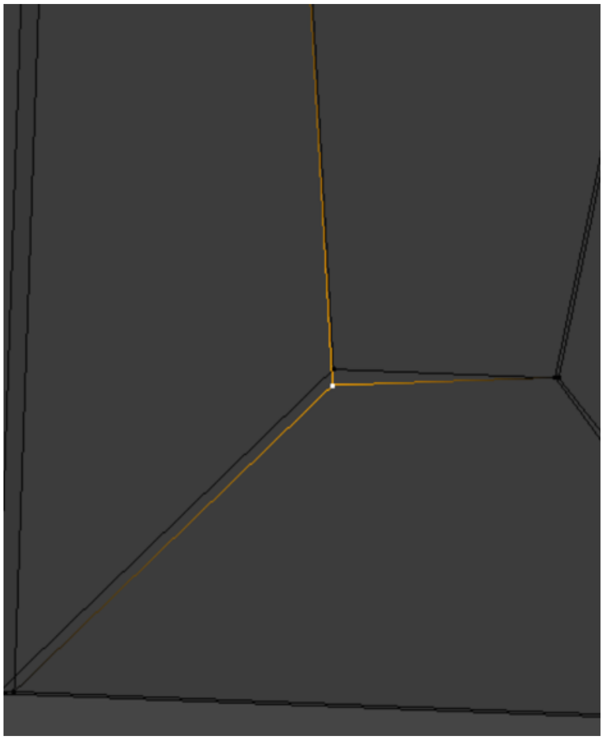


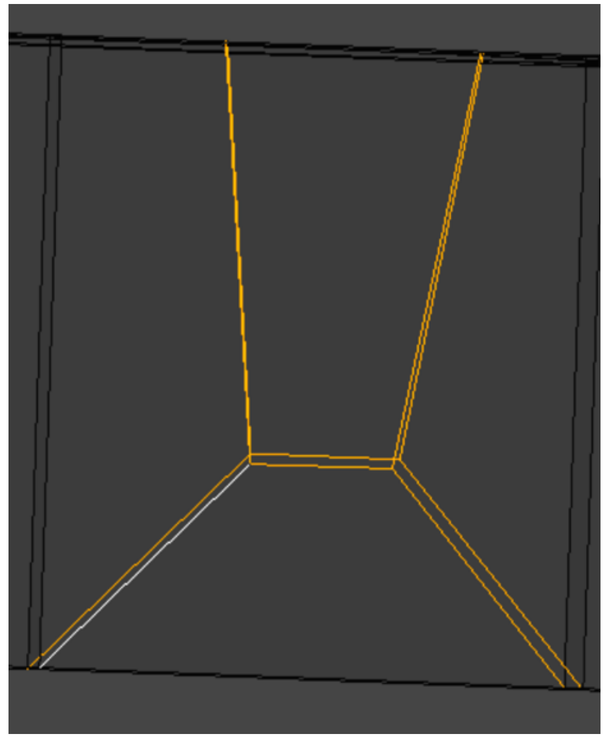
Step 8: Next we select one face on the I.D. wall and press Shift + ‘7’ to get a flat working perspective. In the view shown right, executing this command happens to flip the view.



Step 9: We go into points select mode and select one of the interior vertices. We look at the edge glow and see that it leads to the inside point at the bottom left of the view shown right. We know that inside points belong to the I.D. wall, since it is narrower, and thus we know that we have selected the vertex we want.



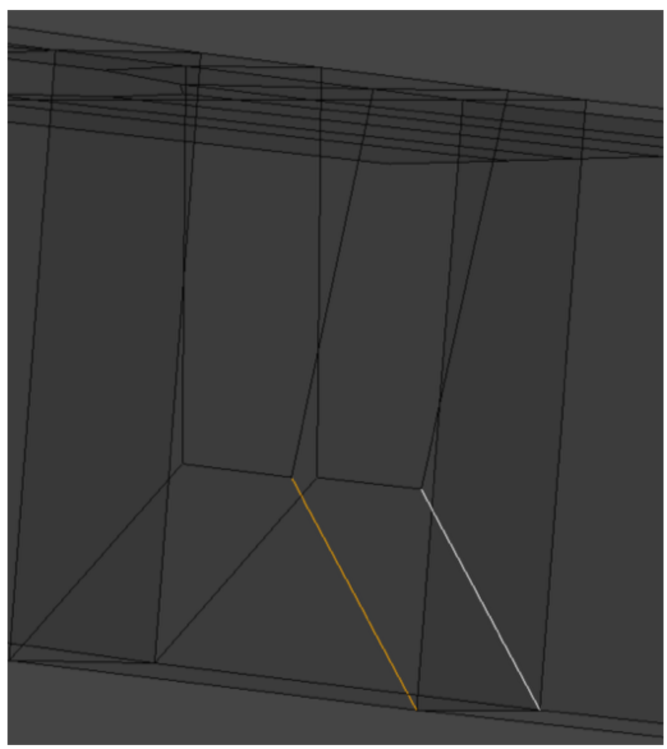




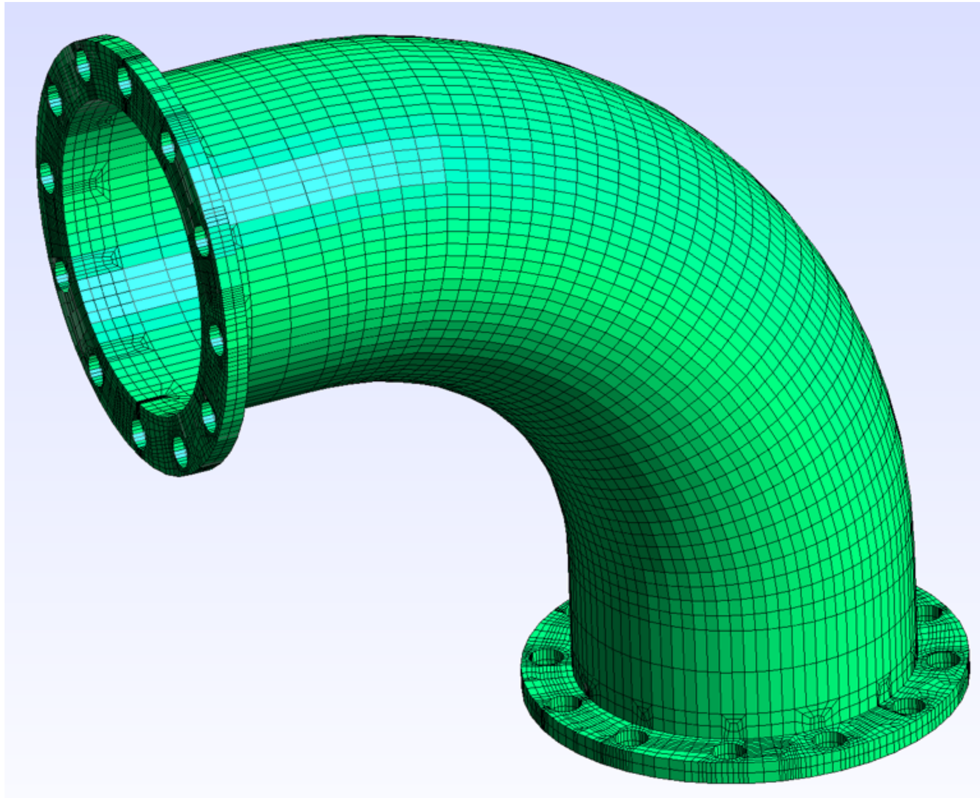
Step 10: We press ‘g’ (grab) and move the vertex freely until the four edges which depend on the vertex’s location are parallel pairs.

We repeat the parallelizing procedure with the other internal vertex, making sure again that we grab the one on the I.D. wall, the plane that we are working perpendicular to.

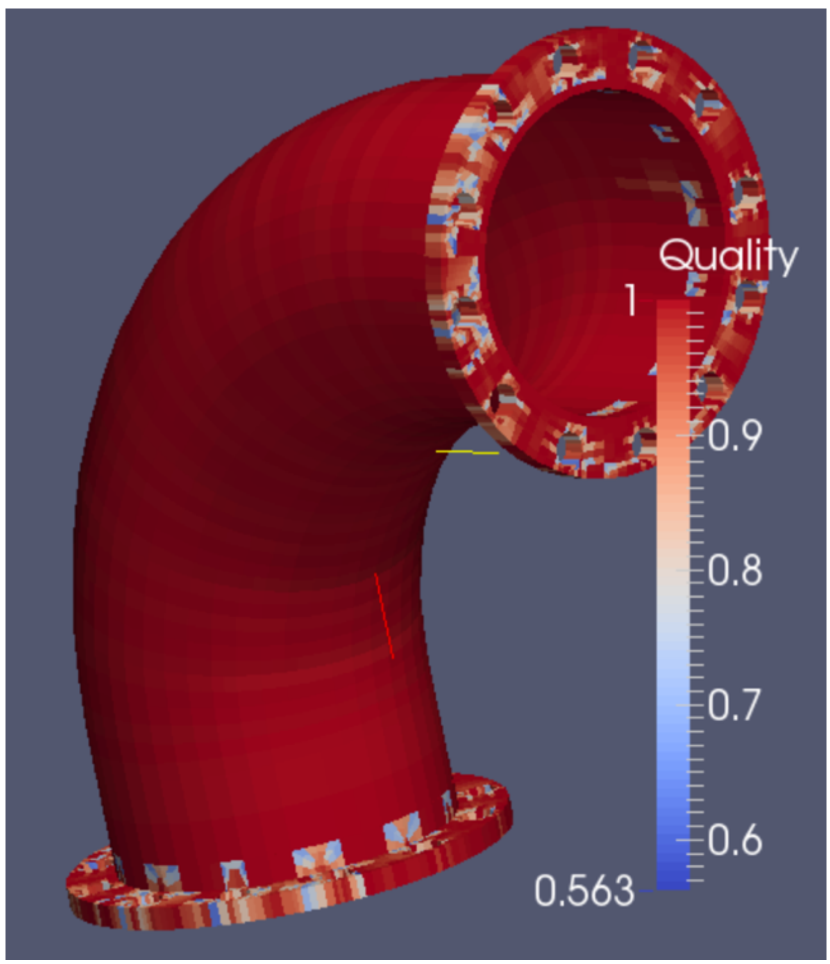
Step 11: The steps above have produced good geometry, but a number of edges do not constitute faces. We can tell in wireframe view by the charcoal shading which faces are missing. We switch to edge select mode and fill in the missing faces. This finishes the wall translation procedure.



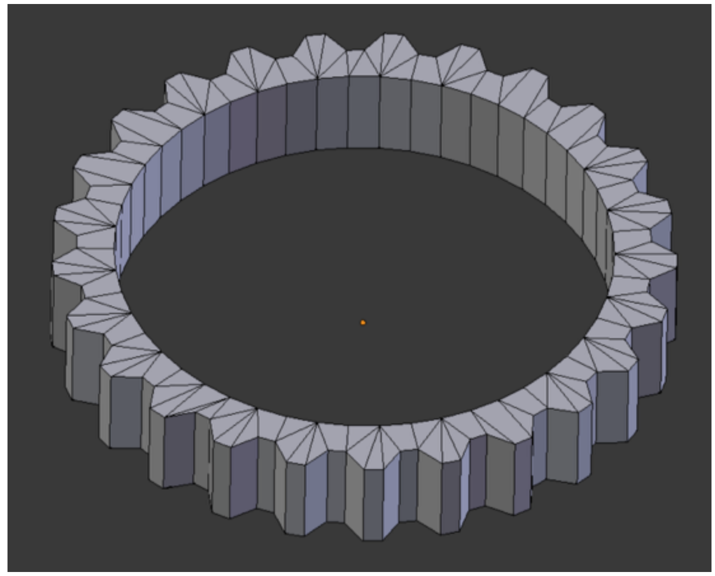
We test one flange subassembly for quality and make a few necessary corrections. Then we use the refined flange subassembly for both ends of the Ell. We connect them with the curved section, then export from Blender, convert to .vtk using Blenbridge, and run the assembly through Gmsh.



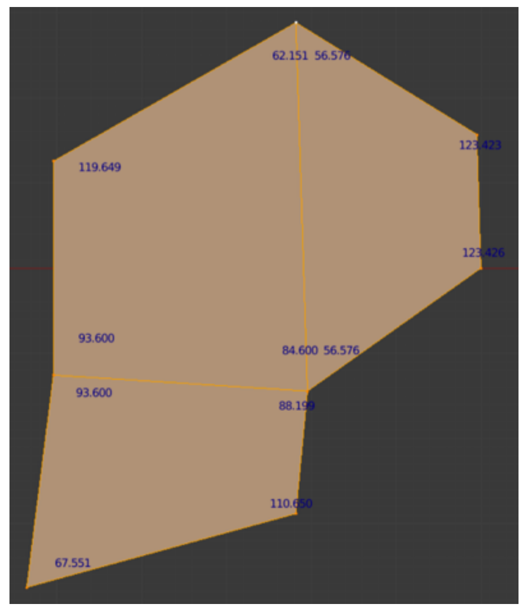
As the Paraview quality filter shows, the final mesh has quality to spare under Scaled Jacobian requirements. The mesh has 26,009 vertices and 16,080 elements.



Demo 12

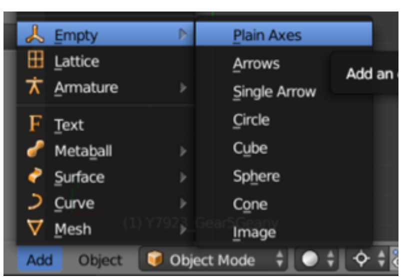


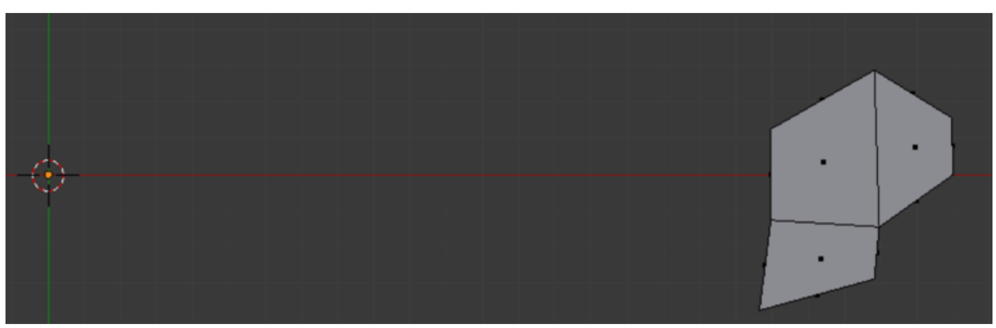
This object is a ring gear; the model can be found in the INRIA model repository as an .obj file, and is imported into Blender with Blender’s .obj import script. Its title is “Y7923” (See References). The Limited Dissolve command is ineffective on such a simple model. An object like this is often divided when stresses are analyzed; however, in this case we will assume that we want to look at the whole object at once.



Fifty equally sized inside edges make a circle with diameter of 400. Everything is deleted except a generating plate. Member quads are created which contain the best-sized angles, which happens to make a very simple design. It includes two inside edges, so we will need to call for 25 array instances.

We will use a different array modifier technique than we used in Demo 11.

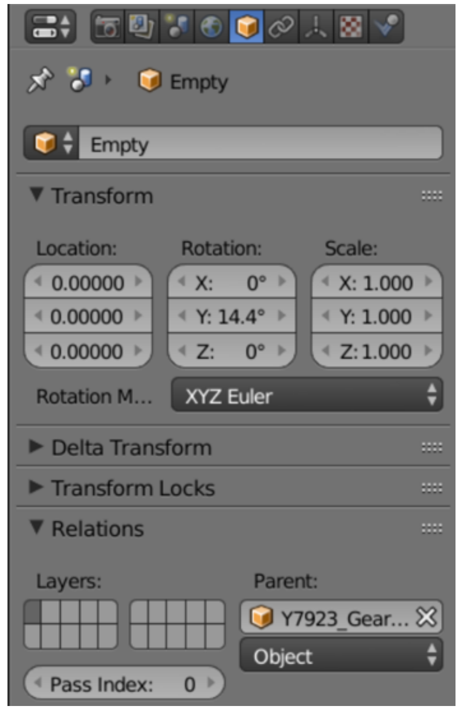




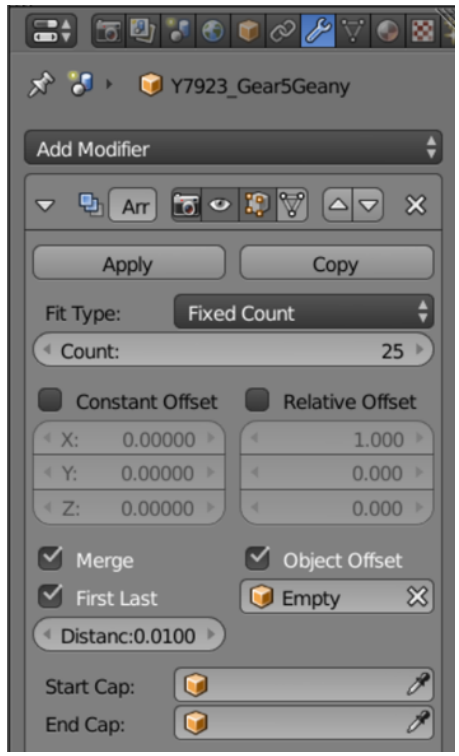
Top view

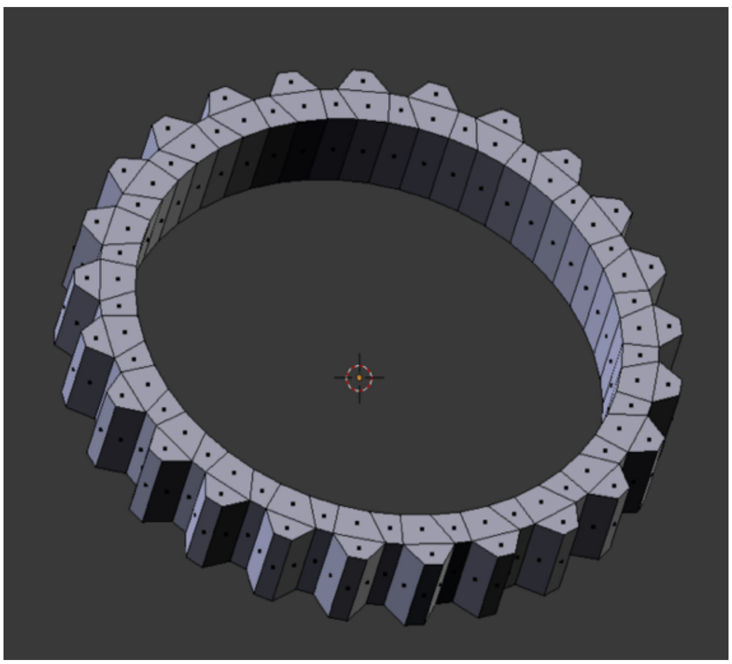
We extrude the plate to get a generating chunk. The object origin has remained at the world center, where the 3D cursor is also located. We add an empty there as well, in Object mode.

The Object panel for the empty is shown right. It is necessary to parent the empty to the chunk before assigning the array modifier to the chunk. The empty is pre-rotated according to the needs of the generating body (the chunk). Since the array rotation is desired around the z-axis, it would seem logical to rotate the empty around this axis. However, we rotate the empty around the axis which produces the desired orientation of generation, determined by experiment, and in this case it turns out to be the y-axis. (The sign on the rotation of the empty does not seem to matter.)



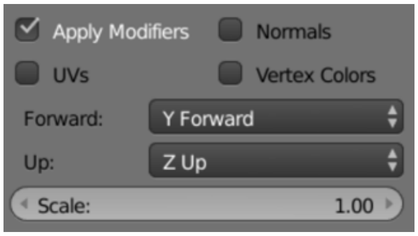
The Modifier panel for the chunk is shown right. The offset method that is chosen is Object Offset, as we see. The Merge and First Last checkboxes are intended to provide continuity and avoid doubles. However, at the end of the operation the Remove Doubles command still removes eight doubles.



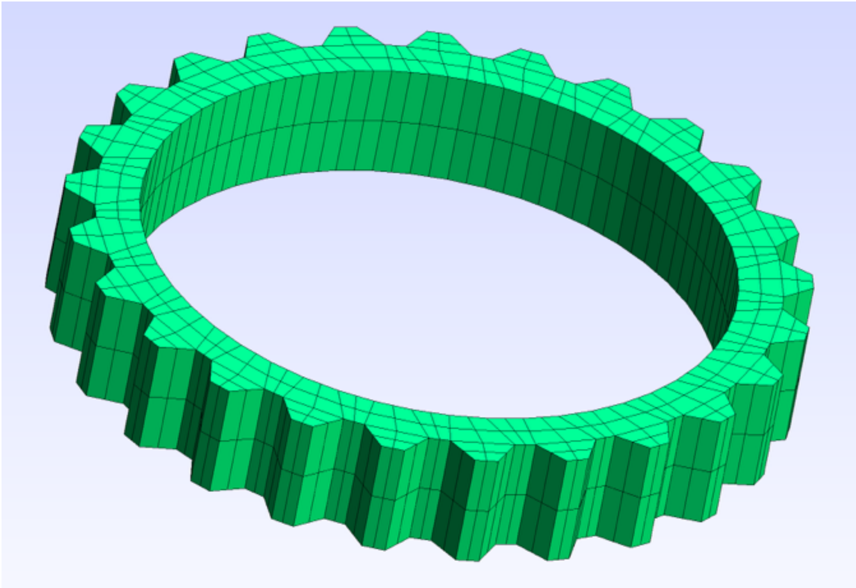


The generated ring after the Apply button has been pressed is shown left in Edit mode. However, it is not strictly necessary to press the Apply button.

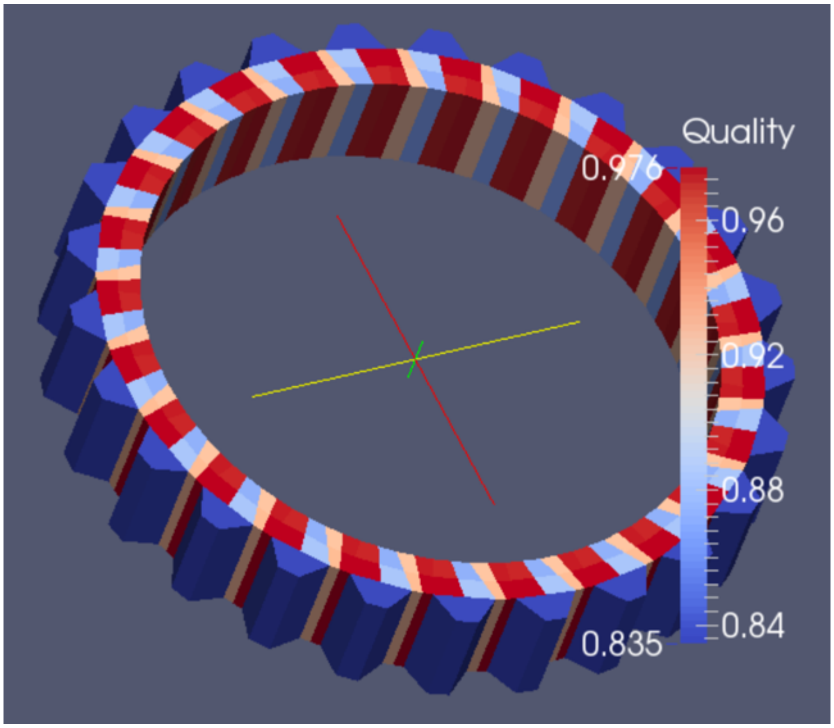
We go on to export the mesh. If for some reason we did not wish to officially Apply the array modifier, we could still export all the generated geometry by selecting the Apply Modifiers checkbox in the export panel. On the other hand, if the Apply button has been pressed, there is no need to mark the checkbox.



The mesh is converted to .vtk format by Blenbridge, then imported into Gmsh. The usual mesh refinement is performed in Gmsh. The mesh as it emerges from Gmsh is shown right. It contains 1350 nodes and 600 elements.

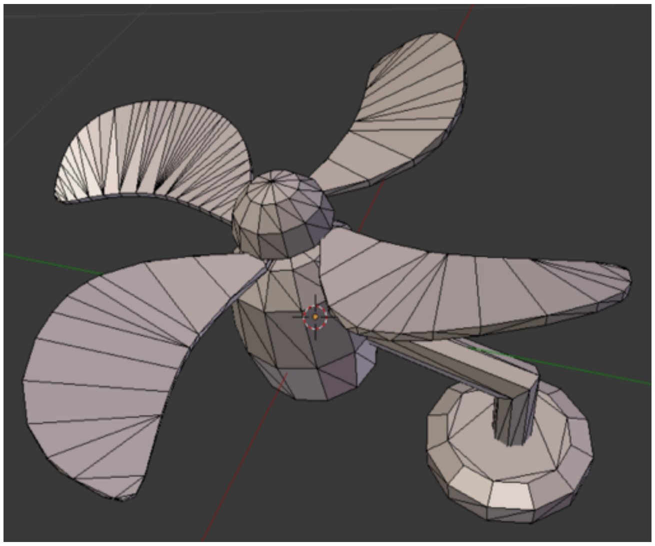


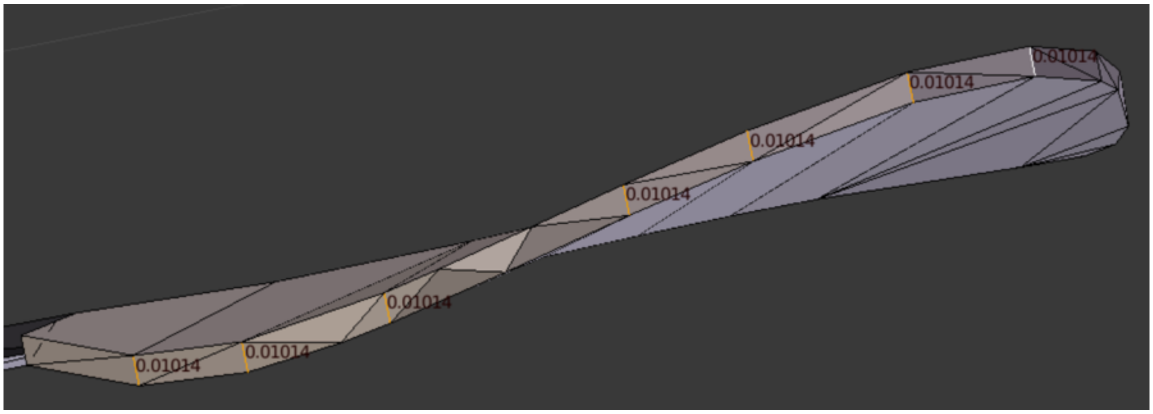
The mesh shows a high degree of quality according to Scaled Jacobian criteria.



Demo 13

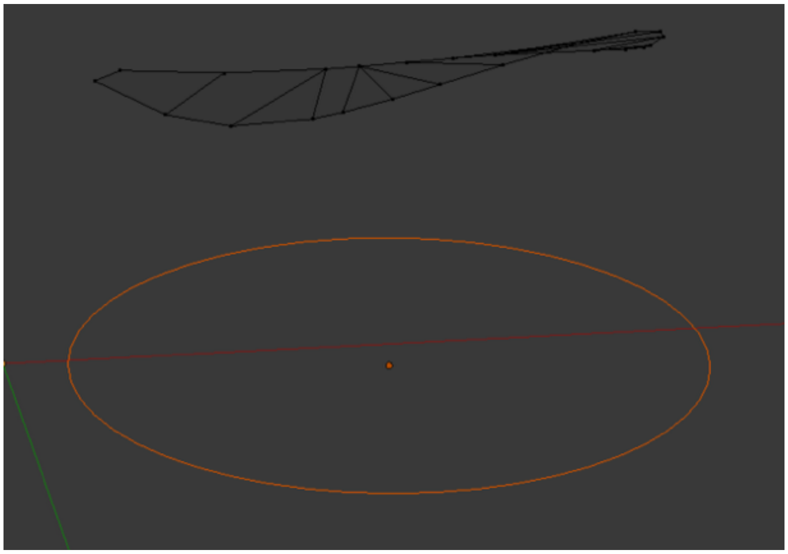
This object is a fan; the model can be found in the INRIA model repository as a .3ds file, and is imported into Blender with Blender's .3ds import script. Its title is "fan_1" (See References).



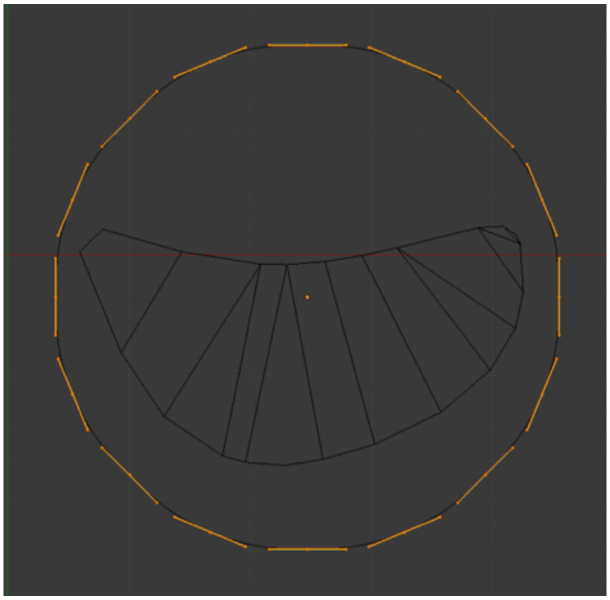


For us, the most interesting things about the fan are its blades. The blade thickness is constant, but the edges are not aligned to a major axis, they run perpendicular to the top and bottom surfaces of the blades.

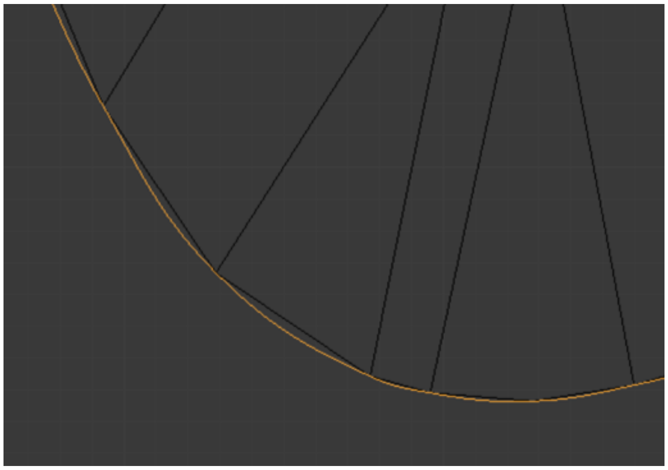
We will suppose that more detail is desired in the shape of the blade edge outlines, and we will undertake an interpolation process to provide it. We start by adding a bezier circle to the scene, and hide all but one fan blade, eliminating its bottom surface.



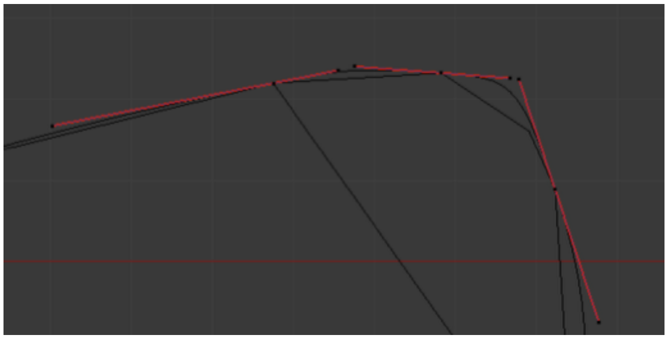
Curves and mesh mix like oil and water, whether they are joined or not. The top surface of the blade has 23 vertices. We use the 'w' (special) menu to subdivide the bezier circle into 16 parts. We can subdivide between any two control points to cover the last 7 vertices. We will snap each control point to a vertex. The procedure is as follows, with EM designating Edit Mode and OM designating Object Mode. (We assume that the 3D cursor has been selected as pivot point.)



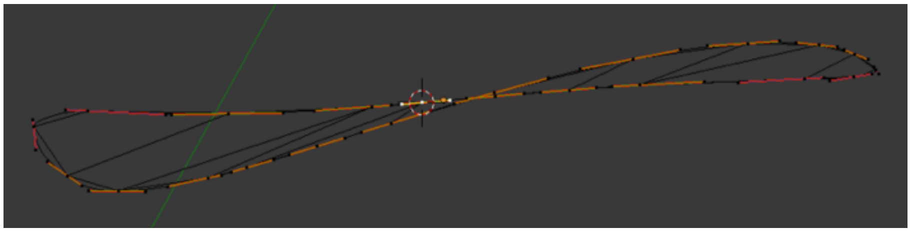
1. OM. Select the mesh.
2. EM. Select a vertex. Use Shift + 's', then 'u' to snap the 3D-cursor to the selected vertex.
3. OM. Select the curve.
4. EM. Select a control point. Use Shift + 't' to snap the control point to the 3D-cursor (and to the desired vertex).
5. OM. Select the mesh again.



With the above procedure all vertices in the surface can be supplied with control points of the bezier circle. The view right is a top view snapshot as the final control point is assigned. We see that some smoothing of the edge outline will result from using the curve to rebuild the blade.

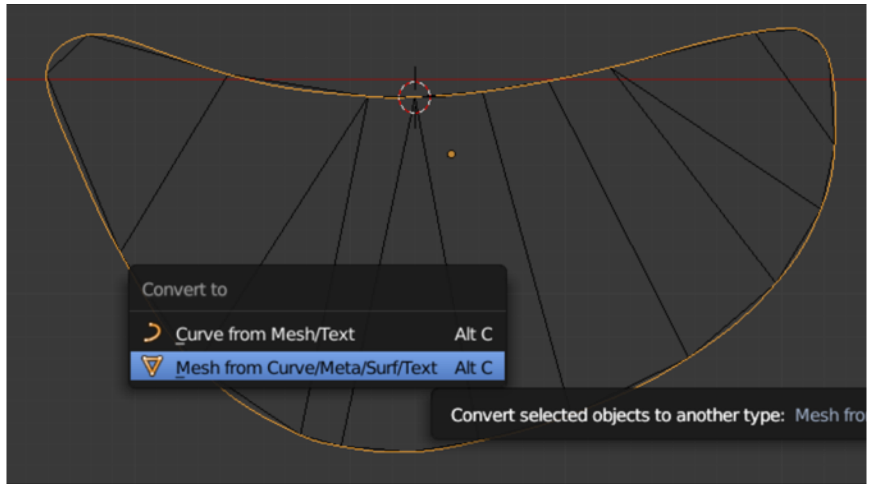


The assumed starting alignment is that defined in the Object Mode 'v' menu, with the Automatic setting. Some adjustment of control handles improves the form of the curve. (This is still a top view.)

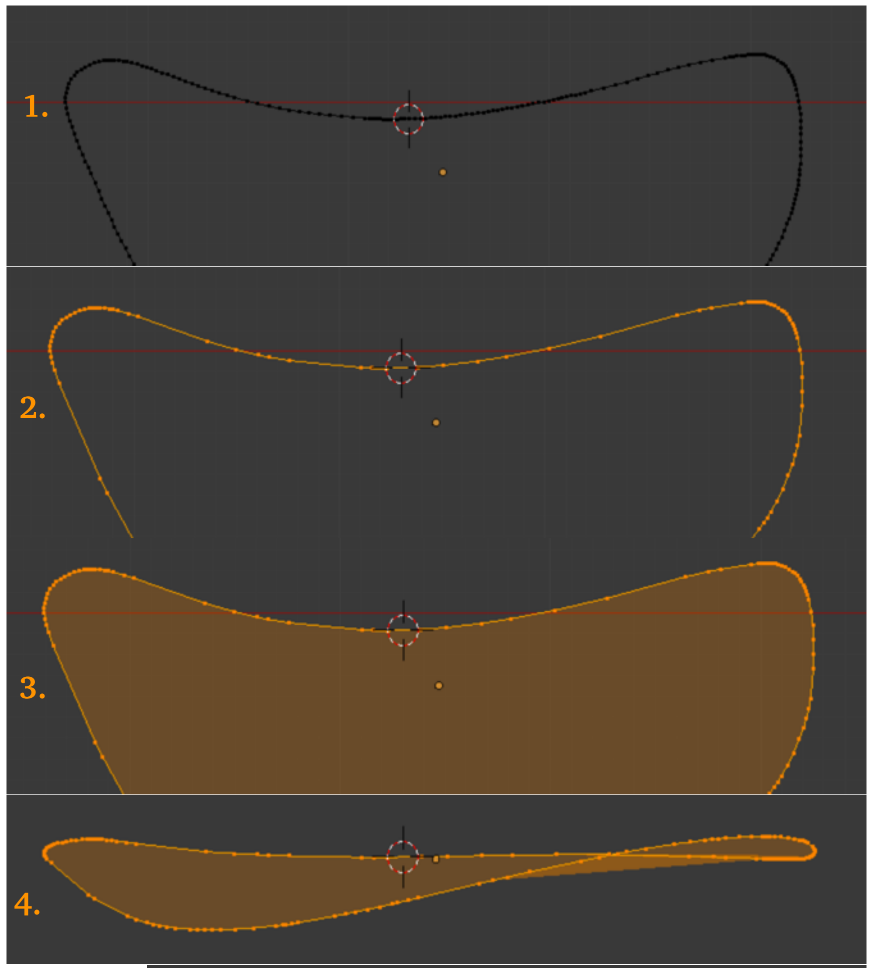


From some arbitrary perspective it can be see that the Automatic control point setting generally follows the path of vertices well.

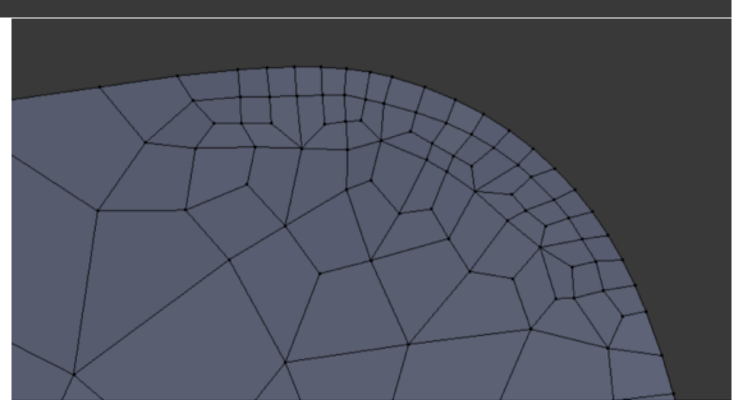
At this point the mesh surface can be deleted. In Object Mode we convert the curve to a mesh, using the Alt + ‘c’ menu.



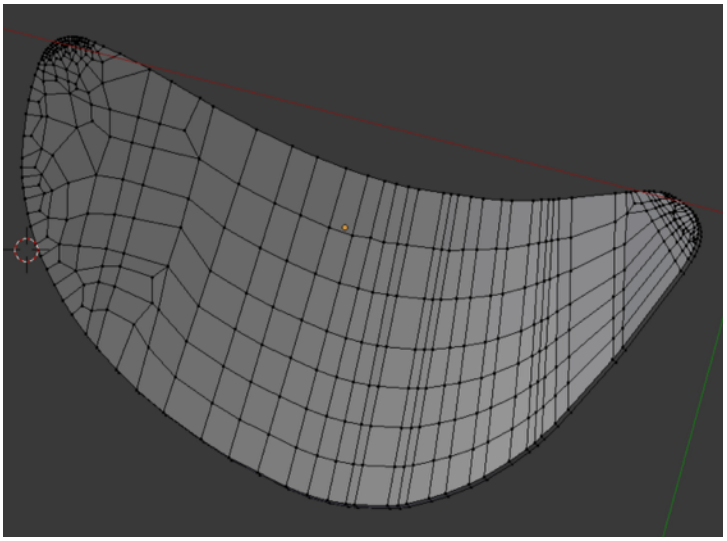
Conditions contained in the pictures: 1. On entering Edit Mode we find many superfluous vertices have been created. The Limited Dissolve command reduces these to a manageable number. 2. We reduce the Max Angle field from 5 degrees to 2 degrees to preserve a few more vertices than retained by the default 5 degrees. 3. Then simply pressing ‘f’ gives a big common n-gon. 4. The n-gon retains the warp of the original surface.



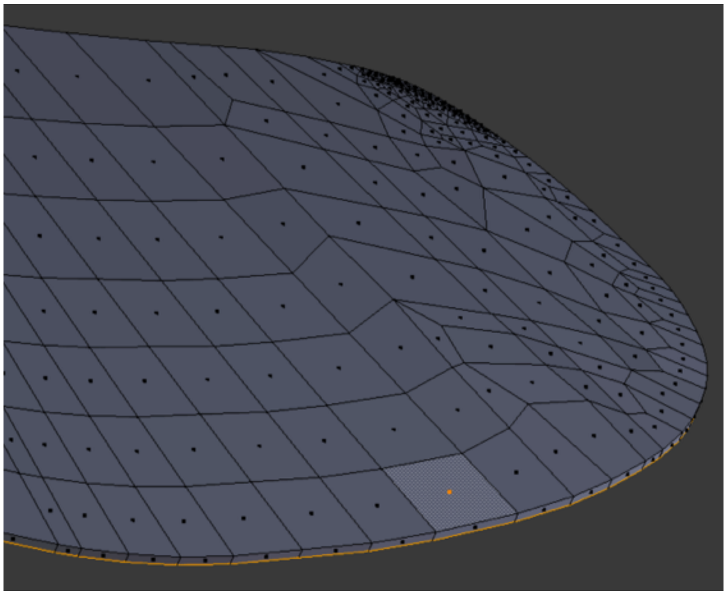
Using the knife tool, we create a pattern of faces. Because of the rapid change of surface contour, all changes in vertex positions are accomplished by sliding, not by grabbing.



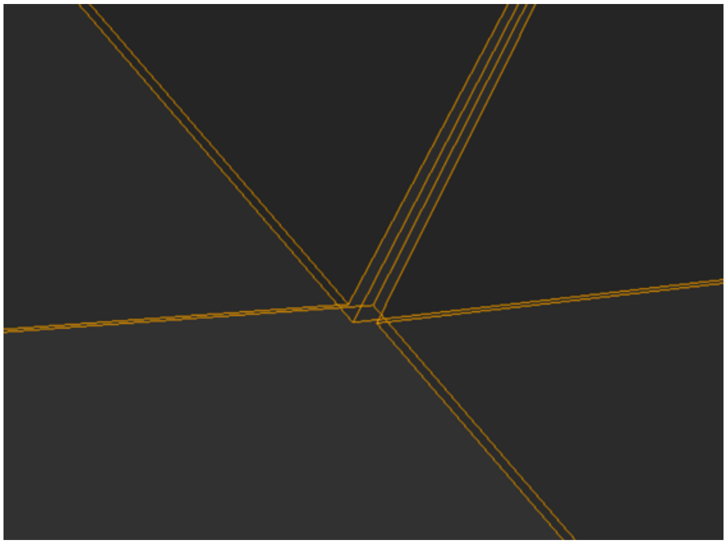
When it is time to test the quality of the elements we created, we extrude along the z-axis. This is not technically correct, but for quality testing it will be acceptable.



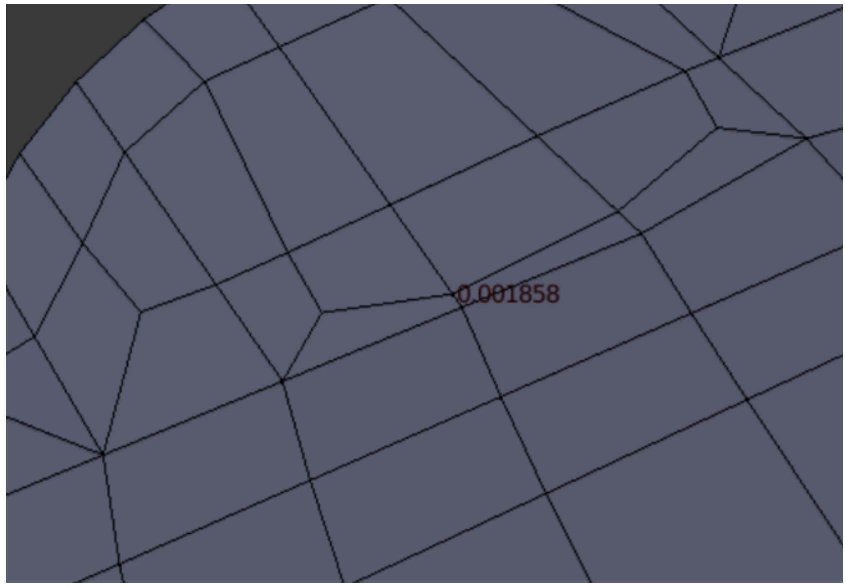
After the mesh meets minimum quality requirements for Scaled Jacobian, we delete all but the primary surface. That surface is duplicated (so that a bottom face will be left behind when the duplicated face moves in its intended extrusion path). Then the Extrude Individual command is pressed in the Toolbox panel. That command has the advantage of extruding in a normal direction, the main thing we need.



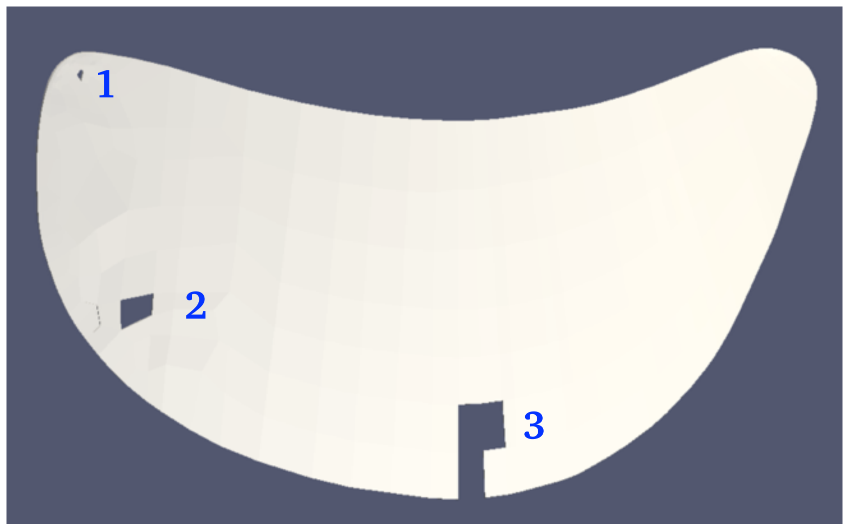
The Extrude Individual command has another nice quality: it extrudes edges as well as faces (unlike the Extrude Region command). This saves a great deal of time. However, there is one drawback to the command, which is that it does not keep vertices integrated with those in neighboring edges. The highly zoomed view right shows the result of this shortcoming. Here corners of elements overlap. Elsewhere there may be gaps.



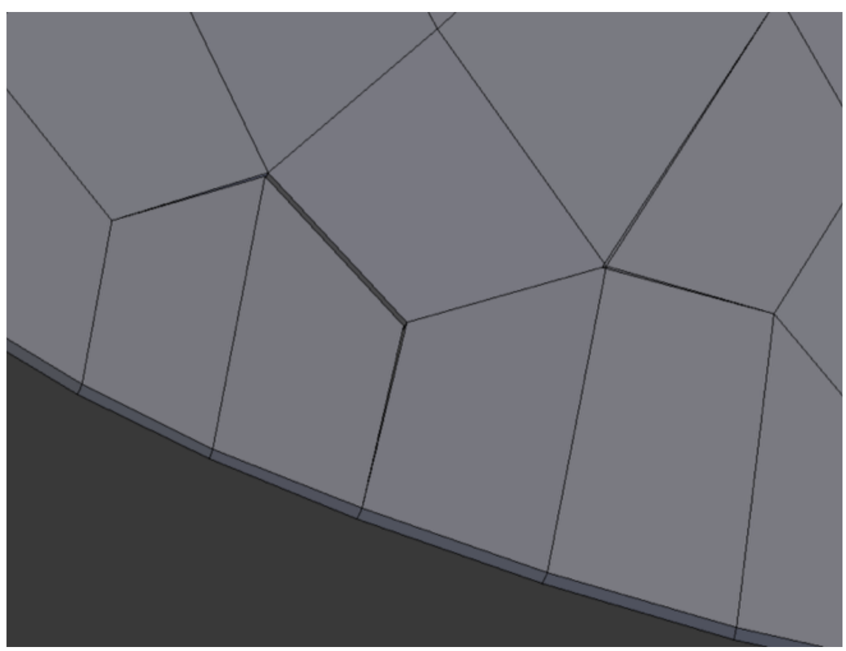
We solve the vertex dislocation problem by inspecting the mesh for the shortest legitimate edge, then setting the Remove Doubles dialog for a distance shorter than this. Then we select all and Remove Doubles. We believe that the Remove Doubles command chooses a central location at which to fix the resultant vertex, which is exactly what we would wish.



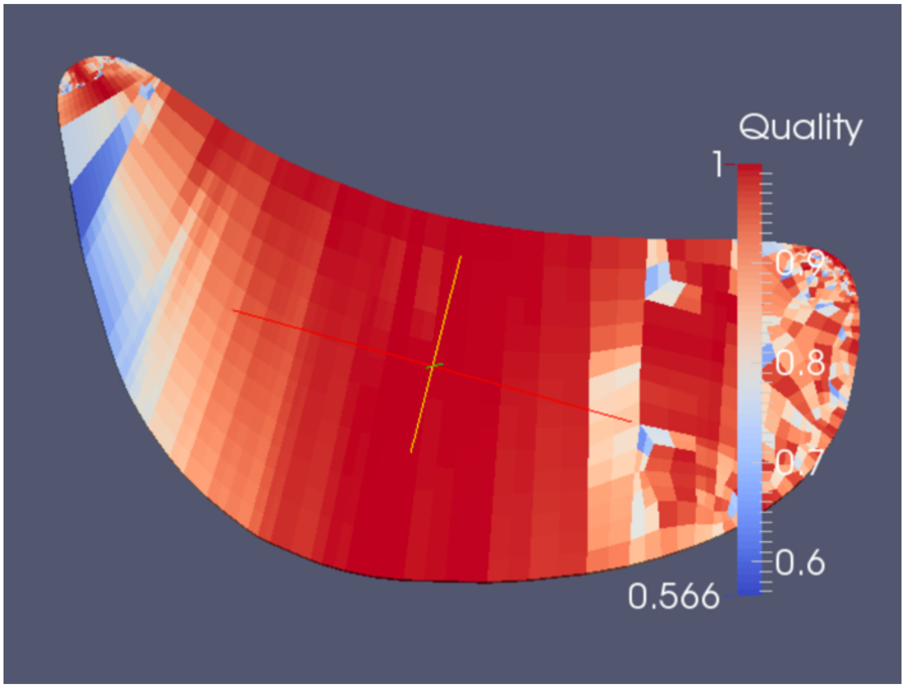
After Blenbridge converts the .ply file to .vtk format, we open it in Paraview and find some deficits. As for element areas 1 and 3 right, we can find no fault with the associated faces, edges, or vertices, but remaking the top and bottom faces in Blender fixes the problem.

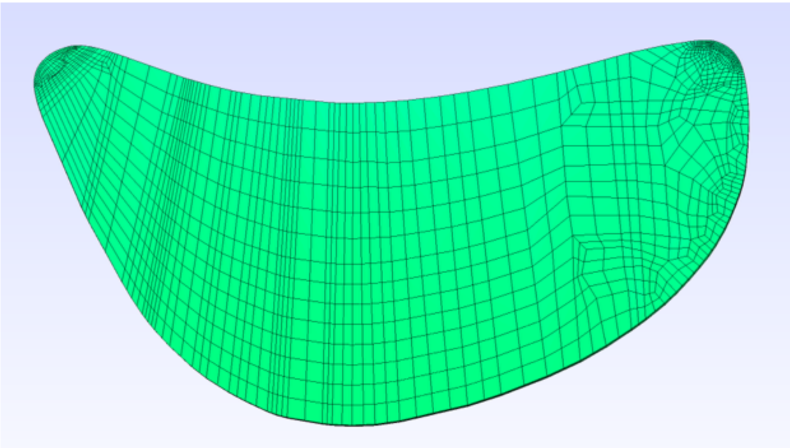


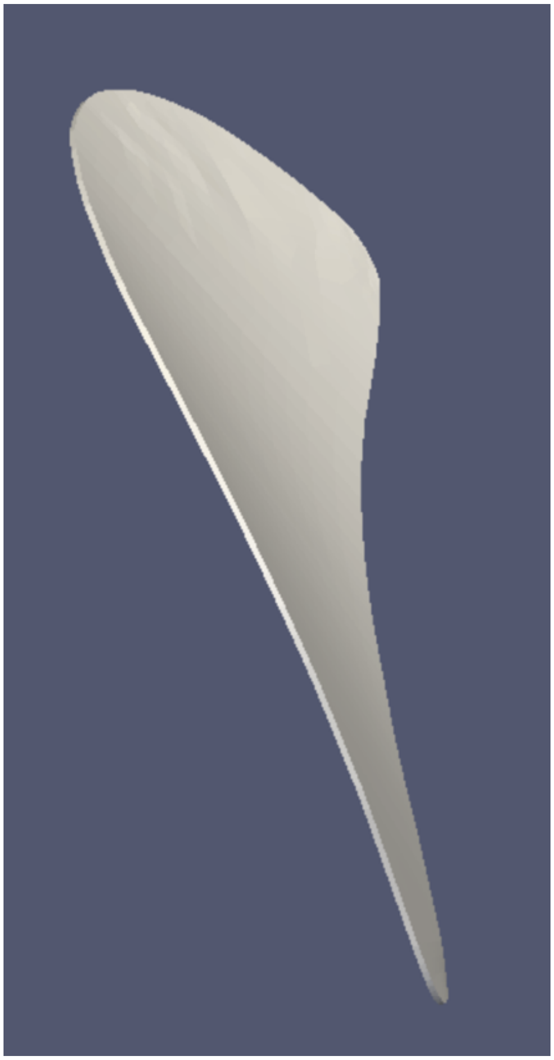
The view right shows Blender's view of the element area number 2. Merging the appropriate vertices (in 3 locations) fixes this problem. The repaired mesh is exported again, then converted by Blenbridge. This time no problems are seen in Paraview.



The final quality picture in Paraview shows acceptable quality for all 3168 elements.



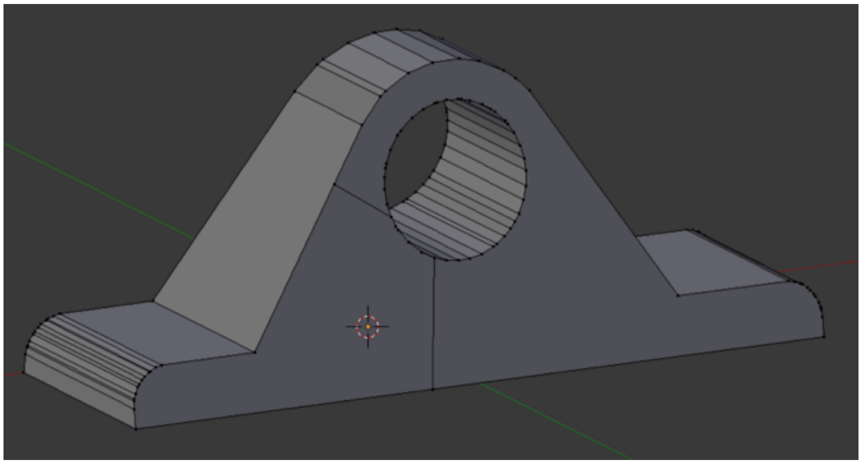




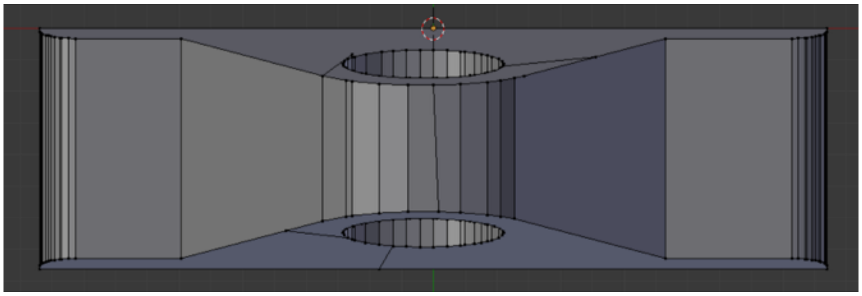
The view at right shows there is some slight surface unsmoothness, in spite of our efforts to preserve the contour. We feel that the exercise achieved at least partial success, in that we were able to create a much refined blade compared with the one we started with.

Demo 14

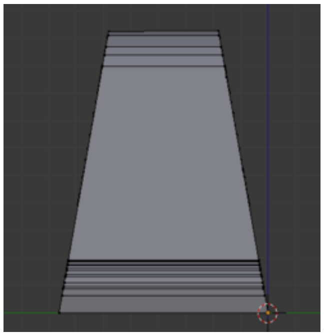
This object is a bracket; the model can be found in the INRIA model repository as a .mesh file, and is converted by Gmsh into a .stl file. Blender imports the .stl file using its .stl import script. Its title is "bracket1" (See References). The ordinary Tris-to-Quads and Limited Dissolve commands are used to prepare the mesh for editing.



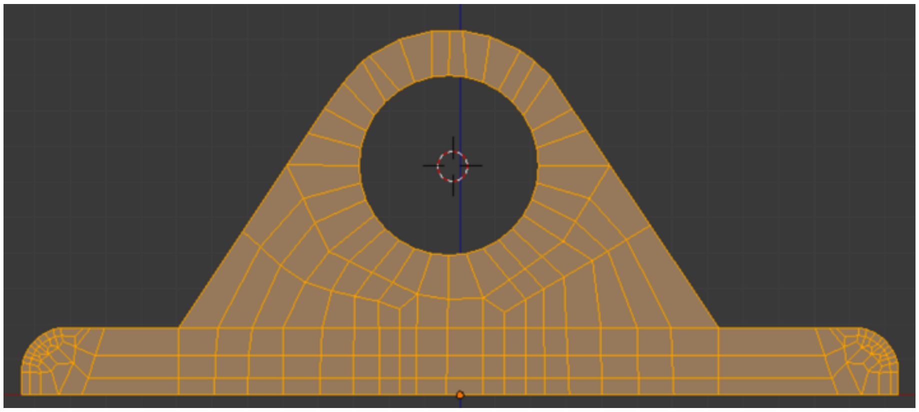
The main interest in this object is its slanted sides, and how to combine them into the geometrical landscape.



The idea of a boolean cut naturally arises. However, we remember that booleans are allergic to non-manifold meshes. The next idea to occur to us is Knife Project.



We note that the left and right flange tabs are not of equal length, a fact which will affect mirroring.

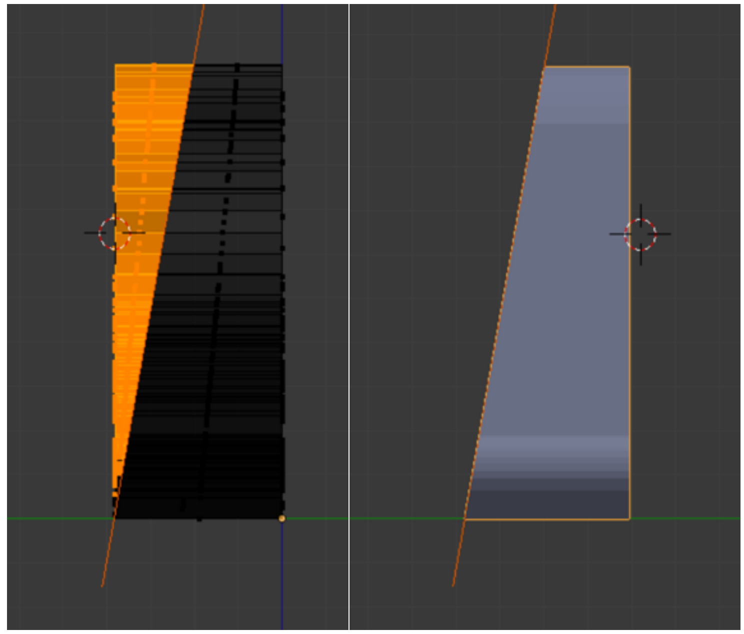


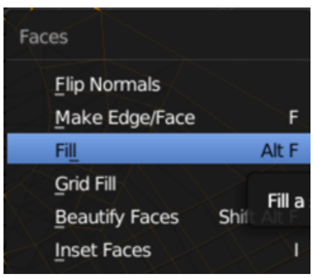
The vertices of the original thru hole are not evenly spaced, and the hole is replaced with a 32-division hole.

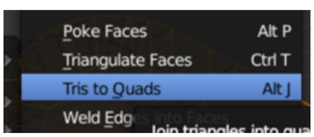
We create a vertical plate from a lengthwise bisection, then equip it with a typical face design and extrude on the y-axis.

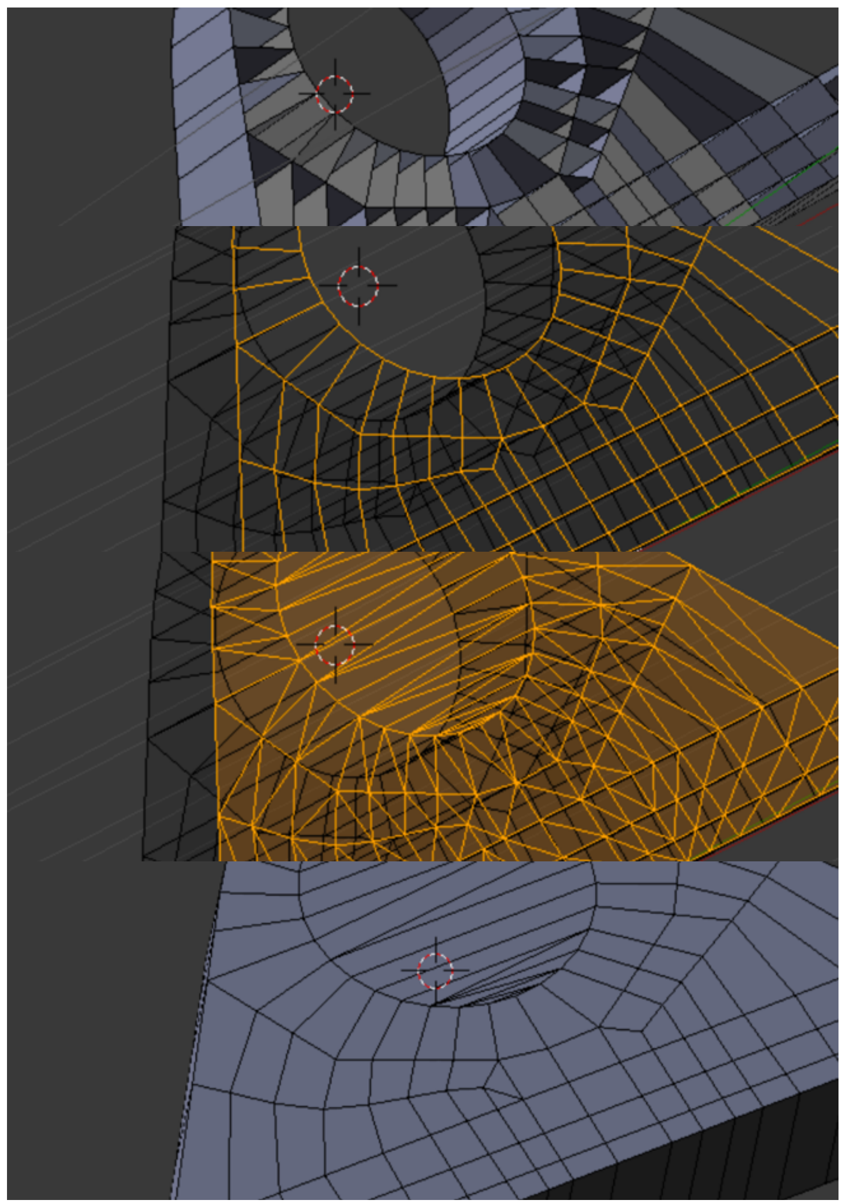
An edge is copied, converted to a separate object, (using ‘p’), scaled, and positioned, then used as a cutter. After pressing Knife Project, we check the ‘Cut through’ checkbox, to cleanly slice all the way through the mesh. Mirroring will save us the trouble of doing this twice.

The slant is 10.000 degrees off vertical. Circle select, ‘c’, selects the cut-off faces for deletion.



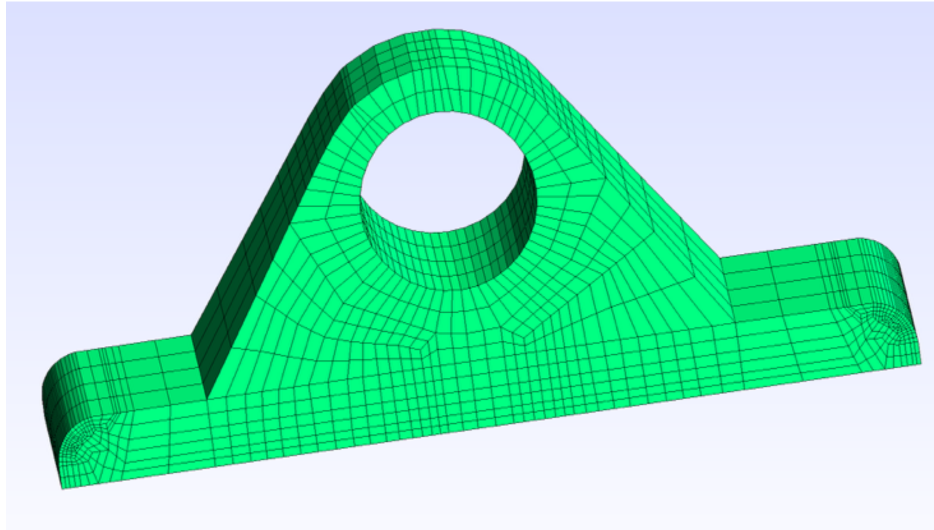




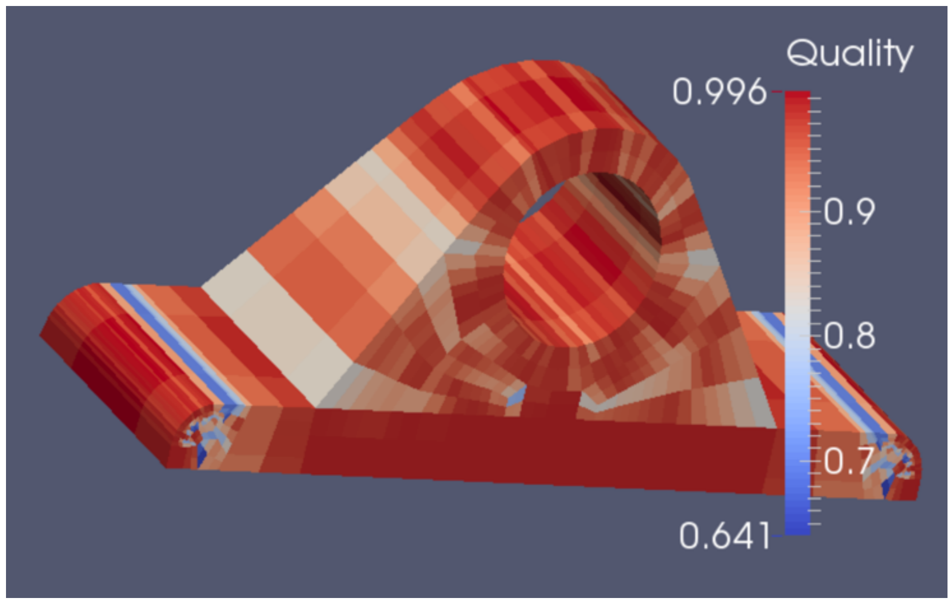


After cutting we have the task of filling in the cut-away faces. There are shortcuts to this process. After selecting the open edges in end view with a border select, ‘b’, we press Alt + ‘f’, and all the faces are filled. We follow this up with the Tris-to-Quads command, Alt + ‘j’, which leaves us with the whole area filled and a little extra. Deleting the unneeded faces and converting a few others is much faster than doing the fill-in by hand. The appearance is also better, since the automated process builds all faces with normals facing in the same direction.

The final mirrored bracket mesh emerges from Gmsh with 3072 elements and 4380 nodes.

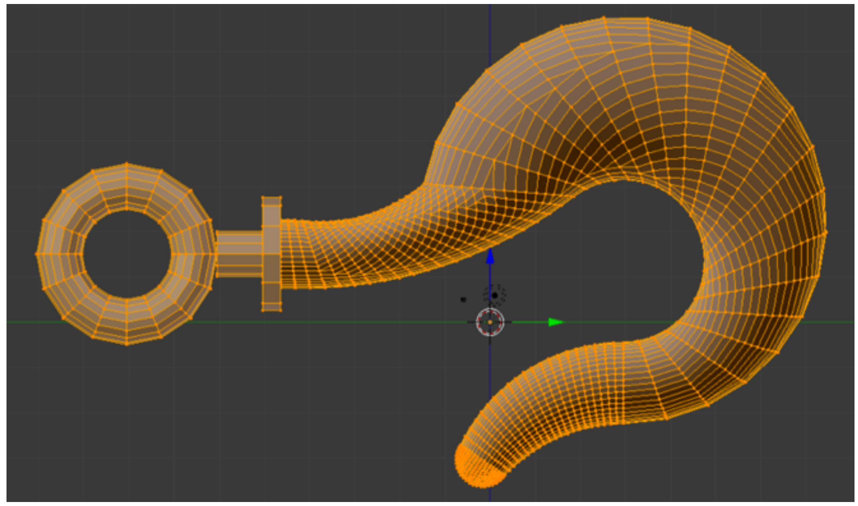


The final quality is acceptable, according to Scaled Jacobian standards. We knew that the slanted surface would cause a reduction of quality compared with the vertical test case, but the decrease in minimum quality level was only from 0.658 to 0.641, a negligible amount. If the slant had been greater, it might have been necessary for us to groom the angles, and possibly reverse extrude.

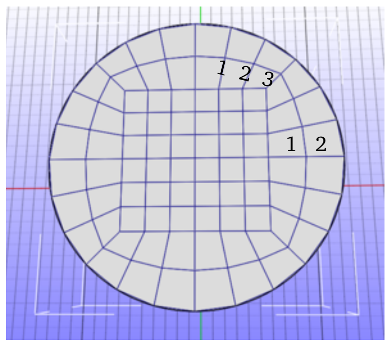


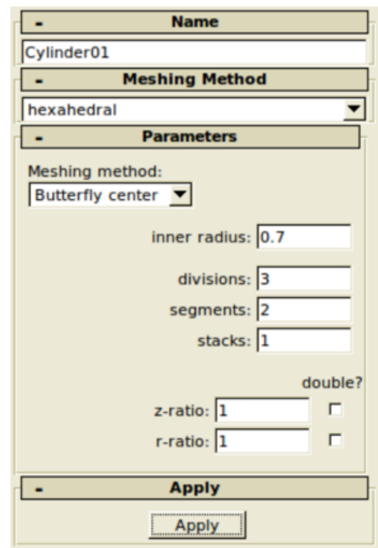
Demo 16

This object is a rigging hook; the model can be found in the INRIA model repository as an .obj file, and is imported into Blender with Blender's .obj import script. Its title is "gancho" (See References). The usual Tris-to-Quads command is given, but there is no need for Partial Dissolve. From experience with this object, we decide to start from the left side and work towards the right.

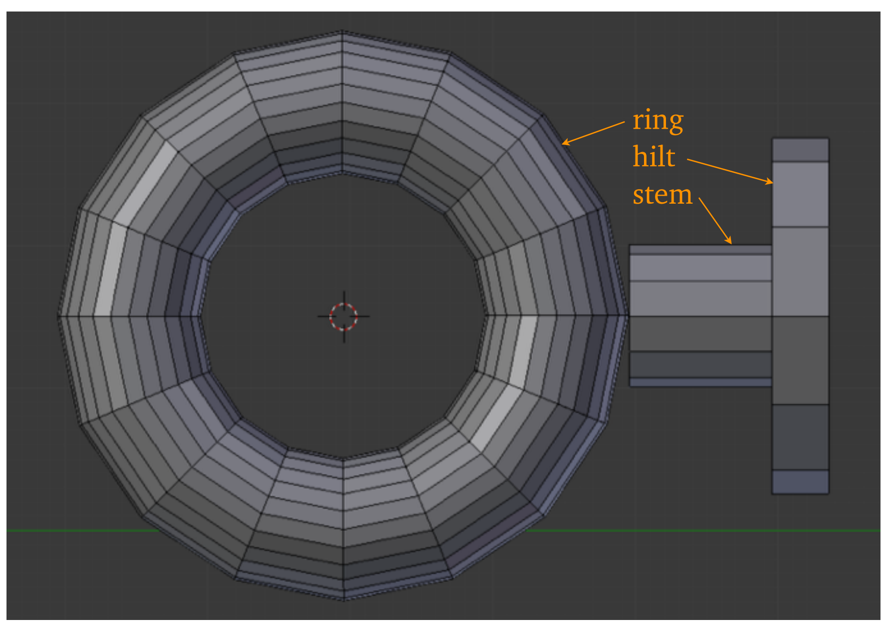


The decision concerning which quad pattern to use is important. It should be flexible enough to adapt to cross sections of varying shapes, but should also be as simple as possible. We settle on a 3-division pattern with 2 segments. (The Preview nomenclature for divisions and segments is shown right.)

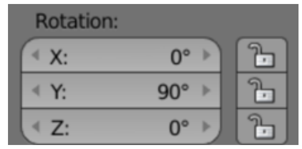


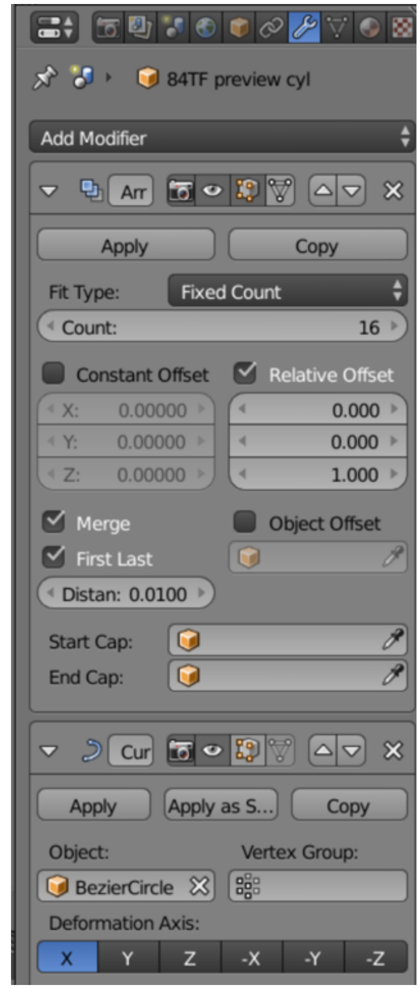


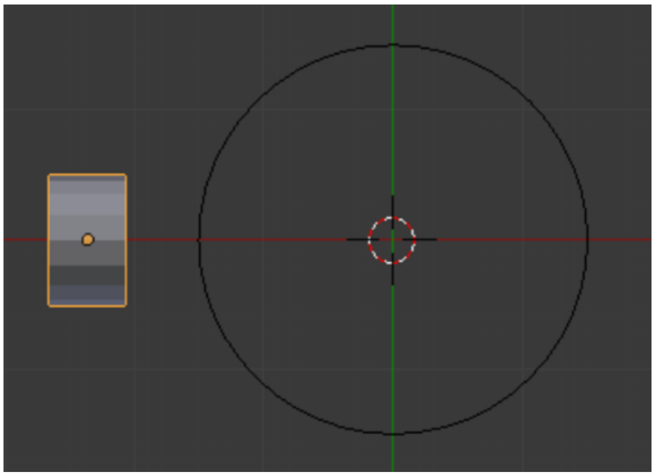
The hook's ring and associated parts are identified right. We notice that originally the depiction of the ring is mere graphic shorthand; it has no real connection with the stem. Preview cannot make the chosen quad pattern in torus form, though it can as a cylinder. Shown right is the array which we created as a substitute ring. It needs to have a tee joint with the stem, which is not a trivial process.



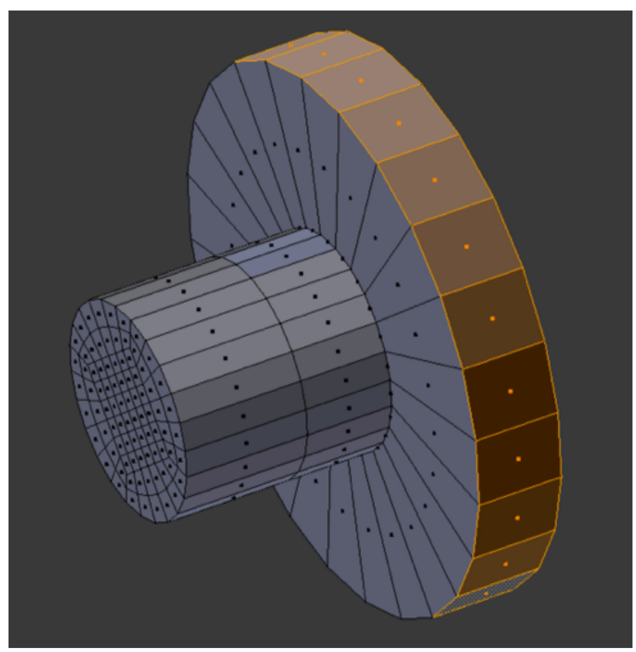
Regarding the array for the ring, we use the bezier circle method of making it. Which variable gets the relative offset, and which Deformation Axis is chosen, are things we determine experimentally, though the rotation of the starting slice seems to have something to do with it.

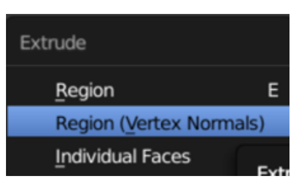






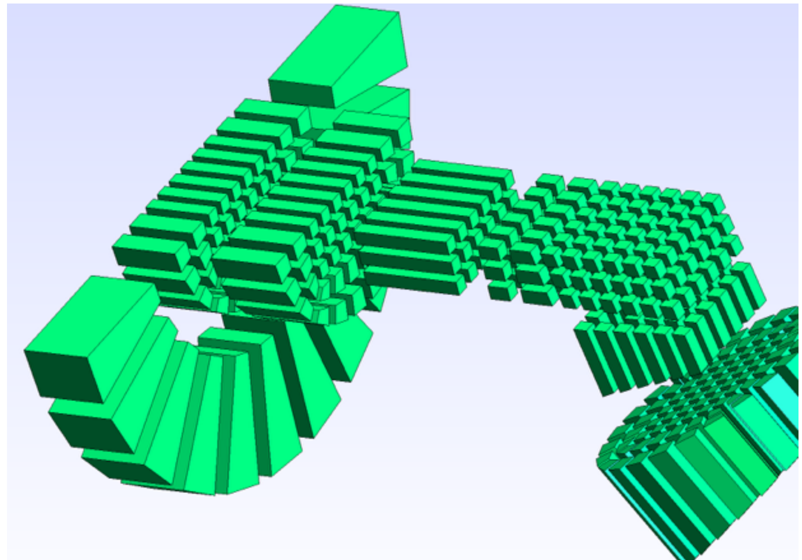
The way we make the stem and hilt is worth mentioning. We lengthen the stem by the thickness of the hilt using the Edges Only method. Then selecting the outer faces, we use the Vertex Normals flavor of extrude to extend the new hilt to the proper diameter. This is as close as we can get Blender to oblige our needs; it still remains for us to fill in some internal faces in the hilt.

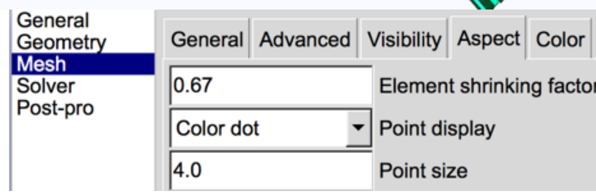




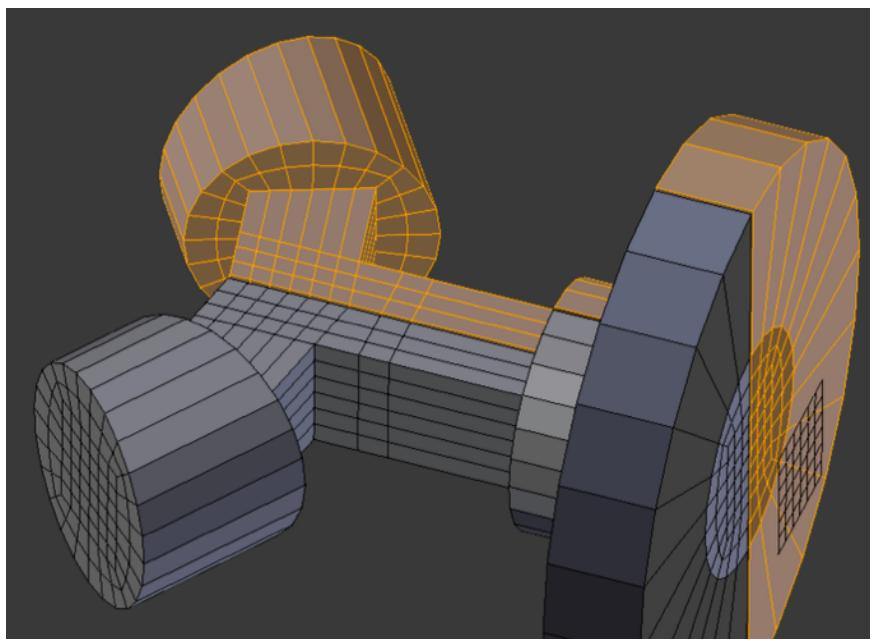
If we want to make sure that all elements in a mesh we create are actually present, we can do a flythrough in Gmsh, using the shrinking factor in the mesh options dialog. Shown is half of the tee we create to tie the ring to the stem.

A similar process can be achieved in Paraview by using the Shrink filter.

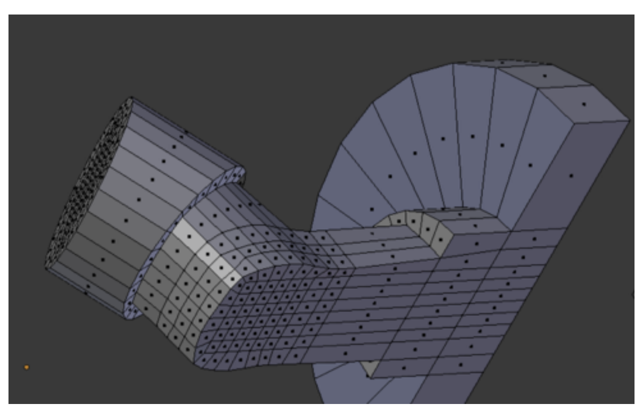




In making the tee, we do the square core first, using a split mesh to save work. At the far right of the picture can be seen the central grid we used for Knife Projecting the juncture of the toroidal array (everything else being hidden at the time). We are able to check the quality of the partially finished piece and report it to be good. This picture anticipates things a bit; it is not time to join things back together yet.

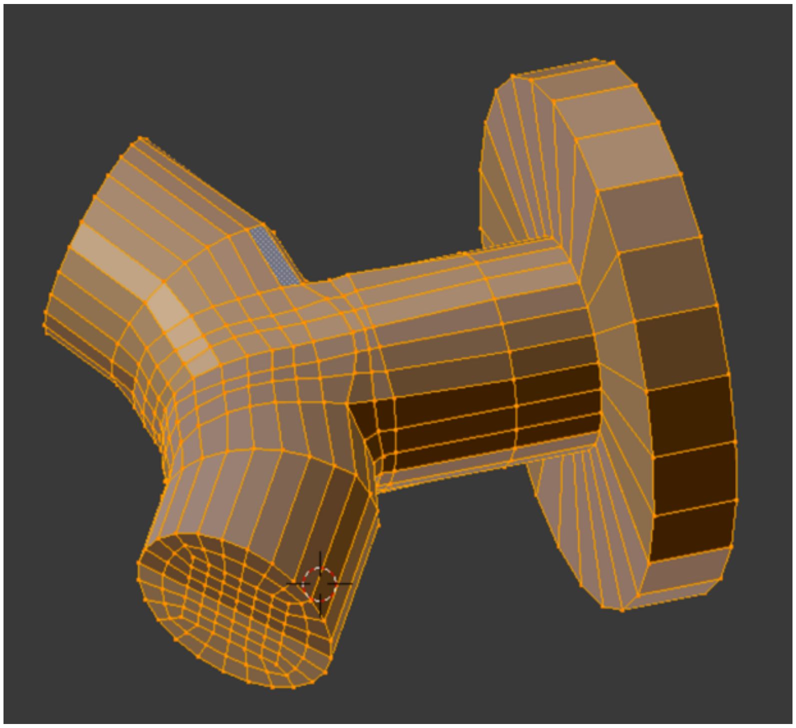


The middle layer is finished. A little late, but we realize that we have symmetry across the z-axis as well as the y-axis, allowing the mesh to be split again, thus saving more time in creating the last layer.

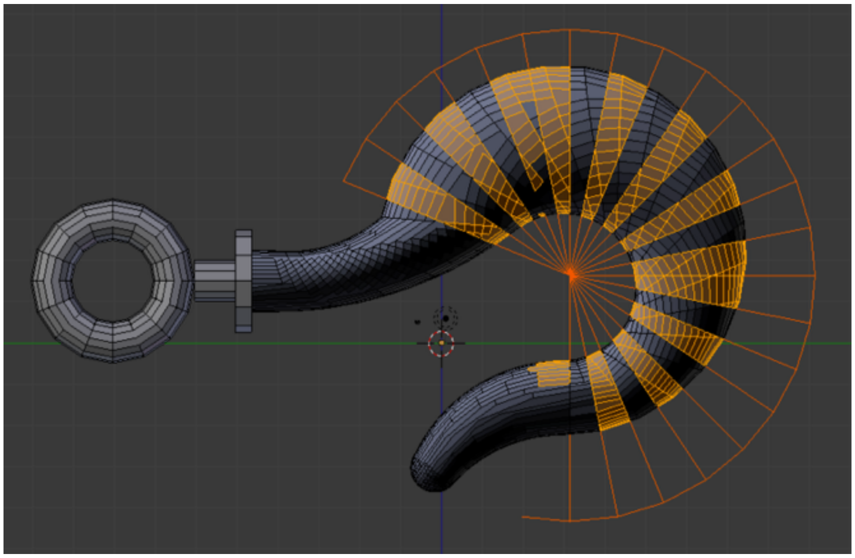


The ring and stem assembly in finished form is shown right. The quality was improved somewhat by grooming a few elements. The Paraview Extract Selection filter used on top of the Mesh Quality filter yielded a bounding box which we used to locate the low quality element in Blender.

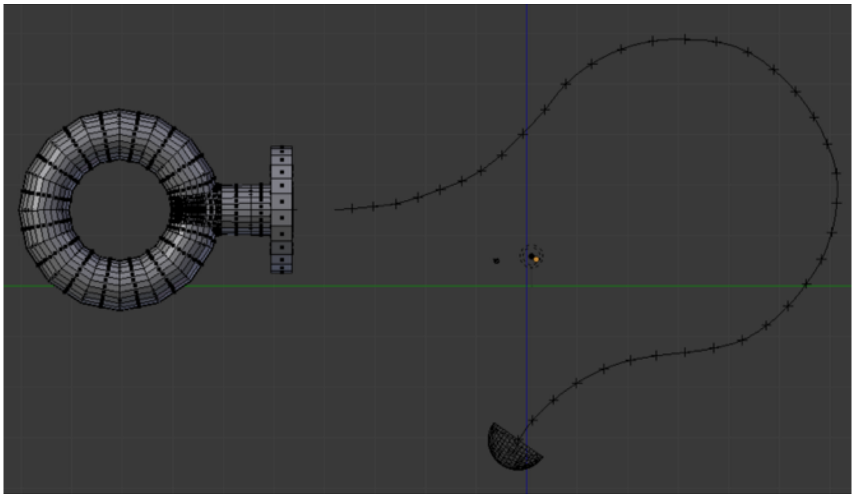
With the quad design chosen and the tricky stem assembly out of the way, we can go on to the serpentine gancho body.



Knife Project cuts are made to extract better section loops than those initially available, and at a later stage we will make a second series of cuts.

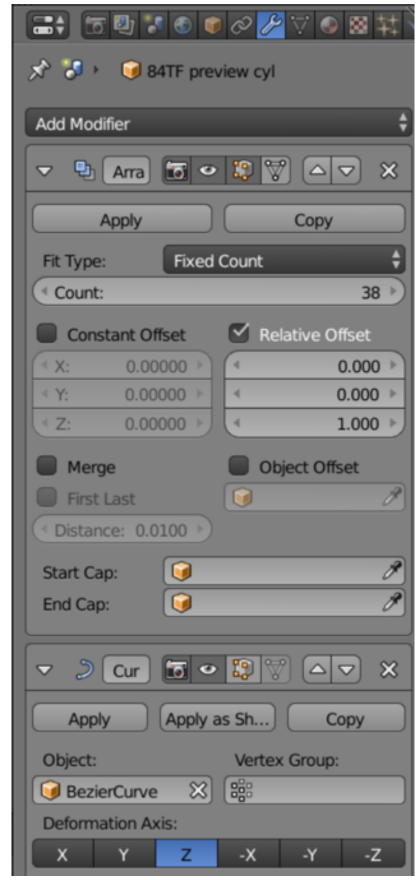


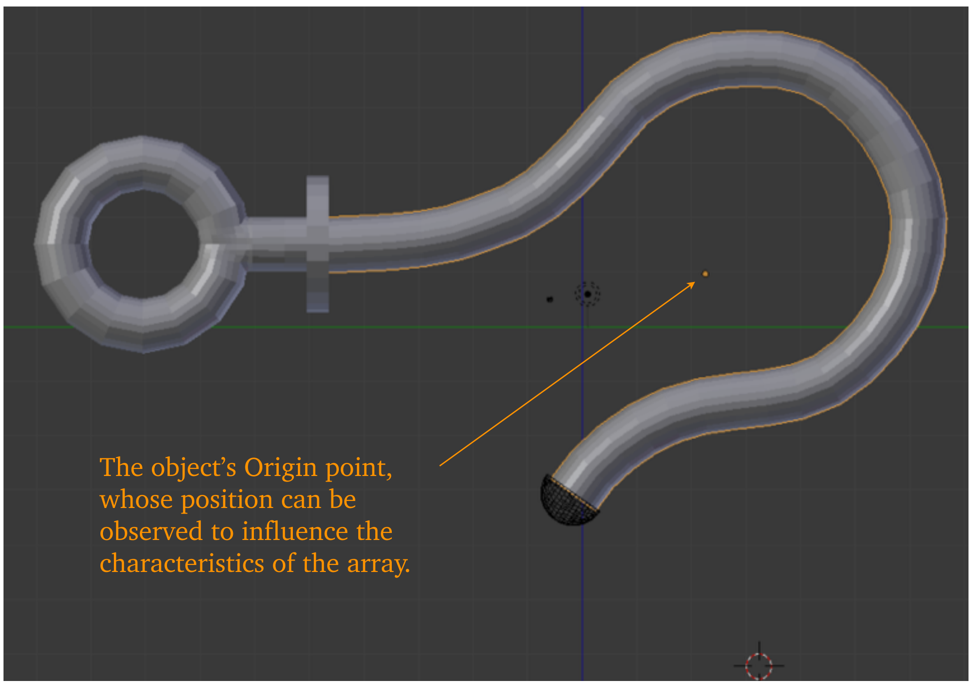
At the centroid of each loop we place an empty. Each empty identifies the location of a control point in a single bezier curve. The bezier curve will govern the behavior of an array, which will advance the progress of a group (or parcel) of elements.

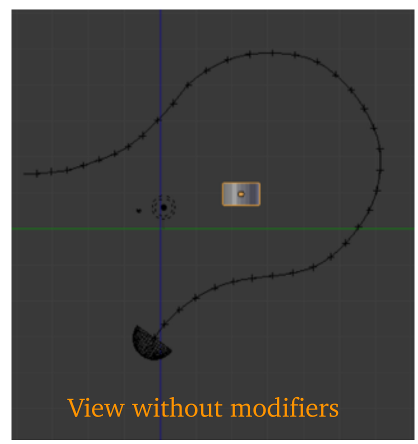


The task before us would seem to lend itself to either a Bridge strategy or a Loft strategy, as contained in the Loop Tools Add-On menu, accessed by the 'w' key. Even the possibility of using Shrinkwrap occurs.

However, the code structure of all these functions protect them against use with non-manifold meshes. Therefore we must resort to a less automated strategy.

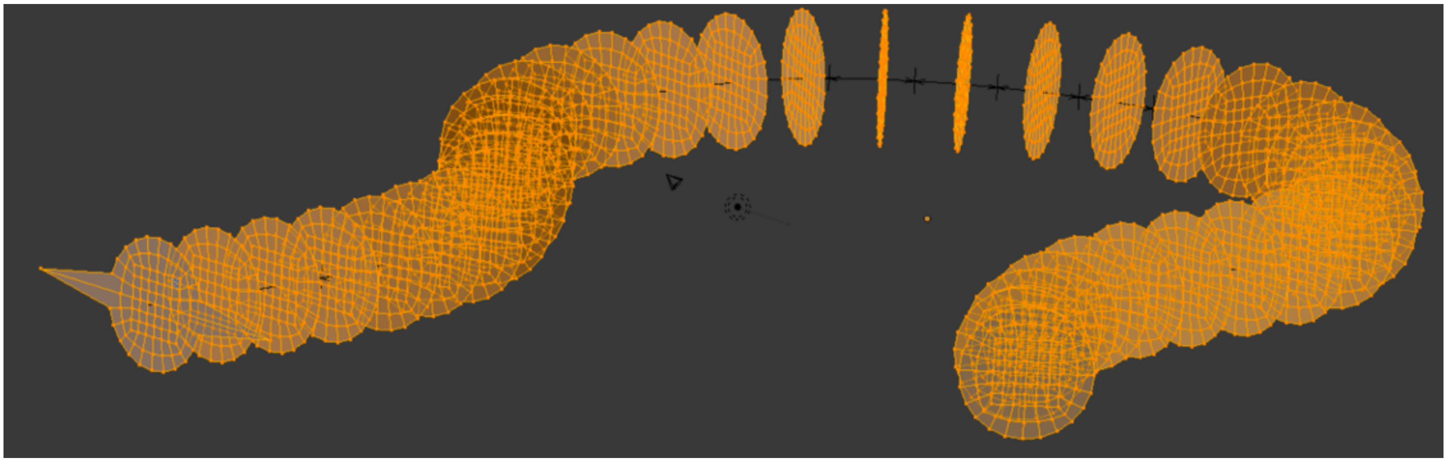




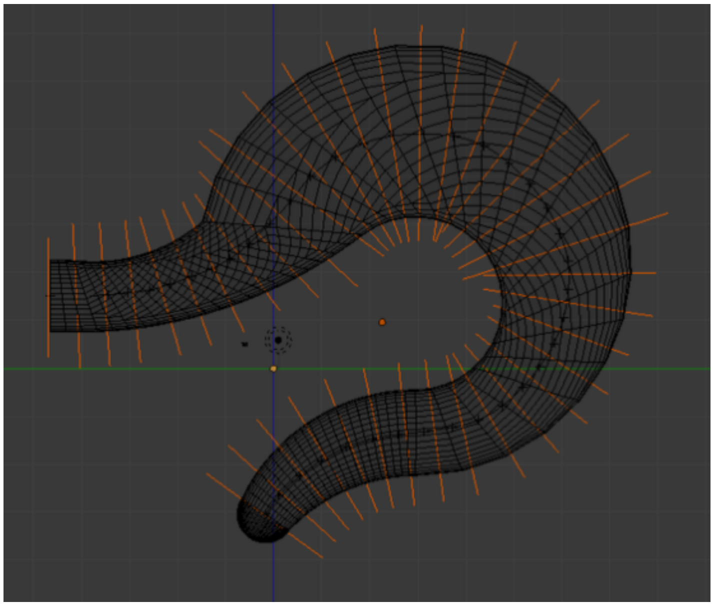


Above is seen the Preview cylinder in the role of an array object subject to the bezier curve. On this occasion we do not try to rotate the array object, and the Deformation Axis and Relative Offset seen in the Modifier panel reflect this. The object Origin is automatically set at the geometrical centroid of the array as it will appear, after Applying. It is necessary to use the 'g' key to adjust the position of the array, until it lines up at its beginning and end points. Note that we have left the Merge function unchecked, because it readily creates triangles. The consequence is that when reconciling sections we will have to remove lots of doubles.

Because of its suggestive shape, we will refer to the precursor array as the 'slinky'.

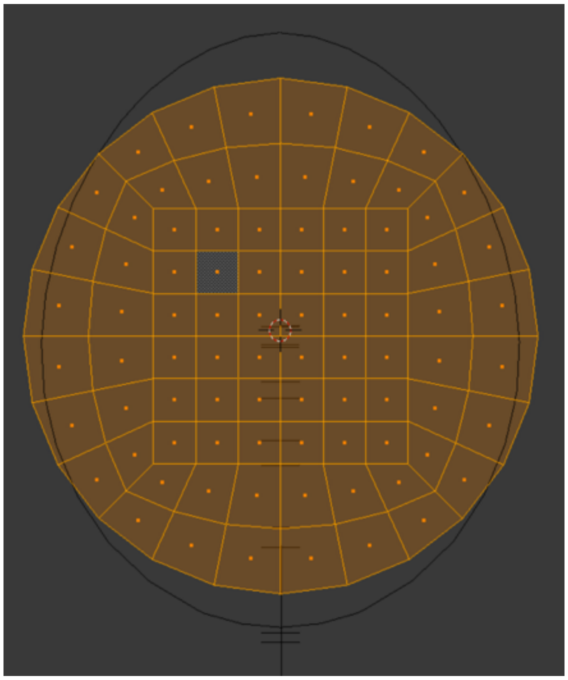


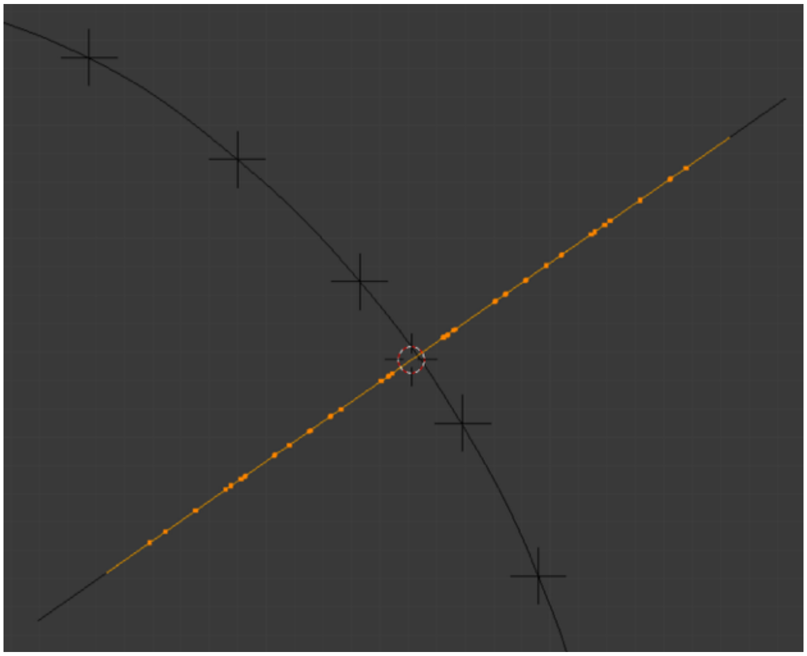
The slinky sections are shown above, with the first showing a special edge. Each section will be equipped with such an edge, and they will make cutters. At right is seen the set of cutters ready to slice a fresh gancho.



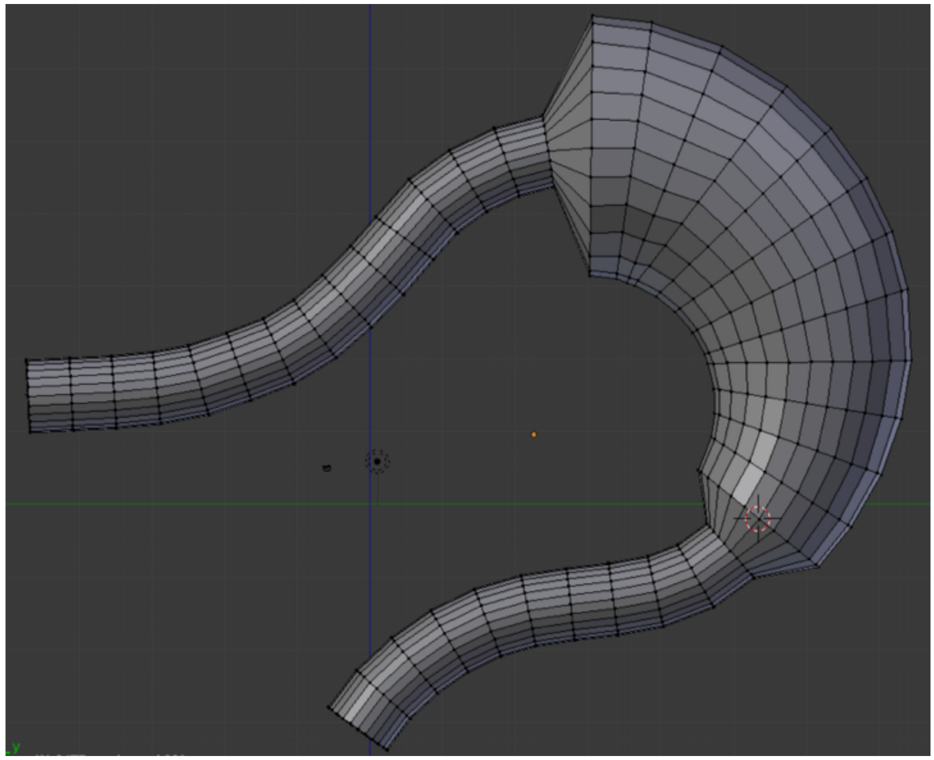
As the gancho has already been cut once, it may seem wasteful to cut it again. However, we feel this is the only way to locate precisely.

Below: Reconciling a section. We can scale and grab, so long as we restore the plane.

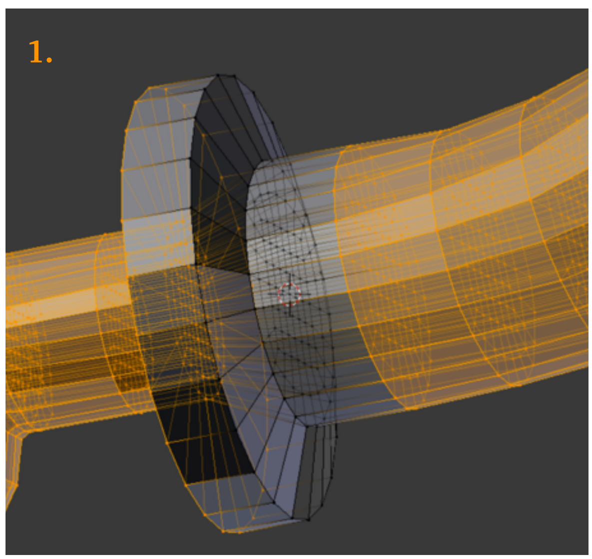


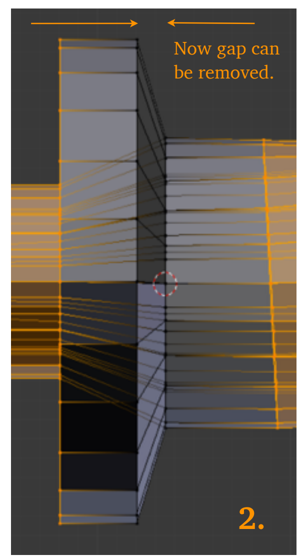


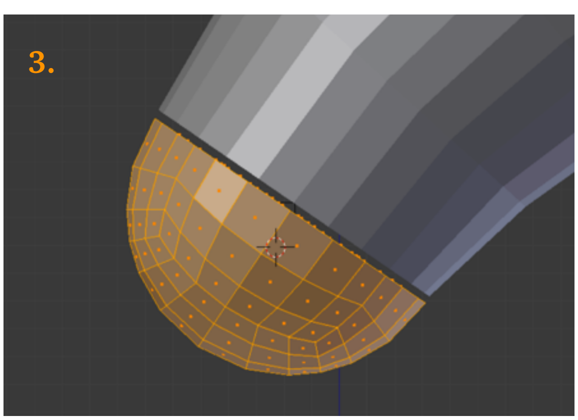
At right is seen the mesh configuration after we have combined 13 sections.

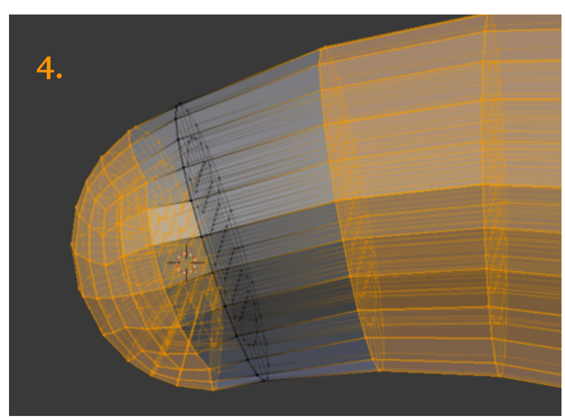


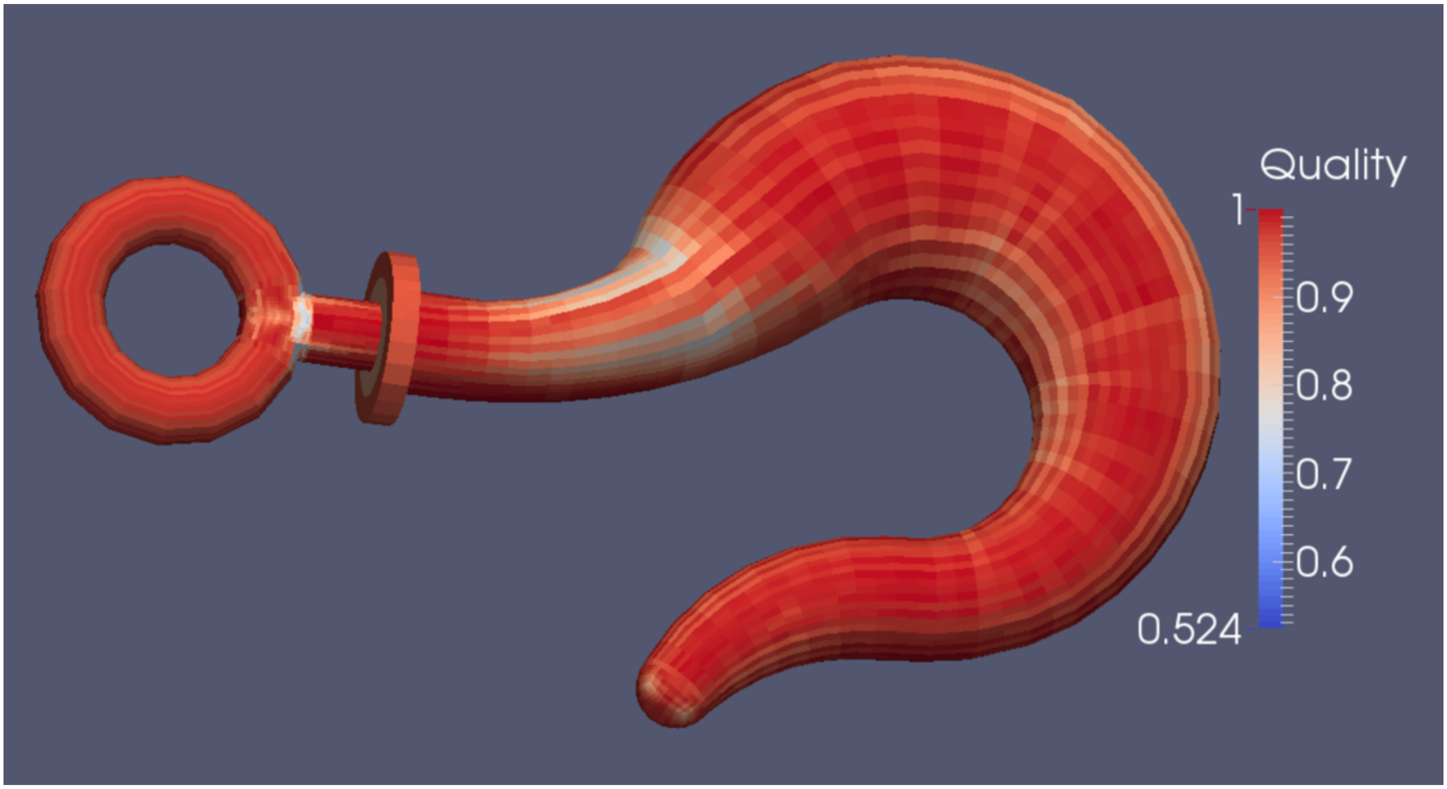
Pictures below: 1. Joining the ring assembly to the gancho. With the quad designs on opposite sides of the joint at different scaling, unpleasantness could arise if the faces are in-plane with each other. Instead, we move the two objects apart a small distance before joining and doing the sewing. 2. The tapered area is collapsed once a successful tie-in has been made. 3. The imported Febio Preview sphere has 3 divisions and 2 segments, but unlike a cylinder, the divisions are not necessarily equal in size. 4. The hemisphere is sewn into the tip area of the gancho.







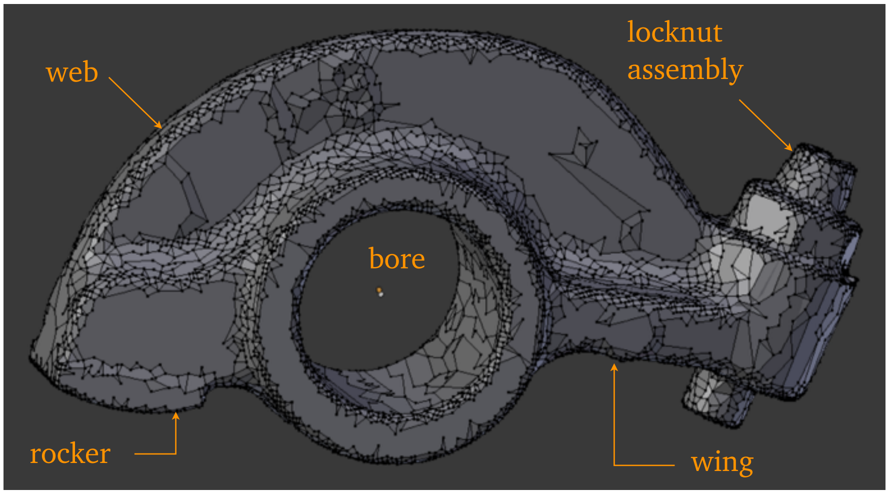




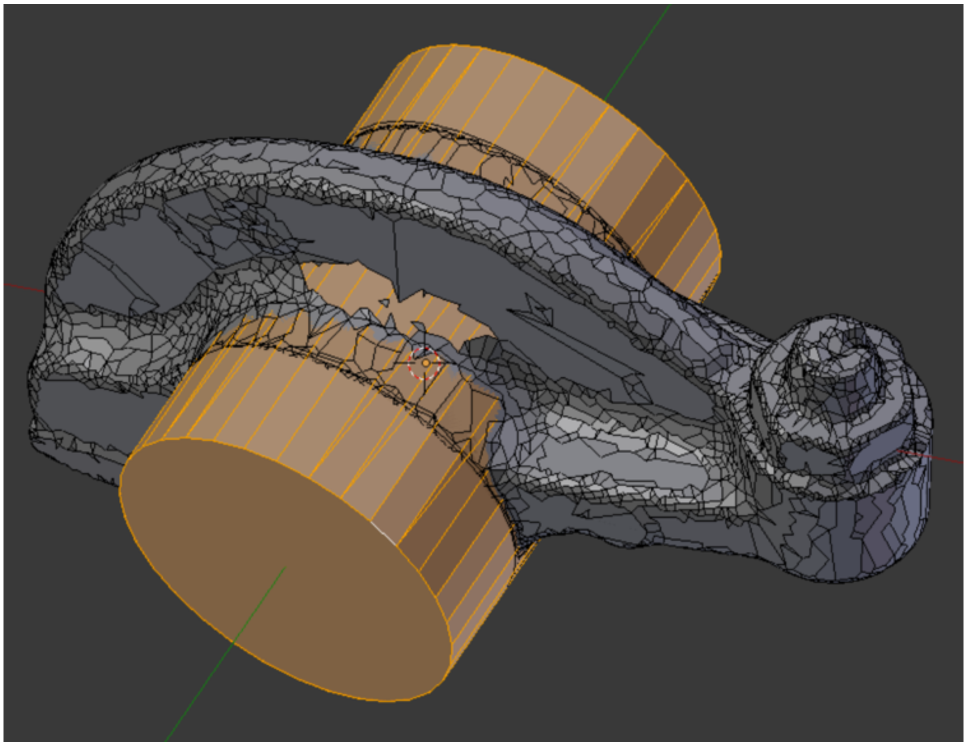
The final hook mesh is shown above. The mesh quality according to Verdict's Scaled Jacobian standard is acceptable. The Gmsh-refined version has 49,357 nodes and 45,736 elements.

Demo 16

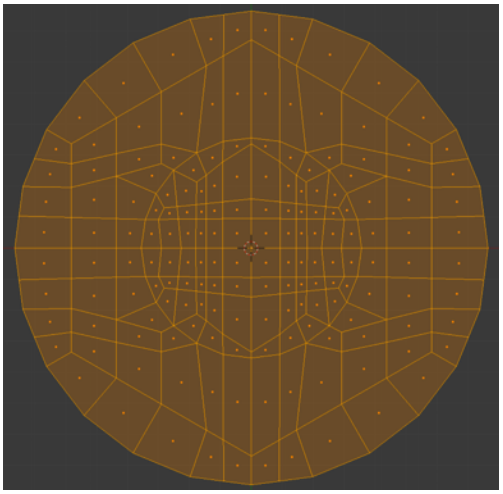
This object is a rocker arm; the model can be found in the INRIA model repository as a .ply file, and is imported into Blender with Blender's .ply import script. Its title is "rockerarm 25" (See References). It is evidently the product of an optical scanning process. The usual Tris-to-Quads and Partial Dissolve commands are given, the latter with a Max Angle of 20 degrees. Our aspirations with this object are limited: merely to carve out a shape that roughly resembles the original, which could be used for ballpark finite element analysis.



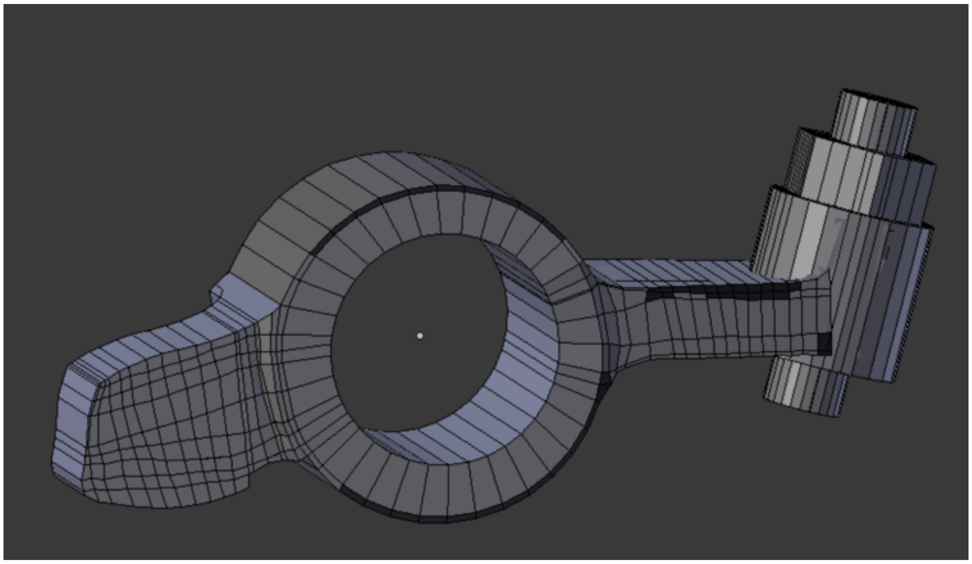
We proceed to introduce a geometrical framework for the mesh. We notice some peculiarities relating to actual samples of machine work. For example, the bore and its lip are not concentric.



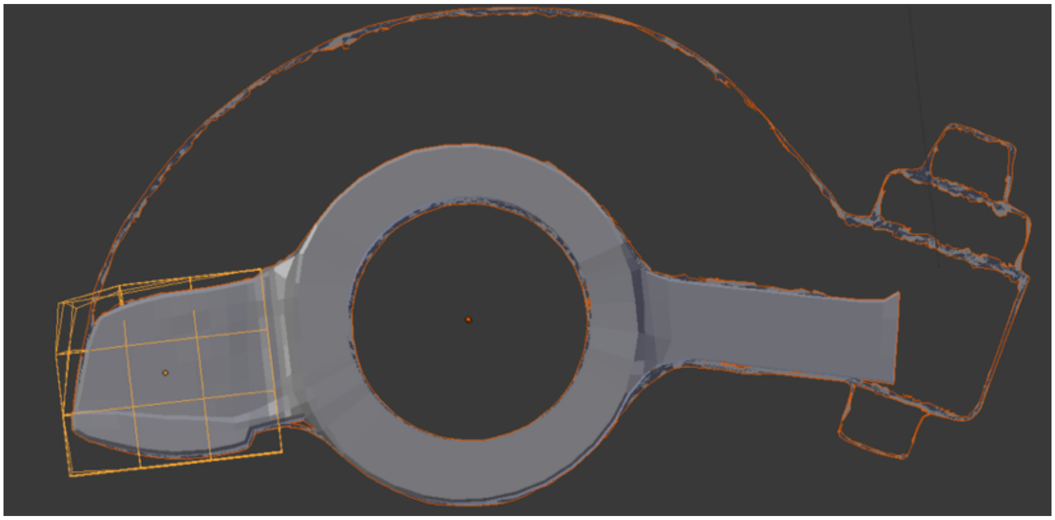
With an appropriate template, we can execute the locknut assembly as a continuous sequence of extrusions.



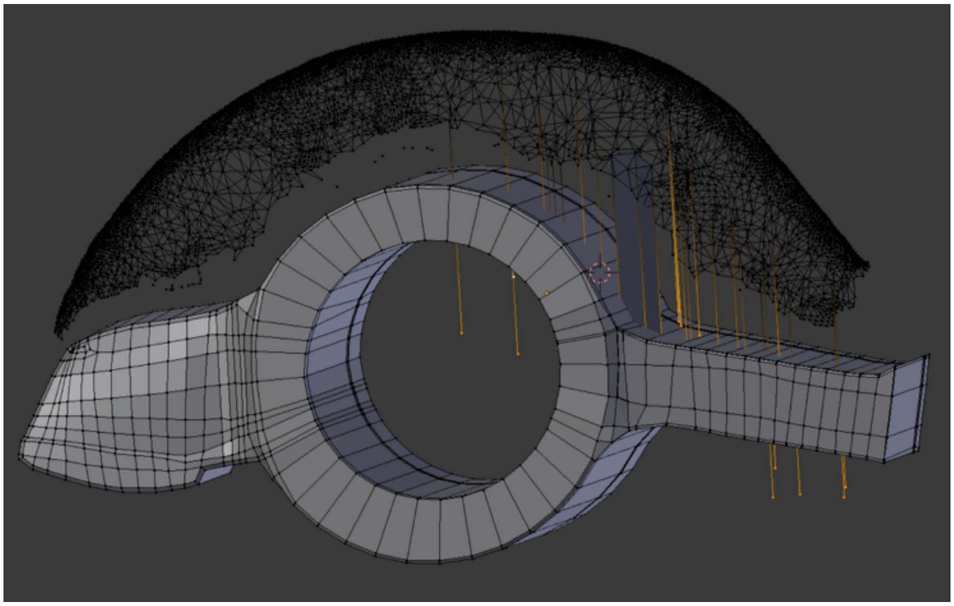
An extruded outline of the rocker/bore/wing section gives us a place to start. An early version of the locknut assembly is positioned to give a rough preview.



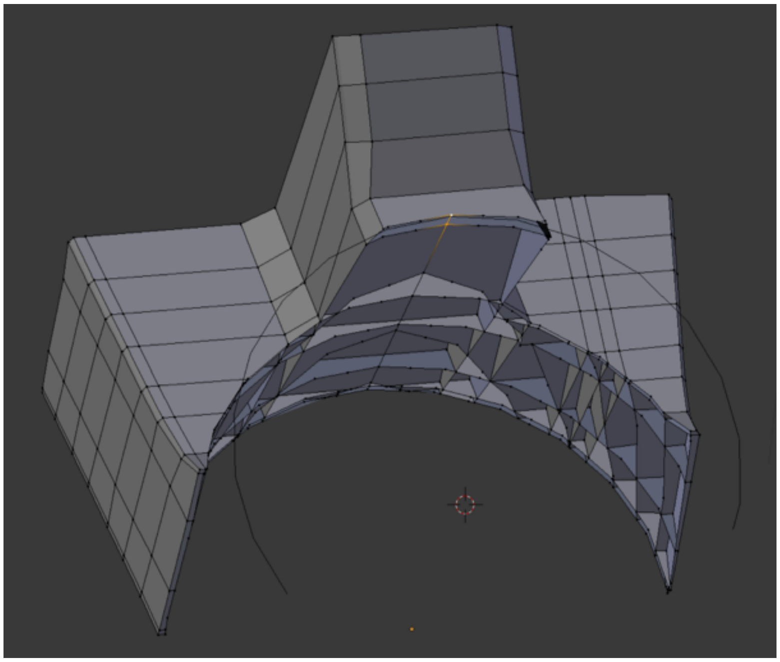
One Blender tool that works with non-manifold meshes is the Lattice modifier. Here it is the last of three objects selected in Object mode. Then the lattice is edited in Edit mode to fine tune the shape of the rocker.



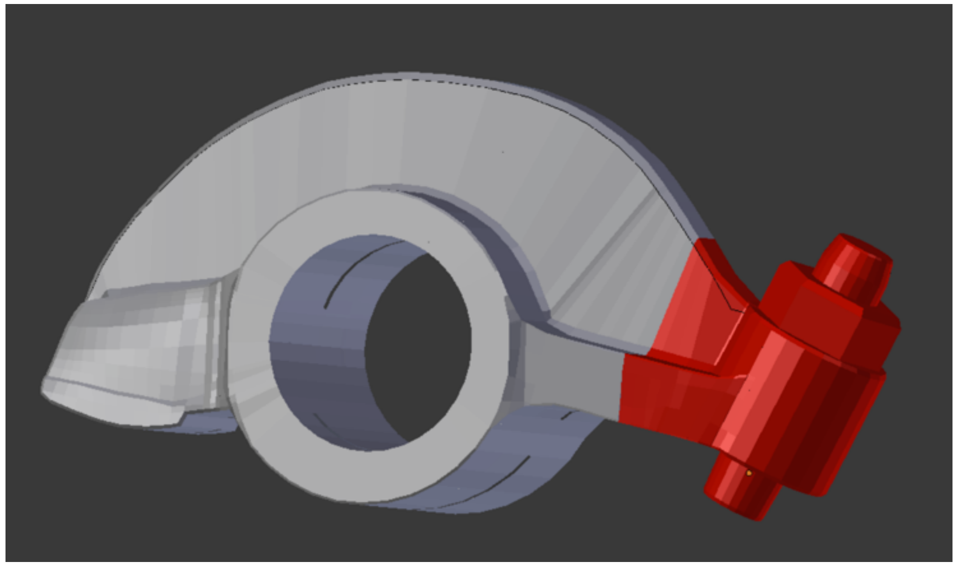
The web is not aligned with a major global axis. Its central plane is also slightly curved. Here we drop points along the z-axis to locate the web's base position on the rocker arm surface.

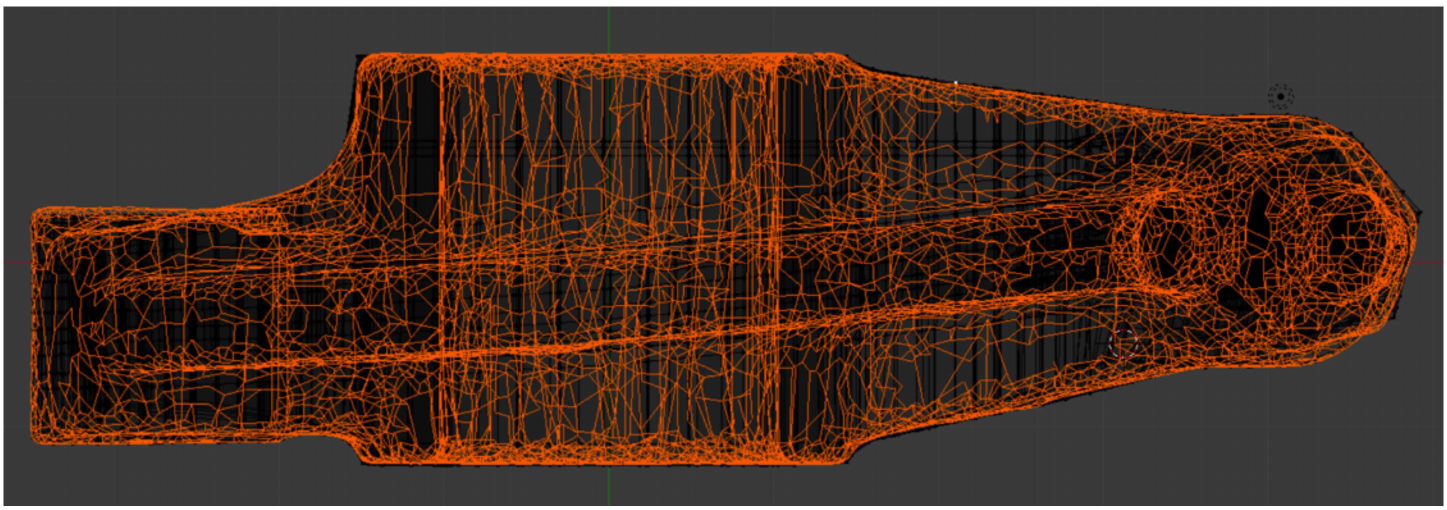


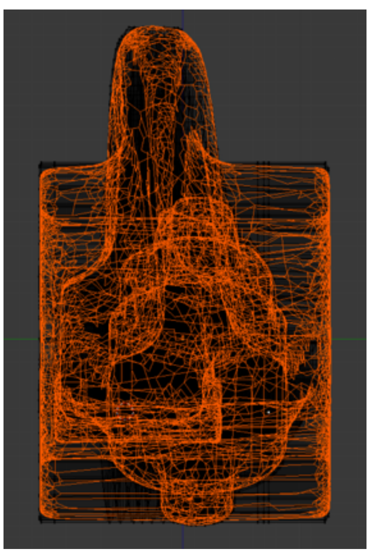
We break off a chunk of wing tip as a separate object and cut it in way of the locknut assembly. Edges from the locknut barrel are seen here out of cutting position. Whenever the Knife Project tool is made up of a series of connected edges, like the arc shape shown here, stray points are scattered along its path. Merging these with existing points after the cut can be a chore.

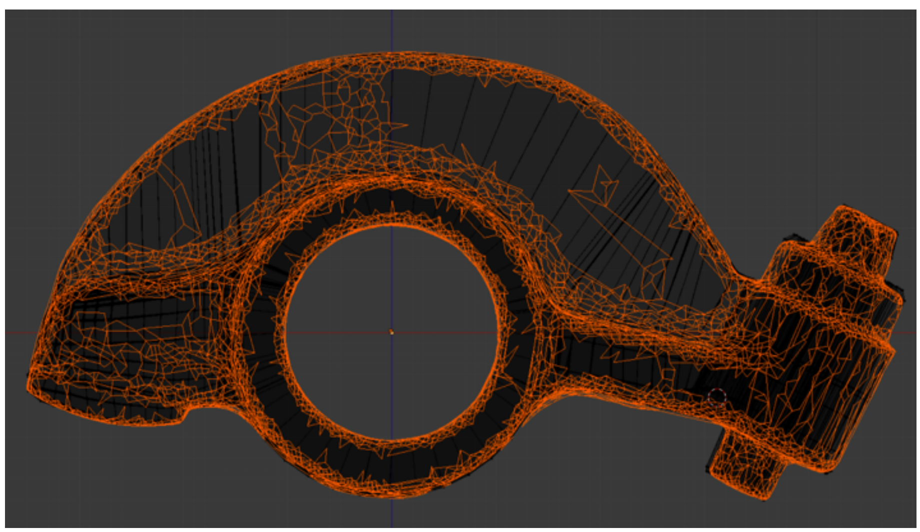


The locknut assembly is fitted to the wing and web. Paraview can be used to identify first triangles, then low quality elements, in the area of the join.

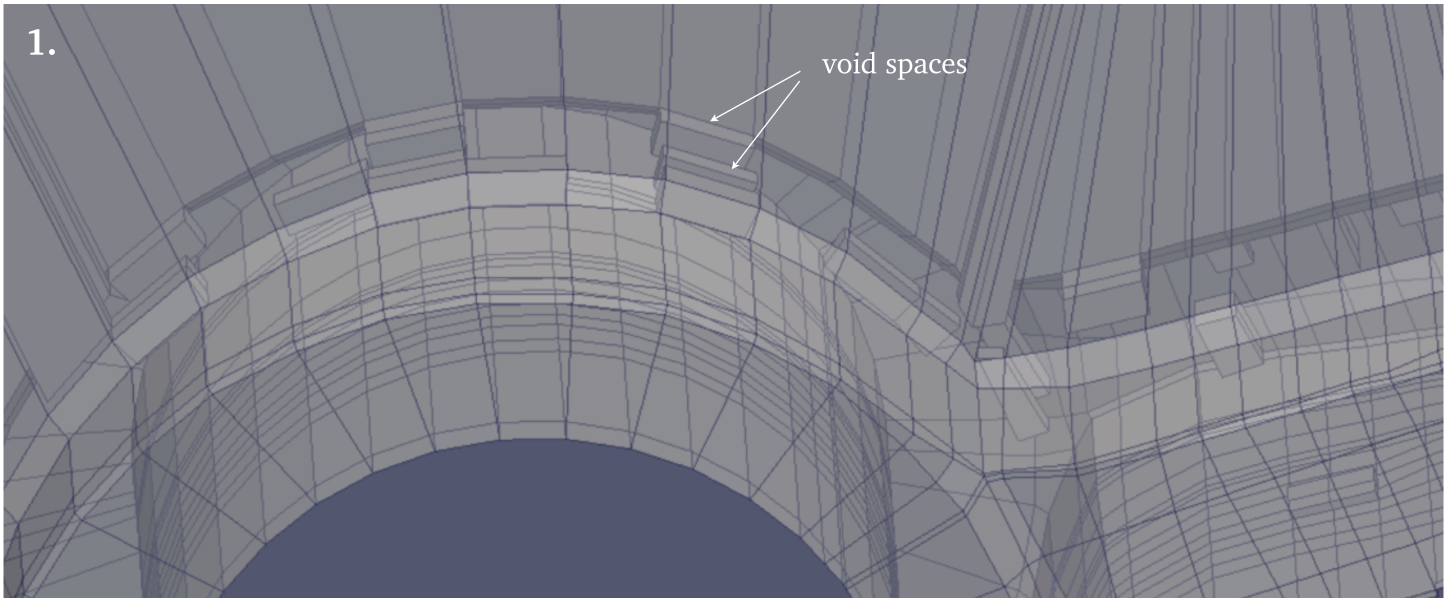


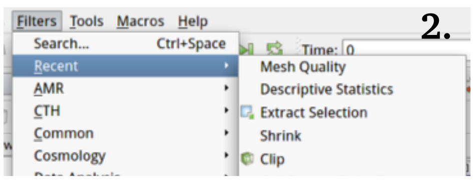


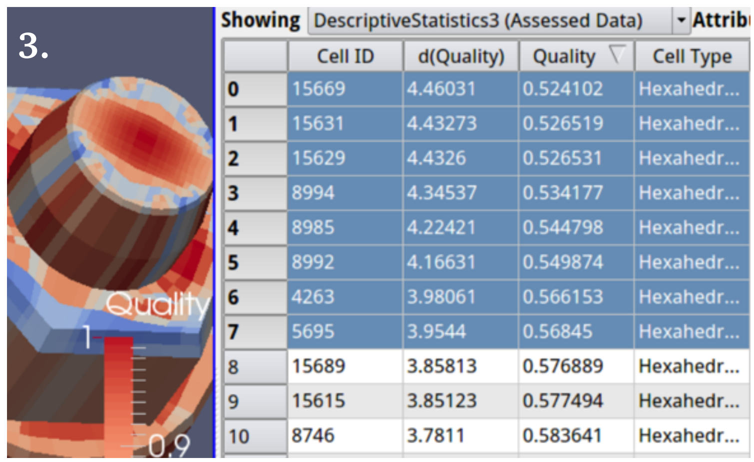




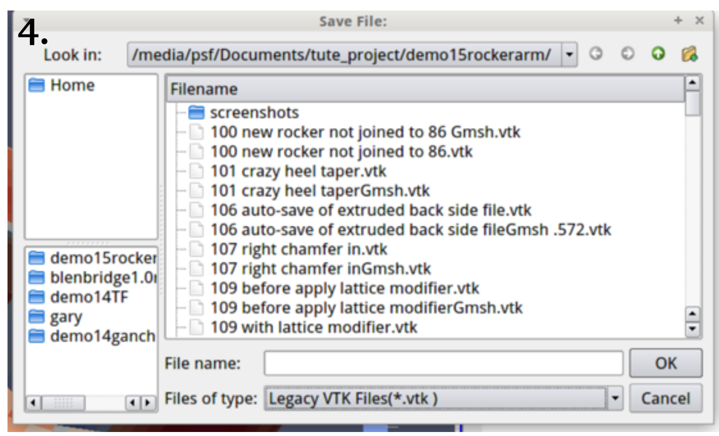
Top, side, and front orthographic views, superimposed on the workpiece, guide the phases of the construction.





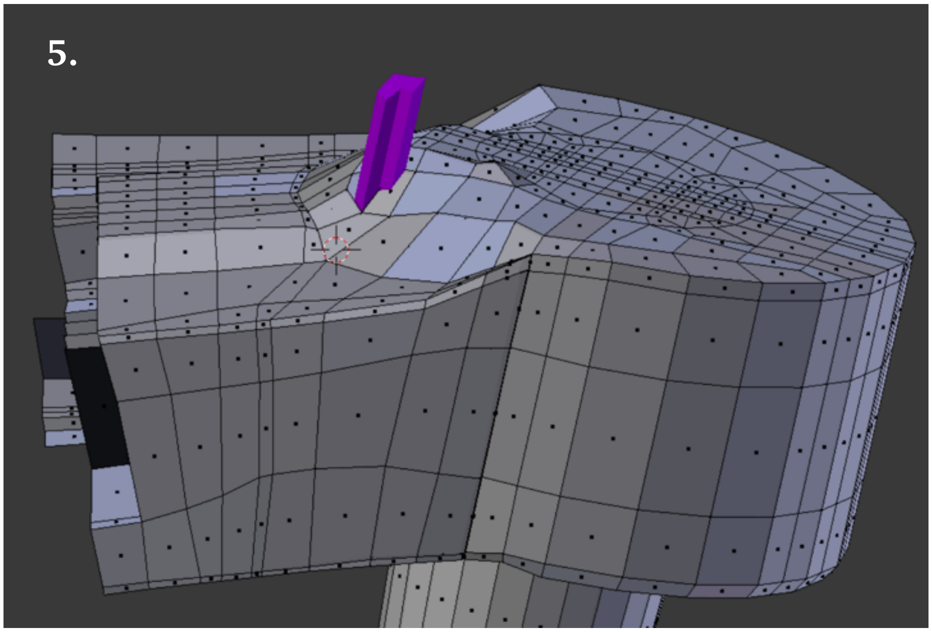


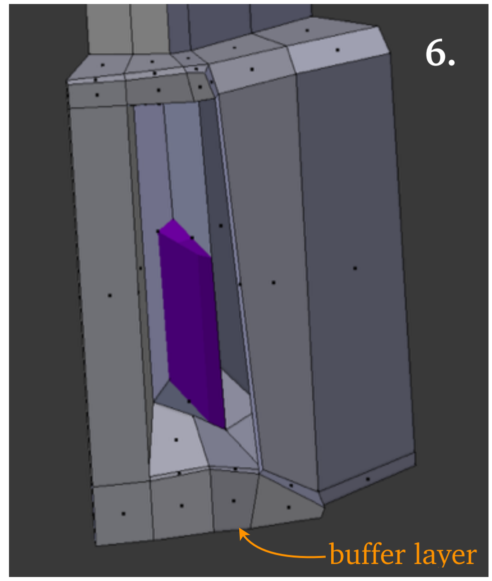
	Cell ID	d(Quality)	Quality	Cell Type
0	15669	4.46031	0.524102	Hexahedr...
1	15631	4.43273	0.526519	Hexahedr...
2	15629	4.4326	0.526531	Hexahedr...
3	8994	4.34537	0.534177	Hexahedr...
4	8985	4.22421	0.544798	Hexahedr...
5	8992	4.16631	0.549874	Hexahedr...
6	4263	3.98061	0.566153	Hexahedr...
7	5695	3.9544	0.56845	Hexahedr...
8	15689	3.85813	0.576889	Hexahedr...
9	15615	3.85123	0.577494	Hexahedr...
10	8746	3.7811	0.583641	Hexahedr...



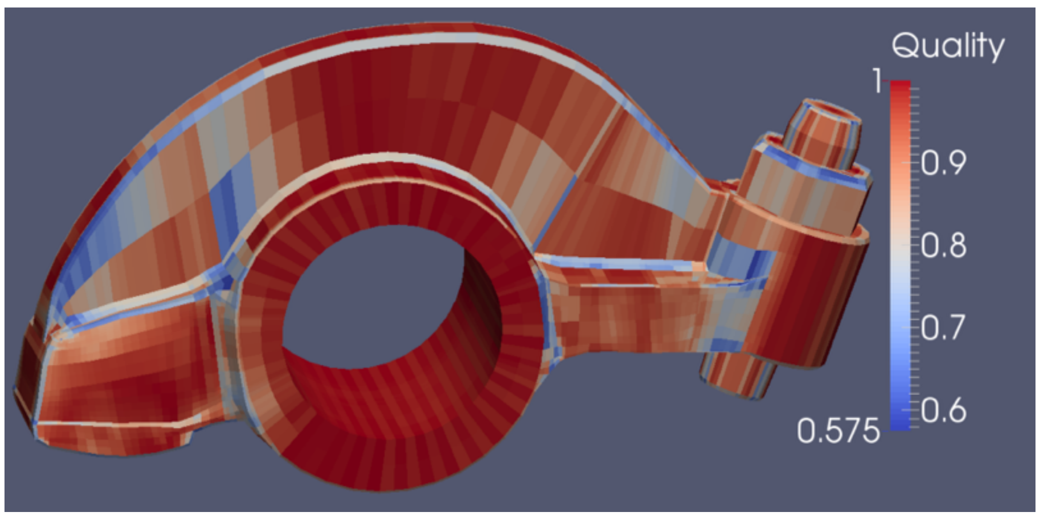
1. By reducing the opacity in the presentation view of the .vtk model in Paraview, we can see inside and identify void spaces, where elements are incomplete.
2. The Filters menu, showing five filters we especially rely on.
3. A common task sequence is to put a Descriptive Statistics filter on top of a Mesh Quality filter, then sort by quality and highlight, automatically making a selection of the highlighted cells. Then an Extract Selection filter applied to the Descriptive Statistics filter grabs the selection as soon as applied.
4. Continuing number 3., the Save Data menu option under the File menu allows the extracted selection to be saved as a legacy .vtk file, which Blenbridge can convert to a Blender-readable .ply file. As an object in Blender, the group of elements stand on top of their Blender counterparts, for perfect identification.

5. The element group described in 4. is non-selectable in Edit mode.
6. Breaking off a chunk for free editing as a separate object and then re-joining is convenient, but it is important to include a buffer layer whose outer boundary will not be altered.

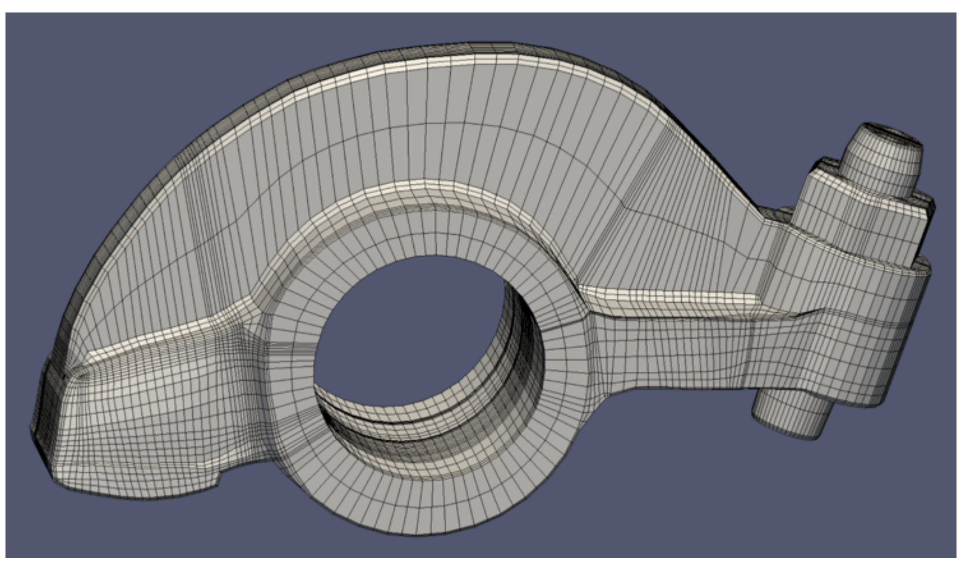




The finished mesh is shown in Paraview. The mesh quality is acceptable.

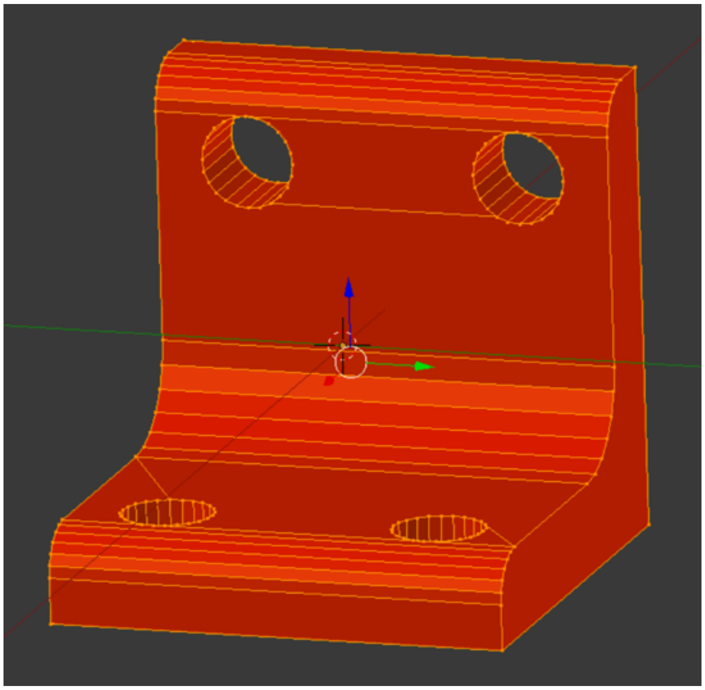


The final mesh contains 62,208 elements and 70,183 nodes.

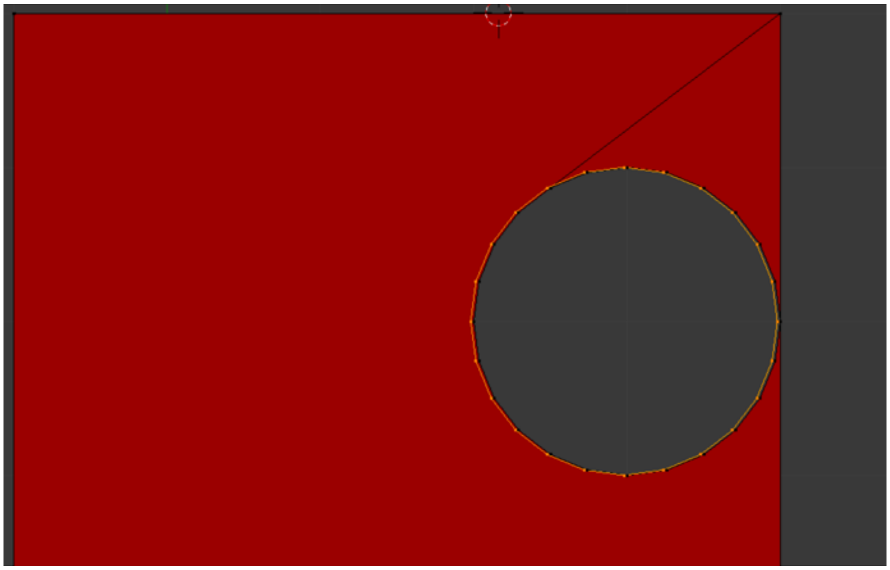


Demo 17

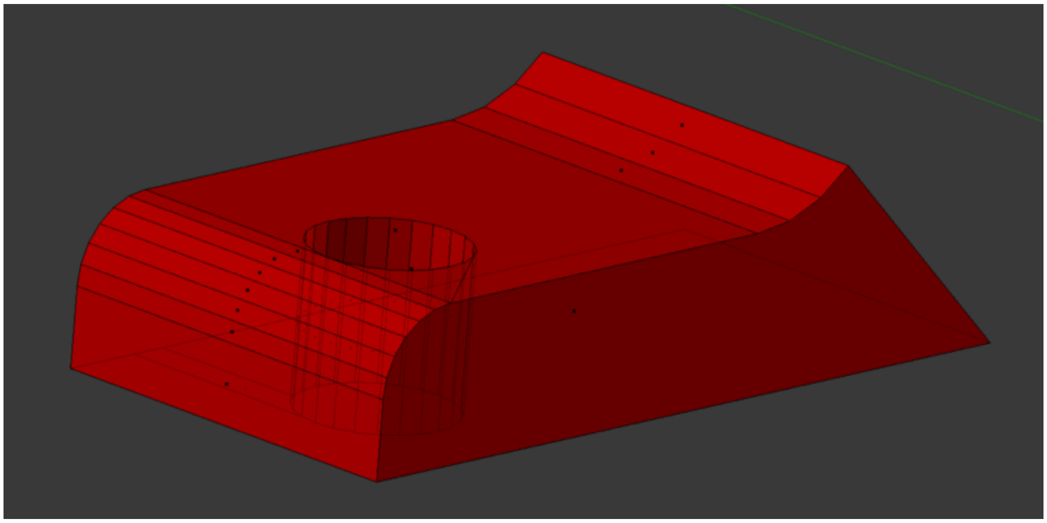
This object is a filleted bracket; the model can be found in the INRIA model repository as a .3DS file, and is imported into Blender with Blender's .3DS import script. Its title is "bracket," and it is in the Drexel section of the repository (See References). The usual Tris-to-Quads and Partial Dissolve commands are given. The object has a deceptively simple appearance, and some caution is warranted in building its mesh. The object is aligned with the major axes. Hole replacement will not be necessary in this case.



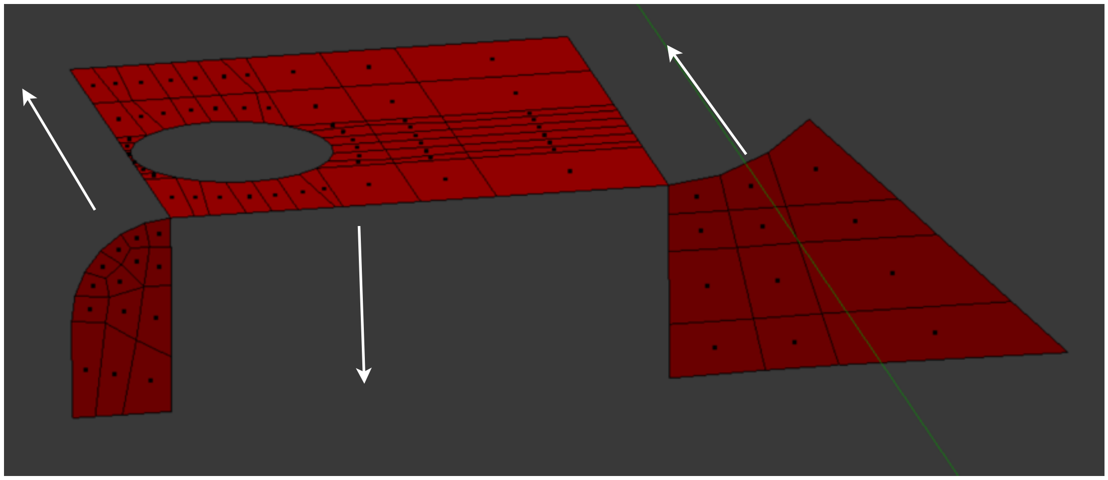
Edges meeting at the extremely small value of 7.5 degrees will constitute a problem for any mesh generating system, and that is what we find here, at the point where the holes meet the fillet tangent. Some fudge is necessary. We choose to nudge the hole over the arbitrary distance of a thousandth part of the hole diameter.



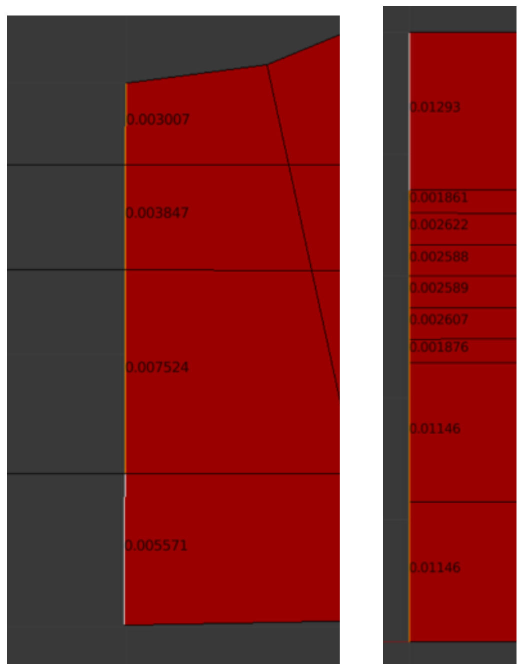
We quarter the object to save time, then decorate the top surface and two front surfaces with faces. There will be two extrusion directions: back, and down.



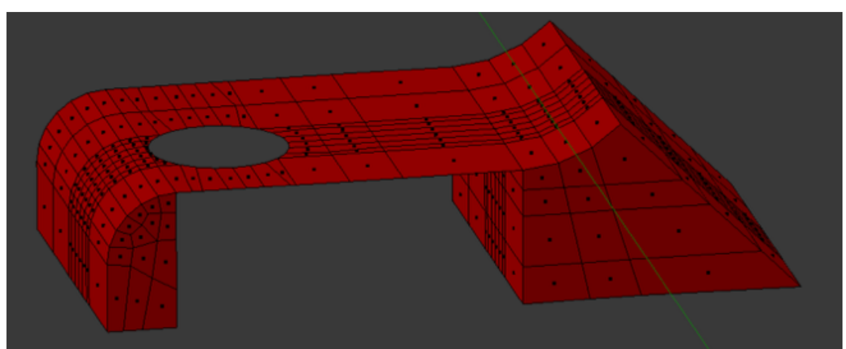
We have checked the face angles of the newly created faces and found them satisfactory. It only remains to do the extrusions.



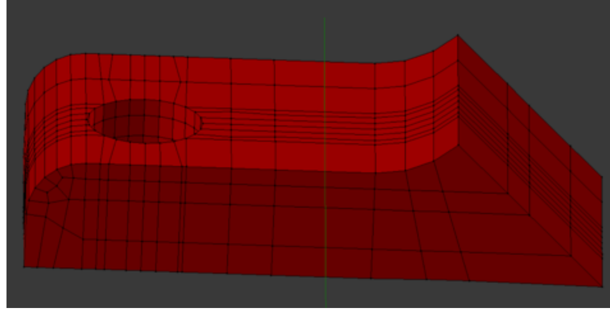
The heights of the vertically oriented edges dictate the extrusion lengths in the vertical direction, and the widths of the horizontally oriented edges dictate the extrusion lengths in the horizontal direction.

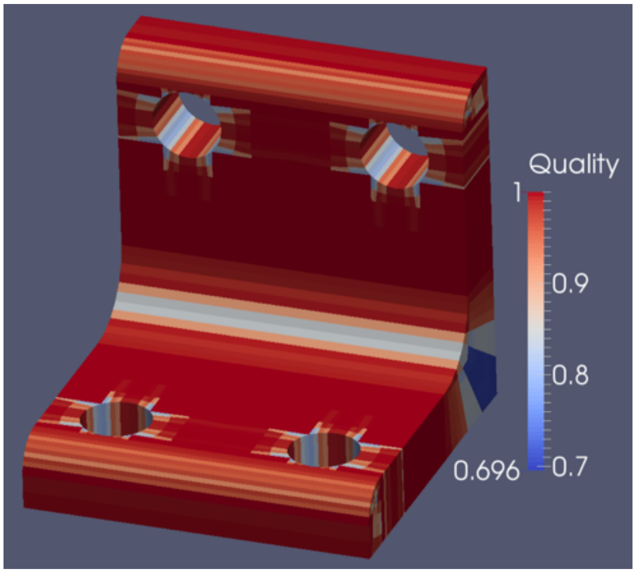


The view after the horizontal extrusions are finished.

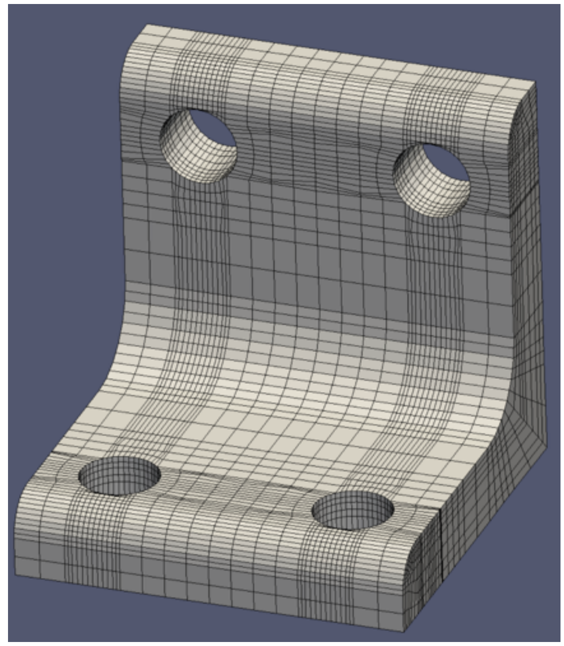


The view after the vertical extrusions are finished.



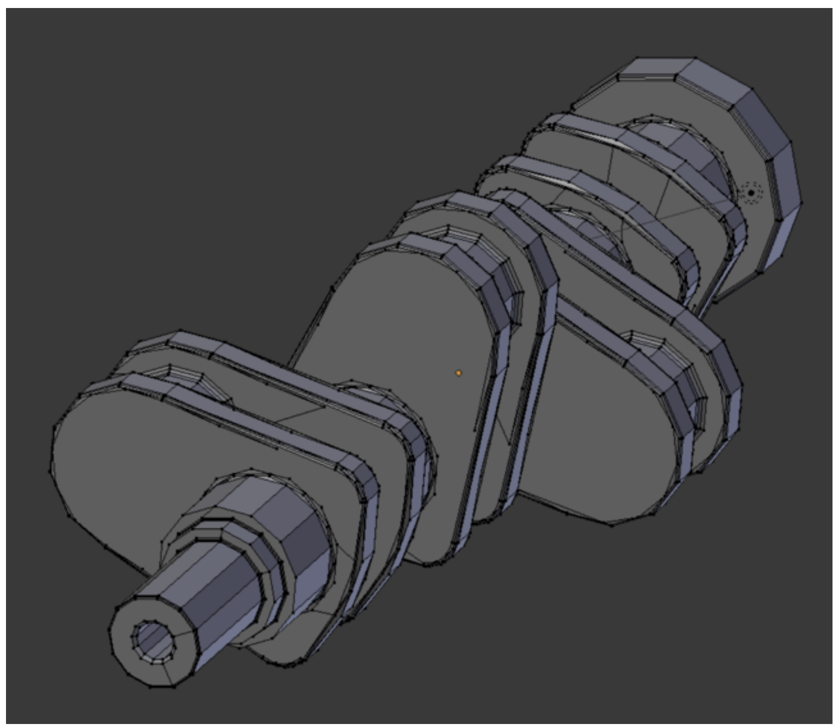


After mirroring and reassembly, the Scaled Jacobian quality is found to be good. Naturally, the mesh was refined in Gmsh.



The final mesh contains 14,400 elements and 17,881 nodes.

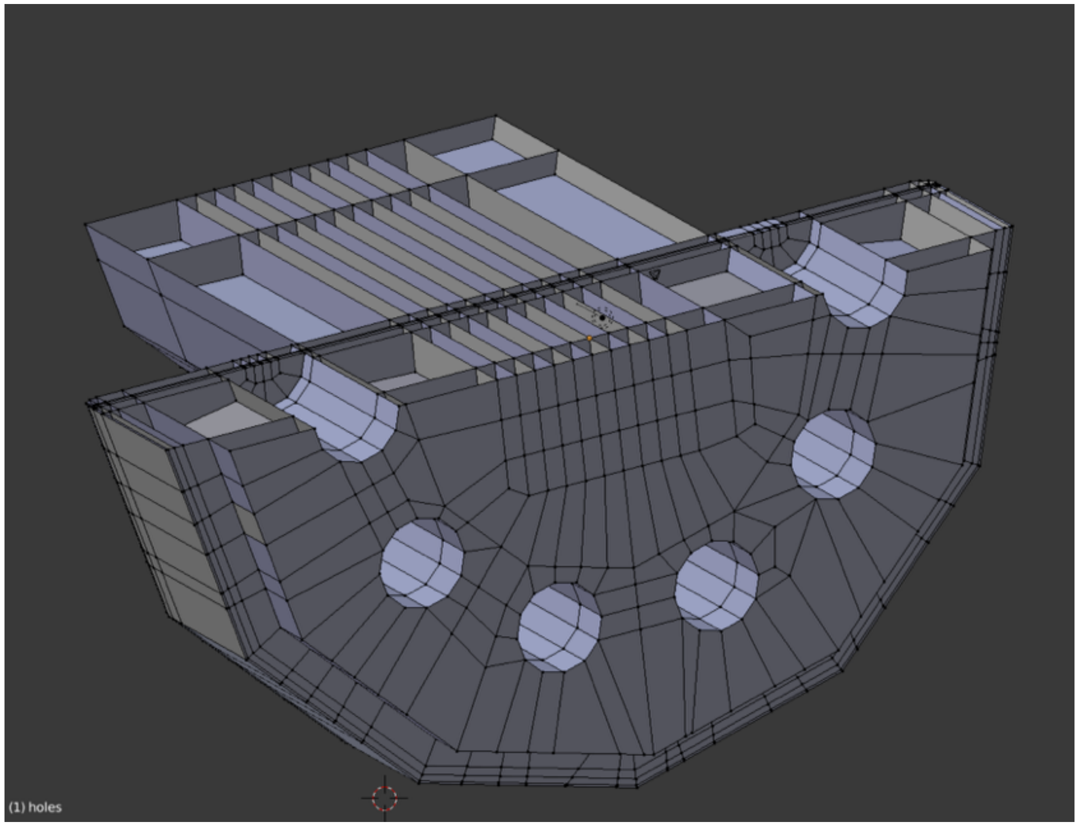
<u>Demo 18</u>

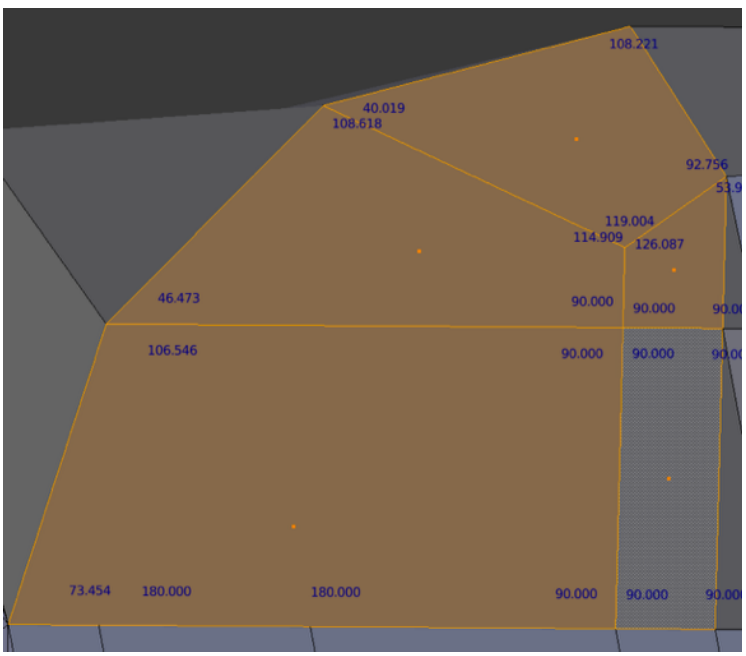


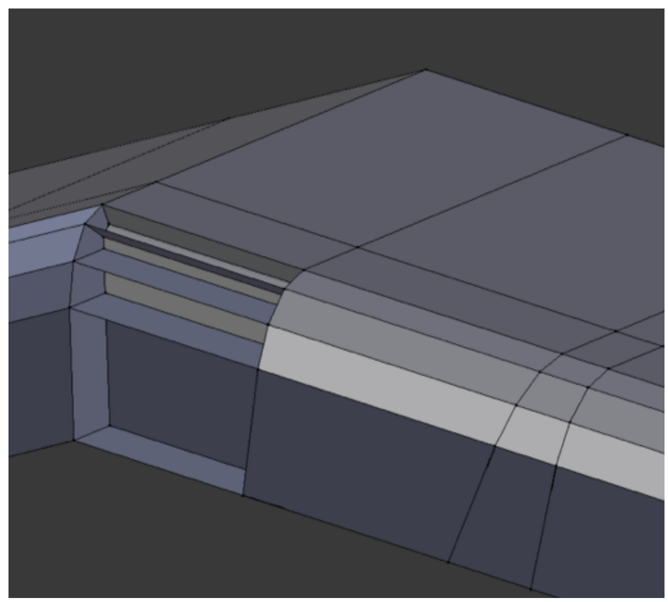
This object is a crankshaft; the model can be found in the INRIA model repository as a .mesh file, and is opened in Gmsh. It is saved as a .stl file and imported into Blender. Its title is "crank3D," and it is in the Mechanical section of the repository (See References). The usual Tris-to-Quads and Partial Dissolve commands are given.

Though the amount of detail could be expanded, we elect to follow the existing 12-part division of the circular elements.

The holed flange. With 12-hole division on the exterior and 10 bottomed holes, it is not possible for us to exploit any symmetries of construction. One of the tasks necessary is to verify the correct placement of the hole bottoms after extrusion.

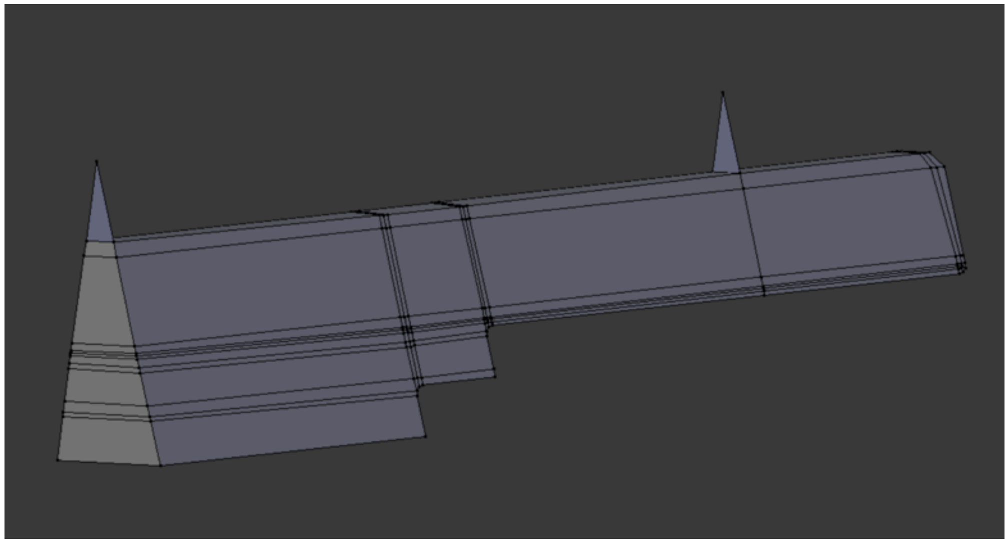




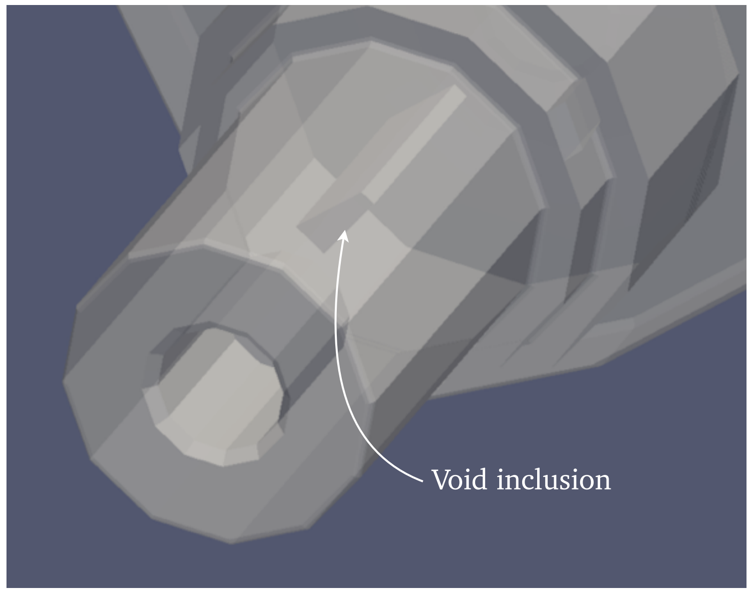


The division plate faces for the outside fillet on the back end of the holed flange are shown above left. This section is made separately from the rest of the flange, using an array, then appended to the waiting holed flange subassembly. The right view shows the fillet after installation.

On the other end of the crankshaft, the central holed stub is also made from an array, one section of which is shown right. After construction, we must replace the triangular portions with standard hole bottoms.

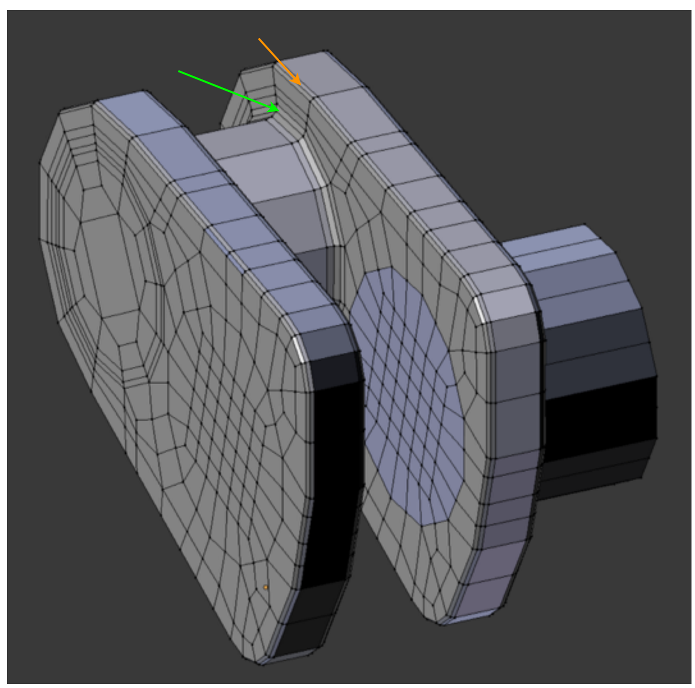


At the stage we show right, the central holed stub contains a void inclusion as seen in the Paraview representation of the unrefined .vtk file. It turns out that the void is caused by an overlapping face, vertically oriented. This is a common defect.

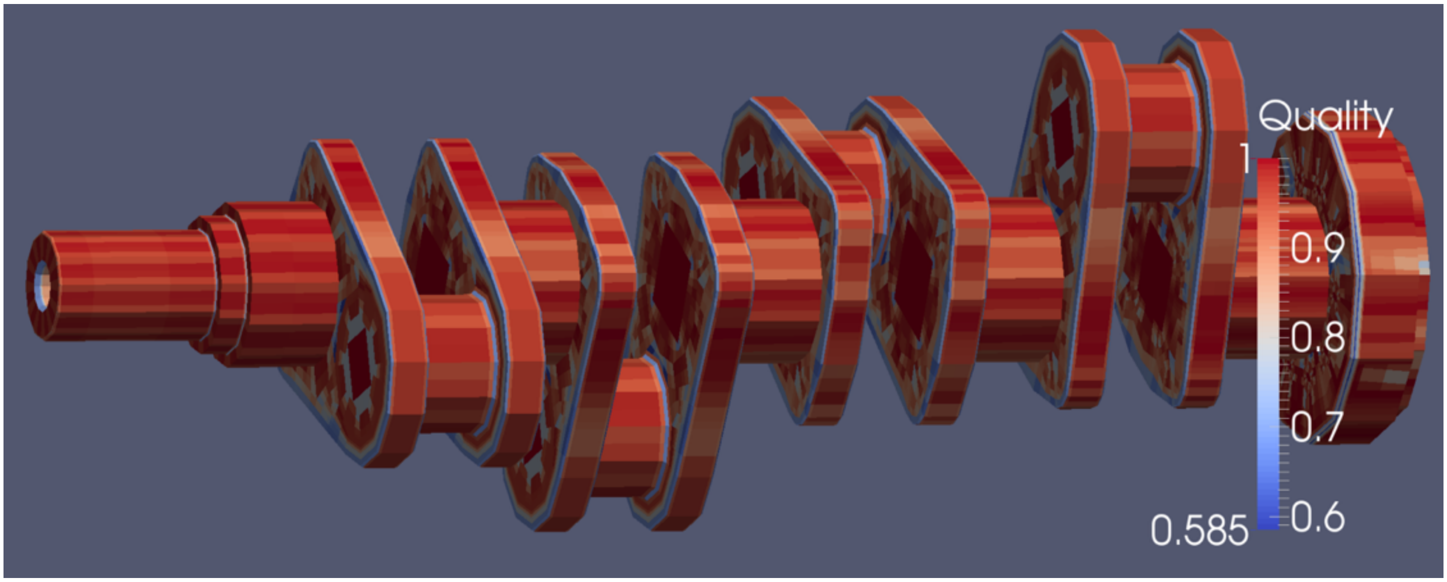


Sometimes the inclusions seen in a translucent Surface with Edges representation in Paraview are not 3D but only 2D. However, it is still important to track down any such defects and eliminate them, as they represent torn areas in the fabric of the mesh.

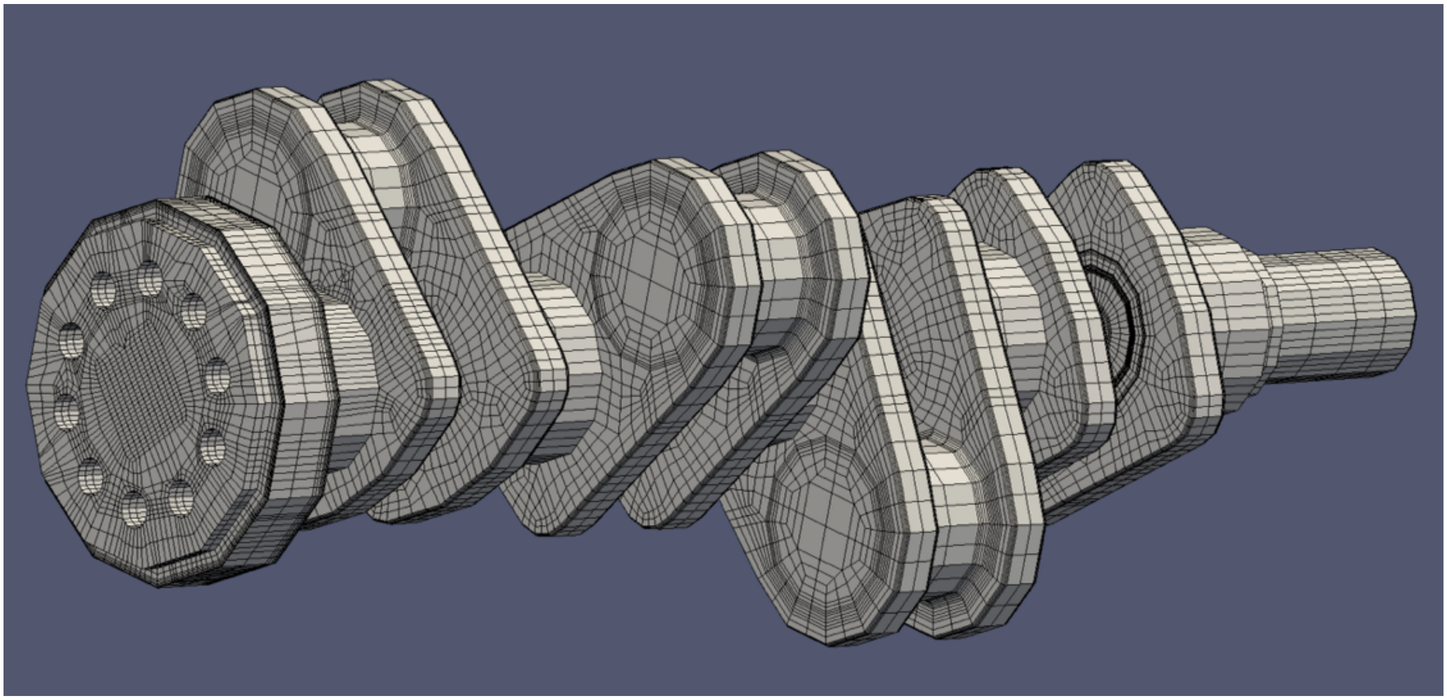
One completed ‘pod’ of the crankshaft is shown at right. By arraying around an empty with an assigned rotation of its own, a crankshaft shape can easily be created. However, precise location of the individual pods using this method is difficult to achieve. For this reason, we elect to do the duplication and assembly steps of the crankshaft by hand.



One section of the three-part inside fillet has a very shallow angle, shown with green arrow, which means we must complete its constituent element with a subsurface layer structure. Fortunately, the outside fillet, orange, also has need of subsurface layers.

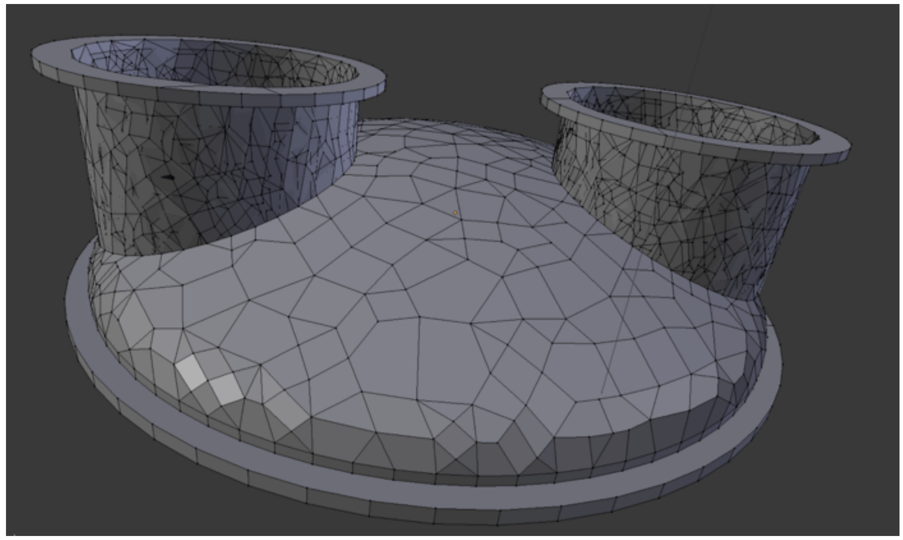


The Verdict quality for Scaled Jacobian is shown for the completed crankshaft. It exceeds the minimum allowable value of 0.5.

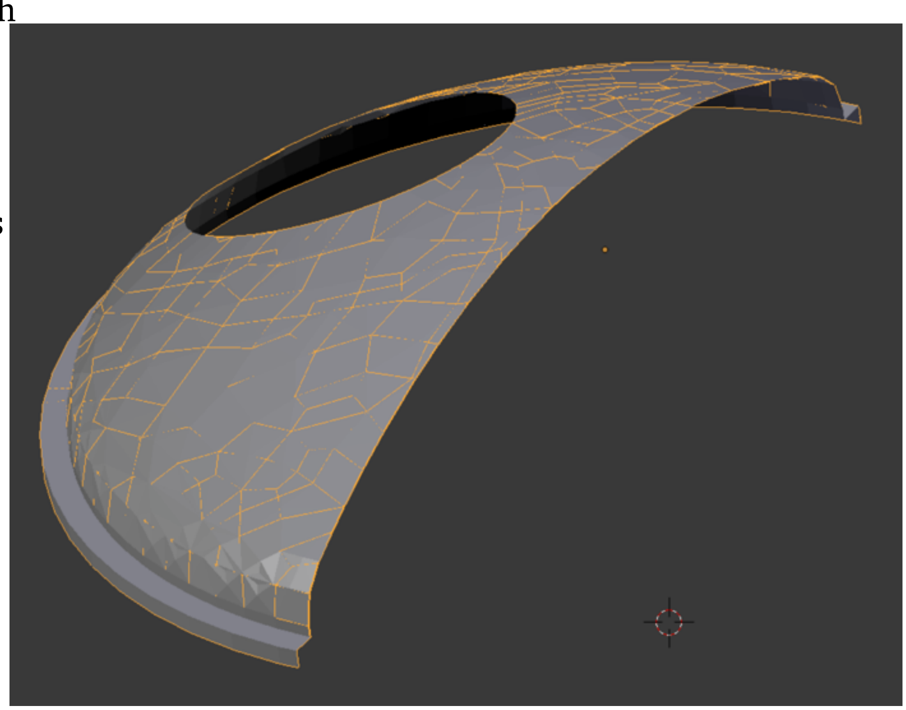


The final mesh, after refining in Gmsh, contains 128,064 elements and 142,790 nodes.

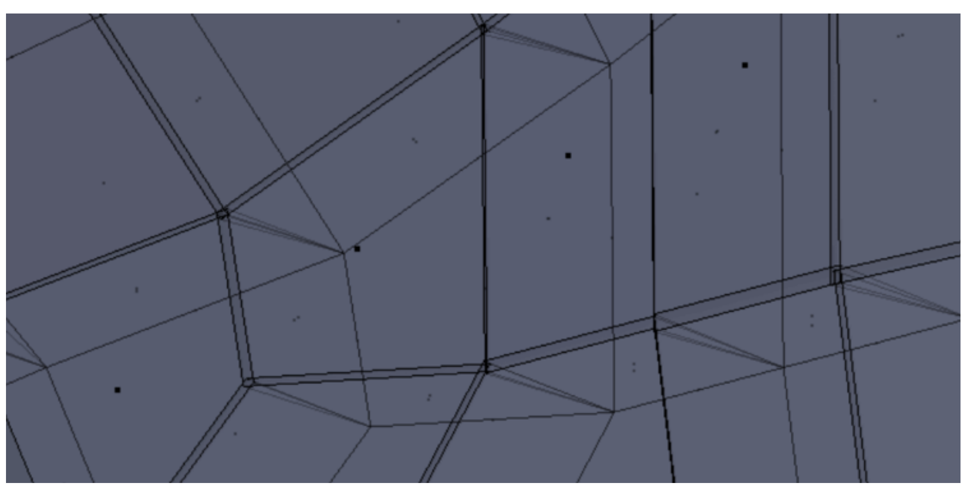
Demo 19



This object is a cooler head; the model can be found in the INRIA model repository as a .mesh file, and is opened in Gmsh. It is saved as a .stl file and imported into Blender. Its title is “Ghs3dDS,” (See References). The usual Tris-to-Quads and Partial Dissolve commands are given. The pipes and body are both of single element thickness.



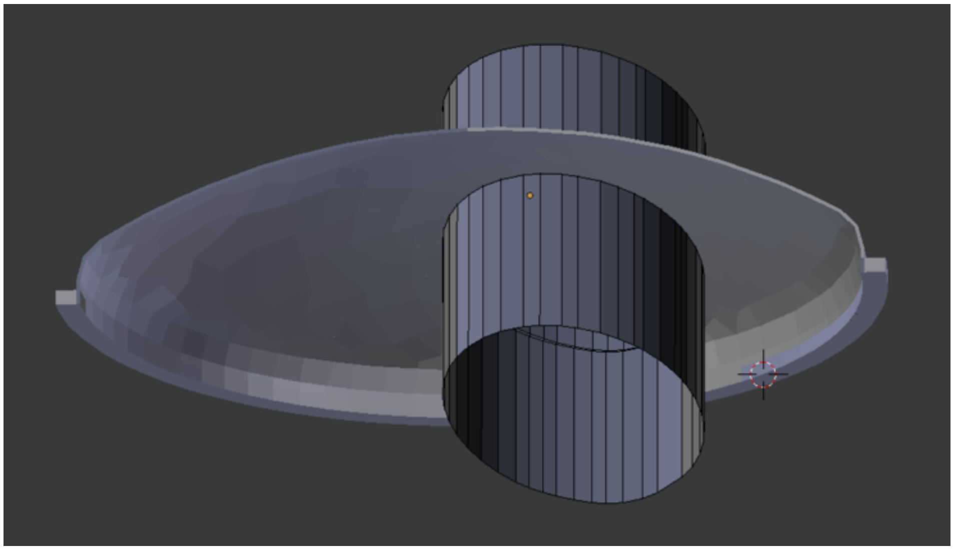
We separate the pipes, cut the mesh in half, and remove the bottom surface. The Tris-to-Quads command does not remove all the triangles, and we finish the operation, edge-sliding the vertices and using the knife tool. Now that we have a single layer, of manifold mesh, we can try using the Shrinkwrap modifier to conform it to the contour of the original surface. The view shown is of both surfaces, after the modifier has been applied. In the case of this model, insufficient quad density limits the usefulness of shrinkwrapping.



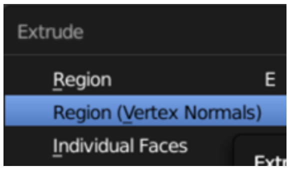
To get the new bottom surface, we Extrude Individual to maintain a normal extrusion direction. The surface curvature is great enough to preclude using the Remove Doubles command to join the separated vertices on the bottom surface, and we must resort to merging the vertices by hand.

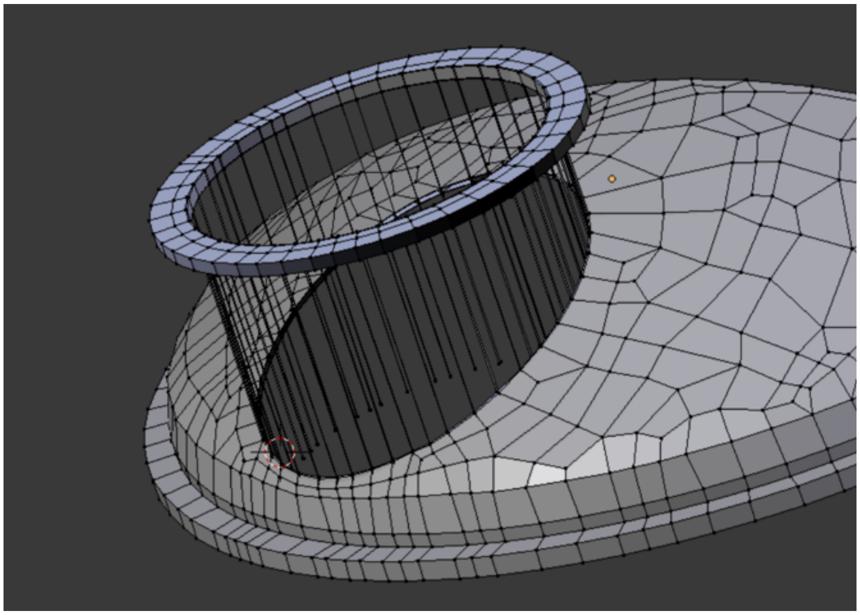
Using the edges lining the penetration, we extrude a new pipe along the z-axis.

At this time we rebuild the lower flange of the body.

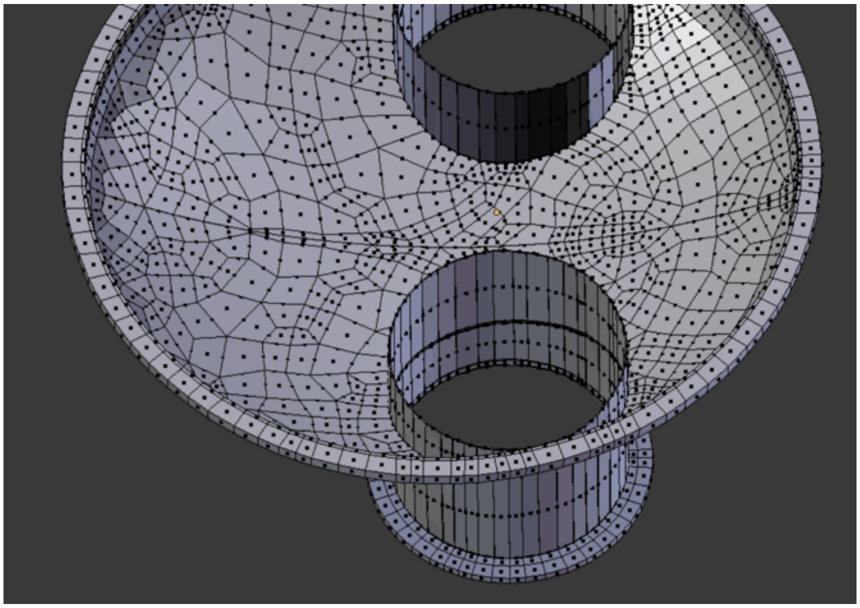


We use the Region (Vertex Normal) command to extrude the outer circle of vertical faces and thus construct the pipe flange.

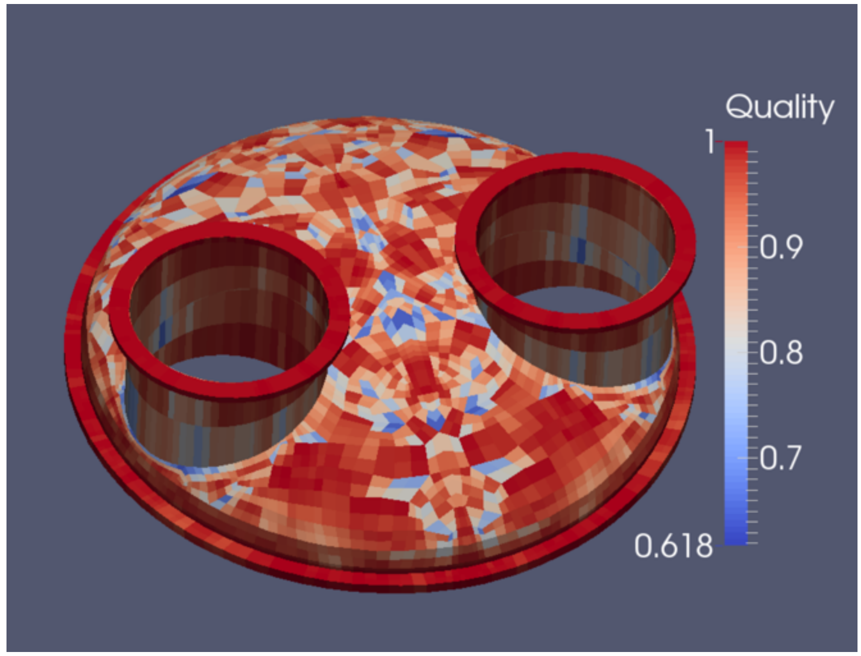




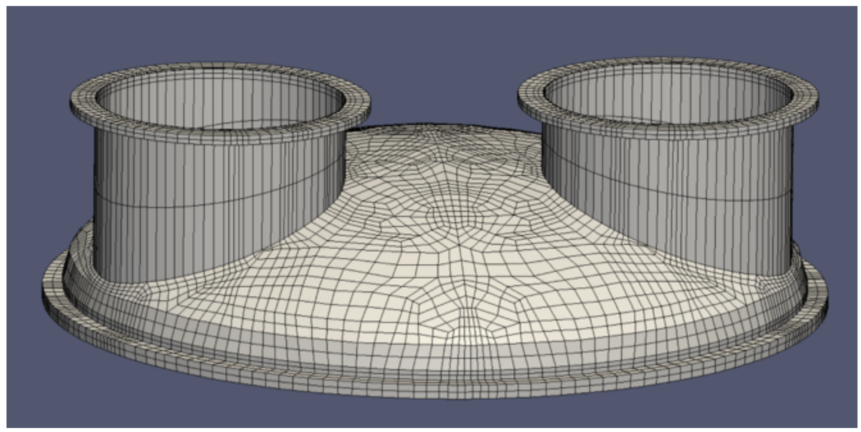
After verifying acceptability of the element quality, we mirror and merge. Because the edges along the seam are not exactly parallel to the x-axis, it proves necessary to ‘sew’ the bottom joint together.



The final quality is acceptable according to Verdict Scaled Jacobian standards.

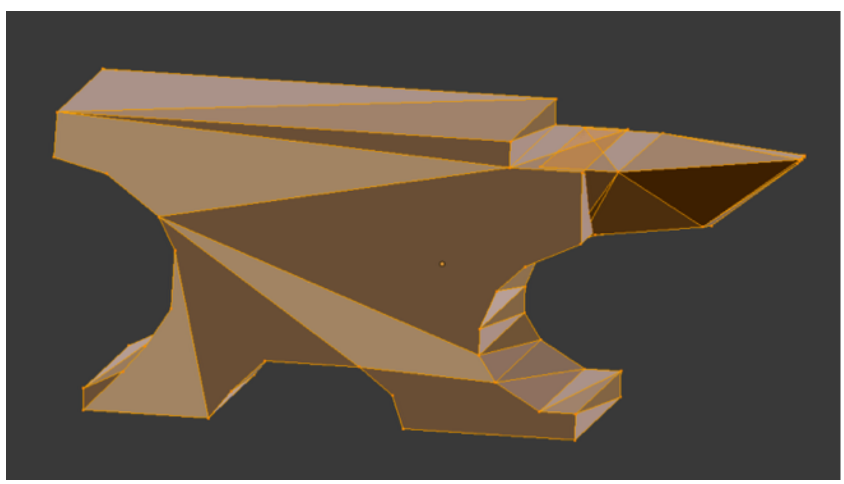


The final mesh contains 11,792 elements and 18,259 nodes.

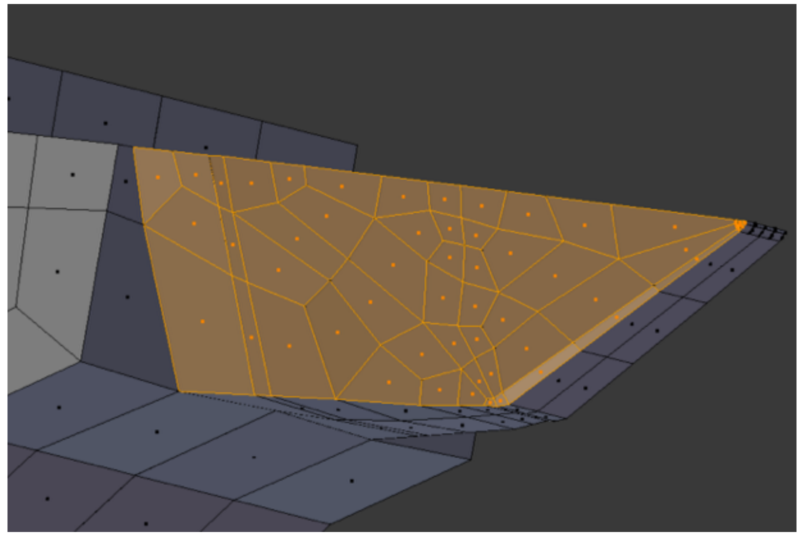


Demo 20

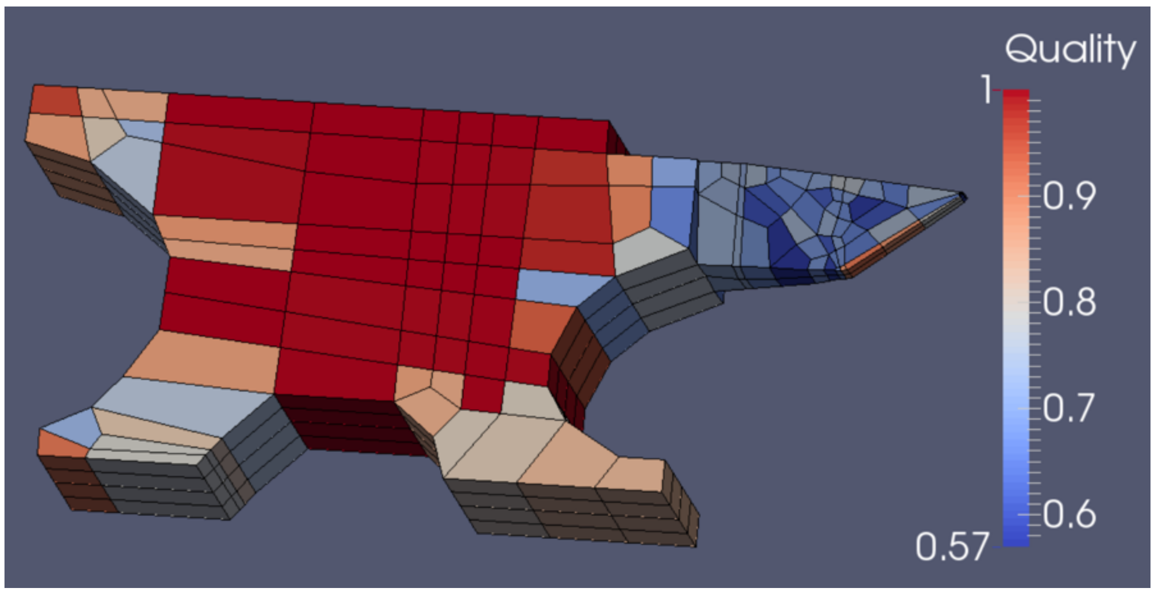
This object is an anvil; the model can be found in the INRIA model repository as an .obj file, and is imported into Blender using Blender's .obj import script. Its title is "Y3492," (See References). The object is too simple in construction to need any simplification commands.



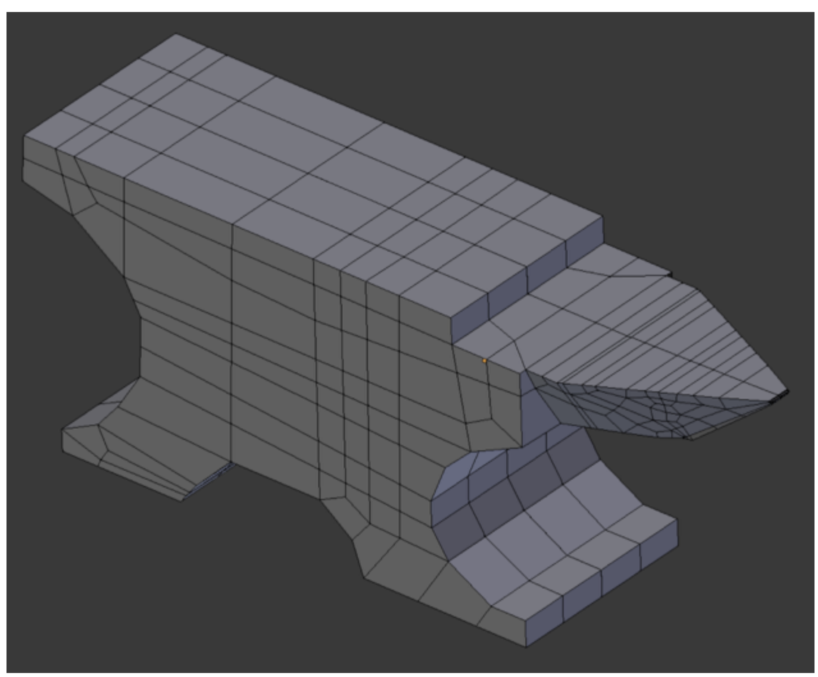
Since it is the most difficult part of the model, the anvil’s horn, after being cut in half, is the first part we divide into elements. Since the mesh’s major axes are aligned with the global axes, extrusion is simple.



The minimum Scaled Jacobian rating is in the range considered satisfactory by Verdict standards.

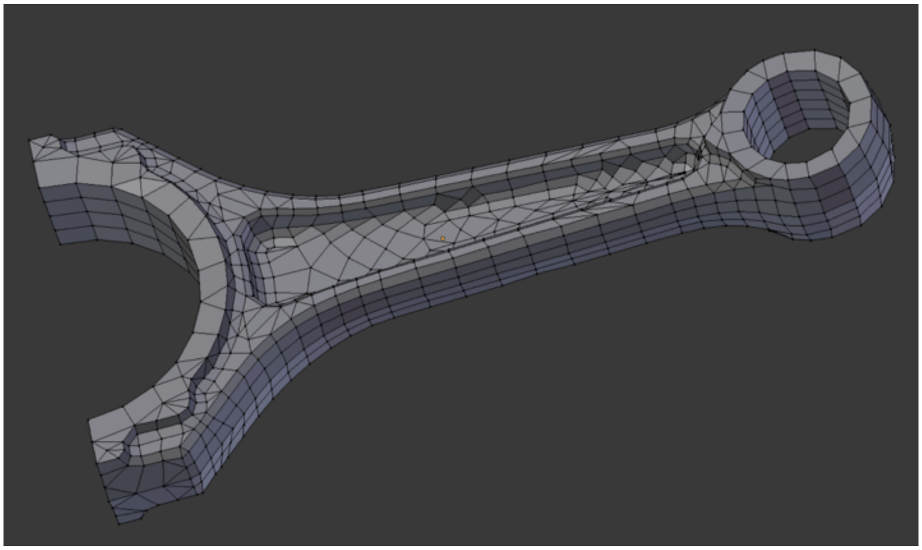


The final mesh contains 398 elements and 747 nodes.

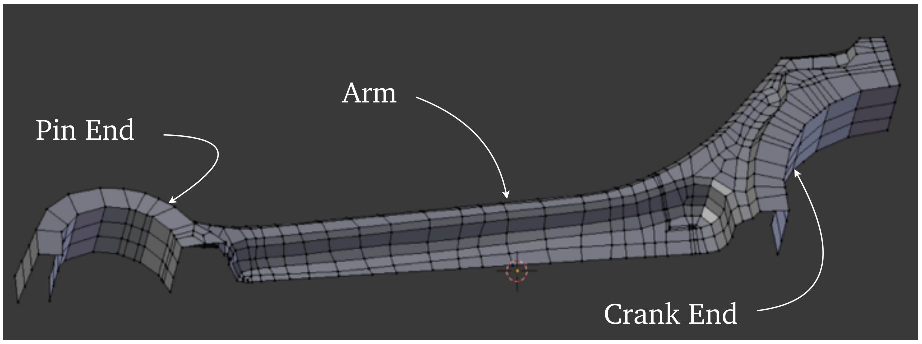


Demo 21

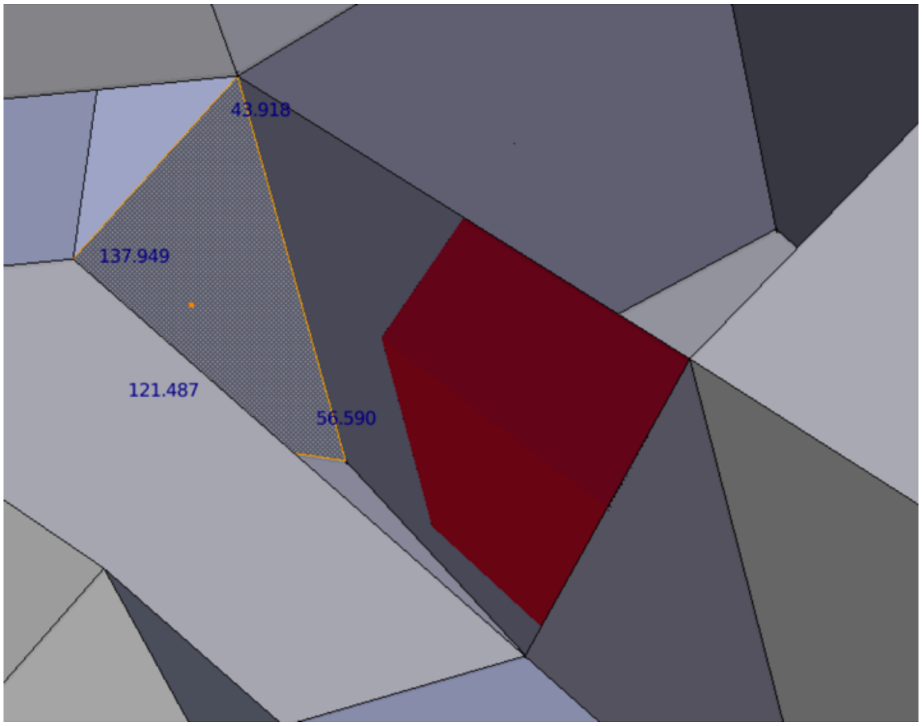
This object is a connecting rod; the model can be found in the INRIA model repository, Mechanical section, as a .mesh file, and is opened in Gmsh, then saved as a .stl file and imported into Blender using Blender's .stl import script. Its title is "bielle_tout," (See References). We give the usual Tris-to-Quads and Partial Dissolve commands.



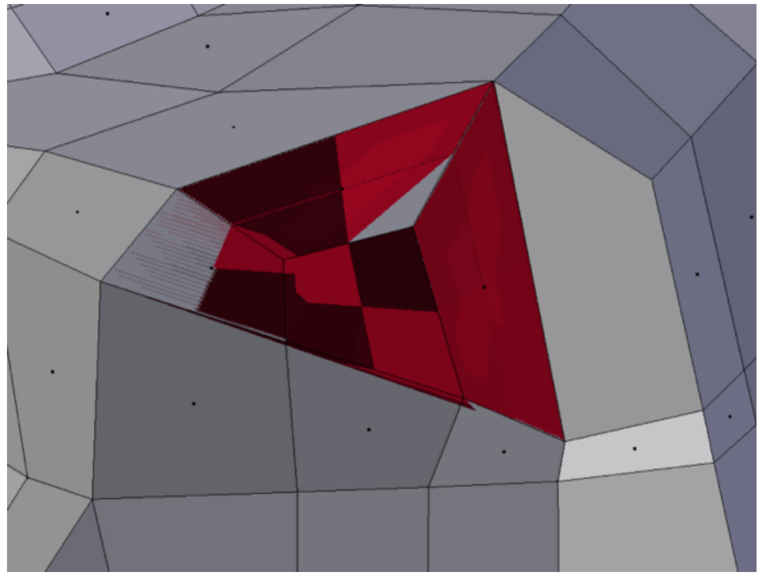
We quarter the connecting rod. Though not fully reflected in the view shown, we liberally cut the mesh with the knife tool, even if the cuts are known to be temporary, so that further face modification can be done by vertex sliding. The ability of the knife tool to follow the surface contours of the mesh makes it invaluable.



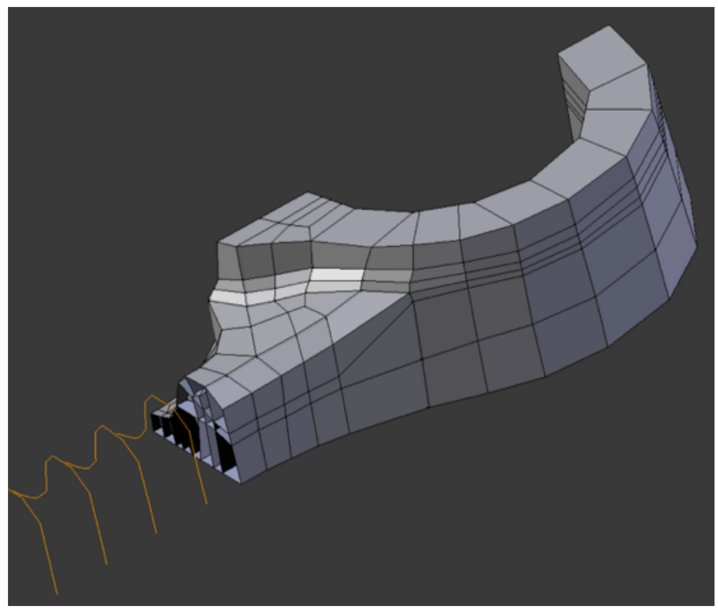
Smoothing the mesh. The red element is part of a collection of low-quality elements imported from Paraview for reference. It is a separate object and not selectable in Blender when the main mesh object is selected. Since it originated in the Gmsh-refined .vtk mesh, it is only about an eighth of the size of its parent. Its smaller size suggests the region of the parent proto-element which needs attention.



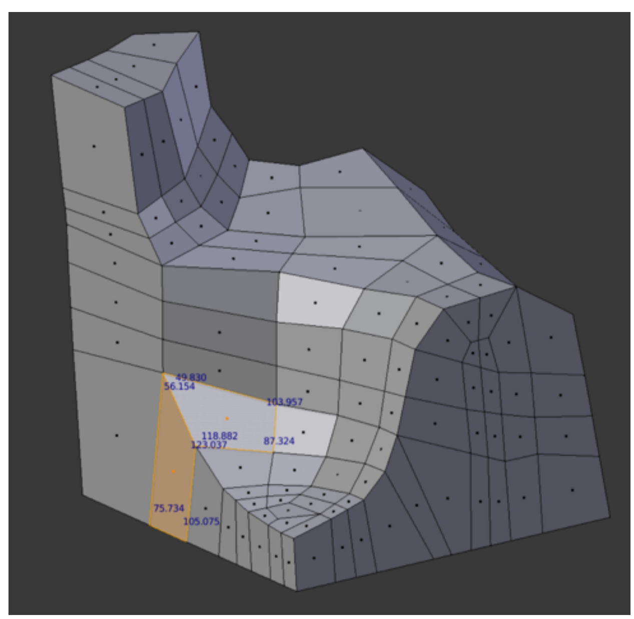
The plane-like group of faces in red is incorrectly created by Blenbridge while converting a .ply file to .vtk. The cause is a Comvol value too low. When the value of the variable is raised above 0.21, the artifact disappears.



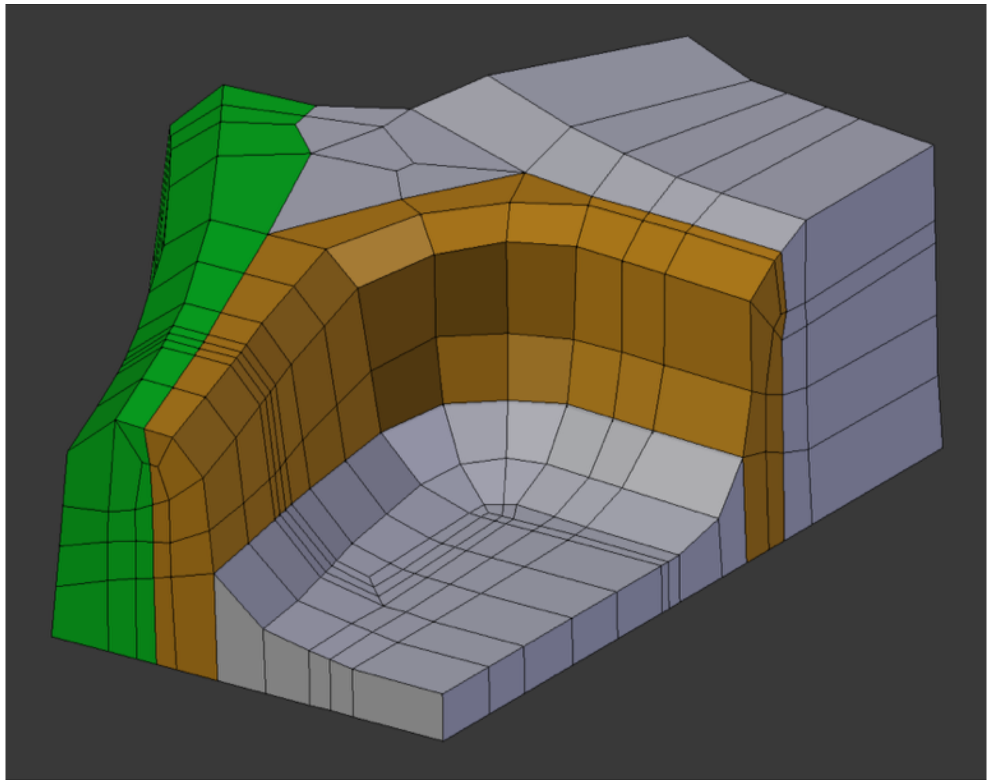
Starting at the pin end, we replace the section walls of the arm with regularly spaced ones, whose edges are planar. The profiles are created from knife cuts. Evenly spaced divisions not only look better, they speed up the face creation process.



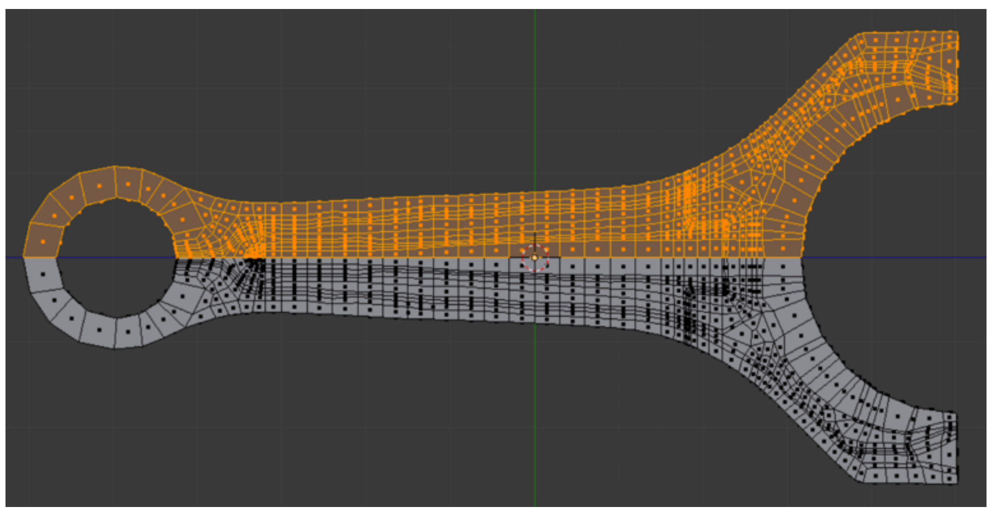
Highlighted in the picture at right is an element in the inside fillet area of the pin end, which has a relatively low quality rating, even though all of its 2D angles appear reasonable to us. We carve off the chunk shown, convert it into a separate object, and perform smoothing on the stubborn element as well as others in its vicinity. Tinkering must be restricted to faces which do not define the exterior. After a number of iterations, we find the elements in the area to be improved to a satisfactory degree.



Here we are at the inside fillet area of the crank end. The green pattern of faces, which starts at the pin end, propagates all the way until it reaches the flat plane of the crank split. Meanwhile, the orange pattern diverges, curves around the inside fillet area, and terminates at the zero Y mirror plane.

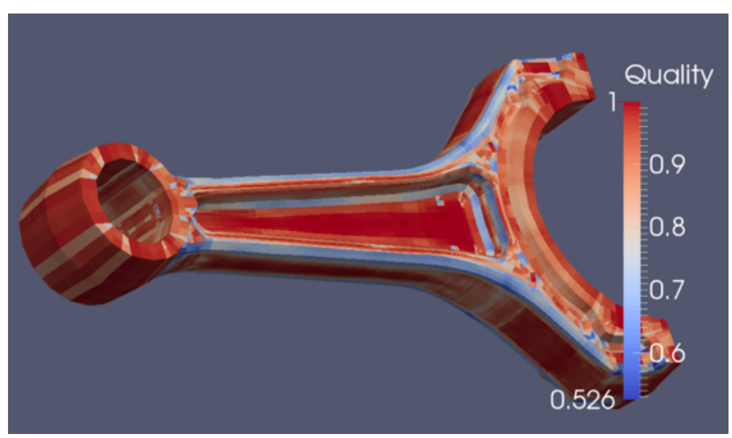


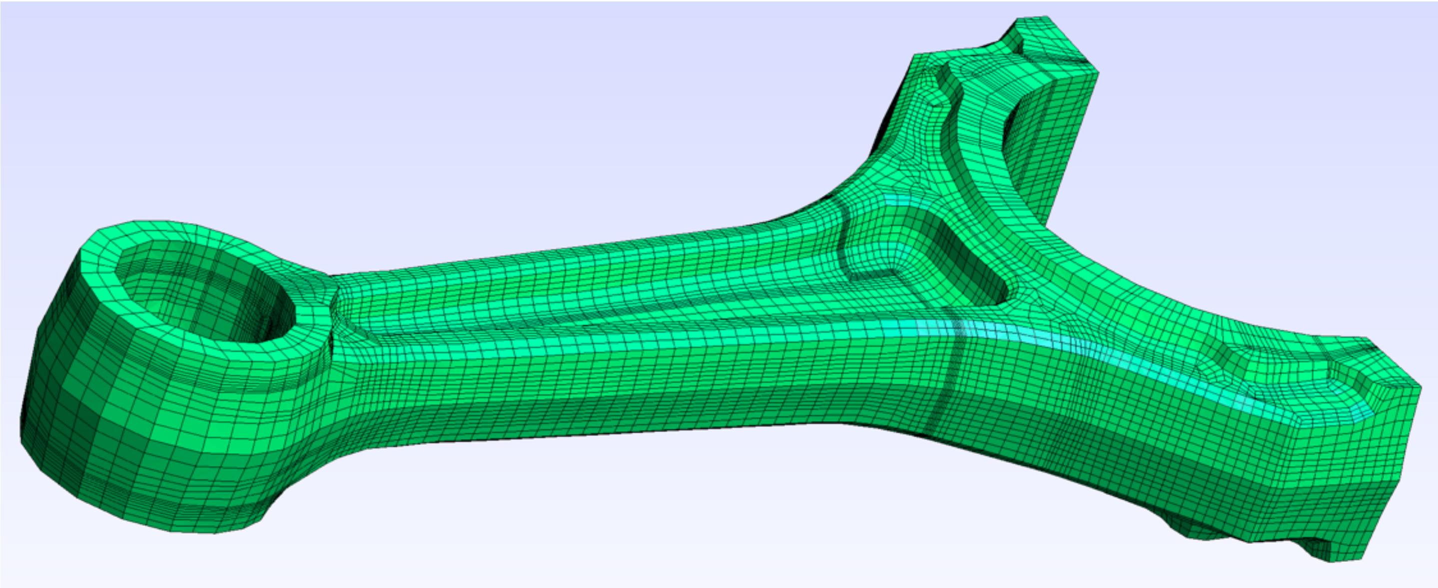
The planes of the mesh which have zero values are Y and X; and it is these planes we will mirror across. We carefully scale to zero all points belonging to these planes. The Y mirroring is referred to global orientation, and the X referred to local. Then more than a thousand doubles are removed.



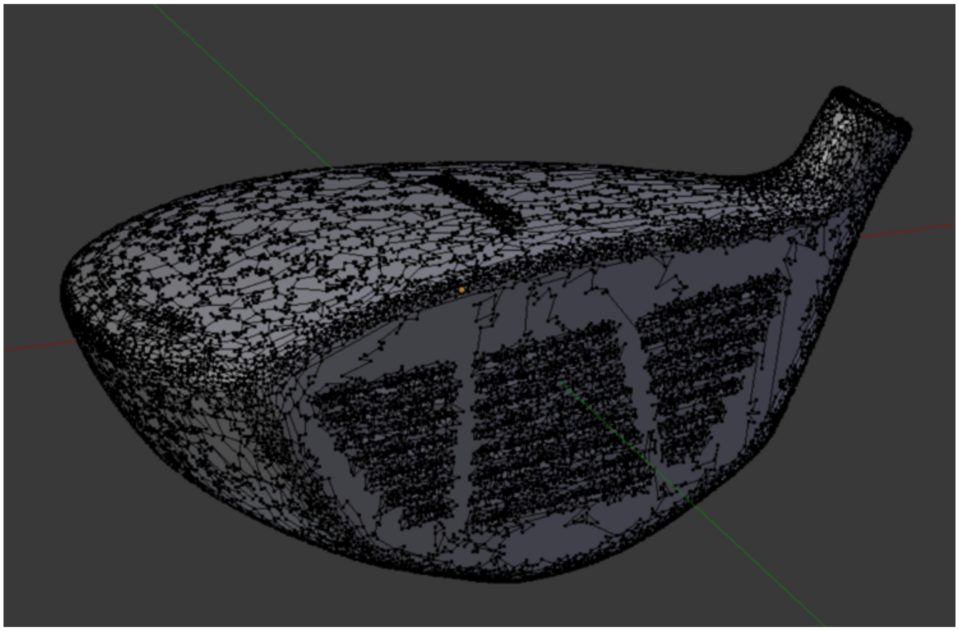
The final quality test finds a minimum value of 0.526, which is acceptable according to Verdict standards.

The final mesh contains 55,616 elements and 64,667 nodes.

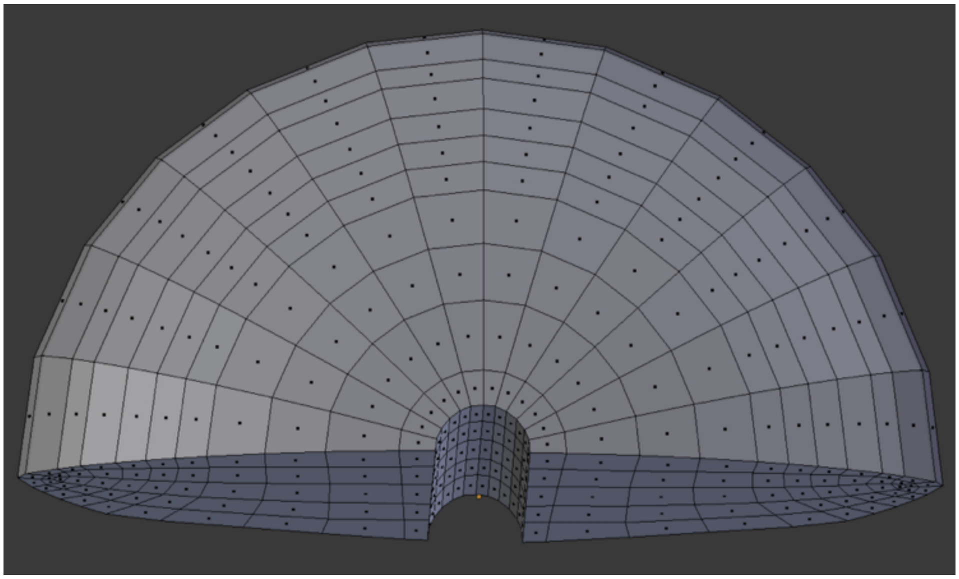




<u>Demo 22</u>

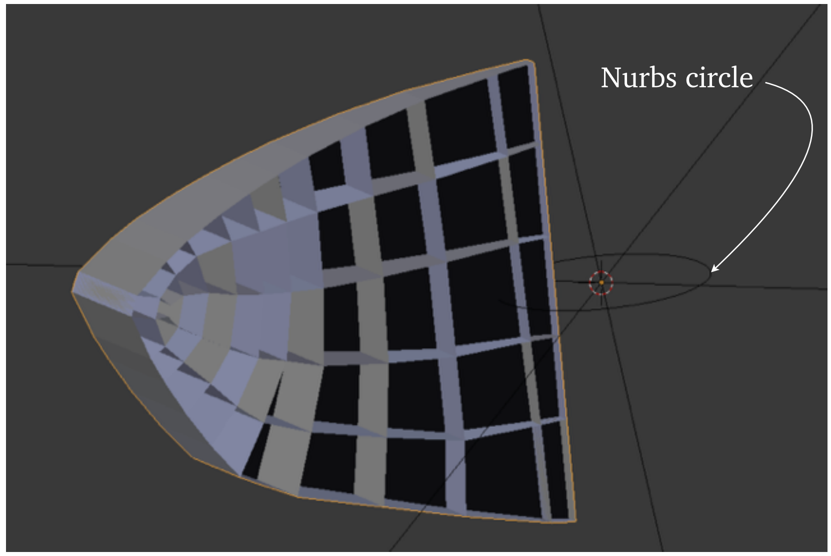


This object is a driver head; the model can be found in the INRIA model repository, Sports section, as a .mesh file, and is opened in Gmsh, then saved as a .stl file and imported into Blender using Blender's .stl import script. Its title is "club," (See References). We give the usual Tris-to-Quads and Partial Dissolve commands.

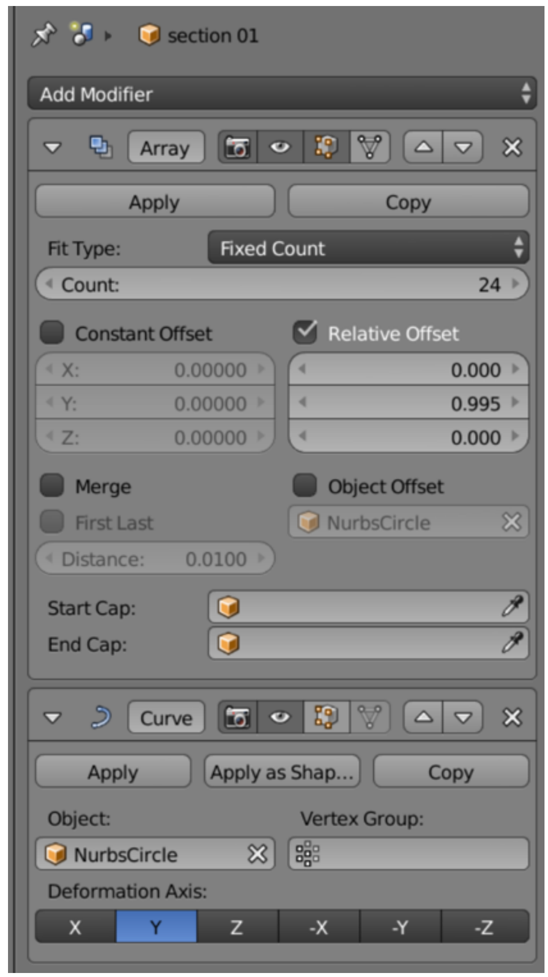


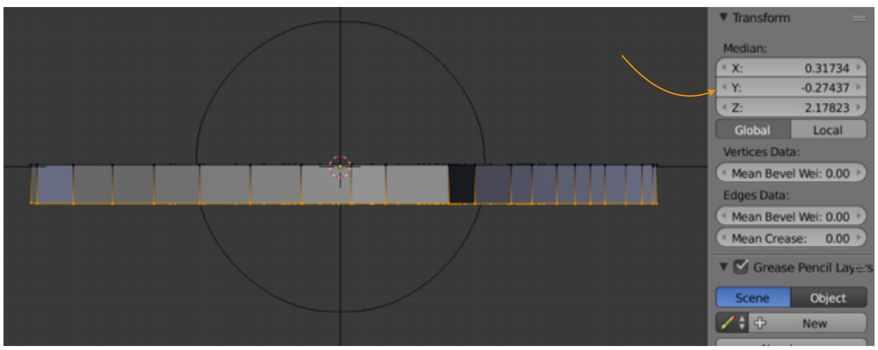
A golf club head may not greatly resemble a cylinder, but we choose to work the problem as if it did. Half of the revolved array is shown at right.

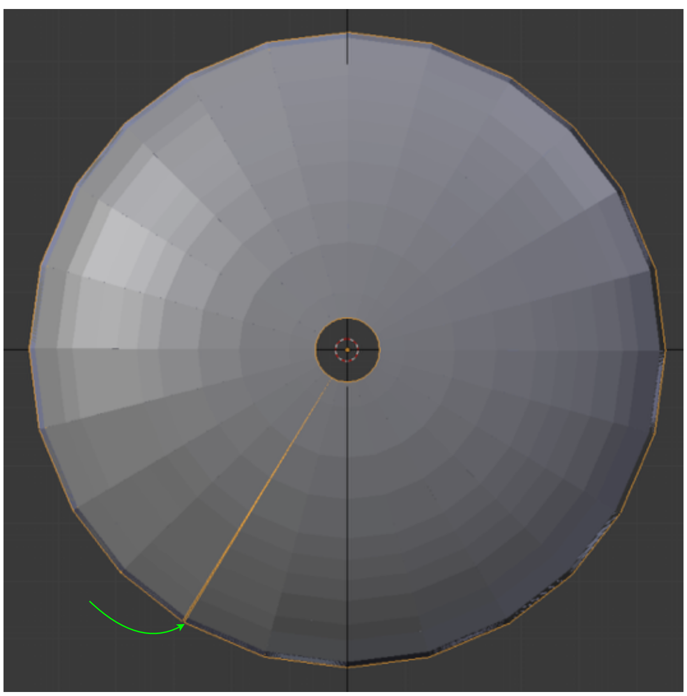
One segment of the 24 in the array can be seen here in Object mode. The back side of the segment consists of flat faces, and the forward side the ribs for the faces in the next segment forward. After trying some other alternatives, we find that the nurbs circle behaves best for this task.



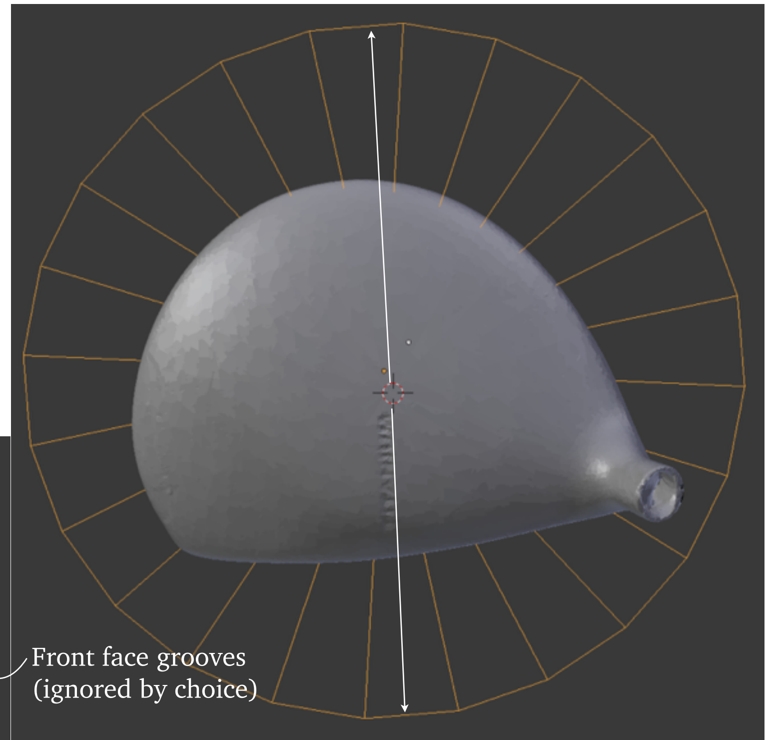
In Edit mode we adjust the width of the segment along the y-axis, using the ‘g’ key with numerical input, watching (orange arrow) as the accumulation of length causes the end point, as viewed in Object mode, to eventually match the start point well (green arrow). The last joint is then welded by hand.

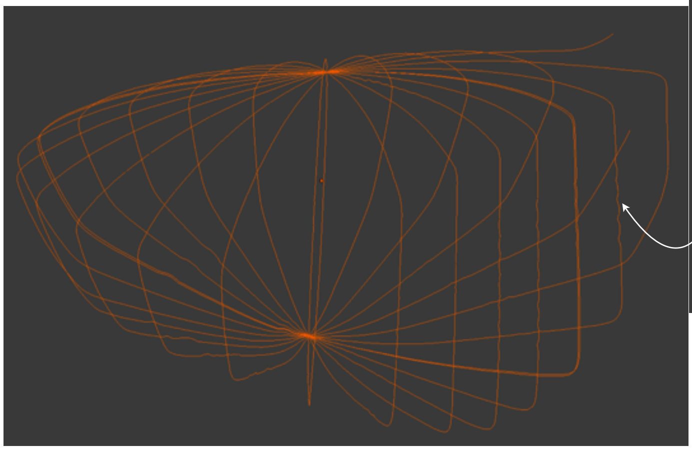


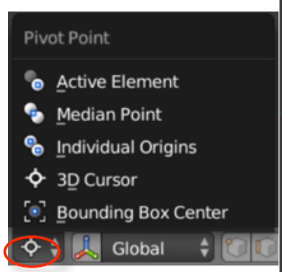


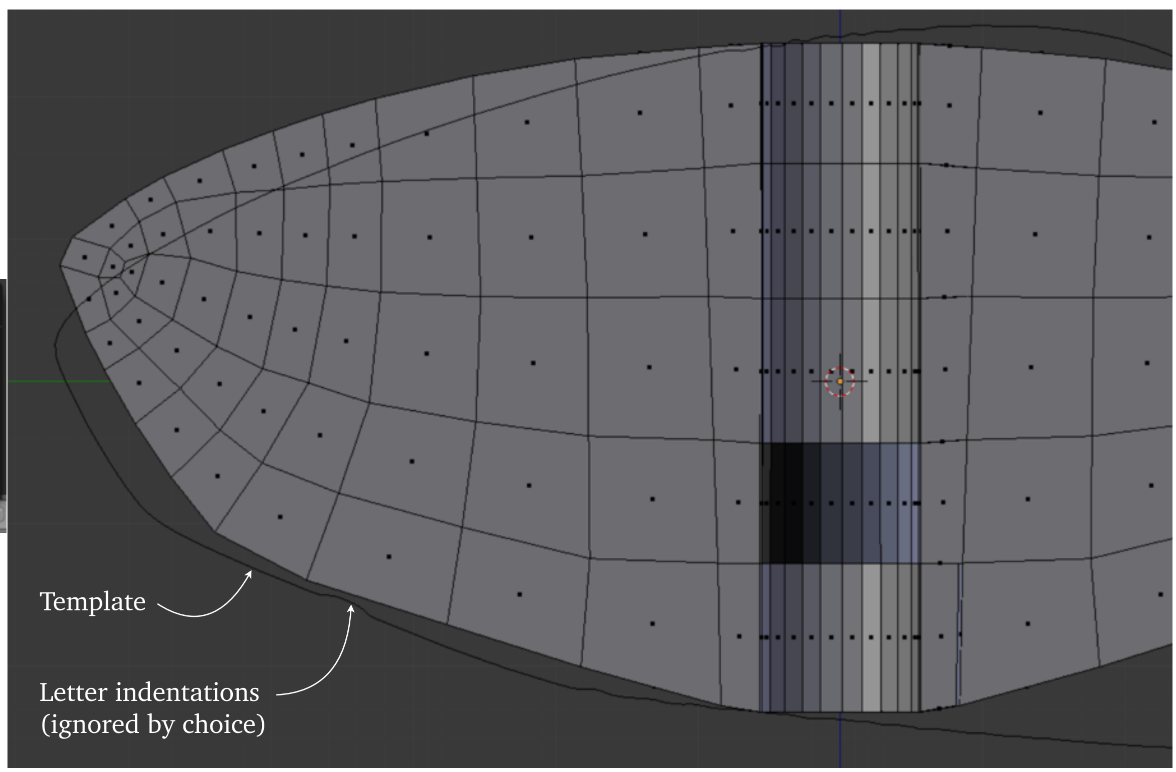


We use a sectioned circle to create 24 section outlines, one for each of the division plates in the array. It is important to position the center of the cut with forethought, so that there is sufficient room for the scaling that will take place on all the sections. The collection of connected edges forming the templates is shown below.



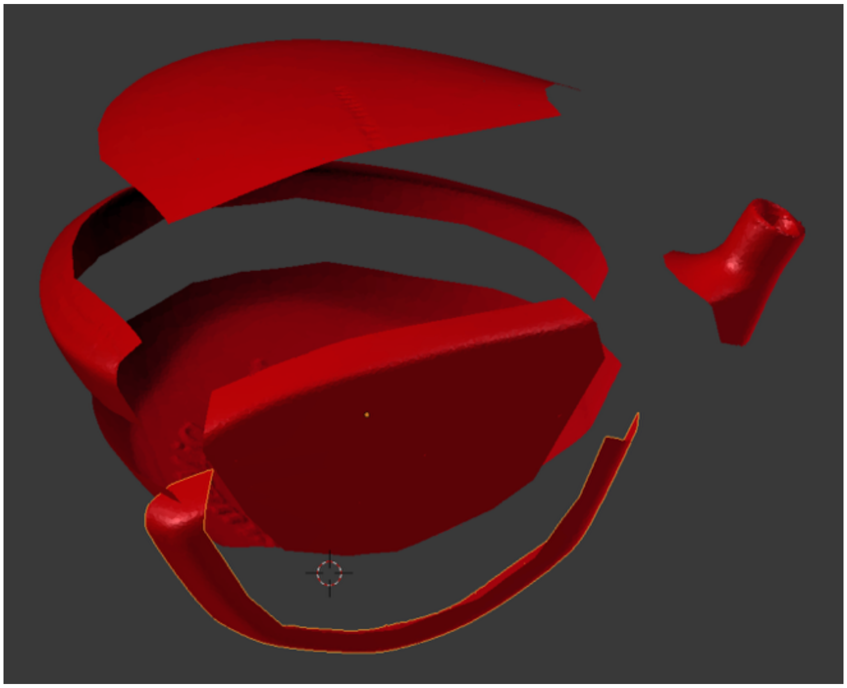


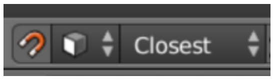




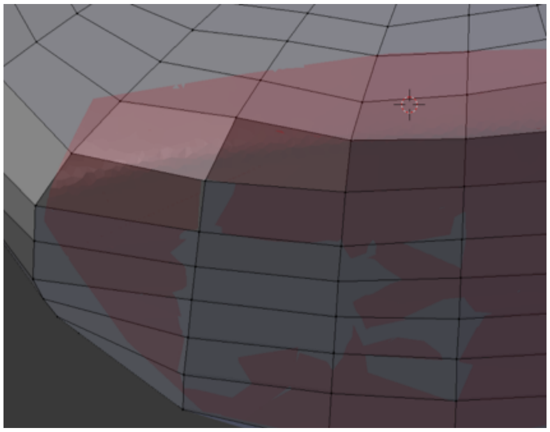
We see above the view of a section plate ready to be scaled. We make sure that the view is perpendicular to the section plate, and that the 3D cursor is the current pivot point. After all faces are selected and scaled within the template, the outer points are slid along the original edges to their required positions to keep a concentric theme in the edges on the outer surface.

An important part of our approach consists of snapping points to the original model. We have found that Blender can get confused when snapping to a complex object, and therefore we cut the model into pieces, placing each in turn on the mesh and snapping to its surface. We snap points to a closest ‘volume’, as indicated in the icons shown. On the active layer, we allow only one non-mesh object, which is then interpreted as target. It is best to continually move the mesh to keep the snap direction aligned to the direction of the view. (Pressing the ‘g’ key activates the snap.)

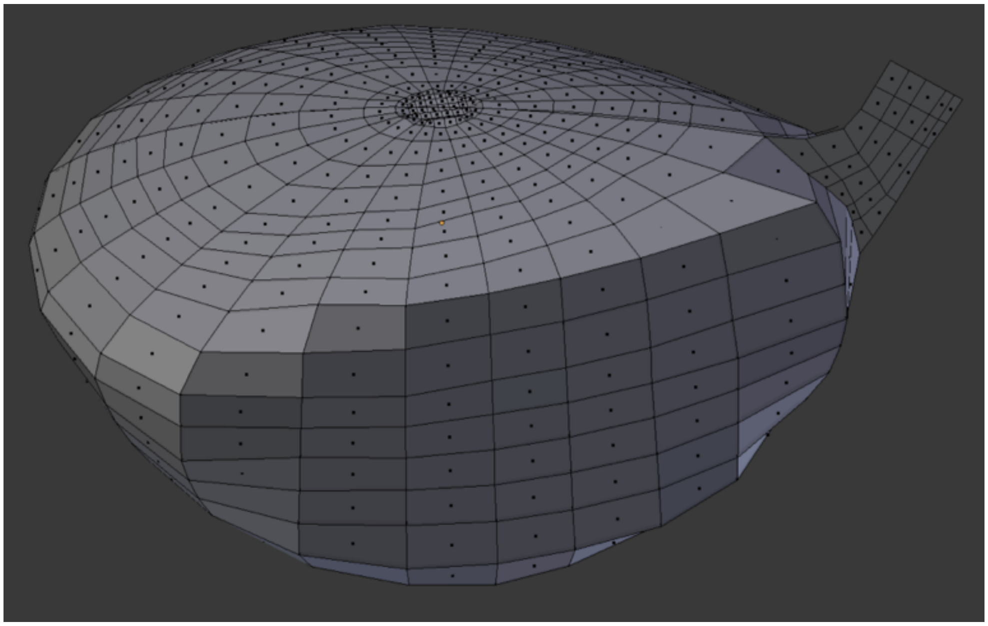




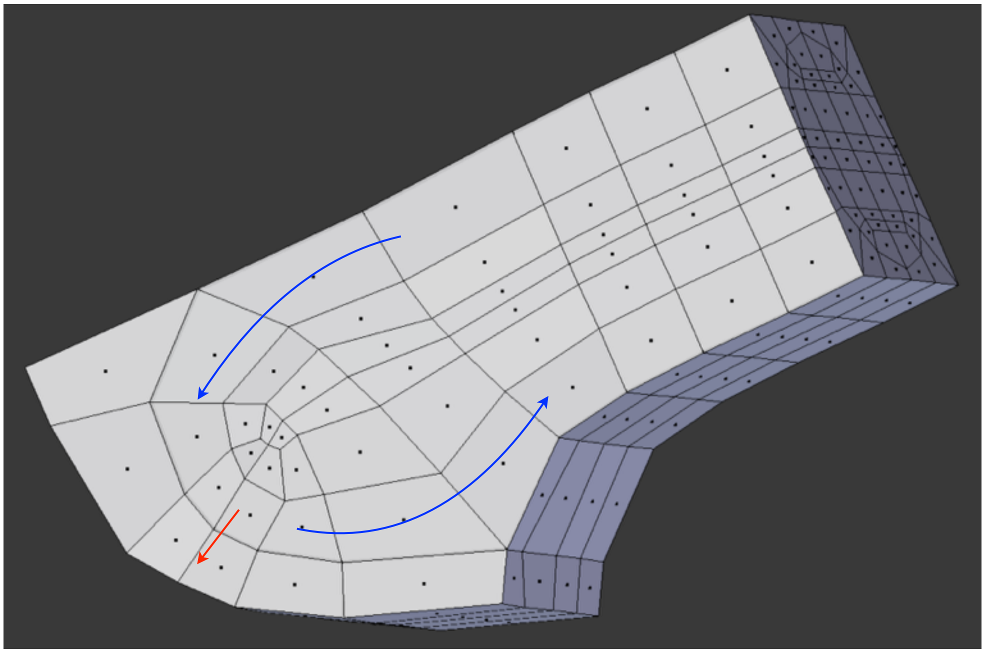
The current snap target can be given a transparent texture, which helps us to see what is happening. A change of reflected light on a face may indicate it has ‘popped out’, but a large disorganized change in appearance means we have slipped off the target surface and need to press the Escape key to abort the snap.



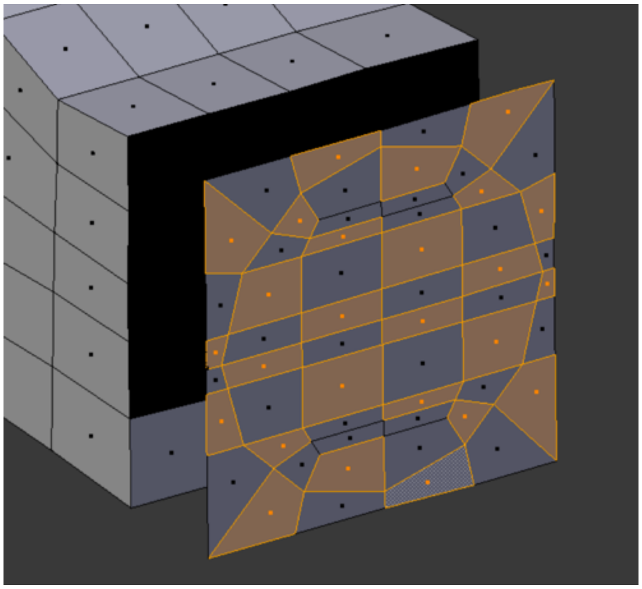
In golf club parlance, the area of attachment between head and shaft is called the ‘hosel’. At right is seen a crude 2D hosel together with a low-resolution mesh, after the first overall snapping. It is useful to always maintain the roughest resolution possible consistent with the tasks to be accomplished during a particular stage.



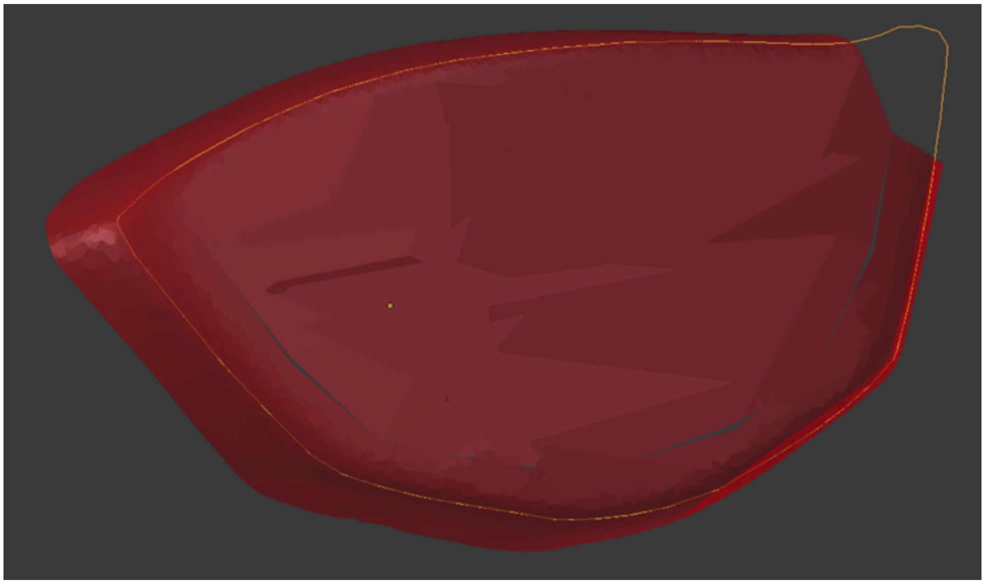
In creating the hosel, we manage the face patterns so that they reflect back on themselves (blue), which prevents propagation of unnecessary detail throughout the main mesh. One central edge boundary does carry through (red), and is necessarily incorporated into the main body.



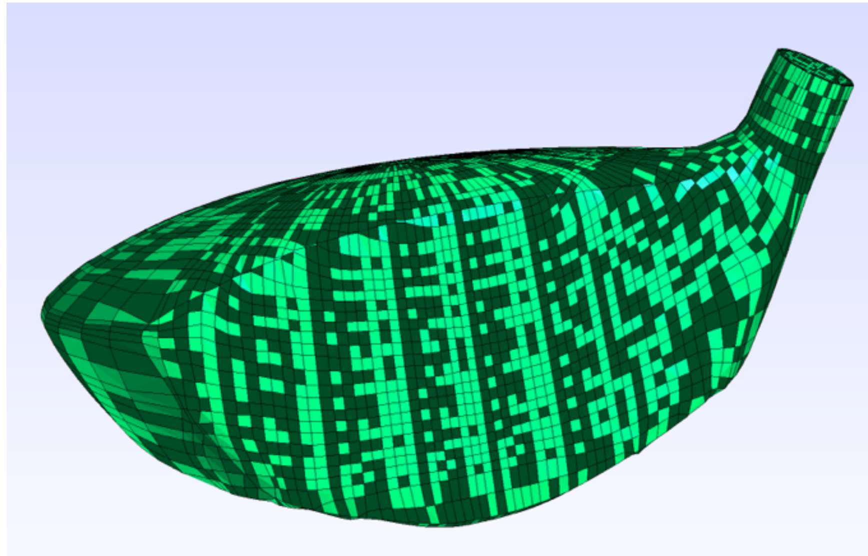
The axis of the hosel is not aligned with a major global axis, so extrusion is problematical. By selecting a checkerboard pattern and using the ordinary Extrude command, we can get extrusion perpendicular to the original faces, and are left with approximately half of the faces to complete along the extruded structure.



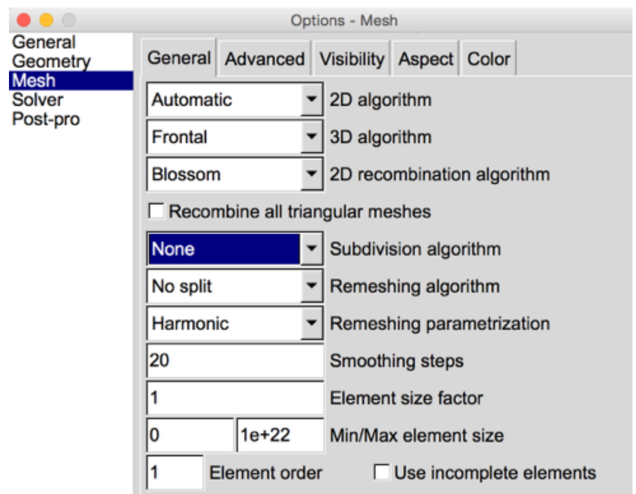
All the points are snapped to the model. But with its randomly created surface, the model lacks definition at its edges. We therefore define a subset of points, a ring, along which we will bend our edges to make the boundary of the club's contact face more regular. This ring, a separate object, is used for visual comparison of point position.



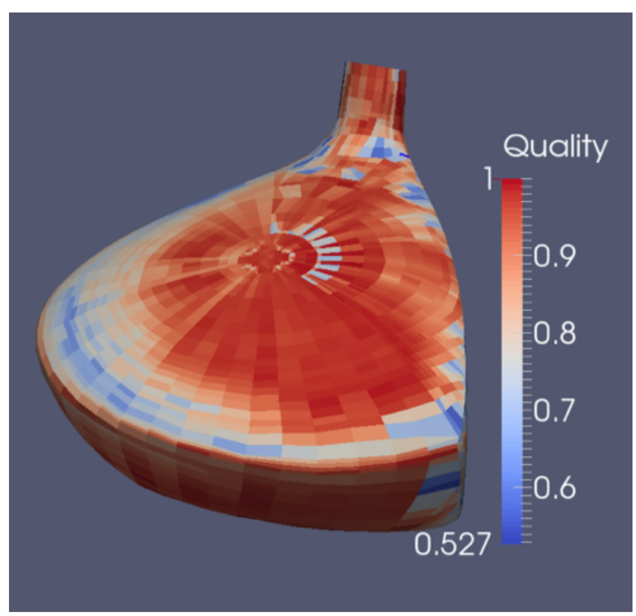
One important thing that Gmsh does for us is to align all the face normals. When the completed mesh enters Gmsh, it appears as shown.

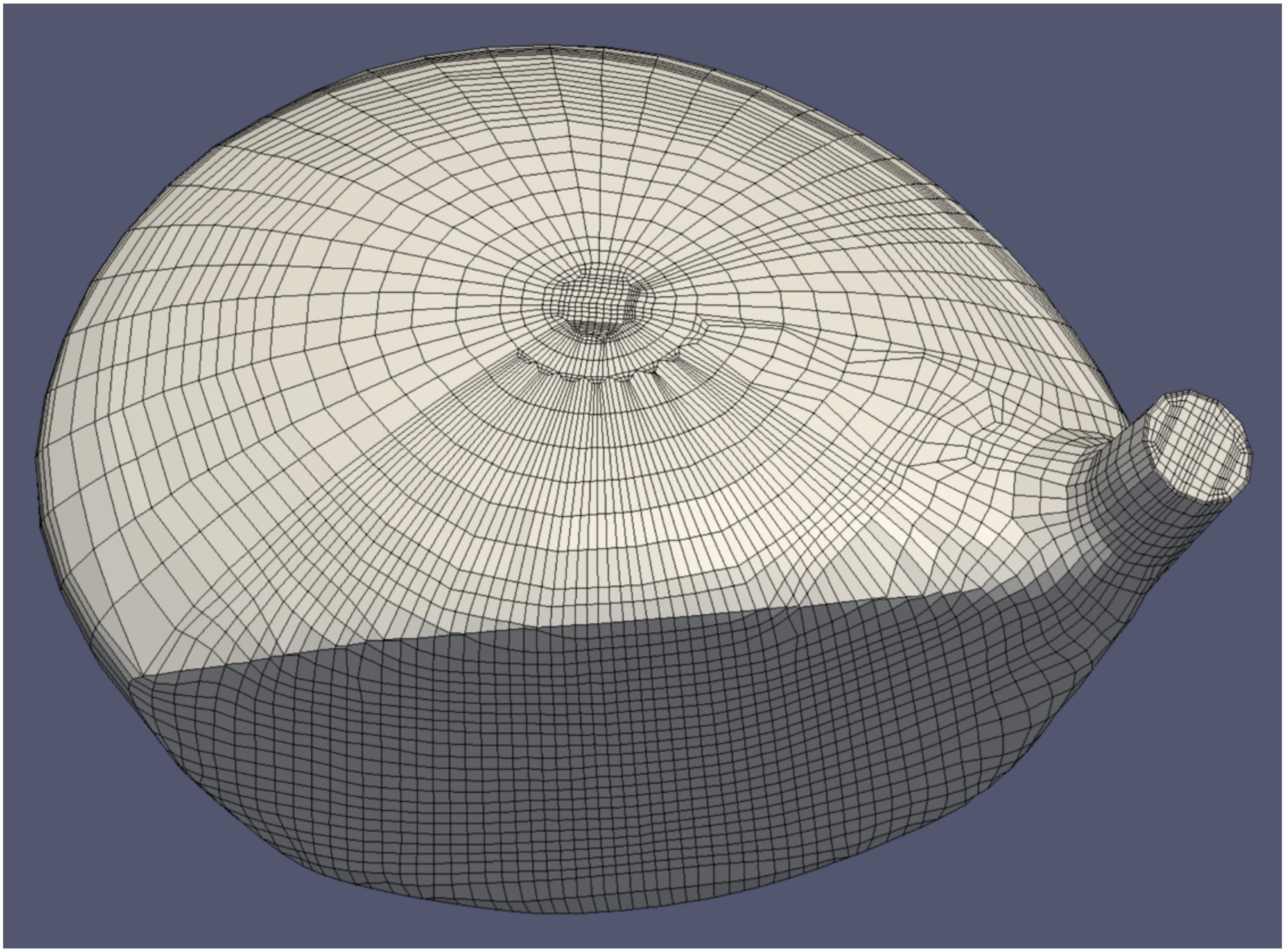


It is desirable to control the mesh density and keep it from getting out of hand. A Gmsh dialog box with selections as shown can deliver a .vtk mesh to Paraview without increasing its density. (Note: in spite of a number of iterations, the 'smoothing steps' take place so quickly that we doubt anything really happens.) Pressing '3D' from the GUI mesh menu creates the mesh.



The Scaled Jacobian mesh quality, with a minimum of .527, meets the required Verdict standards.





The final mesh contains 31,840 elements and 34,651 nodes.

Note that the resolution was increased in the club's contact area, using the cutting and face filling method described in Demo 14.

References:

• Geuzaine, C. and Remacle, J-F. (2017). Gmsh 2.16.0 [Computer software]
• CSC - IT Center for Science. (2016). Elmerfem-CSC 8.2-20170511 [Computer software]
• Stichting Blender Foundation. (2016). Blender 2.78a [Computer software]
• Kitware, Inc. (2013). Paraview 4.0.1 [Computer software]
• Nextwave Software Solutions SRL. (2016). Snappy 1.4.7 (116) [Computer software]
• Free Pascal Lazarus Project. (2015). Lazarus 1.6 [Computer software]
• University of Iowa, Center for Computer-Aided Design. (2008). IA-FEMesh 1.0 [Computer software]
• University of Utah, Musculoskeletal Research Laboratories. (2016). Febio Preview 1.19 [Computer software]
• Dhondt, G. and Wittig, K. (2016). Calculix 2.11 [Computer software]
• Schneiders, R. (2000). Algorithms for Quadrilateral and Hexahedral Mesh Generation. MAGMA Gießereitechnologie GmbH. Aachen, Germany [Document]
• freecadweb.org. (2016). FreeCAD .14 [Computer software]
• INRIA Gamma Team Research Database Website Collections. (2013). 922_pump_carter [3d Model]
• Wendling, C., Treleaven, N., et al. (2014). Geany 1.24.1 [Computer software]
• Visualization Virtual Services Aim Shape Shape Repository (2007). Pump Carter [3d Model]
• Stimpson, C., Ernst, C., et al. (2007). The Verdict Library Reference Manual [Document]
• OneMinute Video Tutorials. (2016). "How to Straighten Inclined Objects in Blender." YouTube NEUa1IA7NBQ [Video]
• INRIA Gamma Team Research Database Website Collections. (2005). DEMO10 [3d Model]
• Schöberl, Joachim (2009). Netgen 4.9.11 [Computer software]
• OpenFOAM Foundation (2016). OpenFOAM 4.1 [Computer Software]
• INRIA Gamma Team Research Database Website Collections. (2005). Aries155 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2002). Y7034 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2002). Y7923_Gear5 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2006). fan_1 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2008). bracket1 [3d Model]

• INRIA Gamma Team Research Database Website Collections. (2008). gancho [3d Model]
• CG Masters. (2015). "Align 2 Faces - Advanced Snapping Blender Tutorial." YouTube Gc8BekthXAQ [Video]
• Rethaller, Tom. (2013). Align by Faces 2.2 [Blender Add-On]
• INRIA Gamma Team Research Database Website Collections. (2005). rockerarm25 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2005). bracket [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2008). Crank3D [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2008). Ghs3dDS [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2005). Y3492 [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2008). bielle_tout [3d Model]
• INRIA Gamma Team Research Database Website Collections. (2004). club [3d Model]